Region Proposal Oriented Approach for Domain Adaptive Object Detection

Hiba Alqasir, Damien Muselet, Christophe Ducottet

Region Proposal Oriented Approach for Domain Adaptive Object Detection*

Hiba Alqasir(✉), Damien Muselet, and Christophe Ducottet

Université de Lyon, UJM-Saint-Etienne, CNRS, IOGS, Laboratoire Hubert Curien UMR5516, F-42023, Saint-Etienne, France

Abstract. Faster R-CNN has become a standard model in deep-learning based object detection. However, in many cases, few annotations are available for images in the application domain referred as the target domain whereas full annotations are available for closely related public or synthetic datasets referred as source domains. Thus, a domain adaptation is needed to be able to train a model performing well in the target domain with few or no annotations in this target domain. In this work, we address this domain adaptation problem in the context of object detection in the case where no annotations are available in the target domain. Most existing approaches consider adaptation at both global and instance level but without adapting the region proposal sub-network leading to a residual domain shift. After a detailed analysis of the classical Faster R-CNN detector, we show that adapting the region proposal sub-network is crucial and propose an original way to do it. We run experiments in two different application contexts, namely autonomous driving and ski-lift video surveillance, and show that our adaptation scheme clearly outperforms the previous solution.

Keywords: Object detection · Domain Adaptation · Deep learning · Faster R-CNN.

1 Introduction

Object detection in images refers to the task of automatically finding all instances of given object categories outputting, for each instance, a bounding box and the object category. Recently, approaches based on deep Convolutional Neural Networks (CNNs) have invaded the field thanks to both their efficiency and their outstanding performances [18,17]. To address a given computer vision problem, these methods require large training datasets with instance-level annotations. However, for most real world applications, few annotations are available due to the lack of image sources, copyright issues or annotation cost. To overcome this problem, a current trend consists in training the network on a large public annotated dataset (source domain), while adapting the network features to the tested dataset (target domain). This approach is called domain adaptation [14,23]. If

* Partially funded by MIVAO, a french FUI project

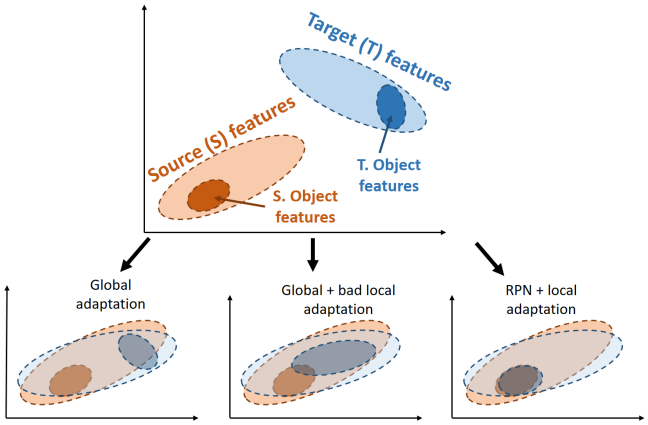

Fig. 1: Illustration of global, local and RPN adaptation (see text for details).

no annotations are available in the target domain, the domain adaptation is referred as unsupervised.

In this context, the case of autonomous driving has been extensively addressed and a variety of datasets exists covering different urban scenes situations, illumination and weather conditions [4,6]. In this paper, we are particularly interested in unsupervised domain adaptation in a ski lift video surveillance scenario. The purpose is to detect dangerous situations during chairlift boarding by detecting relevant objects of the scene (e.g. safety bar, people, chairlift carrier). Instance level annotations are available for a reference chairlift where the chair model, the perspective view or boarding system may be different from the target chairlift [2].

Surprisingly, few works explicitly address the problem of unsupervised domain adaptation for object detection. Most approaches study the supervised case basically by fine-tuning a model pre-trained on the source dataset with few annotated images from the target domain, eventually freezing some layers to concentrate the training on the last layers [8,22]. Other recent approaches try to reduce the domain shift by transforming the source domain to make it close to the target one using style transfer [12]. The most significant contribution of domain adaptive object detection was proposed by [3]. Following [5], they added adversarial training components in the classical Faster R-CNN detector, in order to adapt both globally and locally the detector. Given the features from the two domains and considering the subset of features specific to the object (Fig. 1), a global adaptation, as illustrated in the bottom left of Fig. 1, may not match source and target object features. Thus, Chen et al. [3] also propose to adapt the features pulled from the regions returned by the Region Proposal Network (RPN). We argue that, since the RPN is trained on the source domain, the proposals from the target images may be wrongly detected and the local features used for the adaptation may be outside the target object features set (bottom

center in Fig. 1). In this paper, we propose to adapt the RPN in order to ensure the features extracted from the target images to overlap with the source object features. A local adaptation through adversarial learning will thus better align source and domain features (bottom right in Fig. 1).

Our contributions are threefold: 1) We present a new viewpoint about the domain shift problem in object detection. 2) We propose to adapt the RPN as a global feature adaptation and integrate this new adaptation module in Faster R-CNN. 3) We run extensive experiments in two different applications contexts: autonomous driving and ski lift video surveillance.

2 Related Work

Object Detection The first approaches proposed in the context of CNN were based on the region pooling principle [21,8]. In R-CNN [8], candidate regions detected by selective search were represented by a subset of pooled features and evaluated by an instance classifier. This two-stages principle was further refined in Faster R-CNN [18] with a common CNN backbone to extract the whole image features and two different sub-pipelines: the first one called RPN to generate proposals of regions which are likely to contain objects and the second one which is basically a classification and regression network aiming to refine the location and size of the object and to find its class. Besides these two-stages approaches, one-stage approaches directly predict box location, size and class in a single pipeline either by using anchor boxes with different aspect ratios [13] or by solving a regression problem on the feature grid [17]. Interested readers can refer to the review of recent advances in object detection in [1]. Since Faster R-CNN [18] provides very accurate results and has been largely studied, we propose to consider this network as a baseline in this paper.

Domain Adaptation Unsupervised domain adaptation is needed when we want to learn a predictor in a target domain without any annotated training samples in this domain [14,23]. Obviously, annotations are available in a source domain which is supposed to be close to the target one. Two main types of methods have been proposed in this context. The first one is to try to match the feature distribution in the source and target domains either by finding a transformation between the domains [15] or by directly adapting the features [10]. One noticeable example is the gradient reversal layer approach proposed by Ganin et al. [5] that attempts to match source and target feature distributions. They propose to jointly optimize the class predictor and the source-target domain disparity by back-propagation. The second type of methods relies on Generative Adversarial Networks (GANs) [11]. The principle is to generate annotated synthetic target images from the source images and to learn (or fine-tune) the network on these synthetic target data [12].

Domain adaptation for object detection Few works consider domain adaptation for object detection particularly in the unsupervised setting. [16] proposes class-specific subspace alignment to adapt RCNN [8] and [3] uses adversarial training inspired by [5] to adjust features at two different levels of a Faster

R-CNN architecture. The adaptation at image level intends to eliminate the domain distribution discrepancy at the output of the backbone network while the instance level adaptation concerns the features which are pooled from a Region of Interest (RoI), before the final category classifiers. Following the same adversarial training approach, Saito et al. [19] argue that a global matching may hurt performance for large domain shifts. They thus propose to combine a strong alignment of local features and a weak alignment of global ones. To the best of our knowledge, none of the previous works considers the adaptation of the region proposal sub network of Faster R-CNN. They are then sensitive to any shift in the distribution of object bounding boxes between source and target domains.
In this work, we propose to incorporate two adversarial domain adaptation modules in Faster R-CNN: the first one at RPN-level to address the source-target domain shift of features of the region proposal module and the second one at instance-level to adapt the RoI-pooled features used in the final classification module.

3 Our approach

In order to explain our adaptation scheme, we have to explain in details the work-flow of Faster R-CNN [7], summarized in Fig. 2. Then, we present our approach to adapt this detector between different domains.

3.1 Faster R-CNN

Faster R-CNN is basically composed of two convolutional blocks called C_1 and C_2, providing two feature maps F_1 and F_2, respectively (cf. Fig. 2). Based on F_2, the RPN predicts a set of box positions used to crop the F_1 feature map using the RoI pooling layer (called RP layer, hereafter).

It is worth mentioning that the gradient can not be back-propagated through the RP layer towards the RPN, because this step is not differentiable. The authors of Faster R-CNN resort to an alternating training to cope with this problem [7]. It is crucial to understand this point when one wants to apply domain adaptation to Faster R-CNN. It means that we can not just plug a domain adaptation module after the last layers of Faster R-CNN (namely F_{3i}) and adapt in one shot the classification layers and the convolution blocks C_1 and C_2.

Back to the workflow of Faster R-CNN, the outputs F_{1i}, $i = 1, ..., N_p$, of the RP layer are cropped and resized parts of the F_1 feature map. N_p is the number of proposals returned by the RPN. The feature maps F_{1i} are then sent to shared fully connected layers FC_3 whose outputs F_{3i} are used to take the final decision of class and location.

From this workflow, we note that the classification and regression layers take as inputs either F_2 or F_{3i}, which are the key feature maps of the detector. In the next section, we present how these feature maps can be adapted between the two domains.

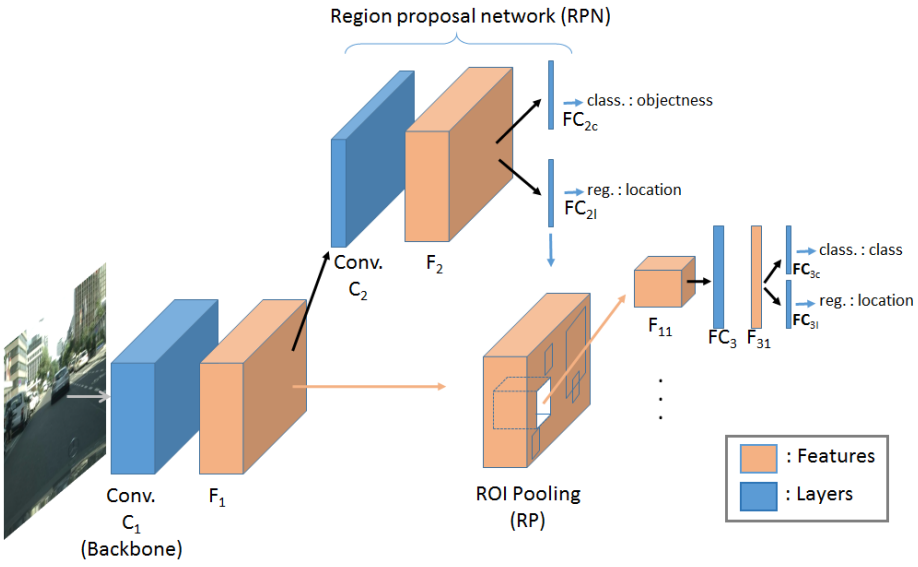

Fig. 2: Faster R-CNN Workflow.

3.2 Adapting Faster R-CNN

Let us consider a source domain $\mathcal{S}$ with $N_{\mathcal{S}}$ images $\{I_i^{\mathcal{S}}\}$, $i = 1, ..., N_{\mathcal{S}}$, each containing $n_i^{\mathcal{S}}$ objects, located at the positions $l_{ij}^{\mathcal{S}}$ and associated with the classes $c_{ij}^{\mathcal{S}}$, $j = 1, ..., n_i^{\mathcal{S}}$. Likewise, we denote $\mathcal{T}$ a target domain constituted of $N_{\mathcal{T}}$ target images $\{I_i^{\mathcal{T}}\}$, $i = 1, ..., N_{\mathcal{T}}$, each containing $n_i^{\mathcal{T}}$ objects, located at the positions $l_{ij}^{\mathcal{T}}$ and associated with the classes $c_{ij}^{\mathcal{T}}$, $j = 1, ..., n_i^{\mathcal{T}}$.

If the two domains are different (cameras, viewpoints, weather conditions,...), there exists a domain shift between the joint distributions $P(I^{\mathcal{S}}, l^{\mathcal{S}}, c^{\mathcal{S}})$ and $P(I^{\mathcal{T}}, l^{\mathcal{T}}, c^{\mathcal{T}})$. In this case, we can not train the detector on the source data and obtain good results on the target data, without adaptation. The aim of domain adaptation is to decrease this distribution discrepancy so that $P(I^{\mathcal{S}}, l^{\mathcal{S}}, c^{\mathcal{S}}) \approx P(I^{\mathcal{T}}, l^{\mathcal{T}}, c^{\mathcal{T}})$. In the context of unsupervised domain adaptation, the labels (locations and classes) of the target data are not available and this is not an easy task to decrease the joint distribution discrepancy. By applying the Bayes' rule on the joint distribution, we obtain, for the source domain:

$$P(I^{\mathcal{S}}, l^{\mathcal{S}}, c^{\mathcal{S}}) = P(l^{\mathcal{S}}, c^{\mathcal{S}} | I^{\mathcal{S}}) P(I^{\mathcal{S}}) \tag{1}$$

Most of the domain adaptation approaches assume a covariate shift, which means that the shift between the source and target joint distributions is caused by the marginal distributions $P(I)$, while the conditional distributions $P(l, c|I)$ are constant across domains, i.e. $P(l^{\mathcal{S}}, c^{\mathcal{S}} | I^{\mathcal{S}}) = P(l^{\mathcal{T}}, c^{\mathcal{T}} | I^{\mathcal{T}})$. Under this assumption, in order to decrease the joint distribution discrepancy, we have just to decrease the marginal distribution shift, so that $P(I^{\mathcal{S}}) \approx P(I^{\mathcal{T}})$. In order to change the

marginal distributions of the images, the classical approaches apply a transform T on the image features, so that $P(T(I^{\mathcal{S}})) \approx P(T(I^{\mathcal{T}}))$. Usually, the transform T is a part of a convolution neural network.

In this paper, we propose to consider and adapt different feature maps extracted from the images. By looking at Fig. 2, we note that two feature maps are used as input for classification and regression layers, namely the F_2 feature map and the F_{3i} feature vectors. So, in order to adapt the detector to the source domain, we have to adapt the marginal distributions of F_2 and F_{3i}, so that $P(F_2^{\mathcal{S}}) \approx P(F_2^{\mathcal{T}})$ and $P(F_{3i}^{\mathcal{S}}) \approx P(F_{3i}^{\mathcal{T}})$.

In order to enforce these distributions to be closer, we propose to resort to an adversarial domain adaptation approach [5] called GRL for gradient reversal layer. Note that any other adversarial domain adaptation algorithms could have been used, we just use this one for a fair comparison with DA-Faster [3]. When plugged on a feature map F_k, the idea of GRL is to minimize the discrepancy between the feature distributions over the source and target domains $P(F_k^{\mathcal{S}})$ and $P(F_k^{\mathcal{T}})$ [5]. If the GRL is able to perfectly overlap these two distributions, we can conclude that the features extracted at this point of the network (F_k) are domain invariant and so can be applied either on the source or target domain with equivalent accuracies.

From the previous analysis, it is obvious that two GRL modules should be inserted in the detector: one after the feature map F_2 and one after the feature vector F_{3i}. It is worth mentioning that, when we plug a GRL module to a feature map, we back-propagate the (reverse-)gradient until the first layer of the C_1 convolutional block. Thus, the main advantage of our approach is that the reversal gradients are back-propagated through all the layers of the detector. Consequently, the backbone, the RPN and the local features are all adapted (see Fig. 3).

Formally, at training time, the total loss corresponding to a given training image $I_k \in I^{\mathcal{S}} \cup I^{\mathcal{T}}$ from domain $d_k \in \{\mathcal{S}, \mathcal{T}\}$ is given by:

$$L = L_{Fst} - \lambda \sum_{i,j} L_H \left(FC_{2a}(F_2^{i,j}(I_k)), d_k \right) - \lambda \sum_{i=1}^{N_p} L_H \left(FC_{3a}(F_{3i}(I_k)), d_k \right) \quad (2)$$

where L_{Fst} denotes the original Faster R-CNN loss activated only if $I_k \in I^{\mathcal{S}}$, L_H denotes the cross-entropy loss, λ denotes the trade-off parameter to balance Faster R-CNN loss and domain adaptation losses, FC_{2a} and FC_{3a} denote the fully connected predictors for domain adaptation, $F_2^{i,j}(I_k)$ denotes the feature vector at location (i, j) of feature map F_2 for image I_k, and $F_{3i}(I_k)$ denotes the feature vector corresponding to the proposal region i of image I_k.

We note that the recent domain adaptive detection approaches ([3,19]) have not tried to adapt the RPN layer and we think that this is a strong weakness of these approaches. Indeed, as mentioned in [3] (called DA-Faster hereafter), the image-level adaptation is enforcing the F_1 target and source feature distributions to be closer but it is very hard to perfectly align them. This is one of the reasons why DA-Faster approach also applies instance level adaptation. But, it is clear in Fig. 2, that if F_1 features are not well adapted between the domains, the output

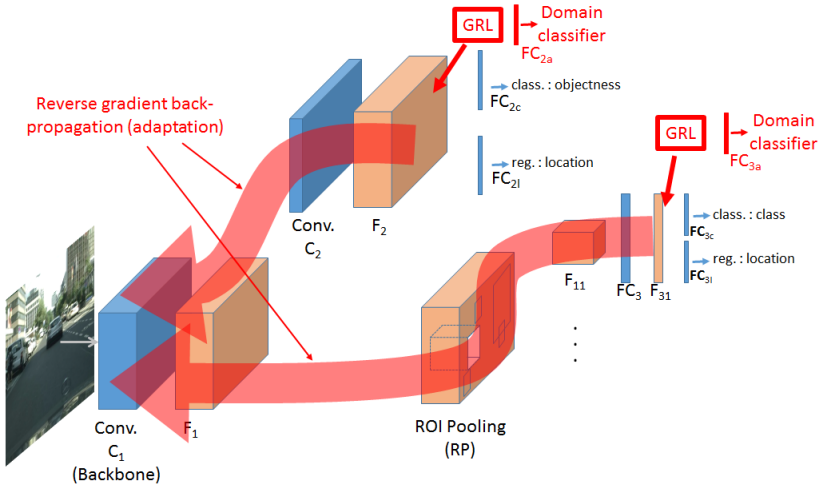

Fig. 3: Our domain adaptation for Faster R-CNN. See text for details.

of the RPN will also be different between the domains and consequently, the locations where the boxes F_{1i} are cropped from F_1 will be domain dependent. Therefore, the instance-level adaptation on F_{3i} features will not help to adapt the object detector between domains, since it will work on local features which are not equivalent between the domains (see Fig. 1).

4 Experiments

4.1 Experiment Setup

To evaluate the efficiency of our approach we conduct experiments in two contexts: autonomous driving and video surveillance of ski lifts. In each case, we train on a source dataset and test on a target dataset from a different domain. During training, likewise the other domain adaptive approaches, we use also images from the target domain, but without any label, while the source dataset images are provided with their bounding boxes instance annotations.

The baseline is Faster R-CNN model trained only on the source dataset. As mentioned earlier, our solution is inspired from DA-Faster [3] but our contribution is in the analysis of the domain shift in Faster R-CNN, conducting to the solution that the domain adaptation module (GRL) should be plugged at the RPN level. Consequently, the aim of these experiments is to compare DA-Faster with our approach in order to check the validity of our contribution in practice. Thus, adapting the RPN in other solutions such as [19], or using other adaptation modules than GRL such as [14,23] might have provided better results, but it is out of the scope of this paper. Thus, for all the experiments, we compare our approach with DA-Faster [3].

As mentioned in [19], the results provided by the authors of DA-Faster are unstable and Saito et al. proposed to re-implement their own code for DA-Faster, conducting to lower results than the original paper [3]. So likewise [19], we report the results of DA-Faster with the implementation provided by [9] with the same hyper-parameters as our solution (results denoted *DA-Faster* hereafter), as well as the results provided by the original paper [3] (denoted *DA-Faster**), when available on the considered dataset.

To evaluate object detection we report the mean Average Precision (mAP) with intersection over union (IoU) threshold at 0.5 (denoted AP50), the mAP with IoU threshold of 0.75 (AP75) and the mAP averaged over multiple IoU from 0.5 to 0.95 with a step size of 0.05 (APcoco). The network is trained in an end-to-end manner using back-propagation and the stochastic gradient descent (SGD) algorithm. As a standard practice, Faster R-CNN backbone is initialized with pre-trained weights on ImageNet classification. We use a learning rate of 0.001 for 50k iterations, and 0.0001 for the next 20k iterations. Each iteration has 2 mini-batches, one from source domain and the other from target domain. The trade-off parameter λ to balance Faster R-CNN loss and domain adaptation loss is set to 0.1 as in [3]. We use a momentum of 0.9 and a weight decay of 0.0005.

4.2 Autonomous driving

In this context we evaluate the domain adaptive detectors for two domain shifts: weather conditions (foggy and not foggy) and acquisition conditions (different cameras, different viewpoints and different scenes).

Fig. 4: One image from each dataset: the Cityscapes dataset (left), its foggy version (center) and the KITTI dataset (right).

Cityscapes → Foggy Cityscapes. In the first experiment we use the Cityscapes dataset [4] as source domain. It is a urban scene dataset with 2975 training images and 500 validation images. The 1525 unlabeled images are not considered. For training the network, we are using the 2975 train images and do not consider the validation images. There are 8 categories with instance annotations in this dataset, namely *person, rider, car, truck, bus, train, mortorcycle* and *bicycle*. The target domain is the Foggy Cityscapes [20] dataset generated by applying fog synthesis on the Cityscapes dataset to simulate fog on real scenes (see Fig. 4). Thus, the number of images and labels are exactly the same as for Cityscapes dataset. For testing the detection, we are using the 500 validation images from Foggy Cityscapes. The results are summarized in Table 1. First, we can note

that, without domain adaptation, the results of Faster R-CNN are very bad, underlying the strong need of adapting the network in case of weather condition variations. Thus, DA-Faster improves the results over Faster R-CNN, but we note that our approach clearly outperforms DA-Faster on this dataset, showing that the RPN adaptation helps in adapting the detector in case of weather condition variations.

Table 1: Detection results on Foggy Cityscapes (trained on Cityscapes dataset). The AP50 is reported for each class as well as the average APcoco, AP50 and AP75 over all classes.

	person	rider	car	truck	bus	train	mcycle	bicycle	APcoco	AP50	AP75
Faster R-CNN	18.8	20.5	24.2	17.0	8.0	6.2	7.2	5.0	06.20	13.35	05.42
DA Faster R-CNN	27.3	35.7	44.1	20.3	35.2	8.9	16.2	23.6	12.28	26.41	10.02
DA Faster R-CNN*	25.0	31.0	40.5	22.1	35.3	20.2	**20.0**	27.1	-	27.6	-
Ours	**27.8**	**35.8**	**45.1**	**23.5**	**42.1**	**26.1**	18.0	**27.6**	**13.70**	**30.47**	**11.04**

Cityscapes → KITTI. In this experiment Cityscapes is the source domain, and KITTI [6] is the target domain (see Fig. 4). KITTI is a benchmark for autonomous driving which consists of 7481 training images. Since the test set is not annotated we use all the training images with their annotations at test time to evaluate the performance. Only one category (*car*) is annotated in KITTI, so we consider this single class for evaluation. The results are summarized in Table 2. Once again, we note that the domain adaptation helps improving Faster R-CNN results. We see also that our approach outperforms DA-Faster for all the criteria when using the same hyper-parameters. The results provided in [3] are better than ours for $AP50$, but note that the implementation and hyper parameters are different from our tests. The comparison is therefore not fair.

Table 2: Detection results in KITTI training set (trained in Cityscapes dataset) for one class (Car) detection.

	APcoco	AP50	AP75
Faster R-CNN	26.73	58.60	21.54
DA Faster R-CNN	27.51	60.38	22.67
DA Faster R-CNN*	-	**64.1**	-
Ours	**28.39**	61.32	**23.59**

4.3 Video surveillance of ski lifts

The MIVAO research project was launched in collaboration with a french start-up Bluecime, based on the needs of ski lift operators to secure chairlifts. MIVAO aims to develop a computer vision system that acquires images from the boarding station of chairlifts, analyzes the important elements (people, chairlift carrier, safety bar, ...) and triggers an alarm in case of dangerous situations. In this paper,

we tackle this problem as an object detection task trying to detect the safety bar in the image, considering that it has to be closed when the chairlift leaves the boarding station. Across the ski resorts, the viewpoint, the background, the carrier geometry and the camera may be different and domain adaptive detectors are required to install new systems without a fastidious and time-consuming step of manual annotation.

Chairlift dataset For this experiment, we have created a dataset with images from two different chairlifts, called hereafter chairlift 1 and chairlift 2. The dataset contains 3864 images from chairlift 1 and 4260 images from chairlift 2. Example images are provided in Fig. 5. We can note that the main differences between the two chairlifts are in the viewpoints which are slightly different and in the presence of a cluttered background in the chairlift 2.

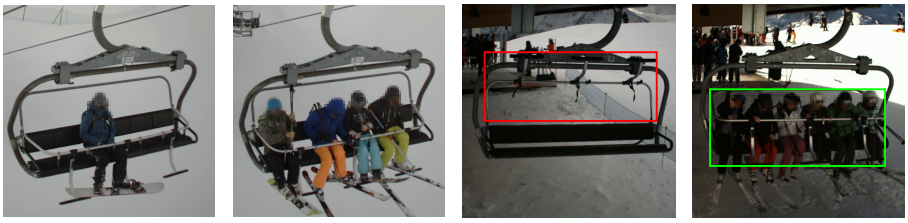

Fig. 5: Example images from our chairlift dataset. The two left images are from chairlift 1 and the two right images are from chairlift 2. The box annotations (open:red and close:green) are provided for the two right images, for illustration.

The images are centered on the chairlift and manually labeled with the position and the dimensions of the bounding box containing the safety bar. From this information, we have created instance annotations with two categories: open safety bar and close safety bar, as illustrated on the two right images from Fig. 5.

Evaluation The results are provided in Table 3. By training the baseline Faster R-CNN using images from one chairlift and test it on images from another chairlift, the results were surprisingly very good in terms of AP50. This can be explained by the important size of the ground truth bounding boxes that have high chance to well overlap random bounding boxes with similar dimensions. Obviously, when looking at the more demanding criteria such as $AP coco$ or $AP75$, the need of domain adaptation is evident for precise object detection. The results show that the two domain adaptive detectors (DA-Faster and ours) are equivalent for the adaptation from chairlift 1 to chairlilft 2, but they also show that our adaptation is much better than DA-Faster for the adaptation from chairlift 2 to chairlift 1. It is difficult to explain why DA-Faster is less accurate in one direction ($ch2 \rightarrow ch1$) than in the other direction ($ch1 \rightarrow ch2$). One assumption could be that in DA-Faster, the RPN is better trained on chairlift 1 since in this case the background is less cluttered. Thus, when applying it on chairlift 2, the adaptation process tends to promote features from the foreground

and both the proposal and the classification are good. On the contrary, if the RPN is trained on chairlift 2, it will rely on cluttered features which are removed with the global adaptation and thus, for DA-Faster, the proposals will be bad on chairlift 1, leading to an important residual domain shift in the results. On the contrary, in our method, since the RPN is directly adapted, the residual shift is lower (see figure 1 and the related explanation in section 1).

Table 3: Detection results on the chairlift dataset. First, adaptation from chairlift 1 to chairlift 2, and second adaptation from chairlift 2 to chairlift 1.

	$ch1 \rightarrow ch2$			$ch2 \rightarrow ch1$		
	APcoco	AP50	AP75	AP	AP50	AP75
Faster R-CNN	30.34	99.49	0.30	36.56	98.98	9.86
DA Faster R-CNN	50.51	99.50	**33.4**	42.56	98.99	11.1
Ours	**50.93**	**99.99**	30.7	**48.83**	**99.00**	**45.6**

5 Conclusion

In this paper, we have tackled the problem of domain adaptation for object detection. After a detailed analysis of the complete workflow of the classical Faster R-CNN detector, we have proposed to adapt the features pulled from this network at two different levels: one adaptation at a global level in the Region Proposal Network and one adaptation at the local level for each bounding box returned by the RPN. We have shown that these two adaptations are complementary and provide very good detection results. We have tested our solution on two different applications, namely the autonomous driving and the chairlift security. As future works, we propose to test more accurate adaptation procedures such as the approaches presented in [14,23]. These methods could help in the learning step to reach stable solutions which is a strong weakness of the domain adaptive Faster R-CNN. Furthermore, it could be interesting to adapt the features at different depth of the network as recommended by [19].

References

1. Agarwal, S., Terrail, J.O.D., Jurie, F.: Recent advances in object detection in the age of deep convolutional neural networks. arXiv preprint arXiv:1809.03193 (2018)
2. Bascol, K., Emonet, R., Fromont, E., Debusschere, R.: Improving chairlift security with deep learning. In: International Symposium on Intelligent Data Analysis. pp. 1–13. Springer (2017)
3. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3339–3348 (2018)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)

5. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495 (2014)
6. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
7. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
9. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. https://github.com/facebookresearch/detectron (2018)
10. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
12. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5001–5009 (2018)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
14. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1640–1650 (2018)
15. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks **22**(2), 199–210 (2010)
16. Raj, A., Namboodiri, V.P., Tuytelaars, T.: Subspace alignment based domain adaptation for rcnn detector. arXiv preprint arXiv:1507.05578 (2015)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
19. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (2019)
20. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision pp. 1–20 (2018)
21. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
22. Singh, B., Davis, L.S.: An analysis of scale invariance in object detection snip. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3578–3587 (2018)
23. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)