Supplementary Material of ICCV 2019: Simultaneous multi-view instance detection with learned geometric soft-constraints

Anonymous ICCV Submission
3964

1. Implementation Details

Our network was implemented using Keras [2] and Tensorflow [1]. The implementation as explained in the main paper was based on SSD [4], but is replaceable with any object detector method. The SSD network was then modified to have a base of our network of ResNet50 [3] which is pretrained on the ImageNet dataset. The IOU threshold was set to 0.45, and confidence threshold of 0.001. Due to the large amount of negative default boxes or anchors, hard negative mining is performed to overcome the positive and negative imbalance with a ratio of 3:1. The input images are resized to 300 x 600 (height x width) being that the images are panoramas to keep the aspect ratio. We optimize our loss functions using ADAM optimizer with the initial learning rate set to 0.001.

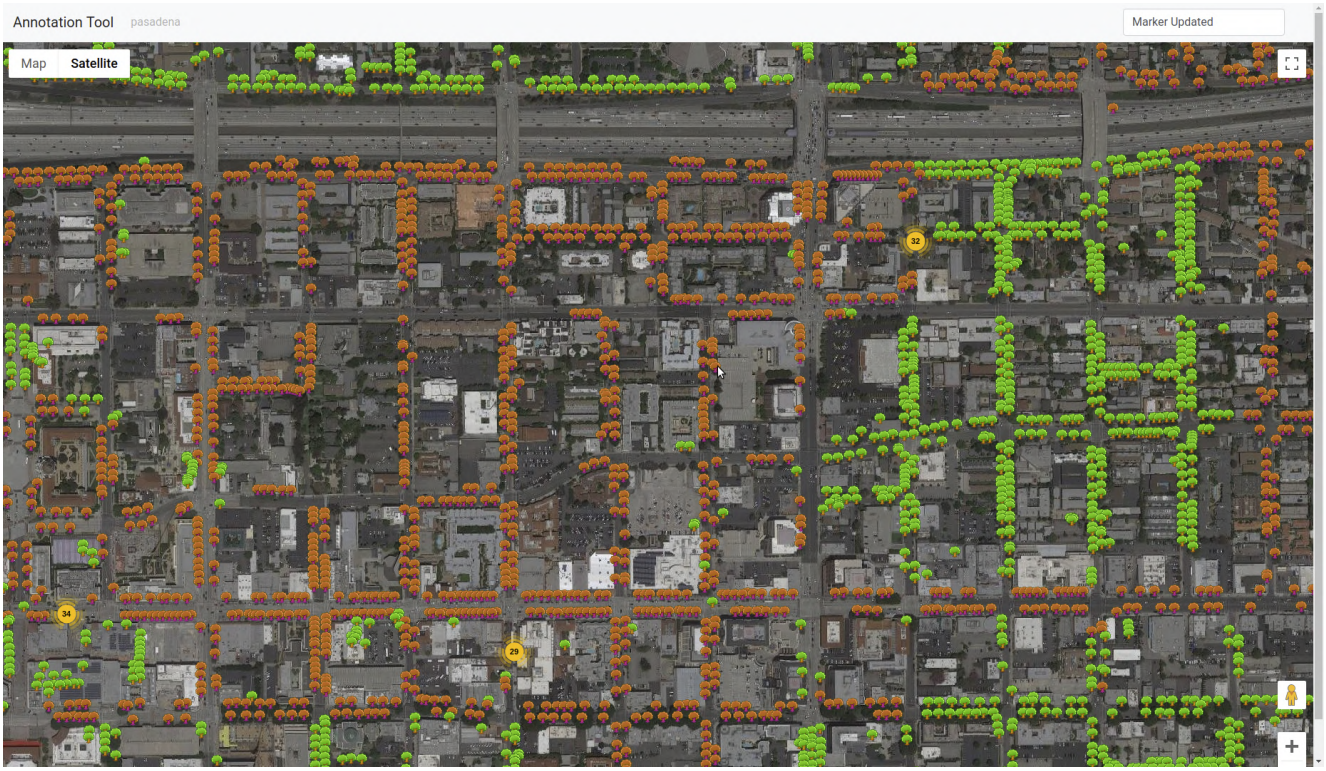


Figure 1: Green tree markers represent trees that are annotated. Brown trees represent annotated labels. Dragging and dropping the tree icon annotates the tree's location by updating it. New trees can be added by clicking the mouse left button in empty areas. Trees can be removed also by the mouse right click.

2. Multi-view object annotation tool

Our annotation tool is the only tool available of its kind as far as we know. It enables the annotation of object both from aerial/satellite view and ground level simultaneously any where in the world where Google Street View imagery exists. The user after creating and naming a dataset is presented with the aerial view Fig. 1. After clicking on the tree or marker, the tool grabs the 4 closest panoramas to the geo-coordinate of the marker, as shown in 2. This view enables the annotation of the Google Street View imagery available in proximity of the annotated aerial view marker. We've implemented an application programming interface (API) that enables the user to request the annotation of choice in VOC XML, or JSON format. The tool is implemented using Flask [1] in Python and HTML to enable developers to customize and build on it extra features. Preliminary boxes are drawn using an object detector (SSD or any python supported object detector) that can be adjusted or modified to help the annotator. This option of providing preliminary boxes can be enabled or disabled.

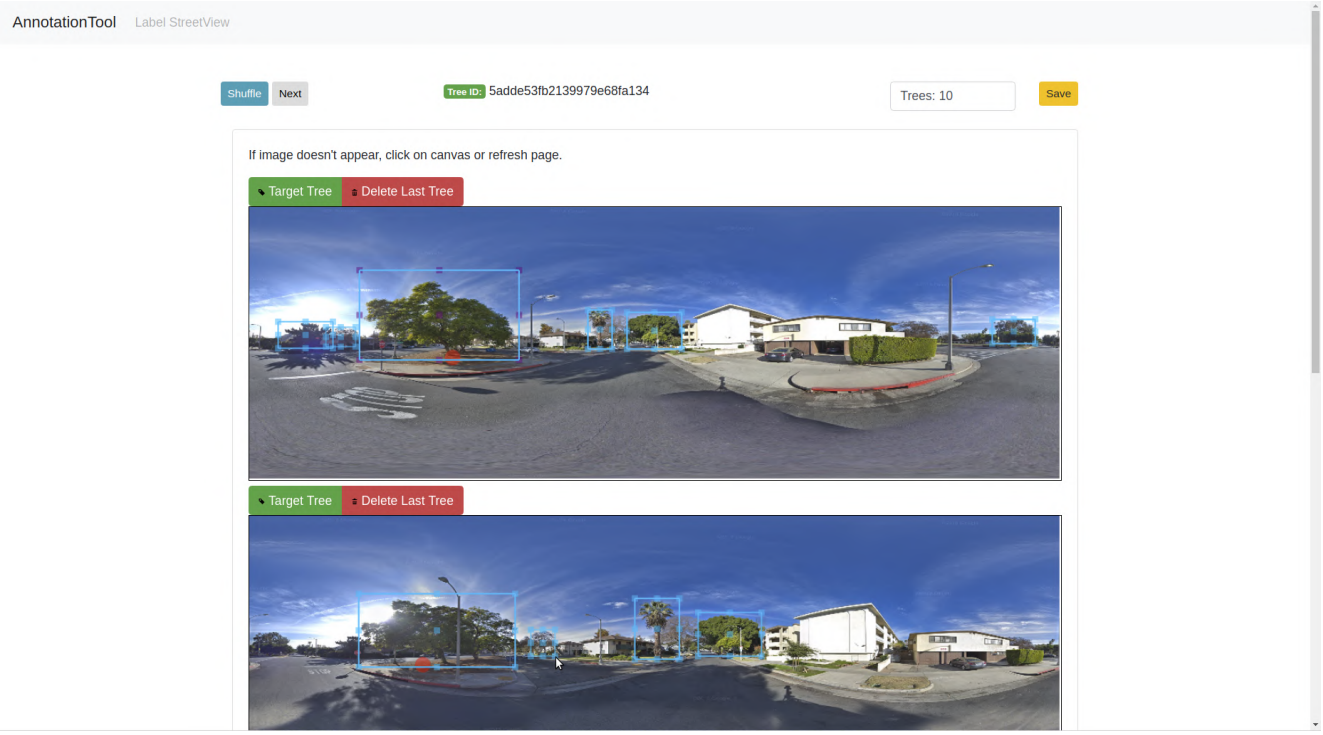


Figure 2: Using the marker places from Fig. 1, we draw a red circle to guide the annotator on which is the target tree we seek to identify.

3. Pasadena Multi-view Re-ID Dataset

In this section, we show a portion of our dataset that will be available for public which is created using "Multi-view object annotation tool" from Google Street View (GSV) images. In Fig. 3 we present an instance of a single annotated identity of a tree. Fig. 4 & 5 shows heavily dense areas with multiple identities annotated.

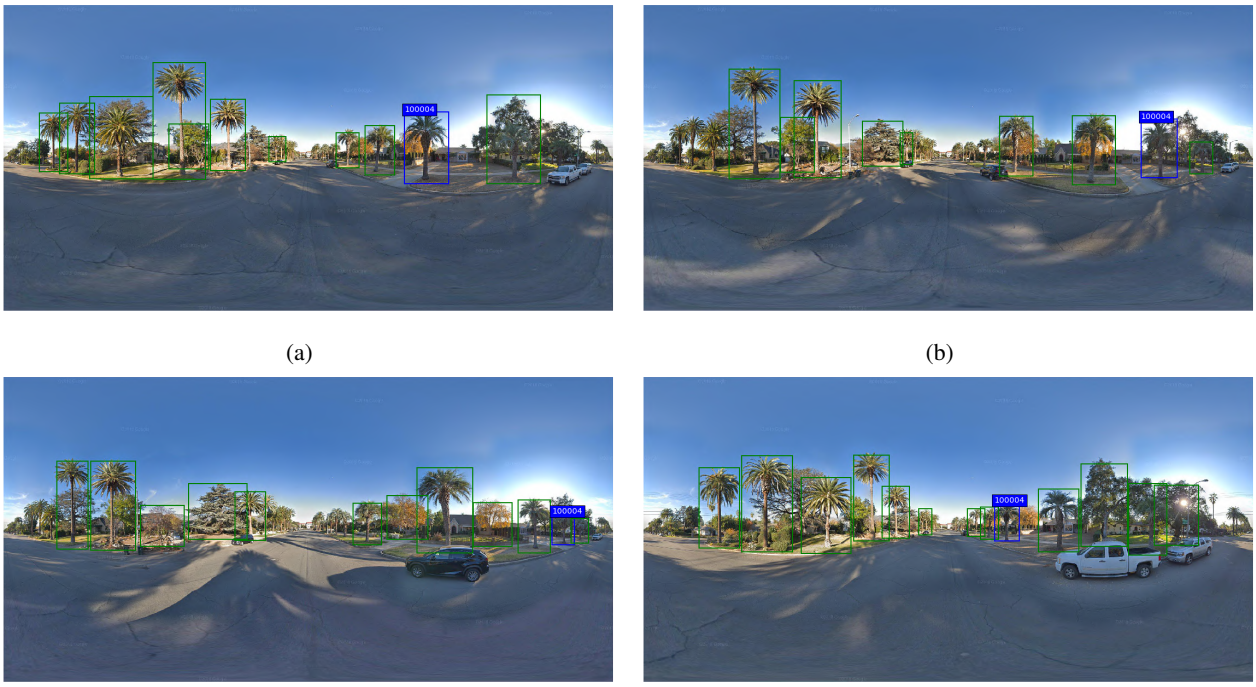
4. Mapillary Dataset

In this section we showcase the dataset provided to us by Mapillary. As presented in Fig. 6, 7 and 8, there are many differences with our "Pasadena Multi-view Re-ID Dataset" shown in Sec. 3.

5. Visualization of Detections

In this section we show some of our detection results on both datasets, and how they are different. Fig. 9 shows the detections on the Pasadena Multi-view Re-ID using a monocular object detection method against our method showing different

[1] http://flask.pocoo.org/



(a) (b)

(c) (d)

Figure 3: GSV images overlaid with ground truth bounding boxes for a single annotated tree instance. Green: annotated trees without identity annotation. Blue: annotated trees with identified instances.

situations and tree types. Similarly, Fig. 10 shows the detections on the mapillary dataset results using a monocular object detection method against our method.

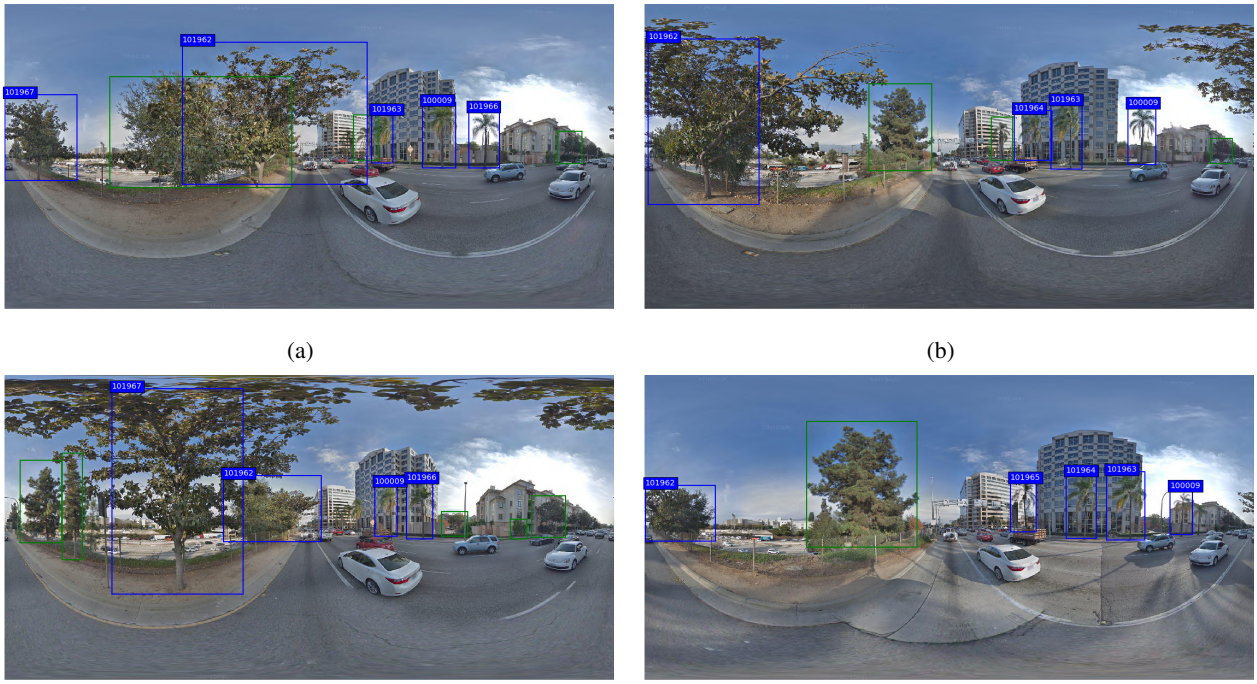


Figure 4: GSV images overlaid with ground truth bounding boxes for multiple annotated tree instances. Green: annotated trees without identity annotation. Blue: annotated trees with identified instances.

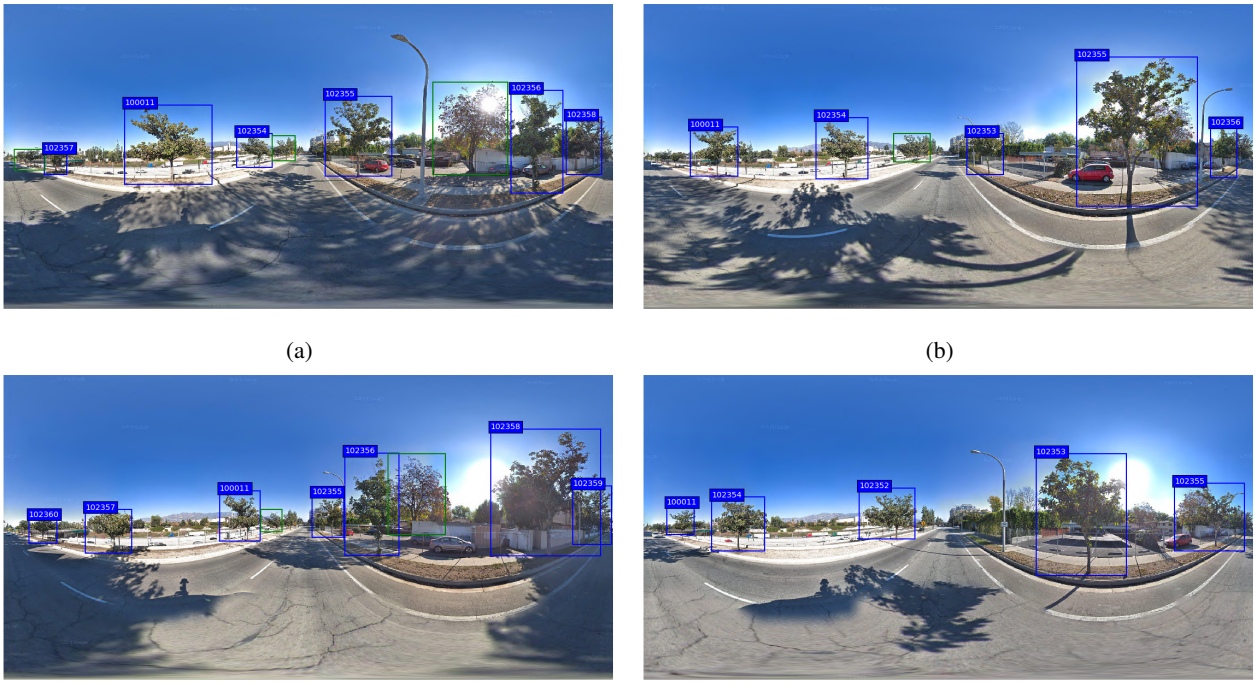
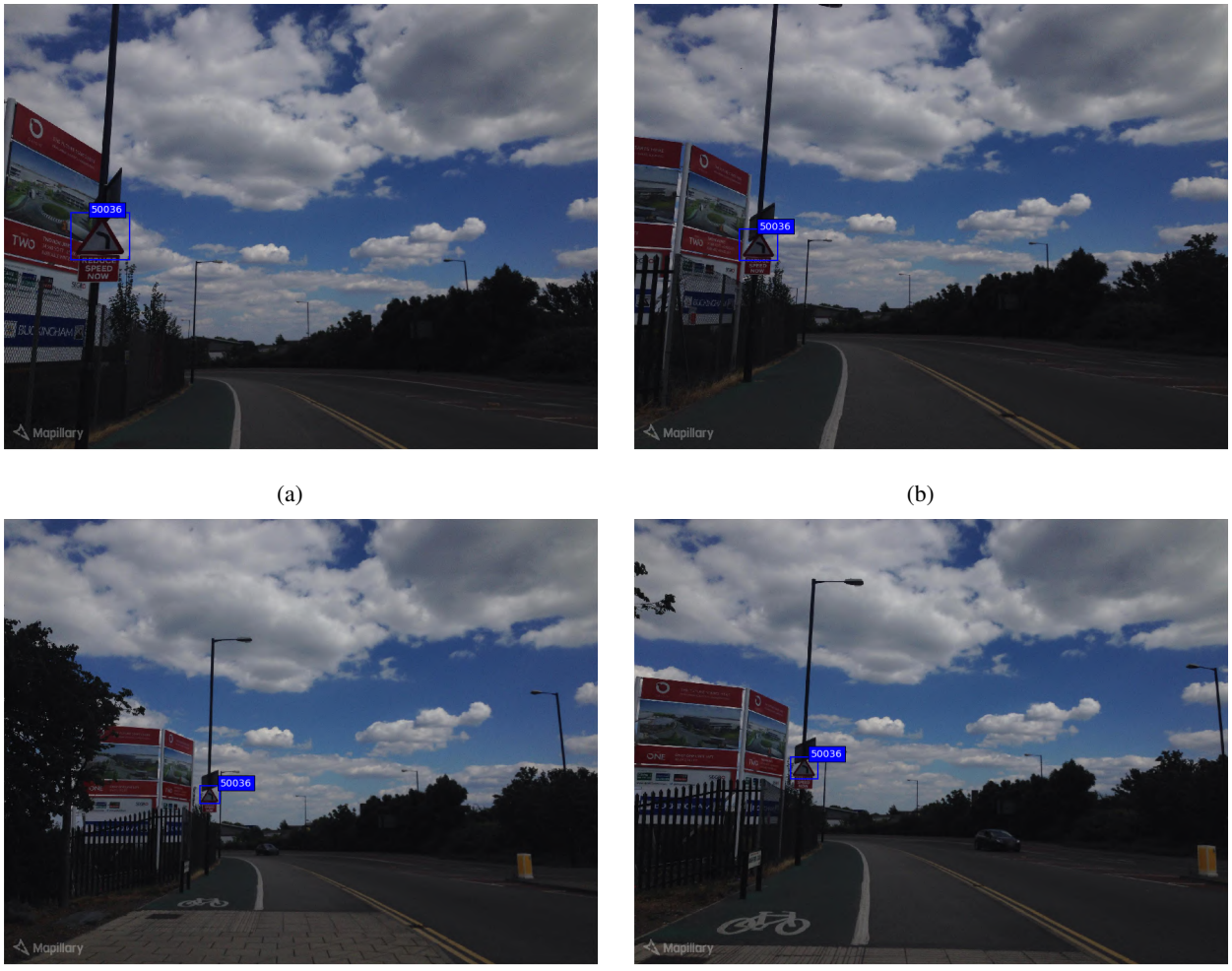


Figure 5: Images overlaid with ground truth bounding boxes for multiple annotated tree instances. Green: annotated trees without identity annotation. Blue: annotated trees with identity instances.

(a) (b) (c) (d) (e) (f)

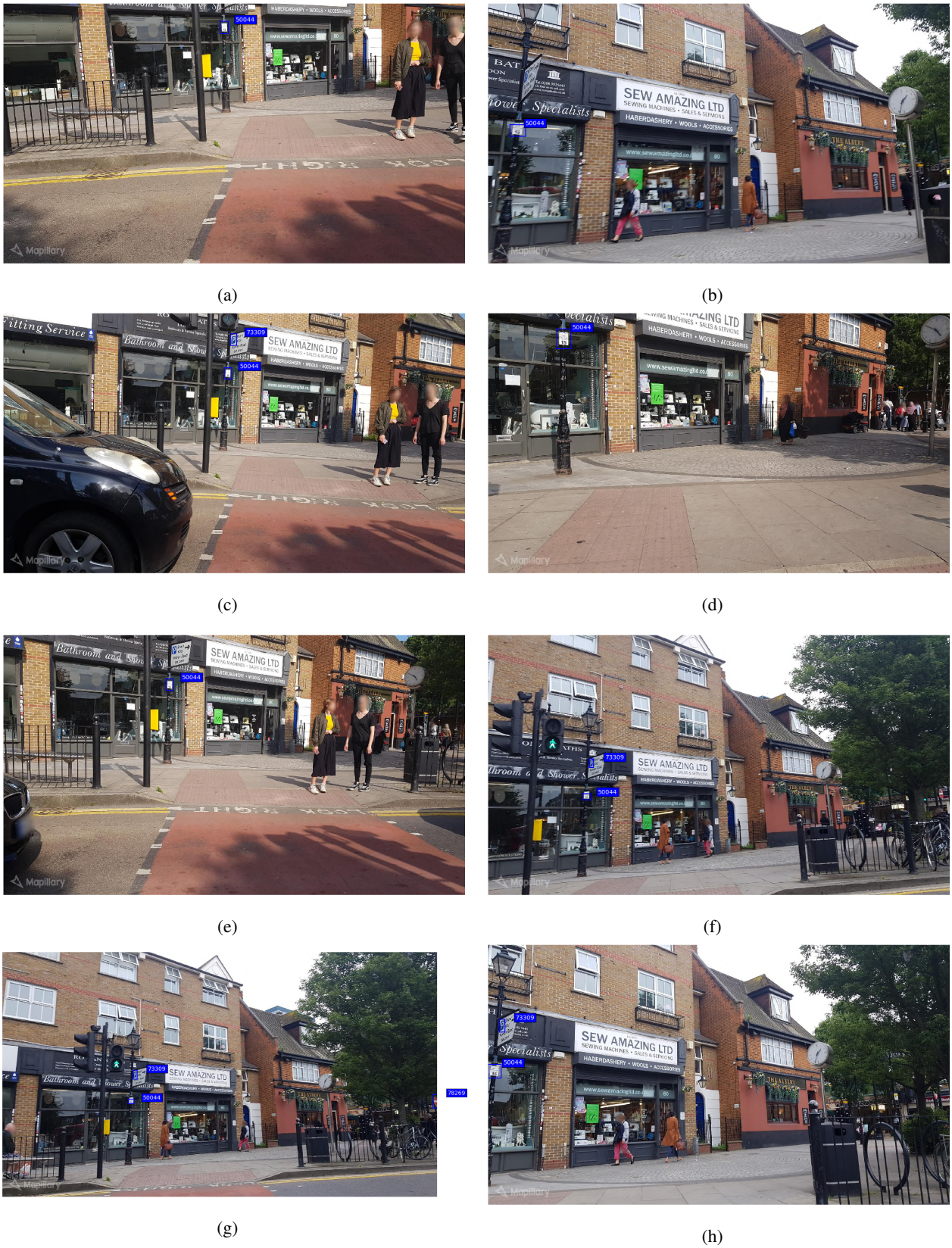
Figure 6: Mapillary images overlaid with ground truth bounding boxes for multiple annotated sign instances. Blue: annotated signs with identity instances. Note how (e) & (f) are taken from a different camera and time to the rest of the images.



(a) (b)

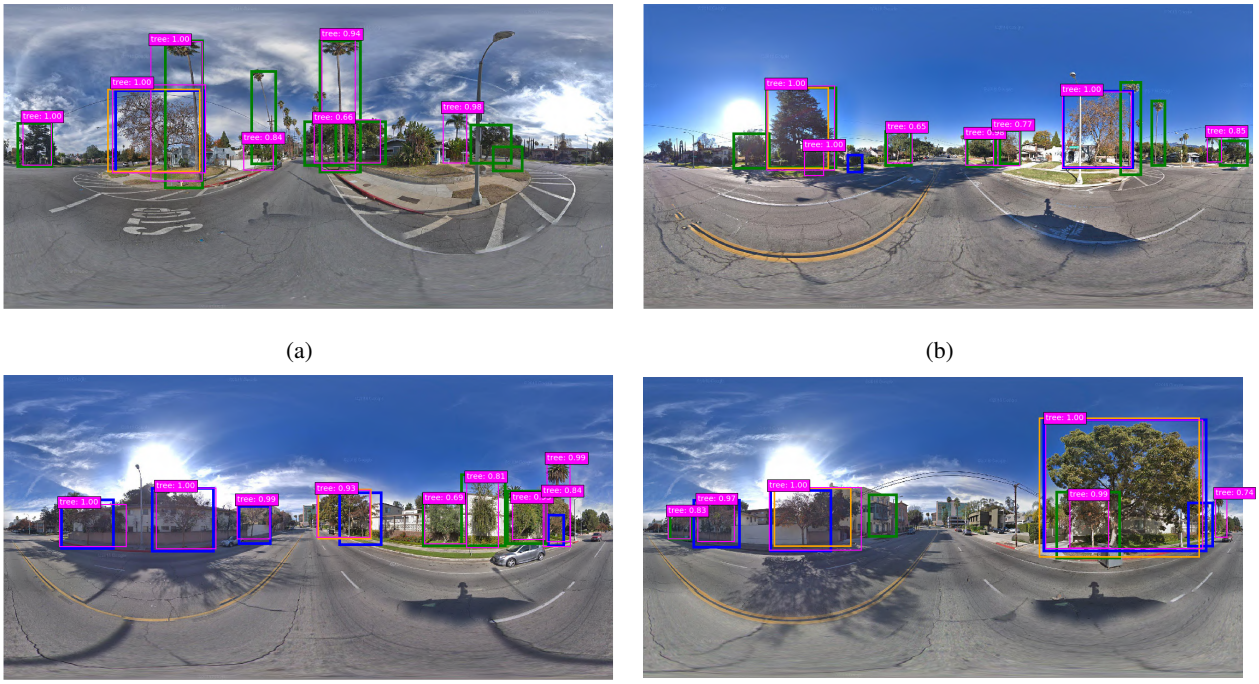
(c) (d)

Figure 7: Mapillary images overlaid with ground truth bounding boxes for multiple annotated sign instances. Blue: annotated signs with identified instances. These images represent the typical scenario of our identified instances which features shorter baselines, tinier objects, smaller field of view, and mostly forward looking cameras as discussed in the main paper.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8: Mapillary images overlaid with ground truth bounding boxes for multiple annotated sign instances. Blue: annotated signs with identified instances. These images show the different instances of signs from different view points.



(a) (b)

(c) (d)

Figure 9: GSV images overlaid with ground truth bounding boxes for multiple annotated tree instances, and predictions using a monocular object detector and our method. Blue: annotated trees with identified instances. Green: annotated trees without identity annotation. Orange: detections using a monocular object detector. Pink: detections using our method. These images show the different instances of signs from different view points.

(a) (b)

(c) (d)

Figure 10: Mapillary images overlaid with ground truth bounding boxes for multiple annotated tree instances, and predictions using a monocular object detector and our method. Blue: annotated trees with identified instances. Orange: detections using a monocular object detector. Pink: detections using our method. These images show the different instances of signs from different view points. (a) & (b) show front facing camera from a car's dashboard camera. (c) & (d) shows a difficult distorted image from a bicycle or motorcycle.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] F. Chollet. keras. `https://github.com/fchollet/keras`, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.