Granular Analysis of Pretrained Object Detectors

Eric Xue
University of Toronto
Toronto, Canada
e.xue@mail.utoronto.ca

Tae Soo Kim
Johns Hopkins University
Baltimore, USA
tkim60@jhu.edu

***Abstract*—Object detectors have become the fundamental building blocks of many real-world machine learning applications. Even though different problem domains require their own unique object detector specifications, it is common practice to take a pretrained object detector off the shelf and either use it as-is or fine-tune it with limited amounts of labeled training data. However, the image distribution that such object detectors are trained on is more often times than not different from the targeted problem domain of interest. In this work, we scrutinize whether existing state-of-the-art object detectors have the ability to generalize across different domains. Specifically, we evaluate whether widely used pretrained state-of-the-art objectors such as Faster-RCNN and YOLOv3 generalize to images sampled from an autonomous vehicle application. For this purpose, we evaluate the performance of detectors on localizing humans and vehicles on images from the KITTI dataset and report results of detailed subgroup analysis on multiple factors. Our analysis shows that the detectors exhibit different levels of performance on varying levels of object-object occlusion and object size. Moreover, we report the performance drop of the object detectors with different image-altering hazardous factors.**

***Index Terms*—(Object Detection, Subgroup analysis, Autonomous Vehicle)**

I. Introduction

The ability to detect objects from images is arguably one of the most important aspects of many modern day vision based applications. The growing capability of recent object detectors [1]–[5] enabled them to be applied to a diverse set of problem domains ranging from applications in autonomous vehicles [6] to medical image analysis [7], only to name a few. What is common to all such object detectors is the use of a deep convolutional neural network based general purpose visual feature extractor backbone [8] combined with modules for detecting objects which often include a module responsible for localizing objects, a bounding-box regressor and a classifier [2].

Such modern state-of-the-art object detectors are very heavily parameterized neural networks which require large amounts of carefully labeled annotated data for training. Thus, the advances in curating larger datasets such as [9]–[11] with accurate object level annotations have fueled the progress in the field of visual object detection. Most of these large scale datasets consist of everyday images which cover a large set of common objects and serve as a general purpose pretraining datasets for object detectors. However, many realistic application of object detectors such as autonomous vehicles focuses on much narrower distribution of images and objects. For example, in autonomous vehicle applications, vehicles and humans are observed from a particular point of view, all objects are viewed in an outdoor setting and actor positions and orientations adhere to a specific distribution which may be different to those of objects found in a common household. However, in practice we often assume that the performance of these object detectors readily transfers to our target applications. Therefore, many times the detectors are used as-is or fine-tuned using small amounts of available training data.

In this work, we wish to scrutinize this assumption by performing an in-depth subgroup analysis of the performance of commonly used pretrained object detectors such as Faster-RCNN and YOLOv3. We take the networks pretrained on MS COCO and test them on images from KITTI to test the detectors' ability to generalize to images drawn from a different distribution. More specifically, we are interested in identifying in detail the strengths and weaknesses of the model with respect to different subgroups. We choose object occlusion level and object size as the main subgroups that we perform analysis on. We identify vehicles and humans to be the most important object types for many applications and perform subgroups analysis separately for the two object classes. Our analysis shows that our common assumption that object detectors transfer well across datasets is not always true. We find that the object detectors perform better for certain subgroups than others and the results provide helpful insights into potential directions to improve existing models as well as datasets.

II. Related Work

Object detections: In the object detection literature, there are two mainstream philosophies in designing object detectors. The first is a region proposal based architectures where the model first generates region proposals and later classifies them. The most notable architecture that follows this pipeline is the Faster-RCNN [2] and the Mask-RCNN [12]. The second type includes object detectors that pose the detection problem as a regression or classification problem by jointly predicting categories and locations directly. For this case, YOLO [3] is a well known architecture with very efficient implementations available. We refer to [13] for a thorough survey on the field of object detection. In this work, we perform our subgroup analysis on the two representative object detectors, Faster-RCNN and YOLOv3.

Faster R-CNN [2] adopts a new region proposal approach, using a Region Proposal Network that share convolution features with the Fast R-CNN detector, rather than using the traditional Selective Search algorithm. The approach allows for the increase in efficiency and accuracy due to the increased region proposal quality.

YOLOv3 [14] is an one-stage detector based on its predecessor: YOLOv2. It simultaneously predicts the class and location, making it considerably faster than some other state-of-the-art methods. YOLOv3 comes with many architectural changes compared to YOLOv2, such as multilabel classification instead of softmax and a new feature extraction network (Darknet-53), which is slower than the previously used Darknet-19, but much more accurate.

Measuring the performance of object detectors: The field of object detection has converged towards an universal metric to measure object detector performance, namely the mean Average-Precision (mAP). One computes mAP by measuring the area under the precision-recall curve for detections over multiple intersection-over-union (IoU) thresholds with which is then averaged over all classes to produce a single evaluation criteria [15]. While mAP provides a great overview of the general performance of a detector on a particular dataset, it hinders analysis of detection errors at a granular level. For example, a practitioner cannot intuitively isolate certain error types and cannot identify different factors that contribute to detection errors. In this work, we isolate different subgroups within the dataset, observe how the performance of a detector is affected by different image perturbations for each subgroup and thus provide much granular analysis of strengths and weaknesses of object detectors.

Analyzing strengths and weaknesses of object detectors: There has been many attempts to diagnose the errors of deep learning based object detectors in recent years. The seminal work of [16] provided tools necessary to perform a more in-depth analysis of false positive detections of the detector. Tools such as the COCO evaluation toolkit[1] extends the analysis of [16] by analyzing errors with respect to their effects on model's precision-recall characteristics. There also exists a recent work [17] that improves usability and interpretability while decreasing dataset dependency of the error analysis. However, all analyses mentioned above assume that the object detector is trained adequately on the target dataset using a adequately large set of annotated training images from the same dataset. However, there lacks detailed error analysis on widely used pretrained object detectors in their off-the-shelf form. In this work, we expose detailed performance characteristics of popular pretrained object detectors and compare how various image perturbations effect detector performance. We also provide granular analysis of object detector performance per different object subgroups such as object sizes and occlusion levels.

Image datasets: The pretrained object detectors used in the experiment were both trained on MS COCO [9]. It contains a total of 2.5 million labeled instances over 328 thousand images covering 91 object types in their natural context. All images in MS COCO were collected from Flickr, a website hosting videos and photos shot by photographers, meaning most images in MS COCO are taken from a typical human eye perspective.

On the other hand, we are testing the object detectors on KITTI. KITTI is a dataset focused on providing annotated images for training and evaluating models in mobile robotics and autonomous driving applications [18]. Its 2D object detection benchmark contains 80,256 labeled instances across 14999 images in total. Unlike MS COCO, all images in KITTI were collected by high-resolution cameras mounted on a vehicle while driving around a mid-sized city. This implies that there will be fundamental differences between the context and perspective of the images between MS COCO and KITTI.

III. Granular Analysis of Pretrained Object Detectors

Existing methods assess object detector performance using mAP which provides an overall summary of detector performance for all defined object classes averaged over multiple operating points. Instead, we wish to provide a more granular analysis of detector performance by measuring the effect of isolated factors. Thus, we fix the operating point of detectors at intersection-over-union (IoU) threshold of 0.5 but measure the performance of the detector across various subgroups. In this section, we define the subgroups and various image perturbations that we perform to measure how robust or fragile the pretrained object detectors are for each category.

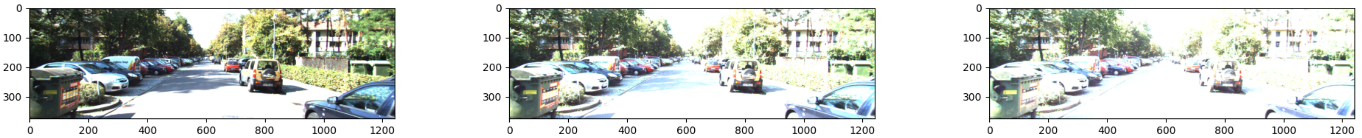
A. Area Under the Curve as the Performance Metric

Area Under the Curve (AUC) is a commonly used performance metric for classification problems. We plot Receiver Operating Characteristic (ROC) curves showing precision-recall trade-offs for each subgroup and type of image perturbation. The precision and recall values of each image is calculated independently. The average precision and recall among images contained in a subgroup is used to plot its precision-recall curve. We then report AUC as the summary of the detector performance.

B. Defined Subgroups

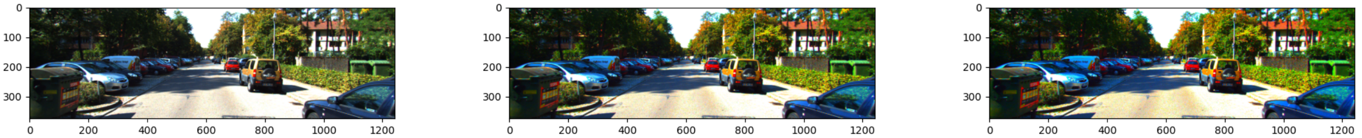
To allow us to examine the performance of the pretrained object detectors in detail, we divided the dataset into many subgroups. On a more general level, all the objects in the dataset are divided into two subgroups: cars and humans. These two subgroups are arguably the most crucial prediction targets in autonomous vehicle applications. Those that aren't in either of the subgroups are excluded from the experiment. Among cars and humans, each object is further categorized into different subgroups according to their occlusion level and relative object size within cars/humans. The KITTI dataset provides each ground truth with four possible occlusion labels: visible, semi-occluded, fully occluded, and truncated, with

[1] http://cocodataset.org/#detection-eval



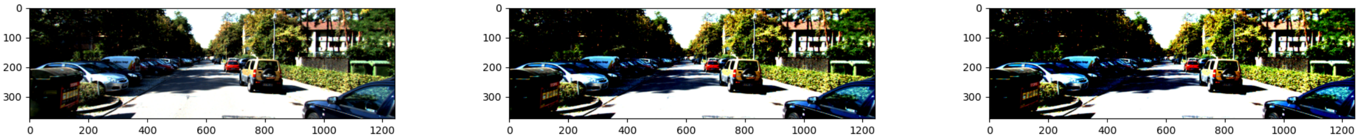
(a) Brightness: 2 (b) Brightness: 4 (c) Brightness: 6

Fig. 1: Effect of Brightness Transformations on the Image



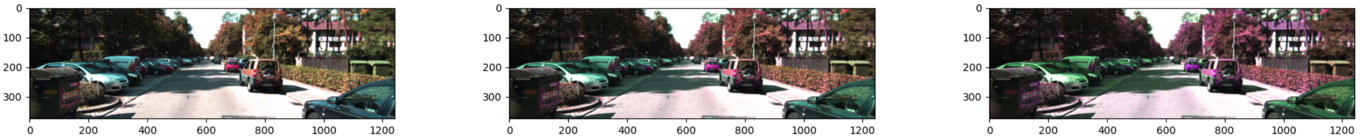
(a) Saturation: 2 (b) Saturation: 4 (c) Saturation: 6

Fig. 2: Effect of Saturation Transformations on the Image



(a) Contrast: 2 (b) Contrast: 4 (c) Contrast: 6

Fig. 3: Effect of Contrast Transformations on the Image



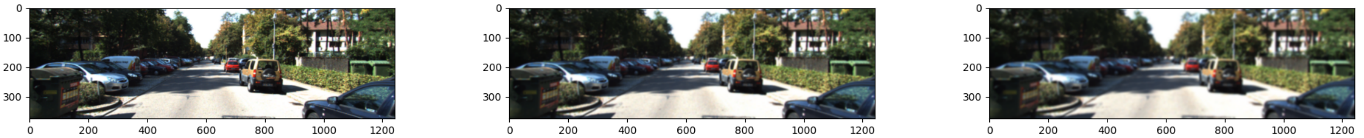
(a) Hue: 0.15 (b) Hue: 0.30 (c) Hue: 0.45

Fig. 4: Effect of Hue Jitters on the Image

each label assigned an occlusion level between 0 to 3, respectively. The occlusion level of an object in an image is defined by the average occlusion level of all ground truths in a given image, rounded down. Objects in a given image are labeled as either visible, semi-occluded, or fully occluded according to their average occlusion levels. No image contained only truncated objects, hence no objects were labeled as truncated. Objects in an image is labeled as being large if the average object size is in the top 50% among the all objects in that particular subgroup. Conversely, objects in an image with an average object size in the lower 50% are classified as being small.

C. *Performed Image perturbations*

We selected a range of different image perturbations to discover how robust pretrained object detectors are when the image quality isn't ideal. More specifically, the image perturbations we employed includes Gaussian blur, brightness scaling, contrast scaling, saturation scaling, and hue jitter. Although color jitter is a common technique in data augmentation, the result it produces would be too inconsistent for analysis, therefore all but hue transformation were done by applying a fixed scaling factor. Three scaling factor values are used for this experiment, ranging from 2, 4, and 6. Hue transformation was done using hue jitter because setting the entire dataset to a certain hue is unquantifiable. The hue jitter value is chosen uniformly from a range of $[-n, n]$, where n is the jitter factor. Since a value of -0.5/0.5 is enough to transform the hue to the opposite side of the color wheel, 0.5 is regarded as the maximum value for the jitter factor. Hence, in this experiment, the values of jitter factor was chosen to be 0.15, 0.30, and 0.45. Lastly, Gaussian blur takes in two parameters: sigma and the corresponding kernel size. The sigma values used were chosen to be 1, 2, and 3, and the kernel size that matches each sigma value ranges from 5, 9, 13, respectively.



(a) Gaussian Blur: σ = 1

(b) Gaussian Blur: σ = 2

(c) Gaussian Blur: σ = 3

Fig. 5: Effect of Gaussian Blur on the Image

Occlusion Level	Vehicles														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	σ = 1	σ = 2	σ = 3	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.686	0.527	0.303	0.675	0.616	0.537	0.562	0.510	0.469	0.697	0.674	0.649	0.696	0.704	0.710
semi-occluded	0.530	0.422	0.273	0.538	0.498	0.450	0.450	0.414	0.382	0.557	0.536	0.521	0.544	0.550	0.557
fully occluded	0.176	0.128	0.077	0.205	0.209	0.212	0.127	0.122	0.094	0.192	0.191	0.172	0.188	0.185	0.195
Object Size															
small	0.604	0.416	0.188	0.598	0.535	0.460	0.474	0.419	0.383	0.622	0.596	0.566	0.621	0.627	0.647
large	0.753	0.690	0.557	0.735	0.699	0.624	0.659	0.632	0.601	0.751	0.735	0.725	0.736	0.744	0.744

TABLE I: Overview of Faster R-CNN Performance on Vehicle Subgroup

Occlusion Level	Humans														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	σ = 1	σ = 2	σ = 3	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.312	0.242	0.156	0.325	0.235	0.138	0.258	0.201	0.171	0.349	0.338	0.322	0.338	0.338	0.338
semi-occluded	0.099	0.072	0.040	0.100	0.087	0.056	0.074	0.058	0.052	0.105	0.101	0.099	0.107	0.112	0.110
fully occluded	0.047	0.033	0.022	0.035	0.020	0.008	0.027	0.022	0.019	0.046	0.044	0.039	0.044	0.045	0.041
Object Size															
small	0.156	0.094	0.042	0.171	0.128	0.072	0.119	0.093	0.076	0.178	0.170	0.162	0.170	0.173	0.172
large	0.547	0.486	0.392	0.489	0.335	0.207	0.435	0.349	0.303	0.540	0.525	0.504	0.532	0.532	0.525

TABLE II: Overview of Faster R-CNN Performance on Human Subgroup

Occlusion Level	Vehicles														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	σ = 1	σ = 2	σ = 3	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.507	0.320	0.147	0.566	0.521	0.467	0.487	0.425	0.381	0.586	0.566	0.548	0.591	0.605	0.611
semi-occluded	0.430	0.299	0.153	0.474	0.436	0.395	0.395	0.351	0.311	0.473	0.460	0.454	0.481	0.481	0.481
fully occluded	0.186	0.121	0.046	0.229	0.181	0.142	0.140	0.082	0.078	0.218	0.214	0.212	0.223	0.213	0.224
Object Size															
small	0.436	0.240	0.082	0.502	0.453	0.403	0.417	0.348	0.311	0.526	0.494	0.478	0.525	0.542	0.554
large	0.606	0.481	0.320	0.620	0.589	0.544	0.560	0.525	0.478	0.635	0.635	0.626	0.641	0.643	0.643

TABLE III: Overview of YOLOv3 Performance on Vehicle Subgroup

IV. Results

In this section, we first report our findings regarding the performance of pretrained Faster R-CNN on KITTI without any finetuning. We experiment with how Gaussian blur and brightness/hue/contract/saturation transformations affect the model performance with respect to varying occlusion levels and object sizes. Tables I and II demonstrate that the effect of image perturbations is similar in both vehicles and humans, other than the fact that the model performance is generally lower for detecting humans. We visualize the effect of all image perturbations in Figures 1, 2, 3, 4 and 5.

We find that Gaussian blur has a large impact on the performance of the model. As the kernel size and sigma gets larger, we observe that there is large drop in the performance. The drop is also shown to be larger when sigma increases from 2 to 3 when compared to the increase from 1 to 2, suggesting that the impact of Gaussian blur increases exponentially as the level of blur increases. This increase is particularly evident in small objects; the highest level of Gaussian blur caused the largest difference in performance between large and small objects. On the other hand, although objects that are completely visible and semi-occluded seem to be rather robust against Gaussian blur, objects that are fully occluded seem to suffer a lot more.

In Tables I and II, we also report the performance of the detector across transformations in brightness, contrast, saturation and hue. The Faster-RCNN model is generally robust against these types of color jitter and the performance is generally higher than that of under Gaussian blur. However, there are subtle differences between the effect of each color transformation. Contrast impacted model performance the most, resulting in the lowest AUC scores among color transformations across all occlusion levels and object sizes. In contrast, hue jitter had the least impact on model performance, resulting in either similar or even higher performance in all subgroups.

Next, we report our findings regarding the performance of

Occlusion Level	Humans														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.224	0.162	0.092	0.242	0.212	0.160	0.177	0.141	0.118	0.238	0.238	0.219	0.237	0.237	0.237
semi-occluded	0.052	0.039	0.016	0.059	0.052	0.040	0.044	0.034	0.026	0.054	0.052	0.052	0.058	0.060	0.065
fully occluded	0.030	0.020	0.008	0.035	0.038	0.031	0.017	0.015	0.013	0.033	0.022	0.022	0.031	0.037	0.027
Object Size															
small	0.088	0.053	0.023	0.101	0.092	0.068	0.059	0.044	0.038	0.092	0.084	0.078	0.095	0.095	0.091
large	0.448	0.372	0.245	0.448	0.403	0.298	0.366	0.310	0.264	0.461	0.462	0.448	0.461	0.461	0.468

TABLE IV: Overview of YOLOv3 Performance on Human Subgroup

pretrained YOLOv3 without any finetuning on KITTI in Tables III and IV. Overall, YOLOv3 shows similar performance characteristics when compared against Faster R-CNN in many aspects. However, we observe that the general performance of YOLOv3 across all subgroups is lower than that of Faster R-CNN. While both being vulnerable to Gaussian blur, all other subgroups (including those from Faster R-CNN) only show a large performance drop from semi-occluded to fully occluded cases, but YOLOv3 already shows a large performance drop when going from visible to semi-occluded in the human subgroup. In the scope of color transformation, the results show that just like Faster R-CNN, YOLOv3 is also most vulnerable to contrast while being least affected by hue jitter.

V. Conclusion

In this paper, we studied the performance of widely used pretrained object detectors, Faster-RCNN and YOLOv3. There are important conclusions that can be made based on our experimental results. First, both detectors show a performance drop from detecting cars compared to when detecting humans. We suspect this is because the humans are inherently smaller than vehicles and this leaves less margins of error for the models to draw prediction boxes that meet the IoU threshold. Secondly, even as some levels of perturbation have been shown to greatly distort the image, the effect on performance is still minimal compared to the effect of high occlusion levels and variations in object sizes. This suggests that we should focus primarily on guaranteeing the model's consistency to detect occluded objects and smaller objects rather than potentially focusing on solving issues regarding color distortion and low-resolution images.

References

[1] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. 2015, vol. 28, Curran Associates, Inc.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, "Ssd: Single shot multibox detector.," in *ECCV (1)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds. 2016, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, Springer.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," *CoRR*, vol. abs/2005.12872, 2020.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo, "Automatic brain tumor detection and segmentation using u-net based fully convolutional networks," in *Medical Image Understanding and Analysis*, María Valdés Hernández and Víctor González-Castro, Eds., Cham, 2017, pp. 506–517, Springer International Publishing.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, "Microsoft coco: Common objects in context," 2014, cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[10] 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*, vol. 88, no. 2, pp. 303–338, June 2010.

[11] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," 2018.

[13] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, "Object detection with deep learning: A review," 2018, cite arxiv:1807.05511.

[14] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement," Tech. Rep., University of Washington, 04 2018.

[15] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman, "Tide: A general toolbox for identifying object detection errors," in *European Conference in Computer Vision (ECCV)*, 2020.

[16] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai, "Diagnosing error in object detectors," in *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., Berlin, Heidelberg, 2012, pp. 340–353, Springer Berlin Heidelberg.

[17] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman, "TIDE: A general toolbox for identifying object detection errors," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds. 2020, vol. 12348 of *Lecture Notes in Computer Science*, pp. 558–573, Springer.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.