Evaluating and Bench-marking Object Detection Models for Traffic Sign and Traffic Light Datasets

Ashutosh Mishra, Aman Kumar*, Shubham Mandloi*, Khushboo Anand*, John Zakkam*, Seeram Sowmya*, and Avinash Thakur

OPPO Research and Development Center, Hyderabad, India
{ashutosh.mishra1,aman.kumar1,shubham.mandloi,khushboo.anand,avinash.thakur}@oppo.com
{sowmya.seeram,johnzakkam2509}@gmail.com

Abstract. Object detection is an important sub-problem for many computer vision applications. There has been substantial research in improving and evaluating object detection models for generic objects but it is still not known how latest deep learning models perform on small road scene objects such as traffic lights and traffic signs. In fact, locating small object of interest such as traffic light and traffic sign is a priority task for an autonomous vehicle to maneuver in complex scenarios. Although some researchers have tried to investigate the performance of deep learning based object detection models on various public datasets, however there exists no comprehensive benchmark. We present a more detailed evaluation by providing in-depth analysis of state-of-the-art deep learning based anchor and anchor-less object detection models such as Faster-RCNN, Single Shot Detector (SSD), Yolov3, RetinaNet, CenterNet and Cascade-RCNN. We compare the performance of these models on popular and publicly available traffic light datasets and traffic sign datasets from varied geographies. For traffic light datasets, we consider LISA Traffic Light (TL), Bosch, WPI and recently introduced S2TLD dataset for traffic light detection. For traffic sign benchmarking, we use LISA Traffic Sign (TS), GTSD, TT100K and recently published Mapillary Traffic Sign Dataset (MTSD). We compare the quantitative and qualitative performance of all the models on the aforementioned datasets and find that CenterNet outperforms all other baselines on almost all the datasets. We also compare inference time on specific CPU and GPU versions, flops and parameters for comparison. Understanding such behavior of the models on these datasets can help in solving a variety of practical difficulties and assists in the development of real-world applications. The source code and the models are available at https://github.com/OppoResearchIndia/DLSOD-ACCVW.

Keywords: Traffic Sign Detection · Traffic Light Detection · CascadeR-CNN · CenterNet · RetinaNet · Yolov3 · SSD · Autonomous Driving

* equal contribution

1 Introduction

Detecting traffic light/traffic sign is one of the most difficult problem for the perception module of an autonomous agent due to factors such as size, imaging resolution, weather etc. Fundamentally, designing systems that can help in achieving automatic TLD/TLR/TSD/TSR (traffic light detection/ traffic light recognition/ traffic sign detection/ traffic sign recognition) would be extremely helpful in substantially reducing the number of fatalities around the world. However, there are multiple hindrances in performing TLD/TLR or TSD/TSR. Some of these hindrances could be size of traffic lights/traffic sign, view distances, weather conditions and objects of confusion such as street lights, house bulbs or other light sources. Pre-deep learning era for TLD/TLR or TSD/TSR uses a combination of techniques such as color thresholding, template matching, pixel clustering etc [13,39,14,19,34]. But they have proven to be robust enough under certain specific conditions to give successful results.

Advancement in deep learning has led to many improvements in the generic object detection performance. The main objective of this paper is to evaluate the performance of recent deep learning based object detection models in detecting small scale objects related to scene understanding task for autonomous driving viz. traffic lights and traffic signs. This evaluation is of utmost importance in order to check the performance metrics that can lead to decrease in false detection rate in real time scenario. For this, we compare all the publicly available datasets on a common benchmark. We club the fine classes to their respective meta class and then compare the metrics of different object detection models. The process of conversion from fine class to meta class is explained in Section III. Therefore, the main contributions of our work are:

- Evaluation and analysis of various state-of-the-art deep learning object detection models on traffic light datasets and traffic sign datasets - LISA TL [20], WPI [5], BOSCH [2], S2TLD [37], LISA TS [26], TT100K [41], GTSD [18] and MTSD [9] (See Figure 1).
- We evaluate the performances of these datasets on common meta classes with six object detectors: Faster R-CNN [30], SSD [23], RetinaNet [22], Yolov3 [29], CenterNet [7] and Cascade-RCNN [4].

To our knowledge, this is the first comprehensive work to compare publicly available traffic sign and traffic light datasets from various geographies on the common meta classes. We choose these models for evaluation because of two reasons. First, they are widely accepted and publicly available across all the deep learning frameworks among industry and academia. Second, the inference time of these models make them deployable ready for real time applications. This evaluation necessitates the need for understanding how state-of-the-art object detection algorithms perform on small road scene objects such as traffic sign and traffic light which is critical for developing practical applications such as self driving vehicles.

Fig. 1. Sample images with ground truth from all traffic light and traffic sign datasets used for bench-marking. The first row (from L to R) contains images from all traffic light datasets: LISA-TL, BOSCH, WPI, S2TLD. The second row (from L to R) contains images from all traffic sign datasets: LISA-TS, GTSD, TT100K and MTSD *(Best viewed when zoomed)*.

2 Related Work

We are interested in bench-marking the performance of various object detection models on publicly available traffic sign and traffic light datasets. The performance obtained can be represented as the state-of-the-art on meta classes of these datasets.

Dataset	TS/TL	Geography	Average Resolution	Training Images	Test Images	Original Classes	Meta Classes
LISA TL [20]	TL	US	1280×960	20535	22481	7	3
WPI [5]	TL	US	1024×2048	1314	2142	21	2
BOSCH [2]	TL	US	1280×720	5093	8334	15	4
S2TLD [37]	TL	China	1920×1080	744	244	5	5
LISA TS [26]	TS	US	880×504	5027	1571	47	15
TT100K [41]	TS	China	2048×2048	6107	3073	128	3
GTSD [18]	TS	Germany	1360×800	600	300	43	4
MTSD [9]	TS	Diverse	3407×2375	36589	10544	313	4

Table 1. Statistics of different traffic light and traffic sign publicly available datasets. **Original Classes**: The classes originally present in the dataset folder. **Meta Classes**: The original classes converted to base class. For LISA TS [26] and S2TLD [37] datasets, the split of training and testing has been created by the authors. The original dataset has more images but we only consider the ones with proper annotations. The test annotations for MTSD [9] and GTSD [18] are not publicly available so we consider val set as test set for reporting results.

2.1 Traffic Sign Detection

The objective of performing traffic sign detection(TSD) is to get the exact locations and sizes of traffic signs. The well-defined colors and shapes are two main cues for traffic sign detection. There have been various works on detecting traffic signs using traditional methods employing histogram of oriented gradients

(HOG) [39,14,19], SIFT (scale invariant feature transform) [15], local binary patterns (LBP) [8]. The main working concepts in TSD using traditional methods revolves around using a sliding window based or a region of interest (ROI) based approach. HOG and Viola-Jones-like detector [25] are examples of sliding based methods. Wang et al. [35] uses a hierarchical sliding window method to detect traffic signs.

With the advent of deep learning and the rise in the use of convolutional neural networks, there have been many works, such as Zhu et al. [40] developed a strategy to detect and recognize traffic signs based on proposals by the guidance of fully convolutional network. R-CNN using a proposal strategy gave good results on a small scale dataset [21]. R-CNN along with an object proposal method [42] was used to further improve the performance on the same dataset. In 2016, [1] proposed a method that implements the multi-scale sliding window technique within a CNN using dilated convolutions. In 2019, the multiscale region-based convolutional neural network (MR-CNN) [24] was proposed for small traffic sign recognition, where a multiscale deconvolution operation was used to upsample the features of deeper convolution layers that were concatenated with those of the shallow layer directly to construct fused feature map. Thus, the fused feature map could generate fewer region proposals and achieve a higher recall rate. multiresolution feature fusion network exploiting deconvolution layers with skip connecting and a vertical spatial sequence attention module was designed 501 for traffic signs detection.

2.2 Traffic Light Detection

Conventional methods of traffic light detection include selecting confident proposals from a probable candidate set of traffic light's generated using color and shape information.

In conventional traffic light detection(TLD), a candidate set of TL is, typically, generated using the colour and shape information [34,13]. Once the candidate TL's are identified, [13] and [6] employ Adaboost algorithm and other morphological operations to segment out the TL regions. But the disadvantage of using such algorithms using hand-crafted features is the lack of generalization ability of methods. Subsequently, methods such as HOG or SIFT tend to lose information which might help in the required task. This can lead to a very lower detection performance.

There have been many attempts to perform traffic light detection using modern convolutional networks as well. For instance, Weber et al. [36] used a convolutional network for traffic light detection modifying Alexnet network. The output of the network is a segmented image which is then given to a bounding box regressor for detection of traffic lights. In [31], the authors use a combination of SVM and CNN for the combined task of traffic light detection and recognition. Similarly, Behrendt et al. [3] uses a combination of detection, tracking, and classification using a convolutional neural network. Yudin et al. [38] propose another a fully convolutional network for traffic light detection. Heat-map is obtained highlighting areas of interest followed by a clustering algorithm to obtain

final traffic light bounding boxes. Besides being a transfer learning approach, it has a very low precision of detection as compared to SSD-based models [27].

Dataset	Meta Classes	Tr. Instance Count	Te. Instance Count
LISA TL [20]	Go	24182	25222
	Stop	26089	30963
	Warning	1555	1464
WPI [5]	Green	1323	2763
	Red	1878	802
BOSCH [2]	Green	5422	7569
	Yellow	444	154
	Red	4164	5321
	Off	726	442
S2TLD [37]	Green	478	164
	Yellow	43	16
	Red	761	239
	Off	2	1
	Wait-on	178	64

Table 2. Tr: Train, **Te**: Test. Publicly available traffic light datasets, their classes along with and it's corresponding number of instances per class. The counts per class for S2TLD [37] is based on the dataset split generated by authors.

3 Deep learning for Object Detection

With the success of convolutional neural networks to outperform traditional methods on classification task, there have been similar trends for other computer vision tasks such as object detection. OverFeat [32] is an example of such a network which outputs bounding boxes along with the scores using a deep network in a sliding-window fashion. Later, R-CNN [12] was proposed which helped in increasing the detection accuracy and was faster than the previous counterparts. The main disadvantage of using R-CNN is that it is expensive both in time and memory because it executes a CNN forward-pass for each object proposal without sharing computation. Spatial Pyramid Pooling Network (SPPNet) [16] was proposed to improve R-CNN efficiency by sharing computation. Due to the multi-stage pipeline in SPPNet, the whole process of detection becomes quite slow. Moreover, the parameters below the spatial pyramid pooling layer cannot be updated while training. After SPPNet, Fast R-CNN [11] was introduced, which proposes a new training algorithm that provides solutions to fix the disadvantages of R-CNN and SPPNet by training in a single-stage using a multi-task approach. But the main bottleneck in this approach is the candidate proposal strategy which is still different from the network training process. To overcome such bottleneck, Faster R-CNN [30], replaced the use of Selective Search with a Region Proposal Network (RPN) that shares convolutional feature maps with the detection network, thus enabling nearly cost-free region proposals. To improve the performance of Faster-RCNN even further, Cascade-RCNN [4]

was introduced. It consists of a series of detectors sequentially feeding in to the output of previous detector trained with increasing threshold values making it very selective for false positives. All these approaches discussed so far are multi-stage methods where there are two or multiple pipelines involved. There are other family of networks known as single stage networks including Single Shot MultiBox Detector (SSD) [23], YOLOv3 [29] and RetinaNet [22] which detects objects using a fully convolutional network rather than having separate tracks for detection and classification. This ability leads to a much faster object detection. Duan et al. have also proposed using anchor less technique for object detection which detects each object as a triplet of keypoints [7].

Standard object detector approaches can be broadly classified in two categories: (i) two-stage object detectors, (ii) one-stage object detectors. Two-stage object detectors combine a region-proposal step, region classification and regression step. On the contrary, one-stage detectors output boxes without a region proposal step. Two-stage and one-stage object detectors can also be called as anchor based since these models employ anchors to perform the detection. For bench-marking different traffic light and traffic sign datasets, we select publicly available and widely used Faster RCNN (2-stage), Yolov3 (one-stage), RetinaNet (one-stage) and SSD (one-stage) and Cascade-RCNN (mutli-stage) networks. Apart from the anchor based models, we also use CenterNet, anchor-less approach to bench-mark the datasets and analyse the quantitative results obtained. As mentioned earlier, we select these models for evaluation because of two reasons. Firstly, these models are known for real time performance. Secondly, these models are widely accepted in academia and industry. Almost all these models have been designed in such a way that they infer in near real-time and also validated through our experiments.

4 Experiments

The aim of the experimentation is to bench-mark different open source traffic light and traffic sign datasets against various state-of-the-art deep learning based object detection models. We consider datasets pertaining to different geographies in order to understand how these deep neural networks perform in different conditions.

4.1 TL and TS datasets

We consider four publicly available datasets for traffic light performance evaluation namely: LISA Traffic Light Dataset, BOSCH Traffic Light Dataset, WPI Traffic Light Dataset and the recently introduced SJTU Small Traffic Light Dataset (S2TLD). For traffic sign detection performance, we consider the following publicly available datasets: TT100K Dataset, Mapillary Traffic Sign Dataset, LISA Traffic Sign Dataset and German Traffic Sign Dataset. Figure 1 has representative images from all the datasets used for the bench-marking in this paper.

All the details for the respective datasets can be obtained from Table 1. We have merged all the fine classes to their respective meta class for all the datasets that we consider. For instance, LISA Traffic Light dataset contains seven classes namely, "*go*", "*goForward*", "*goLeft*", "*warning*", "*warningLeft*","*stop*" which were converted namely to "*go*", "*warning*", "*stop*" for fair comparision amongst the common classes across various datasets. More details regarding the meta classes for various datasets is given in Table 2 and Table 3.

4.2 Training Setup

For training these models, we deploy frameworks such as Detectron2 [31] and MMDetection [5] which are modular and easy to train, validatation and testing on custom datasets. The frameworks have been well documented and implemented in Pytorch [28] deep learning framework. The benchmarking was carried on a Linux machine having 2 Tesla V100 GPU's. Each of the detector model have been trained with some fixed parameters for fair experimentation and trained until convergence. The batch size is set to 4 with a learning rate of 0.00025, momentum of 0.9, and weight decay factor of 0.0001. For FasterRCNN and CascadeRCNN, the backbones considered are Resnet101 and Resnet50. The backbones used for Yolov3 and SSD are Darknet-53 [29] and VGG16 [33] with input resolution of 608x608 and 512x512 respectively.

4.3 Performance Evaluation

The evaluation of all the object detection performance models is done in terms of precision, recall and mean average precision using intersection-over-union (IoU). The mAP calculation is done based on the definition for the Pascal VOC 2007 competition with IoU threshold of 0.5 [10]. Equation (1) describes the formula of the calculation of mean average precision metric at 0.5 threshold. AP_i is the average precision per class in the dataset.

$$mAP@0.5 = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{1}$$

5 Results and Discussion

For the bench-marking task, we use YoloV3, Faster-RCNN, RetinaNet, SSD, CenterNet and Cascade-RCNN object detection model to serve as baselines.

5.1 Traffic Light Results

Table 4 contains quantitative results on the test set using Faster-RCNN, Yolov3, CenterNet, SSD, RetinaNet and Cascade-RCNN. We observe that almost on all datasets, there is a competition of scores between CenterNet and CascadeRCNN. Individually, WPI dataset has ground truth annotations lying in the range of

Dataset	Meta Classes	Tr. Instance Count	Te. Instance Count
LISA TS [26]	Warning	2611	651
	Prohibition	1459	360
	Speed-Limit	107	24
	Stop	1493	362
	Yield	187	49
	School	108	26
	School-Speed-Limit25	81	24
	Zone Ahead	55	14
	Ramp-speed-advisory	41	12
	Round-about	35	12
	Curve-left	27	12
	No-Left-Turn	25	12
	Thru-Traffic-Merge-Left	22	5
	Do-Not-Enter	19	4
	No-Right-Turn	14	4
GTSD [18]	Prohibitory	299	97
	Mandatory	84	30
	Danger	116	40
	Other	143	43
TT100K [41]	Warning	912	456
	Prohibitory	12393	6179
	Mandatory	3444	1626
MTSD [9]	Complementary	9082	1323
	Information	6507	948
	Regulatory	31574	4593
	Warning	14328	2073
	Others	118749	17209

Table 3. Tr: Train, **Te**: Test. Publicly available traffic sign datasets, their classes along with and it's corresponding number of instances per class. For MTSD [9] and GTSD [18], we consider validation set as test set since test set annotations are not publicly available.

Table 4. Results of various object detection models on all traffic light datasets using Faster R-CNN(FRCNN), SSD, Yolov3(Y3), RetinaNet(RN) and CenterNet(CN) and Cascade-RCNN(CRNN) . Mean Average Precision(mAP) at 0.5 threshold is indicated for different classes of the respective datasets. CPU and GPU inference time per image(in seconds) is also indicated in this table. The values in bold font represent the best results in each category across all methods on the respective datasets. Empty row values indicate that meta class is absent in the respective dataset.

	WPI						**LISA TL**					
Class	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**
go	68.13	**91.80**	84.0	43.14	87.50	77.07	54.85	55.70	46.50	54.42	59.50	**62.27**
stop	49.83	**85.40**	75.50	13.09	81.90	81.23	35.40	53.50	8.80	40.61	**58.60**	45.36
warning	-	-	-	-	-	-	12.00	36.30	31.60	16.66	32.30	**39.63**
off	-	-	-	-	-	-	-	-	-	-	-	-
wait-on	-	-	-	-	-	-	-	-	-	-	-	-
mAP@0.5	58.98	**88.60**	79.80	28.12	84.70	79.15	34.09	48.50	29.00	37.23	**50.10**	49.09
GPU Inf Time (sec)	0.70	0.03	0.13	0.08	0.25	0.08	0.69	0.02	0.11	0.07	0.20	0.06
CPU Inf Time (sec)	14.32	3.24	7.27	2.24	5.77	5.79	14.14	3.00	5.15	4.05	5.84	6.65

	BOSCH						**S2TLD**					
Class	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**
go	61.09	73.20	50.10	66.76	**83.50**	67.91	85.89	87.50	91.10	89.80	91.00	**94.15**
stop	60.15	64.40	71.70	46.39	79.00	**79.87**	85.73	87.5	92.3	90.54	**91.00**	86.08
warning	27.66	8.50	64.50	11.98	48.30	**62.51**	58.20	18.60	81.20	53.52	65.80	**92.26**
off	0.01	0.00	0.10	0.00	**0.10**	0.00	0.00	0.00	0.00	0.00	**0.40**	0.00
wait-on	-	-	-	-	-	-	93.76	95.90	90.40	98.20	**98.40**	92.56
mAP@0.5	37.23	36.50	46.60	31.28	**52.70**	52.57	64.72	57.90	71.00	66.38	69.30	**73.01**
GPU Inf Time (sec)	0.69	0.03	0.12	0.08	0.27	0.07	0.60	0.03	0.14	0.13	0.25	0.08
CPU Inf Time (sec)	14.04	3.11	8.75	4.62	5.74	5.69	17.91	2.96	5.76	5.37	5.83	7.42

Table 5. Number of trainable parameters(in million) and number of floating point operations for all the object detection models.

OD Model	Backbone	Parameters (in million)	GFlops
FasterRCNN [30]	ResNet101	104.38	423.16
SSD [23]	VGG16	36.04	355.12
Yolov3 [29]	Darknet53	61.95	171.91
RetinaNet [22]	ResNet50	37.91	215.49
CenterNet [7]	ResNet18	14.44	44.78
CascadeRCNN[4]	ResNet50-FPN	69.10	214.44

medium and large box area. Hence, SSD performs comparatively better among all the competitive baselines because SSD focuses better on large sized objects. SSD's performance is followed by CenterNet which is an anchor-less approach that outperforms other anchor based approaches.

For LISA TL and BOSCH and S2TLD, majority of the ground truth TL instances are located very far away from the camera view, thus pertaining to a small annotation box area. In such scenario, anchors have to adjust a lot to cater the needs of the algorithm in order to match the location of the object of interest. However, CenterNet and CascadeRCNN, both perform better. The reason is that Centernet is an anchor-less approach, eliminating the anchor dependency while CascadeRCNN applies repeated RPN blocks for better detection at uniform thresholds. An observation to note is that class "off" has almost zero mAP even though the class instances are present in BOSCH and S2TLD datasets. The plausible reason is that the class "off" indicates the no traffic light is activated but the network is trained ideally to detect colors since individually in both the datasets, class "off" has very few instances compared to other classes.

Figure 2 shows some qualitative results on few selected frames of the respective traffic light datasets. Visually we infer that all the models are able to detect traffic light at different angles of rotation. On BOSCH dataset, Faster-RCNN and RetinaNet are able to detect the traffic light present on the left in the presence of occlusion and lighting in the image. On the other hand, Yolov3 and CenterNet are able to detect the traffic light on the left. All the models are able to detect horizontal lights for S2TLD dataset and vertical lights for LISA TL and WPI.

5.2 Traffic Sign Results

Table 6 contains quantitative results obtained on the test set using Faster-RCNN, YoloV3, CenterNet, SSD, RetinaNet and CascadeRCNN. From the results, it can be inferred that CenterNet outperforms almost all the models on all the datasets. For LISA TS, CascadeRCNN achieved best mAP because of the presence of evenly distributed traffic signs in medium and small annotation box areas.

Figure 2 shows qualitative results on the frames of respective traffic sign datasets. Visual results indicate that models are able to predict correctly on

Table 6. Results of various object detection models on traffic sign datasets using Faster R-CNN(FRCNN), SSD, Yolov3(Y3), RetinaNet(RN) and CenterNet(CN). Mean Average Precision(mAP) at 0.5 threshold is indicated in the last row for different classes of the respective datasets. CPU and GPU inference time per image(in seconds) is also indicated in this table. The values in bold font represent the best results in each category across all methods on the respective datasets. Empty row values indicate that meta class is absent in the respective dataset.

	TT100K						GTSD					
Class	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**
warning	85.19	79.90	92.3	88.72	**97.90**	95.48	-	-	-	-	-	-
prohibitory	78.65	79.90	92.50	92.40	97.20	**94.46**	66.40	91.60	98.20	81.98	**99.20**	75.13
mandatory	83.71	82.80	88.90	87.04	**97.20**	76.82	65.65	80.10	85.60	77.28	**97.60**	68.42
danger	-	-	-	-	-	-	78.50	96.70	100.00	96.37	**100.00**	86.71
regulatory	-	-	-	-	-	-	-	-	-	-	-	-
complementary	-	-	-	-	-	-	-	-	-	-	-	-
information	-	-	-	-	-	-	-	-	-	-	-	-
speed-limit	-	-	-	-	-	-	-	-	-	-	-	-
stop	-	-	-	-	-	-	-	-	-	-	-	-
yield	-	-	-	-	-	-	-	-	-	-	-	-
school	-	-	-	-	-	-	-	-	-	-	-	-
school-speed-limit25	-	-	-	-	-	-	-	-	-	-	-	-
zone-ahead	-	-	-	-	-	-	-	-	-	-	-	-
ramp-speed-advisory	-	-	-	-	-	-	-	-	-	-	-	-
round-about	-	-	-	-	-	-	-	-	-	-	-	-
curve-left	-	-	-	-	-	-	-	-	-	-	-	-
no-left-turn	-	-	-	-	-	-	-	-	-	-	-	-
thru-traffic-merge-left	-	-	-	-	-	-	-	-	-	-	-	-
do-not-enter	-	-	-	-	-	-	-	-	-	-	-	-
no-right-turn	-	-	-	-	-	-	-	-	-	-	-	-
other	-	-	-	-	-	-	60.06	73.70	90.00	86.81	**92.20**	65.03
mAP@0.5	82.51	80.90	91.60	89.39	**97.40**	88.92	67.65	85.50	93.40	85.68	**97.20**	73.85
GPU Inf Time. (sec)	0.69	0.03	0.10	0.07	0.21	0.07	0.67	0.002	0.13	0.08	0.21	0.07
CPU Inf Time. (sec)	15.86	2.53	9.23	2.09	5.83	6.69	14.90	3.04	9.81	2.14	6.13	8.15

	MTSD						LISA TS					
Class	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**	**FRCNN**	**SSD**	**Y3**	**RN**	**CN**	**CRNN**
warning	66.78	59.20	60.00	66.68	**81.00**	75.98	84.82	90.20	95.00	88.74	88.90	**96.38**
prohibitory	-	-	-	-	-	-	82.07	90.40	93.90	84.94	92.80	**95.77**
mandatory	83.71	82.80	88.90	87.04	**97.20**	76.82	-	-	-	-	-	-
danger	-	-	-	-	-	-	-	-	-	-	-	-
regulatory	57.70	44.40	53.30	54.02	**81.50**	73.94	-	-	-	-	-	-
complementary	50.74	38.90	48.50	41.49	73.00	**74.27**	-	-	-	-	-	-
information	-	-	-	-	-	-	36.89	28.70	38.10	35.13	**67.50**	30.61
speed-limit	-	-	-	-	-	-	25.82	48.90	94.60	**96.70**	80.90	96.78
stop	-	-	-	-	-	-	74.18	84.30	**92.10**	84.20	85.10	94.27
yield	-	-	-	-	-	-	30.92	55.30	**88.70**	44.65	66.80	87.50
school	-	-	-	-	-	-	62.79	91.5	**99.30**	25.74	93.50	87.56
school-speed-limit25	-	-	-	-	-	-	84.73	74.50	95.40	**100.00**	91.10	95.60
zone-ahead	-	-	-	-	-	-	46.37	11.20	80.00	88.6	59.80	**100.00**
ramp-speed-advisory	-	-	-	-	-	-	73.62	74.30	**100.00**	79.86	**100.00**	89.25
round-about	-	-	-	-	-	-	72.70	51.50	71.40	48.67	89.10	**97.69**
curve-left	-	-	-	-	-	-	67.06	27.50	88.30	40.77	88.30	**94.54**
no-left-turn	-	-	-	-	-	-	25.49	46.60	**100.00**	67.85	95.30	80.01
thru-traffic-merge-left	-	-	-	-	-	-	80.19	33.50	**100.00**	60.13	**100.00**	63.51
do-not-enter	-	-	-	-	-	-	88.61	5.00	93.30	83.31	88.60	**96.70**
no-right-turn	-	-	-	-	-	-	0.00	24.00	20.00	75.24	69.50	**100.00**
other	44.24	27.60	30.30	34.00	**55.50**	30.01	-	-	-	-	-	-
mAP@0.5	82.51	80.9	91.6	89.39	**97.40**	88.92	59.96	53.92	87.50	71.27	86.0	**91.71**
GPU Inf Time. (sec)	0.69	0.03	0.10	0.07	0.21	0.07	0.68	0.03	0.11	0.077	0.23	0.06
CPU Inf Time. (sec)	15.86	2.53	9.23	2.09	5.83	6.69	18.03	2.95	8.28	3.72	5.20	2.62

all the classes that they have been trained on. SSD and RetinNet output some false detections on LISA TS and GTSD datasets respectively but CenterNet performs detection on all the datasets with higher confidence, even in slightly darker imaging environment as for the predictions of MTSD dataset.

5.3 Object Detection Model's Statistics

Table 5 shows the number of trainable parameters and the floating point operations(GFlops) of the respective object detection models. From the figures, it is evident that FasterRCNN has the highest number of parameters since it has Resnet101 as the backbone architecture while CascadeRCNN and RetinaNet uses Resnet50 as the backbone architecture. This is also evident in Table 4 and Table 6 for GPU and CPU inference times. SSD operates on lower image resolution compared to Yolov3, hence the run-time for SSD is much lower. CenterNet uses Resnet18 [17] as the backbone having the lowest number of trainable parameters and the number of floating point operations surpassing all anchor based models in terms of detection and also real-time performance.

6 Conclusion

In this work, we used various state-of-the-art models for object detection and evaluate their performance on publicly available traffic light datasets: LISA TL, WPI, BOSCH and S2TLD and traffic sign datasets: LISA TS, GTSD, TT100K and MTSD. To the best of our knowledge, this work is the first detailed analysis of different object detection models (anchor based and anchor-less) on common meta classes of various traffic light and traffic sign datasets. From the results, it is evident that anchor less methods outperform anchor-based methods on almost all traffic light and traffic sign datasets irrespective of the instance location. For traffic light instances located nearby, even SSD performs better compared to any other anchor based model for WPI dataset. Specifically for LISA TS, Yolov3 marginally outperforms CenterNet due to a balance of traffic sign instances in medium and large annotation box areas from the camera view. In future work, we would like to explore the direction of the effect of weather patterns and the role of domain adaptation techniques to increase the detection accuracies of small objects such as traffic sign and traffic light.

References

1. Aghdam, H.H., Heravi, E.J., Puig, D.: A practical approach for detection and classification of traffic signs using convolutional neural networks. Robotics and autonomous systems **84**, 97–112 (2016)
2. Behrendt, K., Novak, L.: A deep learning approach to traffic lights: Detection, tracking, and classification. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE

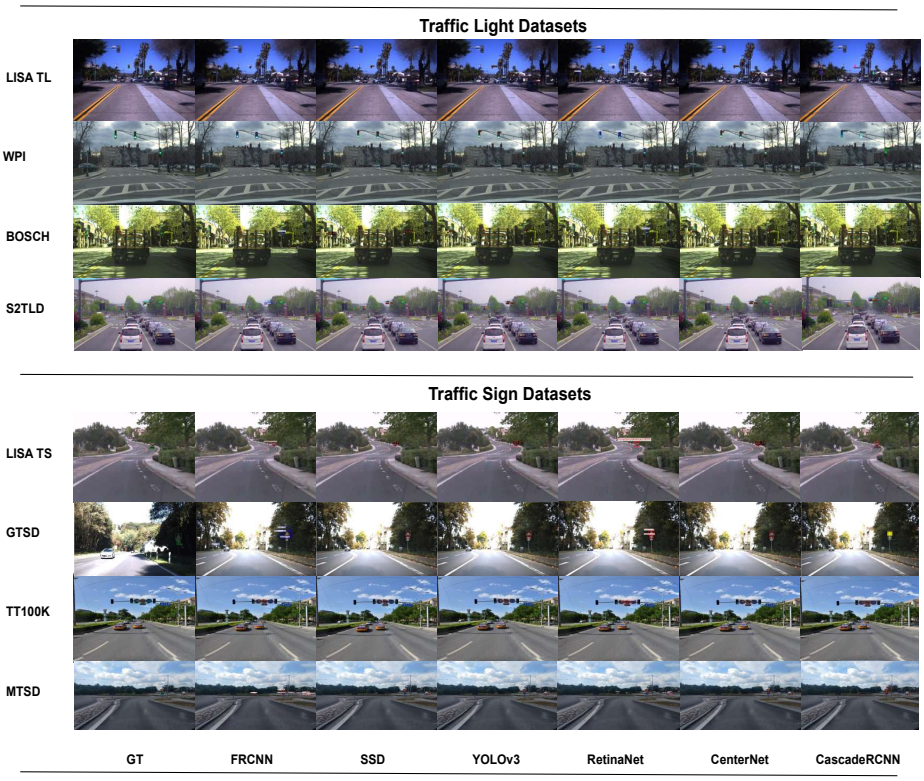

Fig. 2. Qualitative results of various object detection models on all traffic sign and traffic light datasets. The first four rows have visual results from traffic light datasets. The next four rows have visual results from traffic sign datasets. The first column is the ground truth image along with annotations in green color *(Best viewed when zoomed)*.

3. Behrendt, K., Novak, L., Botros, R.: A deep learning approach to traffic lights: Detection, tracking, and classification. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1370–1377. IEEE (2017)
4. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
5. Chen, Z., Huang, X.: Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking. IEEE Intelligent Transportation Systems Magazine **8**(4), 28–42 (2016). https://doi.org/10.1109/MITS.2016.2605381
6. De Charette, R., Nashashibi, F.: Traffic light recognition using image processing compared to learning processes. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 333–338. IEEE (2009)
7. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6569–6578 (2019)
8. Ellahyani, A., Ansari, M., Jaafari, I., Charfi, S.: Traffic sign detection and recognition using features combination and random forests. International Journal of Advanced Computer Science and Applications **7**(1), 686–693 (2016)
9. Ertler, C., Mislej, J., Ollmann, T., Porzi, L., Neuhold, G., Kuang, Y.: The mapillary traffic sign dataset for detection and classification on a global scale. In: European Conference on Computer Vision. pp. 68–84. Springer (2020)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
11. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
13. Gong, J., Jiang, Y., Xiong, G., Guan, C., Tao, G., Chen, H.: The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles. In: 2010 IEEE Intelligent Vehicles Symposium. pp. 431–435 (2010). https://doi.org/10.1109/IVS.2010.5548083
14. Greenhalgh, J., Mirmehdi, M.: Real-time detection and recognition of road traffic signs. IEEE Transactions on Intelligent Transportation Systems **13**(4), 1498–1506 (2012). https://doi.org/10.1109/TITS.2012.2208909
15. Haloi, M.: A novel plsa based traffic signs classification system (03 2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
18. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: International Joint Conference on Neural Networks. No. 1288 (2013)
19. Huang, Z., Yu, Y., Gu, J., Liu, H.: An efficient method for traffic sign recognition based on extreme learning machine. IEEE transactions on cybernetics **47**(4), 920–933 (2016)

20. Jensen, M.B., Philipsen, M.P., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Vision for looking at traffic lights: Issues, survey, and perspectives. IEEE Transactions on Intelligent Transportation Systems **17**(7), 1800–1815 (2016). https://doi.org/10.1109/TITS.2015.2509509
21. Larsson, F., Felsberg, M.: Using fourier descriptors and spatial models for traffic sign recognition. In: Scandinavian conference on image analysis. pp. 238–249. Springer (2011)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
24. Liu, Z., Du, J., Tian, F., Wen, J.: Mr-cnn: A multi-scale region-based convolutional neural network for small traffic sign recognition. IEEE Access **7**, 57120–57128 (2019)
25. Møgelmose, A.: Visual analysis in traffic & re-identification (2015)
26. Mogelmose, A., Trivedi, M.M., Moeslund, T.B.: Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems **13**(4), 1484–1497 (2012). https://doi.org/10.1109/TITS.2012.2209421
27. Müller, J., Dietmayer, K.: Detecting traffic lights by single shot detection. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 266–273 (2018). https://doi.org/10.1109/ITSC.2018.8569683
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
29. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015)
31. Saini, S., Nikhil, S., Konda, K.R., Bharadwaj, H.S., Ganeshan, N.: An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In: 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 606–611. IEEE (2017)
32. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
34. Siogkas, G., Skodras, E.: Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information. vol. 1 (02 2012)
35. Wang, G., Ren, G., Wu, Z., Zhao, Y., Jiang, L.: A robust, coarse-to-fine traffic sign detection method. In: The 2013 international joint conference on neural networks (IJCNN). pp. 1–5. IEEE (2013)
36. Weber, M., Wolf, P., Zöllner, J.M.: Deeptlr: A single deep convolutional network for detection and classification of traffic lights. In: 2016 IEEE Intelligent Vehicles Symposium (IV). pp. 342–348 (2016). https://doi.org/10.1109/IVS.2016.7535408

37. Yang, X., Yan, J., Yang, X., Tang, J., Liao, W., He, T.: Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. arXiv preprint arXiv:2004.13316 (2020)
38. Yudin, D., Slavioglo, D.: Usage of fully convolutional network with clustering for traffic light detection. In: 2018 7th Mediterranean Conference on Embedded Computing (MECO). pp. 1–6. IEEE (2018)
39. Zaklouta, F., Stanciulescu, B.: Real-time traffic-sign recognition using tree classifiers. IEEE Transactions on Intelligent Transportation Systems **13**(4), 1507–1514 (2012)
40. Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., Liu, W.: Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing **214**, 758–766 (2016)
41. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
42. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV (2014)