Evaluating State-of-the-art Object Detector on Challenging Traffic Light Data

Morten B. Jensen, Kamal Nasrollahi, and Thomas B. Moeslund
Section of Media Technology
Aalborg University
Denmark
{mboj, kn, tbm}@create.aau.dk

Abstract

Traffic light detection (TLD) is a vital part of both intelligent vehicles and driving assistance systems (DAS). General for most TLDs is that they are evaluated on small and private datasets making it hard to determine the exact performance of a given method. In this paper we apply the state-of-the-art, real-time object detection system You Only Look Once, (YOLO) on the public LISA Traffic Light dataset available through the VIVA-challenge, which contain a high number of annotated traffic lights, captured in varying light and weather conditions.

The YOLO object detector achieves an AUC of impressively 90.49 % for daysequence1, which is an improvement of 50.32 % compared to the latest ACF entry in the VIVA-challenge. Using the exact same training configuration as the ACF detector, the YOLO detector reaches an AUC of 58.3 %, which is in an increase of 18.13 %.

1. Introduction

In recent years the term *big data* and *machine learning* have gained tremendous momentum, especially the use of big data have been a heavily discussed topic. As a result, data is collected in almost every digital action we do, and is collected like never before. In fact, we create 2.5 quintillion bytes (2,500,000,000 gigabytes) of data each day resulting in 90 % of the current available data have been created for the past 2 years [1]. The data are collected from a large variety of locations, spanning from your social media activities and browsing to various sensors collecting climate data or traffic surveillance data. Collecting traffic data both with the purpose of surveillance and especially autonomous vehicles have gained a lot of media attention as a result of major companies spending large amount money on research in this area. However, making a vehicle drive autonomously have a lot of challenges linked to it, which still requires years of research.

Both industry and academic institutions are looking into research and applications that can be relevant and helpfull in the meantime. This can prove beneficial for the ultimate dream of self-driving cars, but also for the popular driving assistance systems (DAS). DAS applications are already widely implemented in newer vehicles, such as emergency breaking, automatic lane changing, keeping the advertised speed limit, and adaptive cruise control. DAS applications can usually be split into looking-in [28], such as hands activity recognition [19] and looking-out applications, such as detection of other vehicles, pedestrians [5], traffic signs [18] or traffic lights [9]. In 2012, 683 people died and 133,000 people were injured in crashes related to red light running in the USA [26], making traffic light detection a vital part of both self-driving cars and DAS.

In this paper we apply the state-of-the-art, real-time object detection system *You Only Look Once*, (YOLO) [23], which have proven a good competitor to Fast R-CNNs and SSDs both in terms of detections and speed. In this paper, we will apply YOLO on the daytime data from the freely available LISA Traffic Light Dataset used in the VIVA-challenge [11, 16], which have seen a limited use of deep learning methods. The contributions of this paper is twofold:

- Training and applying the state-of-the-art, real-time object detection system *You Only Look Once*, (YOLO) for traffic light detection.
- Deep learning entry in the public VIVA Traffic Light challenge.

The paper is organized as follows: Relevant research is summarized in section 2. In section 3 we present the method used, followed by evaluation of the TL detector in section 4. Finally, the concluding remarks are presented in section 5.

2. Related Work

In this section a brief introduction to the most notable research in relation to TLD is given, for a more comprehensive overview, we refer to the traffic light survey [11]. In [11] TLD is split into two categories: model-based and learning-based.

The model-based methods have been quite dominant and popular in the past decade and are usually created by the use of a heuristically defined model which relies on color and/or shape information. The color information is quite intuitive and a straight-forward approach as traffic lights presents the driver with multiple color cues which corresponds to a driver action e.g. stop or go. The detector is based on a heuristical defined threshold in a selected color space [4, 14]. The color can however vary from scene to scene and thus challenge models relying solely on static thresholds. So rather than looking at color, one could make use of the distinctive shape of traffic lights by applying circular Hough transform on an edge map [20] or by using radial symmetry [25]. Both approaches are challenged in different scenarios, but not entirely the same scenarios, thus shape information is fused with structural information [27, 3], and additionally color information in [29, 15]. Rather than defining static set of rules, [8] propose a Bayesian inference framework relying on color, shape and height to detect traffic lights.

Cascading classifier based on Haar-like features was one of the first learning-based detectors to be introduced in [7, 17], but did however not outperform their Gaussian color classifier. As for most other computer vision research areas, the popular combination of using Histogram of Oriented Gradients features together with a SVM classifier was introduced in [2]. The learning-based Aggregated Channel Features (ACF) detector have seen a large use in TLD, and have shown superior performance over the heuristic models both during day and night time [10, 9]. TLD using Convolutional Neural Network (CNN) is introduced in [13, 12], where a CNN is used detects and recognize the traffic lights using region-of-interest information provided by an onboard GPS sensor.

3. Method

In this section the method used in this paper will be briefly introduced.The method section is split into two sections: firstly the YOLO object detector is introduced. Secondly, training parameters and data specifications used in the evaluation are introduced.

3.1. YOLO

YOLO have been introduced in two versions [22, 23], where the latest version is the one used in this paper which include new features as well as modifications to the existing network. YOLO is an end-to-end single convolutional neural network that detects objects based on bounding boxes prediction and class probabilities. The network divides the input image into a SxS grid, if the center of an object is located within this grid, it is this specific grid's task to detect the object. Each grid predicts bounding boxes and a corresponding confidence, where the confidence is an indicator of how confident the model is that a box contains an object as well as how accurate the box is. The confidence is therefore calculated using the intersection over union (IOU), where a perfect match between a predicted box and a ground truth will provide a confidence of 1, and oppositely if a predicted box is not present in the grid, hence no ground truth overlapping, the confidence will be 0. Finally, the grid cell also predicts the probability of an object belonging to a class.

Unlike many sliding window methods, such as the ACF detector, YOLO examines the entire image during training helping it to learn contextual information about a given class and its surroundings. The original YOLOv2 classification model, called Darknet-19, has 19 convolutional layers and 5 maxpooling layers, and have some resembles to well-known VGG-16 network. It is however a lot less complex as the VGG-16 requires 30.69 billion floating point operations to process a single 224x224 pixel frame, whereas the Darknet-19 only needs 5.58 billion operations whilst improving the top-5 accuracy on ImageNet with 1.2 % compared to VGG-19's 90 %. An additional training where the size is increased from 224 to 448, improves the top-5 accuracy to 93.3 % at the compromise of processing the images 4.24 times slower. This 448x448 model constitutes the Darknet19 448x448 model which have been used as pre-weights for training in this paper.

For using the model for detection, the network is modified by removing the last convolutional network and instead adding three 3x3 convolutional layers with 1024 filters, which is finally followed by a 1x1 convolutional layer with the number of outputs needed for the specific detection. For enabling fine grain features, a passthrough layer is inserted second to the last convolutional layer.

3.2. Training parameters

The *random* parameter enables multi-scale training, resulting in a robustness for detecting objects in different image resolutions. The input size is per default set to a resolution of (416x416), but by enabling the random parameter the network will randomly change the input image size every 10 batch. The YOLOv2 network downsamples by a factor of 32, resulting in a downsampling range between {320, 352, ..., 608}. The smallest input size is thus (320x320), and the largest input size is (608x608). The random parameter is per default enabled in YOLOv2, in this paper we will try to identify the effect. Furthermore, we will investigate varying the input size whilst doing detection.

4. Evaluation

Several models have been trained using different training data and modified in accordance to the parameters described in section 3.2.

The data configuration for each model can be seen in Table 1. The training data used for all the models are from the LISA Traffic Light Dataset [11] and the LARA Traffic Light Dataset [24].

Table 1: Overview of the trained YOLOv2 Traffic Light Detectors. All models have been trained with an input image size of (416,416), with half the models enabled the random parameter varying the input image size between {320, 352, ..., 608}.

		Training Data		
Model name	***Random***	***LISA-dayTrain***	***LARA [24]***	***LISA-daySeq2***
YOLO_V1_0		✓		
YOLO_V1_1	✓	✓		
YOLO_V2_0		✓	✓	
YOLO_V2_1	✓	✓	✓	
YOLO_V3_0		✓		✓
YOLO_V3_1	✓	✓		✓

The LISA Traffic Light Dataset consists of 13 day training clips, hereafter referred to as LISA-dayTrain, as well as 2 longer test sequences, hereafter referred as LISA-daySeq1 or 2. For evaluating, the LISA-daySeq1 has been used, as it was the main evaluation sequence in the VIVA-challenge. The LARA Traffic Light Dataset is also included to create some more variance as it is captured in Paris, France, whereas the LISA TL dataset is captured in San Diego, USA. Furthermore the LARA Traffic Light Dataset is introduced to see how it impacts the model when testing it on a test sequence that is captured in same environment as a large part of the training data. In Table 2, an overview of the used training and test data is seen. In Figure 1 some samples from the data are seen.

Table 2: Overview of the evaluation data.

Dataset	**Frames**	**True positives**	**Resolution**	**Classes**
LARA	11,179	9,168	640 x 480	4 (green, orange, red, & ambiguous)
LISA-dayTrain	14,025	40,764	1280 x 960	6 (Go, go left, warning, warning left, stop, stop left)
LISA-daySeq2	6,894	11,144	1280 x 960	6 (Go, go forward, go left, warning, stop, stop left)
LISA-daySeq1	4,060	10,308	1280 x 960	5 (Go, warning, warning left, stop, stop left)

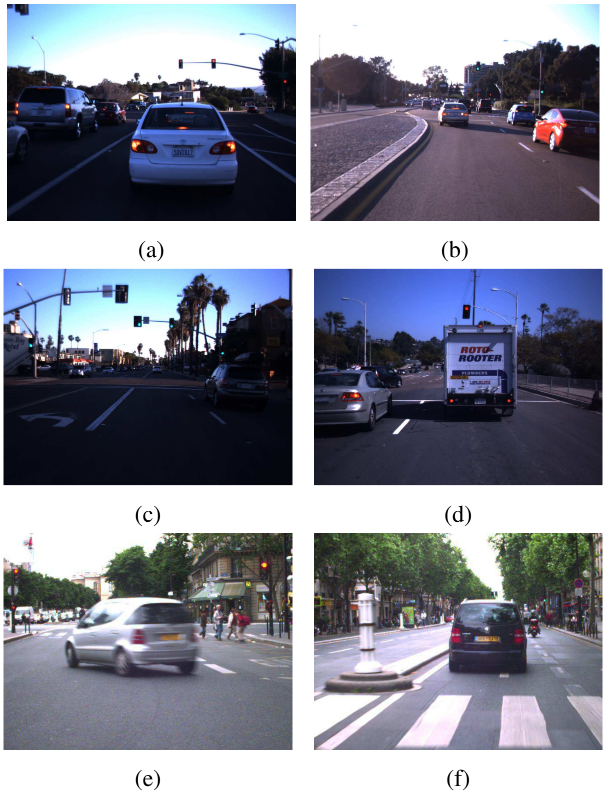

(a)
(b)
(c)
(d)
(e)
(f)

Figure 1: Training samples from the (a-d) LISA and (e-f) LARA Traffic Light database.

A total of 6 YOLO TLD models are trained and applied on the LISA-daySeq1. In order to make the results of above models comparable with previous publications, the results must be evaluated in accordance to the VIVA-challenge [16], where the Area-Under-Curve(AUC) of a Precision-Recall curve(PR-curve) is the final evaluation metric [11]. Furthermore, the true positive criteria in the VIVA-challenge defines a detection as one that is overlapping with an annotation with more than 50 %, as defined in Equation (1).

$$a_0 = \frac{\text{area}(B_d \cap B_{gt})}{\text{area}(B_d \cup B_{gt})} \tag{1}$$

Where a_0 denotes the *overlap ratio* between the detected bounding box B_d and the ground truth bounding box B_{gt}. a_0 must be equal or greater that 0.5 to meet true positive criteria. [6]

Prior to calculating the AUC of the model, we examine the recall of each of the trained models. Models are trained for 80,000 iterations and for every 1000th iteration during training, weights are saved for backup purposes. These weights are used to determine how the performance relates to the number of iterations. This relation is seen Figure 2 and in 3 where the detectors' image size have been changed from (416,416) to (672,672).

In Figure 2 the detectors with an input image of (416,416) are shown. To determine the impact of the random parameter, we compare the versions of the YOLO TL detectors. By enabling the random parameter with only the

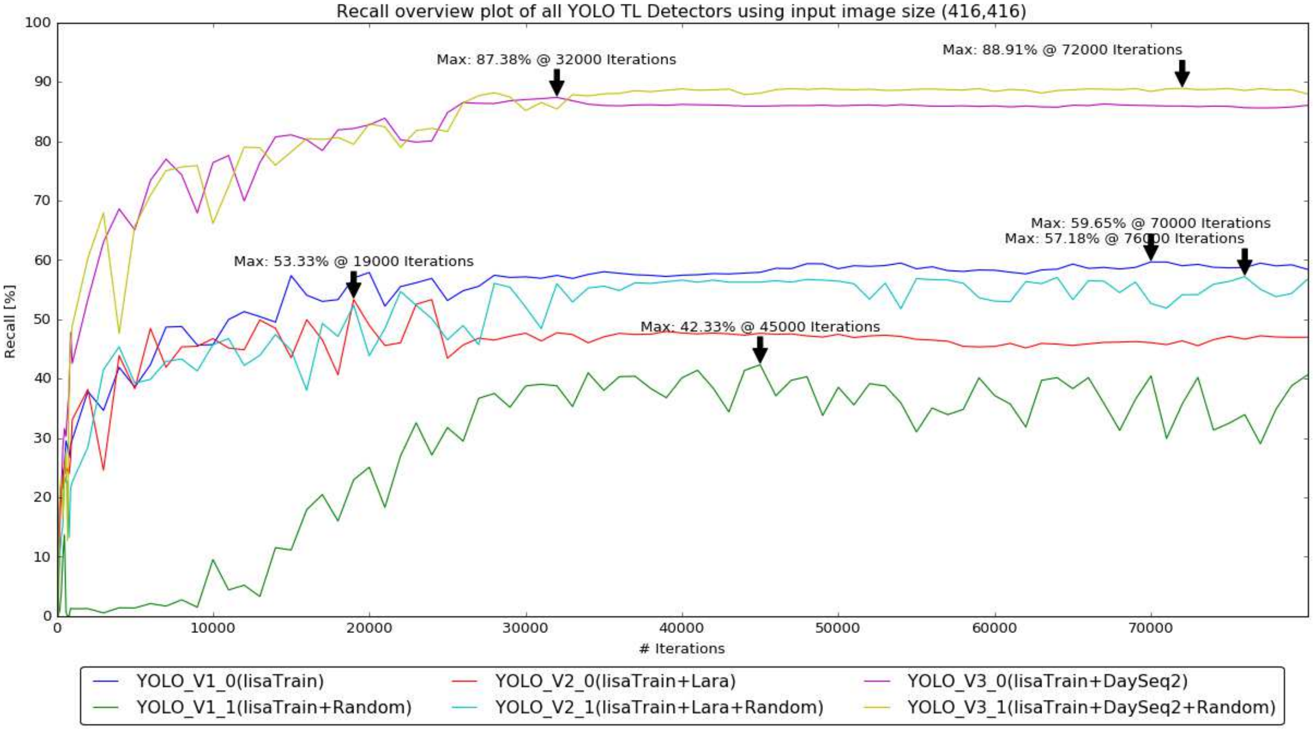


Figure 2: Recall plot for iterations made during training of the models with input image size (416,416).

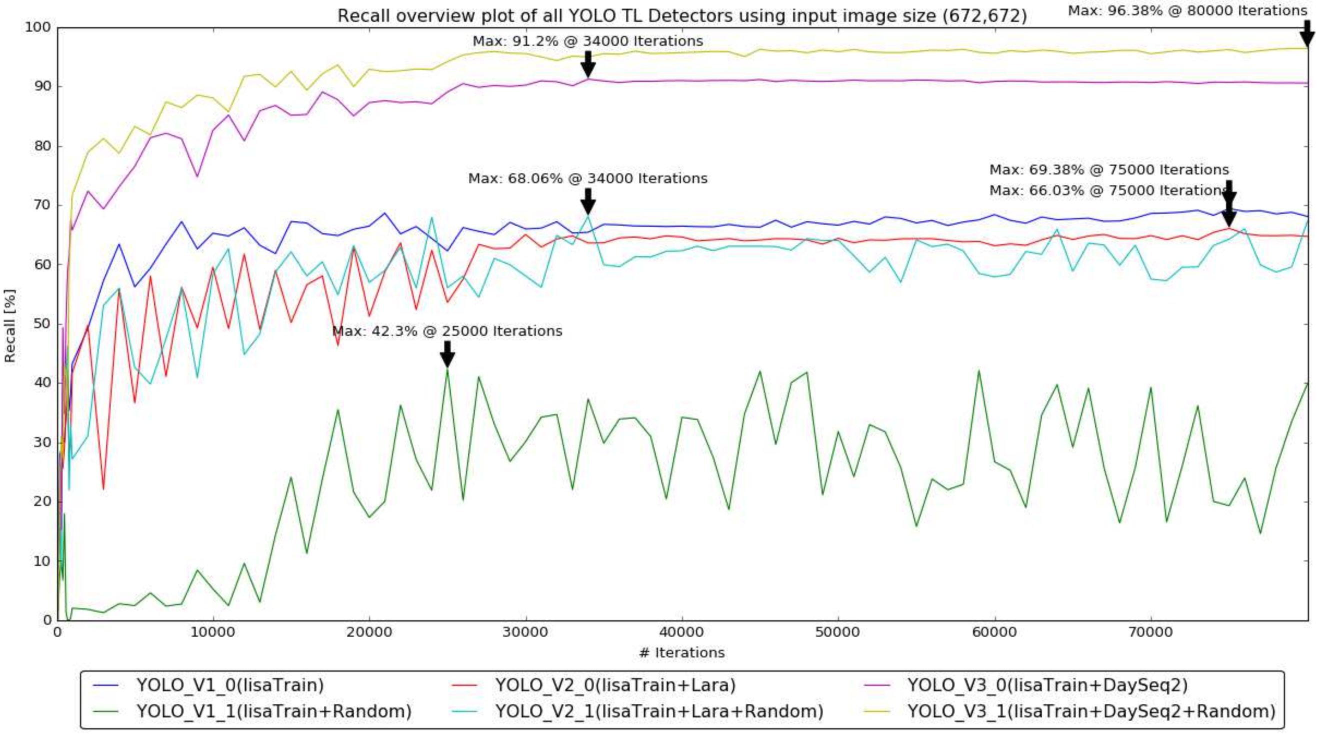


Figure 3: Recall plot for iterations made during training of the models with input image size (672,672).

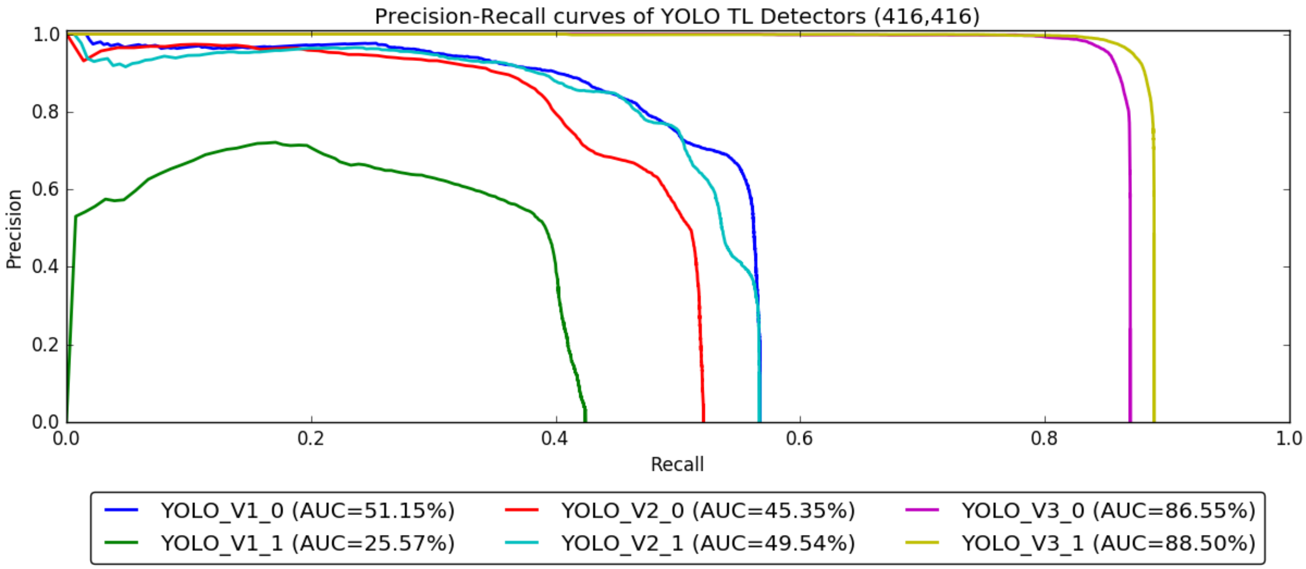


Figure 4: Precision-recall curves of the best recall iterations from (416,416) detectors in Figure 2.

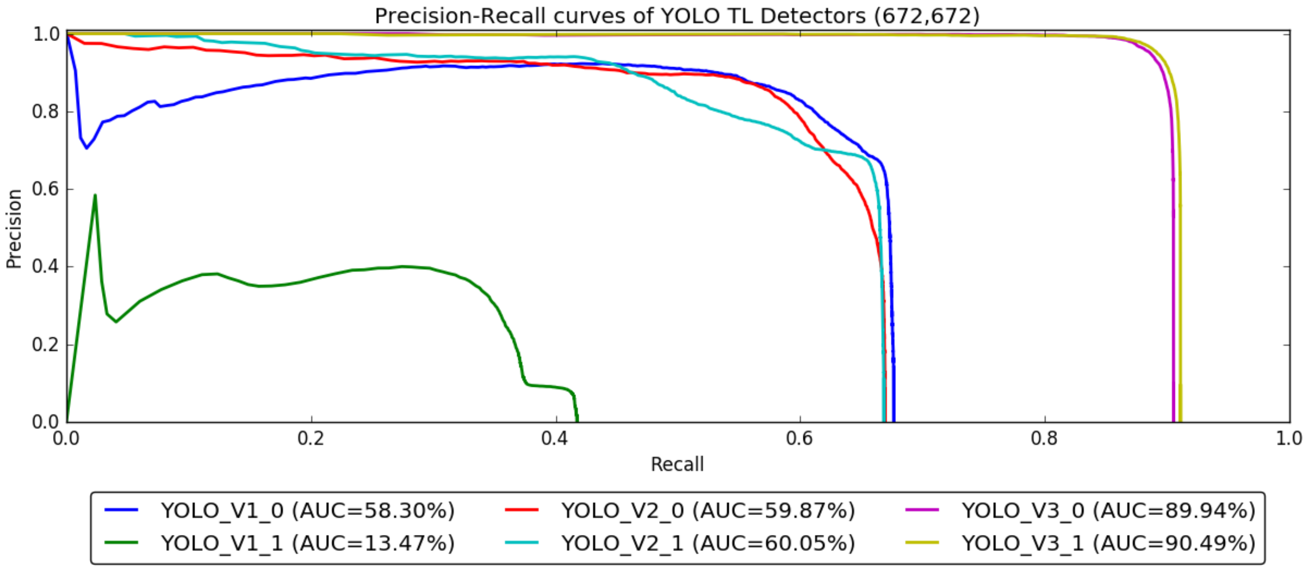
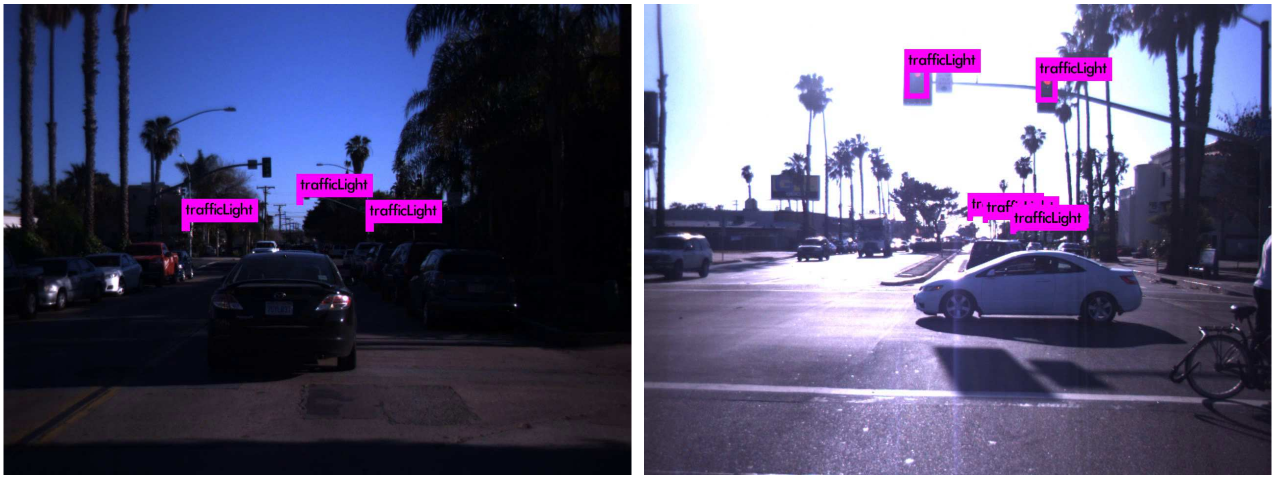


Figure 5: Precision-recall curves of the best recall iterations from (672,672) detectors in Figure 3.

LISA-dayTrain as training data, the recall performance decrease by 17.32 %. By examining the figure, it is clear that YOLO_V1_1 is struggling to reach a stable recall compared to the other 5 models, which suggests that we do not use enough and sufficient varied training data for the varying input image size to make any impact. In YOLO_V2_0 we add the LARA dataset to the training which nearly reaches the same recall as YOLO_V1_0. YOLO_V2_1, with the random parameter enabled, increases the recall with 3.85 % compared to YOLO_V2_0 but is still 2.47 % worse than YOLO_V1_0. Finally, by swapping the LARA dataset with LISA-daySeq2, we reach a recall of 87.38 % and 88.91 % for YOLO_V3_0 and YOLO_V3_1, respectively.

As the detectors only use convolutional and pooling layers we can resize the input image size without retraining. In Figure 5 the detectors with input image of (672,672) are shown. The result of increasing the input image size to (672,672) provides a very similar picture of the detectors as for the (416,416). However, 5 out of 6 models reaches a higher maximum recall after increasing the image input size to (672,672), the exception being YOLO_V1_1 which also struggled in Figure 2. By examining and comparing Figure



(a)

(b)

Figure 6: Results from YOLO V3 1 applied on LISA-daySeq1.

2 and 3, it is clear from a visual analysis, that the (416,416) looks more smooth compared to (672,672). This is due to the larger difference in the recall results between the iterations, suggesting that the input image size of (672,672) might not be completely ideal, at least not for the data configuration of YOLO_V1 and V2. Finally, the best performing model is YOLO_V3_1, which was expected as it is the one with the most training data from the LISA TL dataset, thus looking most identical with LISA-daySeq1.

For each of the detectors seen in Figure 2 and 3, the iteration with the highest recall is used for precision-recall curves and calculating the corresponding AUC. The lowest AUC is in both figures the YOLO_V1_1, which is not surprising as it was also generally performing bad in Figure 2 and 3. In Figure 4, the YOLO_V1_0 is reaching an AUC of 51.51 % and is the best performing of the one not including LISA-daySeq2 in the training data. In Figure 5, YOLO_V1_0 is still performing good, but both YOLO_V2_0 and YOLO_V2_1 surpass it after the image input size is increased. Generally increasing the input image size provided an average AUC increase of 4.29 %, and if we exclude the YOLO_V1_1 we each an average AUC increase of 7.51 %. The average AUC increase caused by enabling the random parameter for YOLO_V2 and YOLO_V3 is 1.72 %, which could indicate that adjusting the input image size provide a larger impact.

The 2 detectors based on both LISA-dayTrain and LISA-daySeq2, YOLO_V3_0 and YOLO_V3_1, reaches the by far highest AUC with both image input sizes. The highest overall AUC is 90.49 % by YOLO_V3_1. In [21], the highest AUC for daySeq1 is 40.17 %, which means that the YOLO_V3_1 have significantly improved the entry on the LISA Traffic Light dataset with impressively 50.32 %. This result do however not form basis for a fair comparison between YOLO and the ACF detector used in [21] as the ACF detector have purely been trained on the lisaTrain data. So to compare the performance of the two methods given the same data, we must compare it to YOLO_V1_0 which reaches an AUC of 58.3 % with an image input size of (672,672), resulting in an AUC increase of 18.13 %.

In Figure 6, detection results from the YOLO_V3_1 detector are shown. Compared to previous work from the ACF detector used in [21], the YOLO_V3_1 handles the varying lighting conditions well as seen from 6a. Generally, the models with the multi-scale training parameter *random* enabled are not surprisingly also able to detect the TLs at a much longer distance, which is illustrated in 6b.

5. Conclusion

We have taken one of the state-of-the-art object detectors and applied in on a challenging traffic light dataset with different model and data configurations. The highest overall AUC on daySequence1 from the LISA Traffic Light dataset is 90.49 % and is unsurprisingly based on all the training data and daySequence2 from the same dataset. This improves the entry from [21] on the LISA Traffic Light dataset with impressively 50.32 %. However, if we use the exact same training data as used with the ACF detector in [21], we reach an AUC of 58.3 %, which is an AUC improvement of 18.13 %. The random parameter that enables multi-scale training did in most cases improve the AUC slightly, whereas increasing the input image size of the detector turned out to have a larger impact than the random parameter.

Further experiments includes using SSD for traffic light detection, creating an ensemble with R-FCN, and do similar evaluation on the nighttime data from the LISA Traffic Light dataset.

References

[1] R. H. Bajaj and P. Ramteke. Big data–the new era of data. *International Journal of Computer Science and Information Technologies*, 5(2):1875–1885, 2014.

[2] D. Barnes, W. Maddern, and I. Posner. Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[3] R. Charette and F. Nashashibi. Traffic light recognition using image processing compared to learning processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 333–338, 2009.

[4] M. Diaz-Cabrera, P. Cerri, and P. Medici. Robust real-time traffic light detection and distance estimation using a single camera. *Expert Systems with Applications*, pages 3911–3923, 2014.

[5] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, Aug 2014.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[7] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich. Making bertha see. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 214–221, 2013.

[8] S. Hosseinyalamdary and A. Yilmaz. A bayesian approach to traffic light detection and mapping. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 125:184 – 192, 2017.

[9] M. B. Jensen, M. P. Philipsen, T. B. Moeslund, and M. Trivedi. *Comprehensive Parameter Sweep for Learning-Based Detector on Traffic Lights*, pages 92–100. Springer International Publishing, Cham, 2016.

[10] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi. Traffic light detection at night: Comparison of a learning-based detector and three model-based detectors. *11th Symposium on Visual Computing*, 2015.

[11] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):1800–1815, July 2016.

[12] V. John, K. Yoneda, Z. Liu, and S. Mita. Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. *IEEE Transactions on Computational Imaging*, 1(3):159–173, Sept 2015.

[13] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita. Traffic light recognition in varying illumination using deep learning and saliency map. In *IEEE 17th International Conference on Intelligent Transportation Systems*, pages 2286–2291, 2014.

[14] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung. Night-time traffic light detection based on svm with geometric moment features. *World Academy of Science, Engineering and Technology 76th*, pages 571–574, 2013.

[15] E. Koukoumidis, M. Martonosi, and L.-S. Peh. Leveraging smartphone cameras for collaborative road advisories. *IEEE Transactions on Mobile Computing*, 11:707–723, 2012.

[16] Laboratory for Intelligent and Safe Automobiles, UC San Diego. Vision for Intelligent Vehicles and Applications (VIVA) Challenge. `http://cvrr.ucsd.edu/vivachallenge/`, 2015.

[17] F. Lindner, U. Kressel, and S. Kaelberer. Robust recognition of traffic signals. In *IEEE Intelligent Vehicles Symposium*, pages 49–53, 2004.

[18] A. Mogelmose, D. Liu, and M. M. Trivedi. Traffic sign detection for us roads: Remaining challenges and a case for tracking. In *IEEE Transactions on Intelligent Transportation Systems*, pages 1394–1399, 2014.

[19] E. Ohn-Bar and M. Trivedi. In-vehicle hand activity recognition using integration of regions. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1034–1039. IEEE, 2013.

[20] M. Omachi and S. Omachi. Detection of traffic light using structural information. In *IEEE 10th International Conference on Signal Processing (ICSP)*, pages 809–812, 2010.

[21] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi. Traffic light detection: A learning algorithm and evaluations on challenging dataset. *18th IEEE Intelligent Transportation Systems Conference*, 2015.

[22] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[23] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[24] Robotics Centre of Mines ParisTech. Traffic lights recognition (tlr) public benchmarks, 2015.

[25] S. Sooksatra and T. Kondo. Red traffic light detection using fast radial symmetry transform. In *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6. IEEE, 2014.

[26] The Insurance Institute for Highway Safety (IIHS). Red light running, 2015.

[27] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi. Tracking both pose and status of a traffic light via an interacting multiple model filter. In *17th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2014.

[28] M. M. Trivedi, T. Gandhi, and J. McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *IEEE Transactions on Intelligent Transportation Systems*, pages 108–120, 2007.

[29] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng. A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles. In *33rd Chinese Control Conference*, pages 4924–4929, 2014.