Geometric Proposals for Faster R-CNN

Sikandar Amin, Fabio Galasso
OSRAM GmbH
{s.amin, f.galasso}@osram.com

Abstract

*Faster R-CNN has established itself as the de-facto best object detector but it remains strongly limited in two aspects: (**i**) it is sensitive to background clutter and its classification performance decreases when it is confronted with more noisy proposals; (**ii**) it suffers when the objects vary largely in scale and specifically for the small objects.*

We address both issues with our geometric-proposals for Faster R-CNN (GP-FRCNN), whereby we re-rank the generic object proposals with an approximate geometric estimate of the scene. But the devil is in the details: the simple extension requires involved scale adjustments (e.g. anchors, layer resolution) which we detail in this paper. Finally, our GP-FRCNN performs equally well on smaller and larger objects, a long standing challenge for any object detection algorithm. The application of GP-FRCNN to surveillance videos is straightforward and does not require an explicit geometric formulation. We extensively test the model on the UA-DETRAC dataset, where GP-FRCNN outperforms the standard Faster R-CNN by 19%.

1. Introduction

The complex city environments are the largest challenge for autonomous driving. While the detection and recognition of objects from the car perspective remains far from the needed accuracy, the autonomous car companies often ameliorate it by the use of accurate maps which provide information on *e.g.* buildings, traffic signs, road-side, when registered to the car views. But the registration is challenging in the cities, *e.g.* the parked cars change at all times and the car view may be occluded by large trucks.

The story changes if we consider cameras mounted on the city infrastructure, such as by the street-light-poles. The infrastructure-based sensing may reach high accuracy because the static cameras are always aware of the city layout. Furthermore, as we show here, prior information about the camera-viewpoint allows reducing the computational load, without compromising performance. This makes infrastructure sensing a precious ally to autonomous cars, also because it allows the cars to see behind the corner.

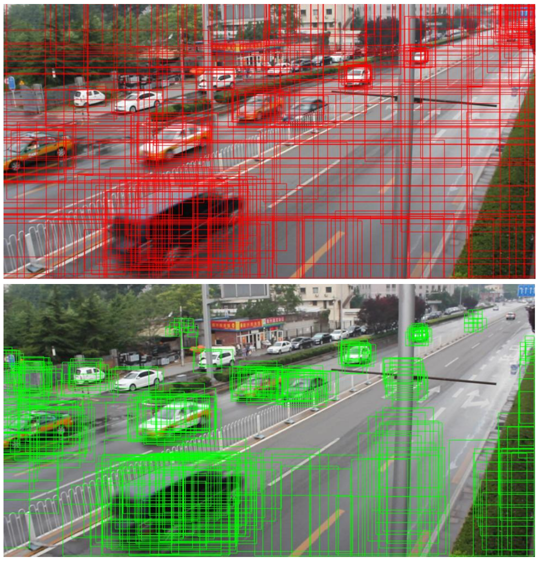

Figure 1. *Top:* Baseline RPN proposals. *Bottom:* Geometric proposals.

Here we address object proposals for surveillance cameras and extend the state-of-the-art Faster R-CNN to consider the scene geometry. Geometry-aware proposals help remove non-plausible detections and increase recall on small objects further away on the road. On the large UA-DETRAC [13] dataset, with a careful model re-design, we improve on the vanilla Faster R-CNN (VGG-16) by more than 19%. Moreover, we maintain the improvement while switching to a much faster VGG-M feature-network.

2. Related work

Our approach mostly relates to [3, 4] which generate 3D object proposals by exploiting the scene geometry from a driving car-viewpoint, while assuming calibrated monocular or stereo camera setup. By contrast, our work leverages the advantages of a static camera, *i.e.* the largest distance allows a simpler 2D formulation, which we cast as an online scale-estimate, without requiring any calibration.

978-1-5386-2939-0/17/$31.00 ©2017 IEEE

Scale invariance in object detection, *i.e.* ability to detect objects regardless of its scale, is a long standing problem in computer vision literature. Especially detection of small scale objects is inherently difficult. In the recent deep learning trend, numerous works have already been proposed to address this problem, *e.g.* SSD-Multibox [9] concatenates detections from various high-end convolutional layers, however recent analysis indicates its weakness on the small scale objects [7]. Other recent approaches such as FPN [8] and TDM [11] go a step further and propose to combine semantically weak high-resolution features with semantically strong low-resolution features, as a result achieving state-of-the-art accuracy.

The importance of high-resolution features has also been described in prior semantic segmentation literature, that addressed the estimation of accurate semantic object boundaries. A notable example is DeepLab [2], which proposed the removal of max-pooling layers for denser feature maps and adopted dilated convolutions to keep the context information (*aka* receptive field size). For improving the detection of people at small scales, Zhang *et al.* [14] also proposed the removal of pooling layers, but did not consider dilated convolutions. In this work, we show that such a strategy is also quite effective for detection of small scale vehicles, however it compromises the detection accuracy of large scale vehicles. Moreover, we show that employing dilated convolution is also unnecessary. We however propose *geometric proposals* which do a very convincing job to improve the scale invariance of the detector. Interestingly, FPN [8] style adjustments to the backbone convolutional architecture are complementary to our work, and we will consider this in the future.

3. Geometric Proposals

We extend Faster R-CNN [10], state-of-the-art in object detection, to incorporate our proposed geometric proposals. Geometric proposals encode the scene layout of a static camera in a simple and effective way.

In the following, we first briefly review Faster R-CNN, then we describe our geometric proposals module. The effective use of the geometric proposals module would require a fine adjustment of the Faster R-CNN model, as detailed in Section 4.

Faster R-CNN. The Faster R-CNN detector works in two stages. The first stage is the fully convolutional region proposal network (RPN), which takes the full image at the input and produces the class-agnostic object proposals. The second stage is the classification network (Fast-RCNN [5]), which classifies the incoming proposals into given object classes. The convolutional layers are shared for both tasks, *i.e.* generating proposals, and classifying them.

Our proposed extension regards the RPN, as we explain in the next section.

3.1. Geometric Extension to RPN

Intuitively, an image contains potentially a few large objects and more smaller ones. This is not reflected into the Faster R-CNN algorithm, esp. due to the RPN, which proposes an equal amount of objects across scales.

We re-establish a relationship between the proposals and the scene geometry by means of an object-scale estimate. First we detect most confident objects, then we estimate a per-pixel-scale estimate as a proxy to the true scene geometry, finally we prune the RPN proposals.

Initial detections. We use the most-confident (high scoring) *car* detections to automatically estimate this layout as shown in Fig. 2. Across a few initial seconds of a video sequence, we gather 10s of reliable detections, which suffice for a reliable scale estimate. Note that the most-confident detections are generally non-occluded and non-truncated.

Per-pixel-scale estimate. Next, we estimate the scale-layout for the image, i.e. an image scale function, which describes the scale of the object given its position in the image. In our case, this corresponds to the expected size of a vehicle in a certain image position.

Let us assume an initial set of detections at positions $\{x\}_1^N$, where x's are the center of detection bounding boxes and N is the number of initial detections. We target the estimation of a scale function $s(x)$ which represents the size of the object bounding box in $pixels^2$ at each pixel coordinate. We assume a second-order polynomial to approximate the function and fit it by least mean square error:

$$s(x) = p_2 x^2 + p_1 x + p_0 \tag{1}$$

where p_2, p_1, and p_0 are parameters to estimate.

Notes on polynomial fit. The second-order polynomial suffices to represent a flat scene layout from a homographic projection. This assumption is plausible with most road scenes visible from a surveillance cameras, which are mostly flat. The approximation of the scale layout compensates for the size variations of the objects (*e.g.* cars of differet sizes) given sufficient number of detections. We illustrate a sample estimate in Fig. 2. Interestingly, the scale layout automatically provides the horizon estimate.

Proposal pruning. Within the RPN module of our proposed GP-FRCNN, we prune object proposals according to the following:

$$\frac{\|s(\hat{x}) - \hat{b}\|}{s(\hat{x})} < \sigma \tag{2}$$

Here, $s(\hat{x})$ is the scale estimate of object at position $\hat{x}$, as described in Eq. (1), and $\hat{b}$ is the actual bounding box size of the object. σ represents the acceptable deviation of the proposal size from the scale function. We set its value to 0.3 for all our experiments, based on the observed variance in the training data.

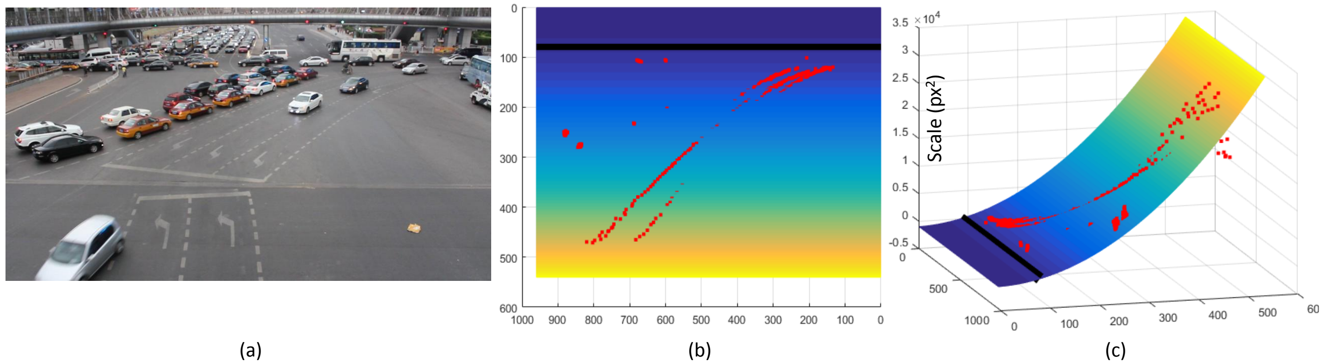

Figure 2. (b) shows the *top-view* of the estimated scale layout of the scene in (a). (c) demonstrates the 3D view of the layout in (b). Black line indicates the estimated horizon. Red points indicate the most-confident car detections that are used to estimate the scale function.

Beyond one object class. For the DETRAC challenge, we only consider cars as target detection for the estimation of scale layout. Note however that Eqs. (1), (2) apply to the estimate of other object sizes.

More interestingly, given a single correct scale observation of another object, *e.g.* a *bus*, one may adapt the scale estimate function $s(x)$ to it, by simply scaling it by the factor $\frac{\hat{b}'}{s(\hat{x})}$, where $s(\hat{x})$ is the original scale function estimation at the pixel position, and $\hat{b}'$ is the size of the *bus* detection.

4. Fine adjustments for Faster R-CNN

The use of geometric proposals is a straightforward extension to the vanilla Faster R-CNN, but the plain integration degrades performance, which we found surprising. We explain in this section the adjustments which are needed, to make the model shine.

Specific anchor scales. During training, Faster R-CNN separates the object bounding boxes into anchor scales and aspect ratios. By default, the scale set of anchor boxes is {8, 16, 32}. This fits most detection benchmarks such as PASCAL VOC. However, applying Faster R-CNN with the default anchor scales to the UA-DETRAC benchmark severely underperforms, because most cars are much smaller than the smallest default scale. As a consequence, the region proposals corresponding to the smallest anchor boxes need to serve any object smaller then its set size, contrasting the very principle of anchor scales.

We fix this issue by extending the set with smaller scales in the sequence *i.e.* {1, 2, 4} (*Option#1*), for which we report results in Fig. 3. An alternative (*Option#2*), which we also experimented with, consists of a set of quantized scales relative to the training data as proposed by [14].

Higher resolution feature maps. A second limitation of Faster R-CNN on smaller objects is given by the coarse resolution of its CNN-feature-block. This issue has been noted several times in semantic segmentation, whereby a coarser granularity limits the per-pixel resolution. Recently, Zhang *et al*. [14] mentioned this issue with regards to detection.

We confirm the need for a finer feature map, prior to the Faster R-CNN proposal and classification machinery. In more detail, we reduce the feature stride from 16 to 8 by removing the last max-pooling layer from the base-feature-networks, across all experimented models. This effectively leads to increasing the number of locations on the image to look for the object, resulting in a significant gain in recall for the small vehicles (*e.g.* Fig. 3).

Note that this additionally results in a smaller effective receptive field on the input image. While this does not affect small objects, for which the model need not look at a larger area in the object, this reduced the recall for the larger vehicles in our experiments, for which the context became too tight. This effect was more evident with smaller feature models such as VGG-M, while larger models seemed more robust to changes in the receptive field size.

Multi-stage training. Learning the parameters for all the convolutional layers on the detection task is not trivial, therefore in the default strategy for training the Faster RCNN model the parameters are initialized with the ImageNet pre-trained model. Moreover, the learning of first 4 convolutional layers is skipped. This means the low-level features in the baseline Faster R-CNN model are still the ones actually trained only for the ImageNet classification task. This is naturally not an optimal setting. In our experiments, however, we adopt a multi-stage training approach and also learn those initial convolutional layers which are skipped in the standard training stage of Faster R-CNN. In the first stage we keep the default strategy and do not learn the parameters of the initial convolutional layers of the networks. And in the second stage we continue the training procedure on the complete network after un-locking the initial convolutional layers as well. Alternatively, one could also investigate a strategy similar to the recently proposed work of Goyal et al. [6], who propose a warmup session with very small learning rates.

	VGG-M									VGG-16	
								GP-FRCNN	GP-FRCNNm		GP-FRCNNm
PASCAL 0712		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Specific anchor scales (Q)				✓							
Specific anchor scales					✓	✓	✓	✓	✓		✓
HR feature maps						✓	✓	✓	✓		✓
Atrous							✓				
GP			✓					✓	✓		✓
multi-stage training									✓		✓
AP (%)	58.9	64.1	64.9	68.3	68.8	72.3	71.8	78.7	80.9	66.8	**82.1**
Inference speed (fps)	**12**	**12**	**12**	11	11	8	8	8	8	7	4

Table 1. Results using different components of our method on the UA-DETRAC validation set. (Q) means quantized anchors [14]. $IoU = 0.7$.

5. Experiments and Results

Here we detail our evaluation on UA-DETRAC [13], which is a very comprehensive dataset for surveillance scenarios. The dataset consists of 100 video sequences (60 for training, 40 for testing) presenting real traffic scenes in different weather conditions.

Network architectures. We report results using two different variants of the VGG network, as the backbone convolutional architectures for our approach. The first one is VGG_CNN_M_1024 [1] with 5 convolutional and 3 fully connected layers. We will refer to this network as VGG-M in our experiments. The second one is standard VGG-16 [12] with 13 convolutional and 3 fully connected layers.

Training strategy. Following the settings described in original UA-DETRAC report [13], we use vehicle annotations with less than or equal to 50% occlusion and 50% truncation. Considering this choice of allowed occlusion ratio, we use intuitive NMS threhold of 0.5. Furthermore, we use a stricter value for the Faster R-CNN parameter FG_THRESH, *i.e.* 0.7 instead of 0.5, for a detection to be considered as a positive class during training of the classification network. The intuition for this change is also quite natural, since the online UA-DETRAC evaluation uses IoU of 0.7 to count a detection as correct.

To make the models robust and avoid overfitting on DETRAC scenarios, we also include PASCAL VOC 2007 and 2012 trainval image sets together with the DETRAC images to train our models. We train for all standard 20 classes of the PASCAL VOC dataset, as we find it slightly better than training only for vehicles. For all our experiments we use the end-to-end approach of Faster R-CNN for model training, which trains both RPN and the classification network jointly.

5.1. Validation

We assume that the distributions of the vehicle data in the train and test sets are similar. Therefore, we take out 36 videos from the training set to quickly benchmark several approaches, including improvements and refinements for the Faster R-CNN framework. The remaining 24 videos are used to train our models in this validation phase.

In Table 1, we detail the effect of all the components of our method in terms of average precision (AP). First, we notice that baseline approach with only DETRAC dataset for training (first column) severely underperforms, however, we improve by 5% by just adding PASCAL VOC datasets for training.

Specific anchor scales. We notice that both options for task specific anchor scales, as discussed in Section 4, improve the AP by a fair amount. This shows that using appropriate scale set is important to learn regression parameters per anchor box, since it makes the task easier during training by allowing them to serve only a narrower portion of the object scales. We notice that the simple extension of the scale set (*Option#1*) works slightly better as compared to the quantized scales as suggested in [14].

Higher resolution feature maps. As argued in the literature, resolution contributes significantly especially in the case of objects with small scales. Overall we notice that the AP improves by 4.5% by removing the last max-pooling layer from the VGG-M network. We see in Fig. 3, that this improvement actually comes from the detection of small scale vehicles. While infact, the recall drops in the case of larger vehicles. Furthermore, higher resolution features notably increase the computation time.

Dilated convolution (Atrous). In order to counter the negative effect of increasing the resolution in the previous step, following [2] we investigate dilated convolutions (atrous) for each subsequent Conv layer in the network after the removed max-pooling layer. Theoretically, this increases the context in the receptive field. However, we notice slight drop in the AP, which indicates its ineffectiveness.

Geometric proposals. First we notice that our proposed geometric proposals does not help when applied to the vanilla Faster R-CNN (Table 1, 3^{rd} column). This tells us that a better distribution of the proposals is not enough if the limitations of the backbone architectures are not addressed. Quite interestingly, the geometric proposals boosts the AP by more than 6% when applied to the modified Faster R-CNN. Analyzing this result further in Fig. 3, we observe

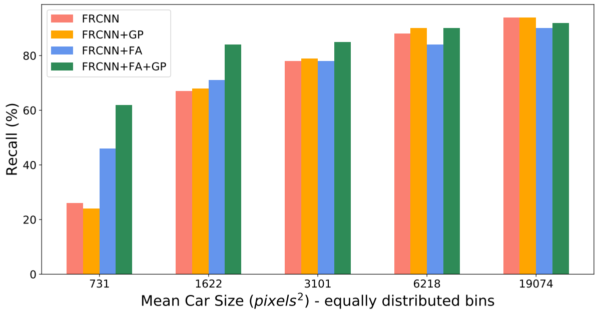

Figure 3. Scale vs Recall, for **VGG-M** network. FA: fine adjustments for Faster R-CNN, GP: geometric proposals.

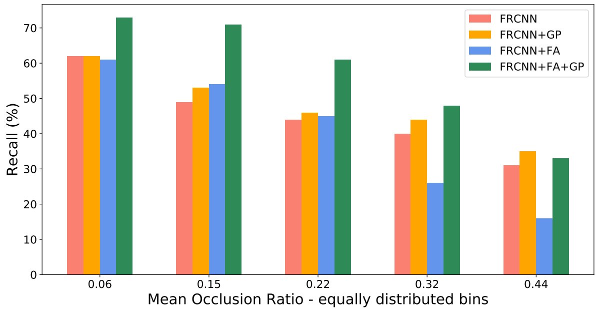

Figure 4. Occlusion ratio vs Recall, for **VGG-M** network. Maximum occlusion ratio in our validation set is 0.5.

that the substantial gain comes for the smaller and medium sized vehicles, while keeping the accuracy on par w.r.t the baseline model for the larger objects.

Multi-stage training. Finally, multi-stage training as described in Section 4 further improves the AP to impressive 80.9%. We demonstrate that the fine-tuning of even the semantically weak features help to build a better representation for the task at hand, if executed reasonably.

In the end, we also evaluate our proposed approach using the larger VGG-16 network. The results are given in last column of Table 1. Overall we get slightly better results but lose half of the frame rate. Here it is interesting to point out that our proposed GP-FRCNNm with smaller VGG-M network significantly outperforms the vanilla Faster R-CNN with larger VGG-16 network, in terms of AP (66.8% *vs* 80.9%), while keeping the frame-rate higher.

In the following, we highlight the important properties of our method which contribute to the higher overall accuracy, including ability to cope with scale changes, and different levels of occlusion.

Scale invariance. In Fig. 3 we demonstrate that our final model is able to enhance the scale invariance property of the original detector. We show that the fusion of geometric proposals (GP) and the adjustments to the Faster R-CNN (FA) is able to significantly improve the recall for smaller and medium sized vehicles while keeping the recall for the larger vehicles intact. We show that the baseline approach significantly underperforms for the detection of cars with smaller scales, and the model with fine adjustments (FA) as detailed in Section 4 is not only ineffective for medium scale cars but also degrades the recall for large scale cars.

Better occlusion handling. In Fig. 4, we see a similar trend when analyzing the performance w.r.t occlusion handling. An interesting finding is that the network adjustments (FA) alone degrade the performance for objects with larger occlusion ratio, explaining the importance of context in object detection, especially when the object is severely occluded in the image. Although, our method with geometric proposals do not modify the receptive field, but propose a better ranking of proposals w.r.t the scene geometry.

5.2. The UA-DETRAC Challenge

In this section we report the results for the UA-DETRAC detection challenge, and as of writing we beat all the currently available results on the website in terms of average precision.

To train the models, we use the compete UA-DETRAC train-set (60 video sequences, all images), together with PASCAL VOC 2007 and 2012 trainval image sets just as done in the validation phase. We observe that some of the traffic scenarios in the UA-DETRAC testset are relatively more dense and cluttered as compared to the video sequences in the training set. However, most results we obtained for the online challenge are consistent with our evaluations during the validation phase. Overall we improve by an impressive 19.5% in terms of AP over the vanilla Faster R-CNN *i.e.* from 57.08% to 76.57%. We notice that the impact of adding geometric proposals is not as strong as we observed during validation. We believe this is due to the fact that a large number of small scale objects are ignored during online evaluation, that lie in the marked ignored regions in the image.

Method	AP (%)	Speed (fps)
Faster RCNN	57.08	**7**
+ specific anchor scales	65.03	6.5
+ HR feature maps	72.80	4
+ GP → GP-FRCNN	73.88	4
+ multi-stage-training → GP-FRCNNm	**76.57**	4

Table 2. Results on UA-DETRAC Test-set using **VGG-16** net.

6. Conclusion

In this work, we showed that our proposed GP-FRCNN approach has the potential to overcome the ranking failures of the baseline RPN, and as a result achieves more or less similar performance independent of the scale of the object. Our findings also suggest that one cannot simply incorporate the geometric layout to re-rank proposals and expect desired improvements, instead a number of scale modifications are essentially required.

References

[1] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[2] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, 2016.

[3] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015.

[5] R. B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448, 2015.

[6] P. Goyal, P. Dollar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *CoRR*, 2017.

[7] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[11] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *CoRR*, 2016.

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[13] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015.

[14] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. *CoRR*, 2017.