# Every Pixel Matters: Center-aware Feature Alignment for Domain Adaptive Object Detector

Cheng-Chun Hsu[1], Yi-Hsuan Tsai[2], Yen-Yu Lin[1,3], and Ming-Hsuan Yang[4,5]

[1]Academia Sinica [2]NEC Labs America
[3]National Chiao Tung University [4]UC Merced [5]Google Research

## 1 Overview

In this supplementary material, we provide more detailed analysis of our approach, including 1) the sensitivity analysis of the weights for the adversarial terms, *i.e.*, $\alpha$ and $\beta$ in equation (1) of the paper, 2) t-SNE visualizations of aligned features, 3) introducing an alternative scheme for center-aware alignment and the comparisons, 4) ablations of the objectness map and centerness map for CA-alignment, 5) ablations of the multi-scale alignment using VGG-16, 6) performance gains comparison of anchor-based and anchor-free detector, 7) the results of weather adaptation using ResNet-50, 8) computational cost in GA- and CA-alignment, and 9) more visualization examples of our predicted results on the benchmark datasets.

## 2 Sensitivity Analysis

The sensitivity analysis of weights $\alpha$ and $\beta$ is reported in Table 1. The first group shows the result of the setting adopted in our paper, *i.e.*, $\alpha = 0.01$ and $\beta = 0.1$. The second group displays the results of varying the value of $\beta$, *i.e.*, $\beta \in \{0.02, 0.05, 0.2, 0.5\}$, while fixing $\alpha$ to 0.01. The third group gives the results of varying the value of $\alpha$, *i.e.*, $\alpha \in \{0.002, 0.005, 0.02, 0.05\}$, while fixing $\beta$ to 0.1.

By fixing $\alpha$, a small value of $\beta$ slightly decreases performance. The performance drops if the value of $\beta$ becomes too large since it may mitigate the effects of global alignment. By fixing $\beta$, increasing or decreasing the value of $\alpha$ leads to a moderate drop in performance, but is still better than the ones only using the GA module (see Table 3 of the manuscript). The results suggest that the GA module is essential to our method, and the CA and GA modules complement each other since combining them leads to better performance.

## 3 Aligned Feature Distribution

This section provides the visualization of feature distributions after applying alignment with the proposed method. We first sample per-pixel features from source and target domains, and then we visualize them using t-SNE in Fig. 1. We observe that foreground and background features without applying feature

**Table 1.** Sensitivity analysis of weights $\alpha$ and $\beta$ using ResNet-101.

| Sim10k → Cityscapes | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha$ | $\beta$ | mAP | $\text{mAP}^r_{0.5}$ | $\text{mAP}^r_{0.75}$ | $\text{mAP}^r_S$ | $\text{mAP}^r_M$ | $\text{mAP}^r_L$ |
| 0.01 | 0.1 | 28.6 | 51.2 | 27.4 | 7.1 | 30.2 | 58.3 |
| 0.01 | 0.02 | 28.2 | 51.3 | 26.7 | 6.7 | 29.7 | 58.4 |
| 0.01 | 0.05 | 28.4 | 51.3 | 28.4 | 6.3 | 30.7 | 57.5 |
| 0.01 | 0.2 | 28.2 | 50.2 | 27.3 | 6.0 | 29.8 | 59.1 |
| 0.01 | 0.5 | 25.8 | 48.6 | 24.8 | 5.9 | 27.2 | 53.9 |
| 0.002 | 0.1 | 27.4 | 50.4 | 26.8 | 7.1 | 28.6 | 56.1 |
| 0.005 | 0.1 | 26.9 | 50.1 | 26.5 | 7.0 | 27.6 | 55.7 |
| 0.02 | 0.1 | 27.7 | 50.3 | 27.3 | 6.2 | 28.8 | 58.5 |
| 0.05 | 0.1 | 27.8 | 49.9 | 27.7 | 6.1 | 29.9 | 57.1 |

alignment are clustered together and thus it leads to inaccurate prediction on the target domain. After applying our alignment, there is a clear separation for feature distributions between foreground and background pixels.

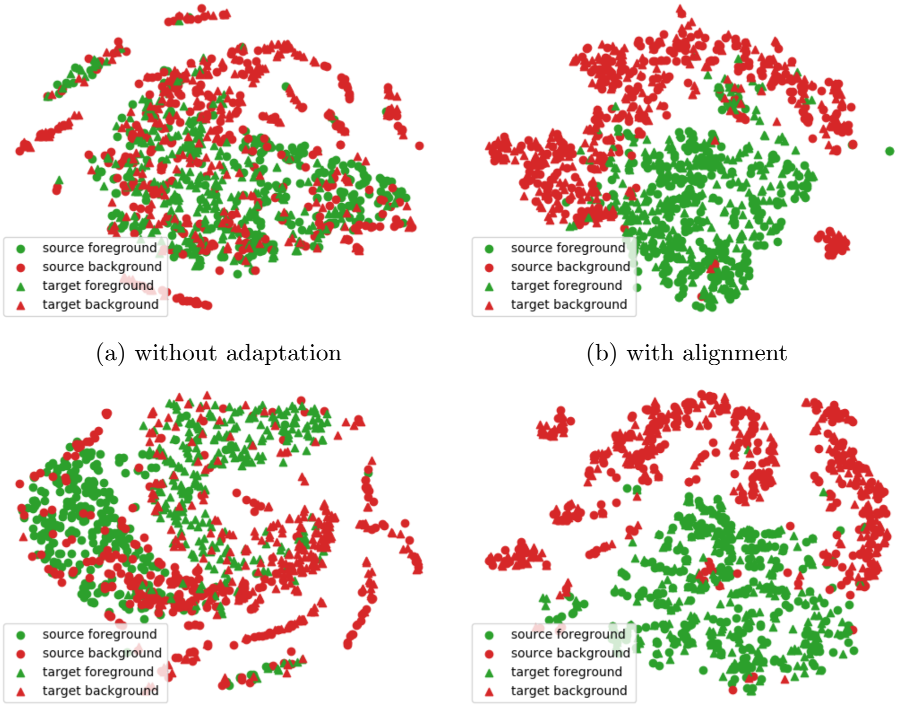
## 4 CA Feature v.s. CA Loss

In this section, we propose an alternative scheme of the center-aware map for alignment and compare it with the original scheme proposed in the paper. Note that we follow all notations used in Section 3 of the paper.

The proposed center-aware alignment described in the paper is denoted by *Center-aware Feature*, because we compute the dot product between the center-aware map and input features during center-aware alignment. In this study, we provide an alternative called *Center-aware Loss*, which also performs center-aware alignment and utilizes the information provided by the center-aware map. However, in this alternative, the center-aware map is utilized in the loss function, which is applied to each location of the map. That is, *Center-aware Loss* estimates the importance of each pixel for alignment by referring to the center-aware map. Accordingly, the discriminator loss for *Center-aware Loss* can be modified as follows:

$$\begin{aligned}\mathcal{L}_{CA}(I_s, I_t) = -\sum_{u,v} & M_{CA}^{(u,v)} z\, log(D_{GA}(F_s)^{(u,v)}) \\ & + M_{CA}^{(u,v)}(1-z) log(1 - D_{GA}(F_t)^{(u,v)}). \end{aligned} \tag{1}$$

As shown in Table 2, *Center-aware Feature* and *Center-aware Loss* both achieve competitive results. It indicates that the effectiveness of center-aware alignment can be carried out in both schemes.



(a) without adaptation

(b) with alignment

(c) without adaptation

(d) with alignment

**Fig. 1.** Visualization of feature distributions. (a) and (b) show the features sampled from Sim10k → Cityscapes, while (c) and (d) are for Cityscapes → Foggy Cityscapes.

## 5 Objectness Map v.s. Centerness Map

We provide an ablation using ResNet-101 to analyze the effectiveness of the objectness map $M_{obj}$ and the centerness map $M_{ctr}$ in Table 3. Compares to the baseline, *i.e.*, ours (w/o adapt.), adding either $M_{obj}$ or $M_{ctr}$ improves the performance, which shows the effectiveness of the proposed alignment. Moreover, combining the two maps obtain the best result.

## 6 Multi-scale Alignment

In addition to Table 5 of the manuscript, we present more results using VGG-16 in Table 4. We show that both center-aware alignment (comparing a. with c.) and multi-level alignment (comparing b. with c.) help achieve performance gains. Note that, adding multi-level alignment also involves center-aware alignment, in which its contribution is consistent with the main idea of the paper. Also, our single-level results are better than SC-DA [3]. In the end, we present another

**Table 2.** Comparison of CA Feature and CA Loss using ResNet-101.

| | Sim10k → Cityscapes | | | | | |
|---|---|---|---|---|---|---|
| Methods | mAP | $\mathrm{mAP}^r_{0.5}$ | $\mathrm{mAP}^r_{0.75}$ | $\mathrm{mAP}^r_S$ | $\mathrm{mAP}^r_M$ | $\mathrm{mAP}^r_L$ |
| CA Feature | 28.6 | 51.2 | 27.4 | 7.1 | 30.2 | 58.3 |
| CA Loss | 28.6 | 50.9 | 27.1 | 6.8 | 30.2 | 59.0 |

**Table 3.** Ablation study of the objectness map $M_{obj}$ and the centerness map $M_{ctr}$ using ResNet-101.

| | Sim10k → Cityscapes | |
|---|---|---|
| Method | mAP | $\mathrm{mAP}^r_{0.5}$ |
| Ours (w/o adapt.) | 23.1 | 41.1 |
| Ours (w/ $M_{obj}$) | 25.3 | 50.4 |
| Ours (w/ $M_{ctr}$) | 26.1 | 49.8 |
| Ours (w/ $M_{ctr} + M_{obj}$) | **26.8** | **51.1** |

**Table 4.** Ablation study of the proposed center-aware alignment and multi-scale alignment using VGG-16.

| | Cityscapes → Foggy | Sim10k → Cityscapes | KITTI → Cityscapes |
|---|---|---|---|
| Method | $\mathrm{mAP}^r_{0.5}$ | $\mathrm{mAP}^r_{0.5}$ | $\mathrm{mAP}^r_{0.5}$ |
| SC-DA [3] CVPR'19 | 33.8 | 43.0 | 42.5 |
| a. Ours (GA, multi-level) | 33.2 | 45.9 | 39.1 |
| b. Ours (GA+CA, single-level) | 33.9 | 45.3 | 42.9 |
| c. Ours (GA+CA, multi-level) | **36.0** | **49.0** | **43.2** |

**Table 5.** Ablation study of the proposed multi-scale alignment using VGG-16.

| | Sim10k → Cityscapes |
|---|---|
| Aligned Scale | $\mathrm{mAP}^r_{0.5}$ |
| w/o adapt. | 41.8 |
| $F^5$ | 45.3 |
| $F^3 \sim F^5$ | 47.1 |
| $F^5 \sim F^7$ | 47.2 |
| $F^3 \sim F^7$ | 49.0 |

ablation study of multi-scale features (similar to Table 5 of the manuscript) using VGG-16 in Table 5, which shows the effectiveness of multi-level alignment.

## 7 Anchor-based v.s. Anchor-free Detector

Consider that the anchor-based and anchor-free detectors might have different characteristics, we report the oracle results of F-RCNN and performance gains compared to the baselines in Table 6 to better analyze the performance. First,

**Table 6.** Performance gains compared to the baseline using the anchor-based and anchor-free detector. The first number in the parentheses is the performance gain compared to the baseline, while the second number in the parentheses is the performance difference compared to the oracle.

| | Cityscapes → Foggy | Sim10k → Cityscapes | KITTI → Cityscapes |
|---|---|---|---|
| Method | $\text{mAP}^r_{0.5}$ | $\text{mAP}^r_{0.5}$ | $\text{mAP}^r_{0.5}$ |
| F-RCNN (w/o adapt.) | 18.8 | 30.1 | 30.2 |
| SC-DA [3] CVPR'19 | 33.8 (+15, -9.4) | 43.0 (+13, -26.4) | 42.5 (+12.8, -26.9) |
| MAF [2] ICCV'19 | 34.0 (+15.2, -9.2) | 41.1 (+11, -28.3) | 41.0 (+11.2, -28.3) |
| F-RCNN (oracle) | 43.2 | 69.4 | 69.4 |
| Ours (w/o adapt.) | 18.4 | 39.8 | 34.4 |
| Ours | 36.0 (+17.6, -5.5) | 49.0 (+9.2, -20.7) | 43.2 (+8.8, -26.5) |
| Ours (oracle) | 41.5 | 69.7 | 69.7 |

**Table 7.** Results of adapting Cityscapes to Foggy Cityscapes using ResNet-50. Note that results of each class are evaluated in $\text{mAP}^r_{0.5}$.

| | | Cityscapes → Foggy Cityscapes | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Backbone | person | rider | car | truck | bus | train | mbike | bicycle | $\text{mAP}^r_{0.5}$ |
| Ours (w/o adapt.) | ResNet-50 | 33.8 | 30.7 | 40.3 | 15.7 | 27.0 | 5.4 | 17.0 | 27.6 | 24.2 |
| MTOR [1] CVPR'19 | | 30.6 | **41.4** | 44.0 | 21.9 | 38.6 | **40.6** | 28.3 | **35.6** | 35.1 |
| Ours (GA) | | 39.8 | 39.6 | 57.1 | 22.7 | 45.2 | 22.0 | 27.7 | 32.5 | 35.9 |
| Ours (CA) | | 39.2 | 40.3 | 57.1 | 27.0 | 45.6 | 35.1 | 26.1 | 34.6 | 38.1 |
| Ours (GA+CA) | | **39.9** | 38.1 | **57.3** | **28.7** | **50.7** | 37.2 | **30.2** | 34.2 | **39.5** |

despite that our baseline is better, our performance gain is still competitive with state-of-the-arts (the first number in the parentheses). Furthermore, our results are closer to the oracle results (the second number in the parentheses), which shows the potential of anchor-free approaches and could motivate future work on using anchor-free detectors for domain adaptive object detection.

## 8 Weather Adaptation using ResNet-50

To compare with MTOR [1], we report the results of adapting Cityscapes to Foggy Cityscapes using ResNet-50 in Table 7. After adaptation, our method (GA + CA) improves our baseline by 15.3% and outperforms MTOR [1] by 4.4% in terms of $\text{mAP}^r_{0.5}$.

## 9 Computational Cost in GA and CA

In Table 8, we show the MACs for our modules, given the input size as (666, 1332). Here, adding single-scale or multi-scale GA+CA does not significantly increase the computational cost. Also, GA+CA module is only required during

**Table 8.** Computational cost of the proposed global alignment and center-aware alignment using VGG-16.

| Module | MACs |
|---|---|
| GA+CA ($F^3 \sim F^7$) | 22.25G |
| GA+CA ($F^5$) | 1.04G |
| Detector | 188.03G |

training, while the computational cost is the same as the original detector during inference.

## 10 Qualitative Results

More visualization examples on different datasets are shown in Fig. 2 to Fig. 4. These examples demonstrate that the proposed center-aware alignment makes the model focus on discriminate areas of objects and produces more promising results. Moreover, the proposed method is robust to some difficulties for object detection, such as object overlapping, small objects, and occlusions.

For example, it can be observed that our method can handle the issue of overlaps effectively in Fig. 3(d) and Fig. 3(j). With the aid of the multi-scale alignment, our method is effective to detect small objects, as shown in Fig. 3(a) and Fig. 4(a). For occlusions in Fig. 3(b) and Fig. 3(h), our method can correctly detect the complete objects.

## 11 Failure Examples

We present some failure examples produced by our method in Fig. 5. In Fig. 5(a), noisy background leads to false positives. In Fig. 5(b), the crowded scene results in redundant predictions. In Fig. 5(c), the model falsely predicts a large object, *i.e.*, a truck, as multiple objects. In Fig. 5(d), due to the domain gap, our model fails to detect the cars reflecting the sunlight. Moreover, it fails to distinguish different instances of the same categories. In Fig. 5(e), the detector incorrectly identifies the sign as a car, which indicates that identifying the differences among small objects of different categories is challenging.

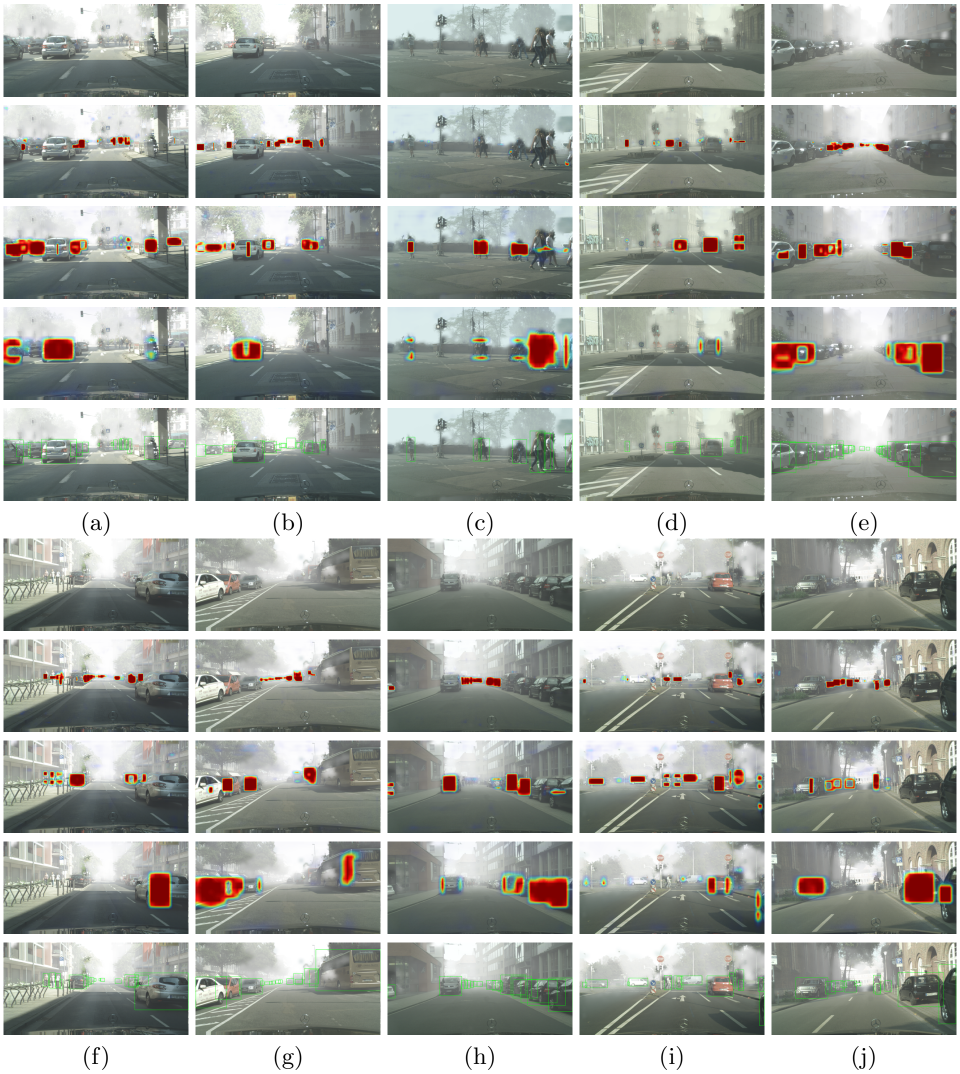


**Fig. 2. (a) $\sim$ (j) Ten examples of the response maps and detection results when adapting Cityscapes to Foggy Cityscapes.** From top to bottom for each example, the input images, the corresponding response maps extracted from the feature layers $F_3 \sim F_5$, and the corresponding detection results are shown, respectively.

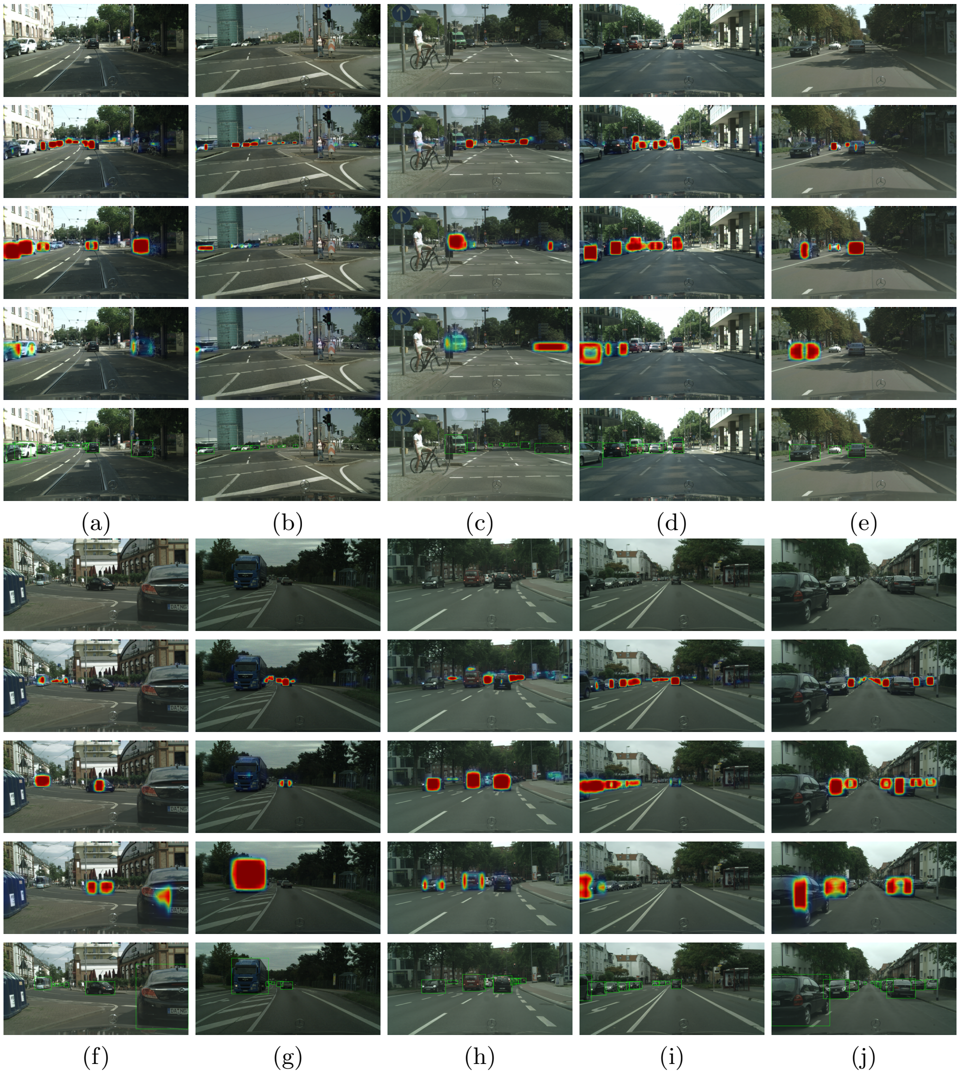


**Fig. 3. (a) $\sim$ (j) Ten examples of the response maps and detection results when adapting Sim10k to Cityscapes.** From top to bottom for each example, the input images, the corresponding response maps extracted from the feature layers $F_3 \sim F_5$, and the corresponding detection results are shown, respectively.

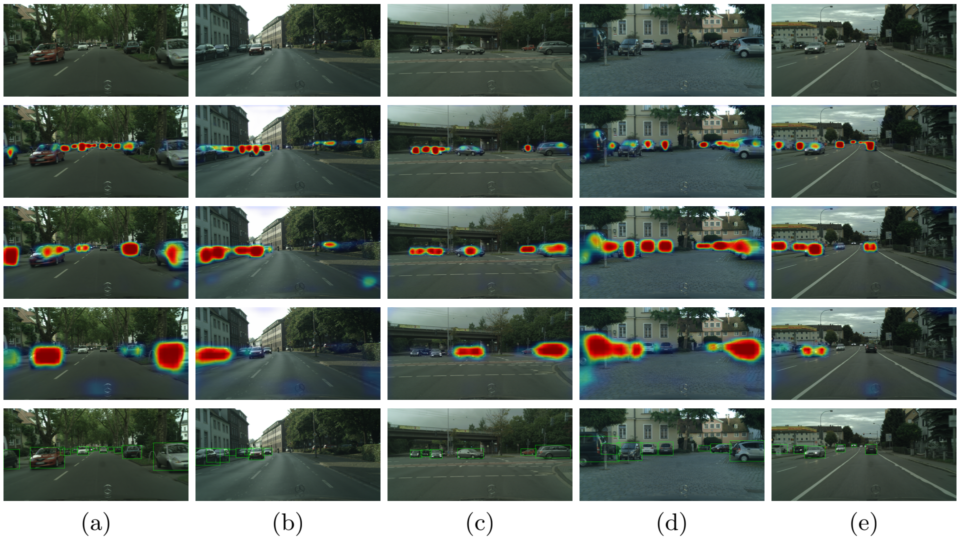


**Fig. 4. (a) $\sim$ (e) Five examples of the response maps and detection results when adapting KITTI to Cityscapes.** From top to bottom for each example, the input images, the corresponding response maps extracted from the feature layers $F_3 \sim F_5$, and the corresponding detection results are shown, respectively.

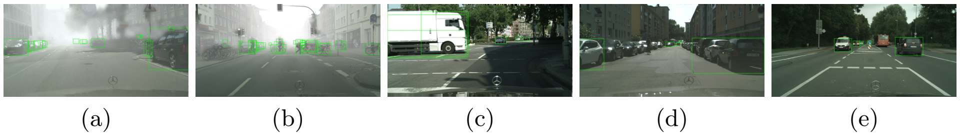


**Fig. 5.** Failure examples predicted by our method.

## References

1. Cai, Q., Pan, Y., Ngo, C.W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: CVPR (2019) 5
2. He, Z., Zhang, L.: Multi-adversarial faster-rcnn for unrestricted object detection. In: ICCV (2019) 5
3. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: CVPR (2019) 3, 4, 5