Comprehensive Parameter Sweep for Learning-Based Detector on Traffic Lights

Morten B. Jensen[1,2(✉)], Mark P. Philipsen[1,2], Thomas B. Moeslund[1], and Mohan Trivedi[2]

[1] Visual Analysis of People Laboratory, Aalborg University, Aalborg, Denmark
mboj@create.aau.dk

[2] Computer Vision and Robotics Research Laboratory, UC San Diego, La Jolla, USA

Abstract. Determining the optimal parameters for a given detection algorithm is not straightforward and what ends up as the final values is mostly based on experience and heuristics. In this paper we investigate the influence of three basic parameters in the widely used Aggregate Channel Features (ACF) object detector applied for traffic light detection. Additionally, we perform an exhaustive search for the optimal parameters for the night time data from the LISA Traffic Light Dataset. The optimized detector reaches an Area-Under-Curve of 66.63% on calculated precision-recall curve.

1 Introduction

The Aggregate Channel Features (ACF) object detector [1], from Piotr's Computer Vision Matlab Toolbox (PMT) [2], has been used for detecting a wide range of objects. Originally it was introduced as as a detector for pedestrians in [1], but have since been applied in several other areas related to driver assistant systems (DAS). The applied areas are not only limited to looking-out of the vehicle [3], where other vehicles [4], signs [5], and traffic lights (TLs) [6] have been popular, but also looking-in areas, such as hands detection [7] has seen use of the ACF detector. General for all areas is that the ACF object detector has been adjusted heuristically in a practical manner. Fine-tuning towards the optimal parameters are a common problem amongst researchers as it can be difficult without any prior experience of applying the given detector or without any prior knowledge of the test data. All of the above DAS areas where ACF has been applied are great challenges and remains important cases as people unfortunately keeps getting injured in the traffic. In 2012, 683 people died and 133,000 people were injured in crashes related to red light running in the USA [8]. Traffic light detection is thus an obvious part of DAS system in the transition towards fully autonomous cars.

A large issue in research is that evaluations are done on small and private datasets that are captured by the authors themselves. For better and easier comparison in DAS related areas, benchmarks such as the *VIVA-challenge* [9] and *KITTI Vision Benchmark Suite* [10] can highly beneficial for determine the prone future research directions.

G. Bebis et al. (Eds.): ISVC 2016, Part II, LNCS 10073, pp. 92–100, 2016.
DOI: 10.1007/978-3-319-50832-0_10

In this paper, we will do a comprehensive analysis of three central parameters for the ACF object detector, applied on the night data from freely available LISA Traffic Light Dataset used in the VIVA-challenge [11]. The contributions of this paper are thus threefold:

1. Exhaustive parameter sweep of ACF.
2. Analysis of correlations between detector parameters.
3. Optimized TL detection results on the night data from the LISA Traffic Light Dataset.

The paper is organized as follows: Relevant previous work is summarized in Sect. 2. In Sect. 3 we present the detector and the three parameters that are investigated. The extensive evaluation of the parameter sweep is presented in Sect. 4. Finally, in Sect. 5 we give our concluding remarks.

2 Related Work

The related work can be split into two parts: model-based and learning-based approaches. For a more comprehensive overview of the related work, we refer to [11].

2.1 Model-Based

Model-based object detection is a very popular approach for detecting TLs. Most model-based detectors are defined by some heuristic parameters, in most cases relying on color or shape information for detecting TL candidates. The color information is used by heuristically defining thresholds for the color of interest in a given color space [12,13]. The shape information is usually found by applying circular Hough transform on an edge map [14], or finding circles by applying radial symmetry [15,16]. In [17,18] shape information is fused with structural information and additionally color information in [19,20]. The output of using above approaches are usually a binary image with TL candidates. BLOB analysis is introduced to reduce the number of TL candidates by doing connected component analysis and examining each BLOB by it's size, ratio, circular shape, and so on [21].

2.2 Learning-Based

One of the first learning-based detectors is introduced in [22,23] where a cascading classifier is tested using Haar-like features, but was unable to perform better than their Gaussian color classifier. The popular combination of Histogram of Oriented Gradients (HoG) features and SVM classifier were introduced in [24], but additionally also relying on prior maps with very precise knowledge of the TL locations. The learning-based ACF detector has previously been used for TLs, where features are extracted as summed blocks of pixels in 10 different channels created from the original input RGB frame. In [25] and [6] the extracted features are classified using depth-2 and depth-4 decision trees, respectively. In [6] the octave parameter, which define the number of octaves to compute above the original scale, is changed from 0 to 1.

3 Method

The method section is two-fold, firstly the learning-based ACF detector is presented. Secondly, the method for conducting the comprehensive parameters optimization for the TL detector is presented.

3.1 Learning-Based Detector

The features for the ACF object detector are extracted from 10 feature channels: 1 normalized gradient magnitude channel, 6 histogram of oriented gradients channels, and 3 channels constituting the LUV color channels. The features are hence created by single pixel lookups in the feature maps. The channels subsampled corresponds to a halving of the dimensions [4].

The training is done using 3,728 positives TL samples (Fig. 1) with a resized resolution of 25 × 25, and 5,772 frames without any TLs and hard negatives generated from 1 execution of bootstrapping on the 5 night training clips from the LISA TL dataset [11]. Examples of these hard negatives are seen in Fig. 2. The number of extracted negative samples varies depending on the configuration, but is limited to maximum of 175,000 samples.

AdaBoost is used to train 3 stages of soft cascades, the three stages consists of 10, 100, and 4000 weakleaners. However, the comprehensive parameters optimization showed that it often converges earlier. The generated AdaBoost classifier is using decision trees as weak learners.

For detecting TLs at greater distances, the intervals of scales can be adjusted by the *octave up* parameter, e.g. changing it from 0 to 1 will define the number of octaves to compute above the original scale. The number of extracted samples from the training will highly depend on the model size, tree-depth, and octave up parameters.

Finally the detection is done by using a sliding window which is moved across each of the 10 aggregated feature channels created from the test frame.

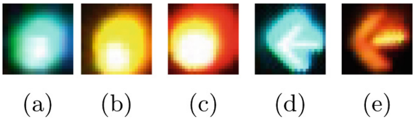


Fig. 1. Positives samples cropped from training data.

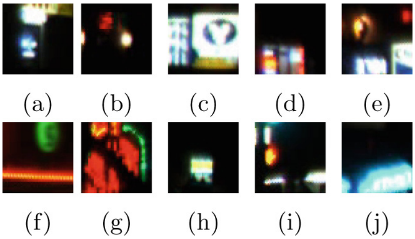


Fig. 2. Hard negatives generated from bootstrapping.

3.2 Parameter Optimization

In this paper, a comprehensive parameter optimization is made by adjusting the dimensions of the sliding window, hereafter defined as *mDs*, the decision tree's depth, hereafter defined as *treeDepth*, and the number of octaves to compute above the original scale, hereafter defined as *nOctUp*. To speed up the parameters optimization, a MATLAB script is developed which uses a FTP connection to communicate with a master web host, such n-computers can work on the parameter optimization simultaneously.

The parameter optimization is done by adjusting one parameter at a time, e.g. creating a TL detector with a nOctUp = 0 and treeDepth = 2, and then vary the mDs size from $[12, 12]$ to $[25, 25]$. A total of $14^2 = 196$ detectors are made with above nOctUp and treeDepth settings. By adjusting the nOctUp and treeDepth and redoing the sliding window variation, a very comprehensive overview of what the optimal mDs size is, and how the performance correlate with the nOctUp and treeDepth.

4 Evaluation

In this paper the parameters optimization will be done according to the parameter variations seen in Table 1. The parameters optimization will be performed on nighttime sequence 1 from the LISA TL dataset which are collected in an urban environment in San Diego, USA and contain 4,993 frames and 18,984 annotations. The data is generated from a 5 min and 12 s long video sequence containing 25 physical TLs split between 5 different types: go, go left, warning, stop, and stop left [11].

The mDs are decreased in the last two iteration in Table 1 as the training time increases significantly when the nOctUp and treeDepth are increased. As the training have been done on multiple different computers, the average training time, defined in Table 1, is calculated from calculated the average training time from the computer being involved in all 6 iterations for the most comparable results. The most involved computer is a Lenovo Thinkpad T550 with an Intel i7-5600U CPU @ 2.6 GHz, 8 GB of memory, and a SSD page file. The parameter sweep was done using MATLAB R2015b on Windows 7 Enterprise, both 64-bit.

Table 1. ACF detector parameter variation

mDs Start	mDs End	nOctUp	treeDepth	# Detectors	Avg Time [DD:HH:MM]
[12, 12]	[25, 25]	0	2	196	00:02:59
[12, 12]	[25, 25]	0	4	196	00:04:40
[12, 12]	[25, 25]	1	2	196	01:02:43
[12, 12]	[25, 25]	1	4	196	01:06:34
[15, 15]	[22, 22]	2	2	64	02:19:28
[15, 15]	[22, 22]	2	4	64	02:21:19
				912	

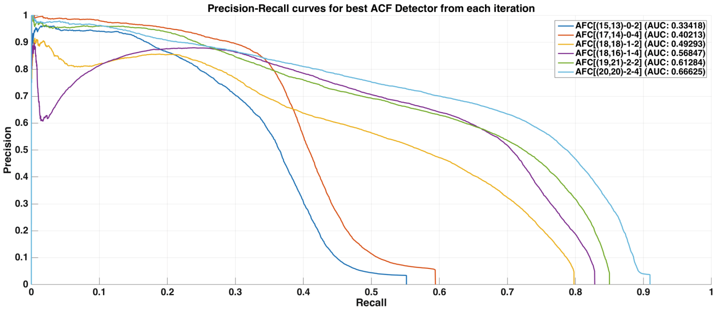


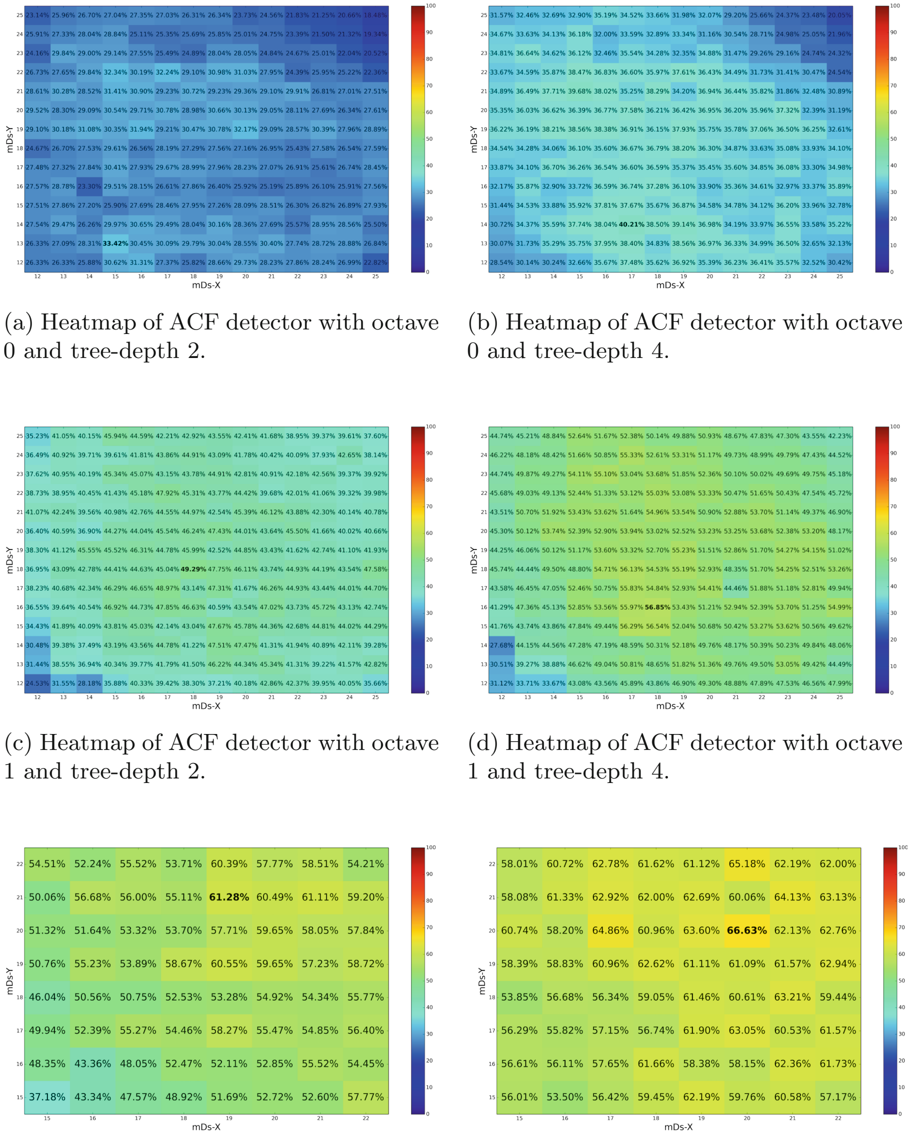
Fig. 3. PR-curves of best ACF detector from each heatmap.

Each detections will be quantified in accordance to the VIVA-challenge [9], where the Area-Under-Curve (AUC) of a Precision-Recall curve (PR-curve) generated from the ACF results is used as 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 Eq. (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 [26].

In Fig. 4, the 6 different parameter variation sweeps, defined in Table 1, are seen. All of the heatmaps are plotted with the same color range, spanning from dark blue to dark red indicating a detection rate of 0% and 100%, respectively. For each heatmap plot in Fig. 4, the model dimension with the highest detection rate is marked with bold. By examining the figures in pairs, e.g. 4a+4b and 4a+4c, one can determine the effect of changing tree-depth or octave, respectively. Increasing only the octave from 0 to 1 increases the best performance from 33.42% to 49.29%. Furthermore, the average AUC of the entire heatmap is also increased significantly as a result of the octave increment, which is best illustrated by the increase of more bright green areas in Fig. 4c compared to Fig. 4a.



(a) Heatmap of ACF detector with octave 0 and tree-depth 2.

(b) Heatmap of ACF detector with octave 0 and tree-depth 4.

(c) Heatmap of ACF detector with octave 1 and tree-depth 2.

(d) Heatmap of ACF detector with octave 1 and tree-depth 4.

(e) Heatmap of ACF detector with octave 2 and tree-depth 2.

(f) Heatmap of ACF detector with octave 2 and tree-depth 4.

Fig. 4. Heatmaps of ACF detector with varying octaves and tree-depths. (Color figure online)

Increasing the tree-depth from 2 to 4 increases the best performing mDs with 6.79%, and the overall average AUC is also increased by comparing the color schemes of Fig. 4a and b. In Fig. 4d both the octave and tree depth is increased to respectively 1 and 4, resulting in an AUC of 56.85% with a mDs of $[18, 16]$. There are no clear tendency of a groupings of mDs where the detection rate is good in Fig. 4a. In Fig. 4a–d, a grouping with a lower detection rate is present in the upper right corner and the lower left corner, which suggests that the optimal mDs is found between a size of 15 and 22. Finally, the octave increased in Fig. 4e and f, where only the detection with mDs from 15 to 22 have been executed due to time restrictions and the previously mentioned low detection rate grouping analysis. Increasing the octave to 2 increases the AUC to 61.28 with a tree-depth 2, and finally 66.63%, which is the highest achieved AUC in this parameter sweep.

In Fig. 3, the Precision-Recall curves of the best performing mDs from each heatmap are seen. The precision is decent when the recall is under 0.35 for all of the detections, meaning that we have high confidence in our detections until this point. The detections with octave 0 detects less than 60% of the true positives, by increasing the octaves the recall, and number of true positives detections, are greatly improved reaching over 90% with octave 2 and tree-depth 4. By increasing the octave all detections reaches a recall above 79% resulting in a higher AUC.

5 Conclusion

Increasing only the octave provides us with better capabilities of detect a larger size range of TLs, resulting in the most significant AUC increments. The increments of the tree-depth improves the results when keeping the octave unchanged, however, the AUC increase is not as high as increasing the octave while keeping the tree-depth the same. The AUC is nearly doubled by increasing both of tree-depth and octave in Fig. 4a and d, leading to conclusion that these parameters are correlated, as the color scheme strongly show the overall AUC increase. Finally, the AUC is improved by increasing octave and tree-depth additionally, as seen in Fig. 4e and f, respectively. As in the first 4 iteration heatmaps, the best performing AUC is increased when increasing both octave and tree-depth simultaneously, which supports the conclusion that the parameters are highly correlated. By examining Fig. 4f it is clear that the best performing AUC is increased additionally and found at a mDs of $[20, 20]$ with 2 octaves and a tree-depth of 4.

Further experiments includes finding the convergence points by keep increasing the parameters. Additionally, a similar parameter sweep on the daytime data from the LISA TL dataset would be interesting.

References

1. Dollár, P., Belongie, S., Perona, P.: The fastest pedestrian detector in the west. In: BMVC (2010)
2. Dollár, P.: Piotr's Computer Vision Matlab Toolbox (PMT) (2016). http://vision.ucsd.edu/pdollar/toolbox/doc/index.html

3. Trivedi, M.M., Gandhi, T., McCall, J.: Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety. IEEE Trans. Intell. Transp. Syst. **8**, 108–120 (2007)
4. Dollar, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1532–1545 (2014)
5. Mogelmose, A., Liu, D., Trivedi, M.M.: Traffic sign detection for US roads: remaining challenges and a case for tracking. In: IEEE Transactions on Intelligent Transportation Systems. pp. 1394–1399 (2014)
6. Jensen, M.B., Philipsen, M.P., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Traffic light detection at night: comparison of a learning-based detector and three model-based detectors. In: 11th Symposium on Visual Computing (2015)
7. Das, N., Ohn-Bar, E., Trivedi, M.: On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), pp. 2953–2958 (2015)
8. The Insurance Institute for Highway Safety (IIHS): Red light running (2015)
9. Laboratory for Intelligent, Safe Automobiles, UC San Diego: Vision for Intelligent Vehicles and Applications (VIVA) Challenge (2015). http://cvrr.ucsd.edu/vivachallenge/
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
11. 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 Trans. Intell. Transp. Syst. **PP**, 1800–1815 (2015)
12. Diaz-Cabrera, M., Cerri, P., Medici, P.: Robust real-time traffic light detection and distance estimation using a single camera. Expert Syst. Appl. **42**, 3911–3923 (2014)
13. Kim, H.K., Shin, Y.N., Kuk, S.g., Park, J.H., Jung, H.Y.: Night-time traffic light detection based on SVM with geometric moment features. In: 76th World Academy of Science, Engineering and Technology, pp. 571–574 (2013)
14. Omachi, M., Omachi, S.: Detection of traffic light using structural information. In: IEEE 10th International Conference on Signal Processing (ICSP), pp. 809–812 (2010)
15. Siogkas, G., Skodras, E., Dermatas, E.: Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information. In: VISAPP (1), pp. 620–627 (2012)
16. Sooksatra, S., Kondo, T.: Red traffic light detection using fast radial symmetry transform. In: 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1–6. IEEE (2014)
17. Trehard, G., Pollard, E., Bradai, B., Nashashibi, F.: Tracking both pose and status of a traffic light via an interacting multiple model filter. In: 17th International Conference on Information Fusion (FUSION), pp. 1–7. IEEE (2014)
18. Charette, R., Nashashibi, F.: Traffic light recognition using image processing compared to learning processes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 333–338 (2009)
19. Zhang, Y., Xue, J., Zhang, G., Zhang, Y., Zheng, N.: A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles. In: 33rd Chinese Control Conference (CCC), pp. 4924–4929 (2014)
20. Koukoumidis, E., Martonosi, M., Peh, L.S.: Leveraging smartphone cameras for collaborative road advisories. IEEE Trans. Mob. Comput. **11**, 707–723 (2012)

21. Nienhuser, D., Drescher, M., Zollner, J.: Visual state estimation of traffic lights using hidden Markov models. In: 13th International IEEE Conference on Intelligent Transportation Systems, pp. 1705–1710 (2010)
22. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.: Making Bertha see. In: IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 214–221 (2013)
23. Lindner, F., Kressel, U., Kaelberer, S.: Robust recognition of traffic signals. In: IEEE Intelligent Vehicles Symposium, pp. 49–53 (2004)
24. Barnes, D., Maddern, W., Posner, I.: Exploiting 3D semantic scene priors for online traffic light interpretation. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea (2015)
25. Philipsen, M.P., Jensen, M.B., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Traffic light detection: a learning algorithm and evaluations on challenging dataset. In: 18th IEEE Intelligent Transportation Systems Conference (2015)
26. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)