Boxy Vehicle Detection in Large Images

Karsten Behrendt
Bosch Automated Driving
karsten.behrendt@us.bosch.com

Abstract

Camera-based object detection and automated driving in general have greatly improved over the last few years. Parts of these improvements can be attributed to public datasets which allow researchers around the world to work with data that would often be too expensive to collect and annotate for individual teams. Current vehicle detection datasets and approaches often focus on axis-aligned bounding boxes or semantic segmentation. Axis-aligned bounding boxes often misrepresent vehicle sizes and may intrude into neighboring lanes. While pixel level segmentations are more accurate, they can be hard to process and leverage for trajectory planning systems. We therefore present the Boxy dataset for image-based vehicle detection. Boxy is one of the largest public vehicle detection datasets with 1.99 million annotated vehicles in 200,000 images, including sunny, rainy, and nighttime driving. If possible, vehicle annotations are split into their visible sides to give the impression of 3D boxes for a more accurate representation with little overhead. Five megapixel images with annotations down to a few pixels make this dataset especially challenging. With Boxy, we provide initial benchmark challenges for bounding box, polygon, and real-time detections. All benchmarks are open-source so that additional metrics and benchmarks may be added.

1. Introduction

Perception systems and especially vision-based object detection systems are integral parts of self-driving cars. Camera images generally offer a higher resolution compared to various sensors such as lidar or radar. This allows an understanding of a vehicle's complete surrounding and object detections over long distances. Color information can additionally be used to deduce attributes, such as brake lights and turn signals, which are not available in other sensors.

A lot of advances in computer vision and vehicle detection are possible because of public datasets and benchmarks.

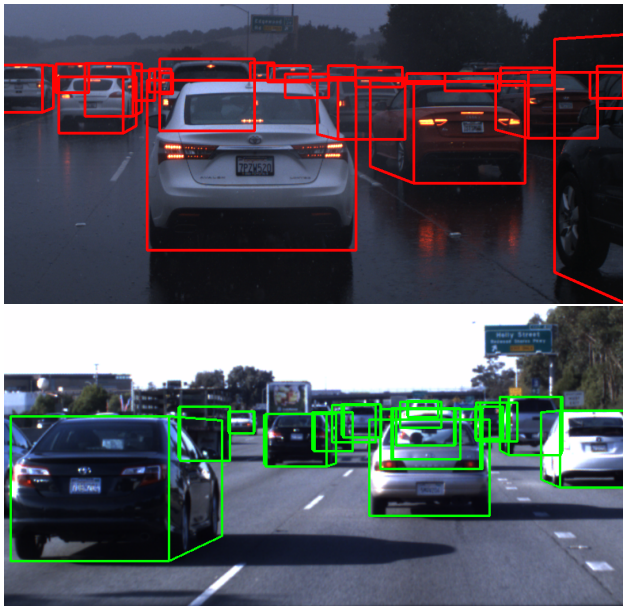

Figure 1. Top: Sample annotations in rainy weather. Bottom: Detections provided by our baseline method.

1.1. Vision Datasets

One of the most impactful datasets, the ImageNet Large Scale Visual Recognition Challenge's (ILSVRC) [25], saw a top-5 classification error reduction from 28.2% to $\tilde{3}$% within only six years. Over the same time frame, object detection accuracy on the Pascal Visual Object Classes (VOC) Challenge [6] and the detection part of the ILSVRC also increased significantly [25]. These datasets, containing tens of thousands to millions of annotated images, allowed researchers to train new, much larger, and more powerful neural network architectures such as Faster-RCNN [22], Single Shot MultiBox Detector [15], You Only Look Once [20, 21], and various ensembles. In addition to bounding boxes for object detection, the Pascal VOC [6] and Microsoft Common Objects in Context (COCO) [14] offer pixel level annotations. This enabled the creation of models which accurately stimate object locations in images down to individual pixels [16, 23, 1]. One promising application for the

advances in object detection and semantic segmentation are driver assistance systems and fully automated vehicles.

1.2. Vehicle Detection Datasets

Fast, accurate, and reliable detections of other traffic participants are crucial for automotive applications. This demand has already led to a number of datasets for vision-based vehicle detection [7, 5, 32, 29, 2, 19, 24, 17, 31, 28, 29].

The KITTI Vision Benchmark Suite [7] is one of the first large datasets to offer a variety of annotations for automated driving topics such as odometry, optical flow and object detection. Vehicles are annotated as 3D boxes within KITTI. Cityscapes [5] offers full-scene pixel-level annotations for 5000 images with an additional 20,000 coarsely annotated images. The BDD100k dataset contains 100,000 images with vehicles labeled with both 2D bounding boxes and pixel-level annotations.

Additionally, there exist a few datasets with axis aligned bounding box (AABB) labels for vehicles such as the Toyota Motor Europe Motorway Dataset (TME) [2], two Udacity datasets [31], the Nexar Challenge 2 [17], Mapillary Vistas [19], and the Lisa Vehicle Dataset [28]. See Table 2 for the respective dataset sizes.

In addition to manually annotated datasets, it is possible to train detection models on simulated data. The Synthia dataset [24], for example, contains 200,000 images with pixel level annotations including vehicles. Another research area focuses on creating photo-realistic images from simulation [3].

We present the Boxy dataset for image-based vehicle detection specific to freeway driving. All vehicles are split into their visible sides which creates a 3D-like boxy impression for a more detailed representation compared to AABB. To our knowledge, the dataset is the largest public vehicle detection dataset with 1,990,806 manually annotated vehicles in 200,000 images. It includes different weather conditions and high resolution, five megapixel images which make this dataset especially challenging.

We publish Boxy with benchmark challenges on AABB detections with and without runtime restrictions, and 3D-like detections to allow comparing vehicle detection methods on a large amount of difficult annotations.

2. The Boxy Vehicles Dataset

2.1. Key figures:

- 200,000 images, full resolution about 1.1 TB
- 5 megapixel resolution of 2464x2056
- 3D-like and 2D bounding boxes
- 1,990,806 annotated vehicles
- Average vehicle annotation covers only 0.3% of an image

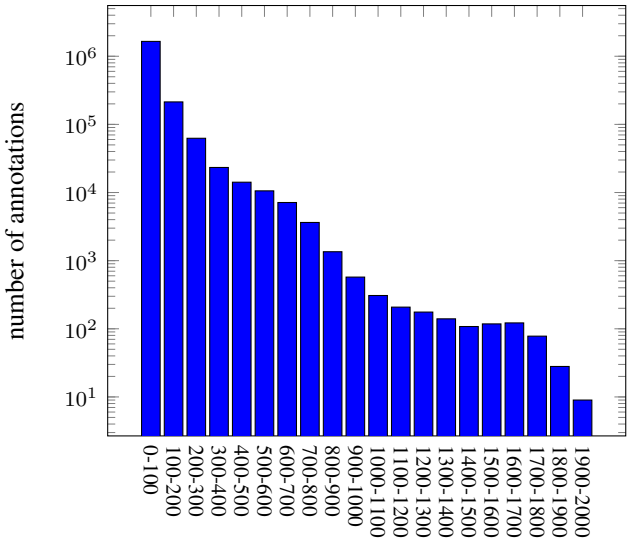

Figure 2. Distribution of annotated vehicle heights in pixels.

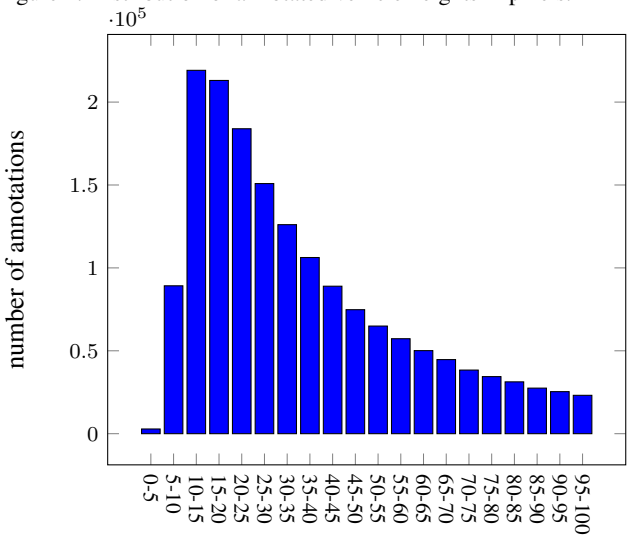

Figure 3. Distribution of vehicle heights for annotations smaller than 100 pixels.

- Sunny and rainy conditions at daytime, dawn, and dusk
- Traffic jams and empty freeways

2.2. Dataset Overview

Boxy is a large and challenging computer vision dataset for vehicle detection.

One challenging aspect is the fairly small object annotations compared to the image size, which results in a large search space. The average annotation only covers approximately 0.3% of its respective image and the majority of annotations are less than 50 pixels in height as displayed in Figures 2 and 3. We also note that Boxy also contains

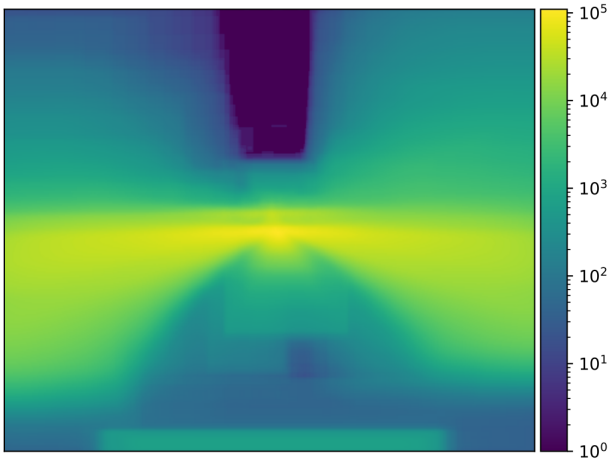

Figure 4. Number of vehicles that occupy each pixel. A large percentage of vehicles is represented by small annotations towards the center of the image close to the vanishing point.

annotations that are larger than the complete image resolution of most existing datasets. Especially for realtime detections, there need to be input resolution, runtime, and accuracy trade-offs.

Generally, vehicles are visible within every part of the image. An overall majority, however, are clustered around the vanishing point with a distinct distribution around the average location of neighboring lanes as visualized in Figure 4. Vehicles outside the densest regions also need to be reliably detected, especially because large annotations refer to close and therefore safety critical ones. The camera view cannot only be optimized for vehicle detection, but also needs to be able to, for example, capture traffic signs, traffic and metering lights.

2.3. Image and Sensor Specifications

All images are collected using a mvBlueFOX3-2051 with a Sony IMX250 chip using a global shutter [10]. The data is stored at 15Hz as 8 bit color images at a resolution of 2464×2056 pixels. At 3x8 bits per pixel value, each image array requires approximately 15.2 MB or 228 MB per second per camera which may need to be streamed, processed, and stored. As part of Boxy, we provide the images lossless portable graphics files at about 5.5 MB per image. For faster downloads and easier handling, we also provide equalized and downsampled versions.

2.4. Recordings and Environment

All sequences were recorded on San Francisco Bay Area freeways, namely the California state routes 85 and 92, and interstates 101 and 280. Despite its limited regional scope, the different traffic scenarios, 3D-like annotations, their sizes, times of day, and weather conditions should ensure that Boxy is a challenging dataset.

Sequence	# Frames	Conditions
Training:		
2016-09-30-14-41-23	9313	sunny
2016-09-30-15-03-39	6349	sunny
2016-09-30-15-19-35	6199	sunny
2016-10-04-13-52-40	12854	sunny
2016-10-04-14-22-41	6494	sunny
2016-10-10-15-17-24	2779	sunny
2016-10-10-15-24-37	3126	sunny
2016-10-10-15-32-33	373	sunny
2016-10-10-15-35-18	4940	sunny
2016-10-10-16-00-11	835	sunny
2016-10-10-16-12-20	11054	sunny
2016-10-10-16-43-45	7456	sunny
2016-10-10-18-25-04	5592	sunny
2016-10-10-18-41-33	7898	sunset to dark
2016-10-26-12-49-56	178	sunny
2016-10-26-13-00-25	1031	sunny
2016-10-26-13-04-33	16045	sunny
2016-10-26-17-55-06	191	sunset
2016-10-26-17-57-22	1890	sunset
2016-10-26-18-03-11	3375	sunset
2016-10-26-18-22-27	2380	sunset to dark
2016-10-26-18-38-03	1423	dark
2016-10-30-09-53-48	3559	rain and traffic jam
2016-10-30-10-01-47	1224	rain and traffic
2016-10-30-10-04-51	7956	rain
2016-10-30-10-24-32	83	rain
2016-11-01-10-07-39	5239	sunny, different lens
2016-11-01-10-20-23	5562	sunny, different lens
Validation:		
2016-09-27-14-43-04	21475	sunny
2016-11-03-15-40-30	7271	sunny, light traffic
Testing:		
2016-11-02-18-05-08	12767	sunny to dark
2016-11-03-15-03-15	11180	sunny and traffic
2016-11-03-15-28-03	5614	sunny and traffic
2016-10-30-10-26-40	6295	rain

Table 1. Overview of the individual sequences within Boxy. There are 135,398 training, 28,746 validation, and 35,856 test images.

An overview of the different sequences in the training, validation, and test sets is given in Table 1. The recordings consist of mostly sunny conditions with non-negligible parts of overcast, heavy rain, dusk and nighttime driving. Traffic conditions range from light to heavy congestion and should reflect typical freeway driving.

2.5. 3D Boxes and Annotation Specifications

Axis aligned bounding boxes (AABB), 3D bounding boxes, and pixel level segmentation are the current standard in vehicle detection. AABB often do not tightly capture vehicles and may intrude into neighboring lanes (as dis-

Figure 5. The difference in accuracy between 2D and 3D annotations. The 2D axis-aligned bounding box clearly includes parts of a neighboring lane.

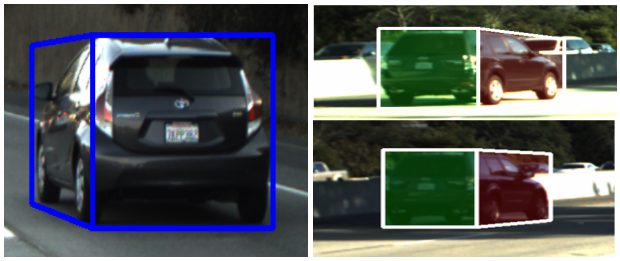

Figure 6. Left: Typical annotation of a car using a rear rectangle and a trapezoid for the side. Note the shared edge reduces the number of required points to six. Right: Visible difference in orientation of the upper side edge.

Dataset	# Images	# Vehicles	Resolution	Label
[32] BDD100k	100,000	1,095,289	1280x720	pixel
[29] BoxCars	116,286	27,496	<200x200	3D-like
[5] Cityscapes	25,000	88,305	2048x1024	pixel
[7] KITTI	15,000	32,750	1392x512	3D
[28] Lisa Vehicles	2,200	8,217	704x480	AABB
[19] Mapillary	25,000	<175,000	>1920x1080	AABB
[17] Nexar	55,000	148,000	1280x720	AABB
[24] Synthia	200,000	pixel-level	960x720	pixel
[2] TME	31,850	135,100	1024x768	ABBB
[31] Udacity 1	9,423	72,064	1920x1080	AABB
[31] Udacity 2	15,000	93,086	1920x1080	AABB
Boxy	**200,000**	**1,990,806**	**2464x2056**	**3D-like**

Table 2. Overview of vehicle detection dataset sizes.

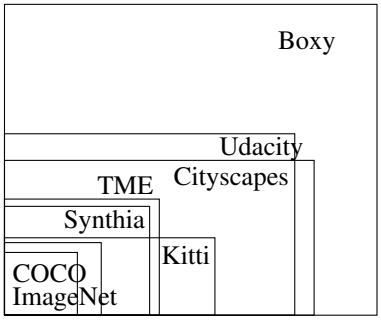

Figure 7. Visualization of different dataset resolutions.

played in 5) and therefore may impede planning capabilities. Pixel-level segmentations can be computationally intensive to process for planning methods and may be noisy.

Boxy contains 3D-like annotations with visible sides split into individual quadrilaterals. The annotations are image only and do not contain 3D points. For a simplified annotation process and quality control, we label vehicle rears with AABB and sides with trapezoids. Figure 6 displays example annotations. This simplification works for all vehicles within the dataset but does not for corner-cases such as vehicles positioned orthgonal to driving lanes.

One difficulty in the annotation process is the definition of the upper front. The upper side edge is supposed to align with the roof of the car, but with a variety of vehicles this can be ambiguous. Figure 6 displays different side annotations for the same car. 3D information is not accurate enough to fix the height for distant cars and having the upper and lower side edge parallel is not accurate. One possible fix could be to incorporate the images' vanishing points.

2.6. General Annotation Requirements

All vehicles going in the same direction as the camera have to be annotated. This includes on-ramps, off-ramps, and parallel roads. Most of the sequences are recorded on fully divided freeways which makes it unlikely for oncoming traffic to affect our trajectory.

All vehicles within the freeway are annotated as a single vehicle class. This includes passenger cars, trucks, campers, boats, car carriers, construction equipment, and motorcycles. The rear bounding box should contain the complete rear without containing the sides or front mirrors. For vehicles that carry or contain other vehicles, see Figures 5 and 9, only the larger vehicle has to be annotated.

Partially occluded vehicles have to be labeled with an estimate of their complete size and position. Finally and importantly, only vehicles that can clearly be seen and identified as vehicles are annotated. Especially tiny, blurry objects where it is unclear if they are vehicles are not added.

2.7. Dataset Evaluation and Comparison

Boxy is one of the largest vehicle detection datasets in terms of number of images, annotated vehicles, and vehicles per image, as displayed in Table 2. To our knowledge, only general datasets like the ILSVRC Detection [25], OpenImages [13], and COCO [14] surpass it in terms of number of images.

Usually, cameras for automotive systems have a resolution of one to two megapixels [5, 7, 17, 2, 31]. With five megapixel images, we provide a higher resolution than most datasets, see Table 2. Some of the annotated vehicles are larger than the complete images in competing datasets. Figure 7 gives a visual comparison of the different image reso-

lutions. Additionally, the ratio of object to image size is on average only 0.3% compared to 1.0% in Cityscapes, 1.65% in Kitti, and 17% in Imagenet.

Another distinct feature of our dataset is the 3D-like bounding boxes. The Kitti annotations exceeds these by having real 3D points, but do not reach the same annotation distance.

However, we group all types of vehicles into a single class, do not offer annotations in urban environments, simplify the annotations slightly, and do not have 3D information. Boxy also does not offer the highly accurate calibrations and sensor-set that Kitti has to offer or the pixel-level semantic segmentations that are available in Cityscapes and BDD100k. A small subset of our annotations are incorrect and the level of detail in the annotations may slightly vary between images. Overall, the dataset should be one of the largest and most challenging for object detection and especially vehicle detection.

3. Vehicle Detection Baselines

For our benchmarks, we split the dataset into training, validation, and test sets such that no recordings are split and a variation of conditions is reflected in the test set, see Table 1. The starting benchmarks will cover 2D, 3D-like, and real-time detections with the test set's annotation being private. All benchmarks are initially evaluated based on average precision.

3.1. AABB Baselines

Over the last years a variety of object approaches, for example, Overfeat [27], R-CNN [9], Fast R-CNN [8], Faster R-CNN [22], the YOLO architectures [20, 21] and the Single Short Detector (SSD) [15] were developed. For these general methods, the underlying base networks can be selected based on accuracy, speed, latency, convience and memory requirements. The different base networks can range from a MobileNet [26] over the ResNet familiy [11] up to the Inception [30] and NASNET architectures [33]. Additionally, there are a number of image-based 3D box specific methods [18, 4].

As our baseline methods, we select an SSD [15] with MobileNet V2 [26] for speed and a Faster R-CNN [22] with NASNET-A (6@4032) for a higher accuracy. We train both networks using the Tensorflow Object Detection API [12] and initialize them using models pretrained on COCO [14].

3.2. Refinement by Keypoint Regression

A second step optimizes the axes-aligned bounding boxes to better represent the real shape of vehicles. We train a MobileNet V2 [26] to detect the eight visible corner points of a 3D box for each detected vehicle. For this, all detected objects are scaled to an input resolution of 256×256 pixels and used as input to the second network. We pose the regression problem as a classification problem by sampling the bounding box corners uniformly over the image. For the baseline model, we use 50 bins and add a random margin of 10% to 30% to account for inaccuracies in the detection step. During inference, a constant margin of 20% is added.

Figure 8. Left: Sample detections within image at dusk. Right: The trained model incorrectly classified the left side as visible

The geometric representation of the annotations, described in Section 2.5, namely that each vehicle is represented by a rectangle with connected trapezoid allows us to reduce the number of values to regress. The rear can be represented by two points at opposing corners of the rectangle, totaling in four scalar values. The sides are added to the rear by two additional points that share a common axis, resulting in three additional scalar values.

$$L = L_V + L_R \tag{1}$$

$$L_V = -\sum_{v_i}^{V} \log(v_i) v_{i_l} \tag{2}$$

$$L_R = -\sum_{r_i}^{R} v(r_i) r_{i_l} \log\left(\frac{e^{r_i}}{\sum_{r_s}^{R} e^{r_s}}\right) \tag{3}$$

We minimize the overall loss L (1) which is the sum of a visible side classification loss L_V (2) and the regression loss L_R. For each corner point bin $r_i \in R$, the cross-entropy loss is calculated and summed up if the corner point belongs to a visible side, i.e., $v(r_i)$ is 1.

We add a classifier to determine which of the sides are visible for a given object. The visibility of each vehicle side V is posed as a binary classification problem and evaluated using binary cross-entropy as part of the visible side classification loss L_V in (2).

3.3. Baseline Results

We evaluate the detection methods based on the average precision (AP) based on an intersection over union (IOU) of at least 70%. Table 3.3 shows the accuracies and frames per second (FPS) on an Nvidia GTX 1080 TI for the different models on the test set. Each model receives a downscaled image with an input resolution of 1232x1028 and only is evaluated against objects that are larger than 5x5 pixels at that resolution.

	AP	FPS
AABB SSD MobileNet	29.5%	13.2
AABB Faster-RCNN NASNet	41.3%	0.5
Faster-RCNN NASNet + Refinement	43.4%	0.4

Table 3. Average precision and frames per second for the baseline models on the scaled test set.

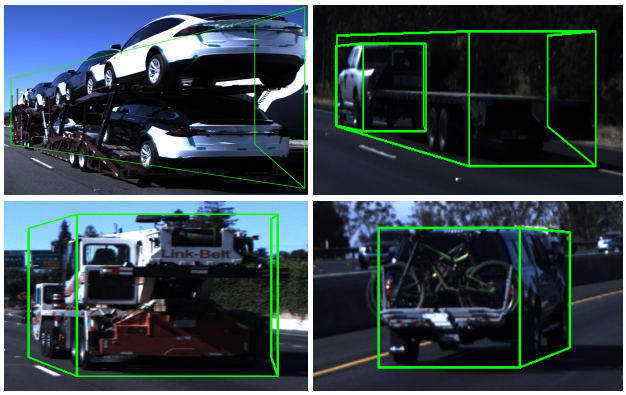

Figure 9. Detection examples on a few interesting cases. Depending on the required IOU, the first three larger vehicles may be correctly detected, but their sides are not classified correctly.

For a qualitative impression of the baseline results, we refer the reader to the supplementary video. Visually, the 3D detections tend to be far more appealing and lead to a noticeable improvement in detection accuracy. There are some reoccurring deficiencies in the detections such as incorrectly classified visible sides as displayed in Figure 8, detections in oncoming traffic, and false positives at greater distances. A few more challenging samples are shown in Figure 9.

Future work includes looking into different approaches, underlying models, increasing overall accuracy, and investigating speed/accuracy trade-offs.

4. Conclusion

We presented the Boxy vehicles dataset, the largest publicly available dataset for vehicle detection with almost 2 million annotated objects in 200,000 images. The small, 3D-like detections within 5 megapixel images in different weather and traffic conditions make for a challenging dataset. The average annotation only covers approximately 0.3% of its camera image.

With the dataset, we presented benchmarks for AABB, 3D-like, and real-time detections. The benchmark evaluation and website are fully open source so that additional metrics and challenges can be added. With enough feedback and submissions, we plan to extend the different objectives and metrics. We encourage all kinds of benchmark suggestions.

For future datasets, we would look into annotating objects in multiple sensors and additionally urban environments.

There are a number of research directions to explore with this dataset such as speed, accuracy trade-off analyses, testing different input resolutions, combining different datasets, inferring vehicle control based on camera images, domain adaptation, and better metrics than average precision for automotive applications.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[2] C. Caraffi, T. Vojíř, J. Trefnỳ, J. Šochman, and J. Matas. A system for real-time detection and tracking of vehicles from a single car-mounted camera. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 975–982. IEEE, 2012. 2, 4

[3] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, 2017. 2

[4] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 5

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 2, 4

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 1

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 4

[8] R. Girshick. Fast r-cnn. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 1440–1448. IEEE, 2015. 5

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 5

[10] M. V. GmbH. mvbluefox3-2. https://www.matrix-vision.com/USB3-vision-camera-mvbluefox3-2.html, 2016. 3

[11] 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. 5

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama,

et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7311, 2017. 5

[13] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Open-images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 4

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 4, 5

[15] 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. 1, 5

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[17] N. Ltd. Nexar challenge 2. https://www.getnexar.com/challenge-2, 2017. 2, 4

[18] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017. 5

[19] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017. 2, 4

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 5

[21] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017. 1, 5

[22] 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. 1, 5

[23] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 2

[24] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. 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*, 2016. 2, 4

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 4

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 5

[27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014. 5

[28] S. Sivaraman and M. M. Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):267–276, 2010. 2, 4

[29] J. Sochor, A. Herout, and J. Havel. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 4

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 5

[31] Udacity. Udacity self-driving car. https://github.com/udacity/self-driving-car/tree/master/annotations, 2016. 2, 4

[32] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 2, 4

[33] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 5