Semantic segmentation based traffic light detection at day and at night.

Vladimir Haltakov[1,2], Jakob Mayr[1,3], Christian Unger[1,2], and Slobodan Ilic[2]

[1] BMW Group, Munich, Germany
{vladimir.haltakov,christian.unger}@bmw.de
[2] Technical University Munich, Germany
slobodan.ilic@in.tum.de
[3] Munich University of Applied Sciences, Germany
jamayr@web.de

Abstract. Traffic light detection from a moving vehicle is an important technology both for new safety driver assistance functions as well as for autonomous driving in the city. In this paper we present a machine learning framework for detection of traffic lights that can handle in real-time both day and night situations in a unified manner. A semantic segmentation method is employed to generate traffic light candidates, which are then confirmed and classified by a geometric and color features based classifier. Temporal consistency is enforced by using a tracking by detection method.
We evaluate our method on a publicly available dataset recorded at daytime in order to compare to existing methods and we show similar performance. We also present an evaluation on two additional datasets containing more than 50 intersections with multiple traffic lights recorded both at day and during nighttime and we show that our method performs consistently in those situations.

1 Introduction

In the past decade various advanced driver assistance systems (ADAS) have found their way into series production and today almost all car manufacturers offer a wide variety of comfort and safety features like for example speed limit information, adaptive cruise control and automatic emergency breaking. In addition, safety organizations like the EuroNCAP and the equivalent institutions in other countries, traditionally performing crash tests to assess passive safety, are developing and introducing new test procedures for active safety systems, which further promote the usage of ADAS in commercial vehicles. While there are multiple sensors that can be used in such systems, cameras are usually the most universal and cheapest choice, because they have the highest spatial resolution and are able to detect the highest variety of object types, e.g. traffic signals, lane markings and other road users.

The position and the current state of the traffic lights in front of the vehicle is a valuable information for many safety and comfort driver assistance functions.

Traffic lights detection is needed in order to enable autonomous and highly automated driving in cities and on country roads. Furthermore, red light running is a major safety problem, with estimated 165,000 motorists, cyclists and pedestrians injured in the USA every year, a lot of which fatal [15, 1]. Similar studies in Germany [2] show that 7,356 incidents with people or property damaged happened in 2013 because of disregarding traffic signals at intersections.

In this paper, we focus on the problem of detecting the presence and the state of traffic lights from camera images both at day and at night. Day and night scenes pose fundamentally different challenges for visual traffic light recognition. At day, the structure of the traffic light is well visible, but the light source can be difficult to detect due to the presence of many other bright image regions especially in sunny weather. Furthermore, traffic lights are relatively small in width compared to traffic signs or other road users, which makes the detection at large distances difficult. In contrast, at nighttime, light sources are visible from a very high distance, but since the traffic light box is usually not visible in the camera image (or only at very short distances), there is no textural support to distinguish the traffic lights from other light sources like street lamps and advertisements. Fig. 1 shows examples of such difficult situations.

Fig. 1. The same scene recorded on a sunny day and at night showing the challenges for a traffic light detection system that needs to operate in all conditions.

The method presented in this paper is based on machine learning and can handle both day and night situations in a unified manner, such that only the trained classifier parameters are different for day and night, while the whole method remains unchanged. While there is a vast amount of literature on traffic light recognition, only very few vision methods deal with both day and night situations [18, 23]. A detailed overview of the related work is given in Section 2.

Our method consists of two main stages. First, we use a pixel-wise semantic segmentation method similar to [13] to find image regions that are potential traffic light candidates. While similar image segmentation steps, usually based on color thresholding, are used by many other systems, we show that more advanced machine learning methods like our semantic segmentation approach can provide more robust candidates. In a second step, we compute multiple color and geometric features on the regions found in the first step, which are then used by another classifier to confirm or reject the candidates and to also

determine the current color of the traffic light. Additionally, a tracking algorithm is used to enforce temporal consistency.

The proposed system is evaluated on two datasets with 57 intersections recorded both at day and at night in order to show that we can handle both scenarios using the same approach. Furthermore, we also present our results on the publicly available dataset of [4], which contains only daytime recordings.

Our main contribution is a unified framework for real-time traffic light detection both at day and at night based on semantic segmentation to generate traffic lights proposals and the subsequent classifier used to confirm or reject those candidates based on geometric and color features.

2 Related work

We divide the related methods in three groups based on the situations they operate in: at day, at night or both. While there are works that rely on high-accuracy maps as a prior for the traffic lights position in the camera image [8, 10], here we focus on purely vision based systems, because they pose unique challenges.

2.1 Detection at day

Most of the related works focus on traffic light detection at day. Many methods rely only on pure image processing by applying color segmentation followed by geometric and visual filters [3, 7, 11, 21, 22, 24]. Those methods may deliver good results if the light shape is clearly visible, but it is not clear if they can scale well to various traffic light types and night conditions. The evaluation provided on those methods is also very limited and sometimes only qualitative.

The methods described in [4, 26] rely on template matching in addition to image processing techniques, which increases the robustness of the system in some situations, but they work only during the day. Both methods are evaluated on the dataset or part of it that is introduced in [4], which we also use for the quantitative evaluation of our method.

More powerful machine learning methods are employed by [5, 12, 16, 20] in order to learn the appearance of the traffic lights at day. However, at night most parts of the traffic light are not visible, so it is not clear if those methods can be extended to also work in all situations.

Most of the works above provide very limited evaluation based on short sequences of couple of minutes or done only qualitatively, which makes comparison of performance difficult.

2.2 Detection at night

The big challenge for traffic light detection methods at night is to filter out light emitting objects that are not traffic lights. Several works exist that explicitly focus on the night detection problem either by using template matching methods

[9, 19], support vector machines classification [17] or just image processing [6]. However, due to the lack of a publicly available datasets for traffic lights detection with night recordings, those methods are tested only qualitatively or on small non-public datasets.

2.3 Detection at day and night

The methods that are most strongly related to ours are those that are designed to deal both with day and night conditions [23, 18].

The authors of [18] use a pipeline consisting of image adjustments in the RGB space, thresholding and applying a median filter to detect traffic lights in different weather and illumination conditions. However, the scenarios where this method is applied are limited, because only suspended traffic lights are detected, while at many smaller intersections, only supported traffic lights are available.

Another system designed to handle both day and night situations as well as adverse weather conditions is presented by [23]. The authors employ a color pre-processing step, followed by a fast radial symmetry transform to extract candidates and a spatio-temporal consistency check to reduce false positives. While the detection at day is quantitatively evaluated on the dataset of [4], the night detection is evaluated only qualitatively, which makes comparison of the performance in different situations impossible.

Our method is evaluated quantitatively both on the dataset of [4] and on two new datasets recorded at night and at day. In this way, we are able to analyze the performance of our method in different lighting conditions.

3 Method

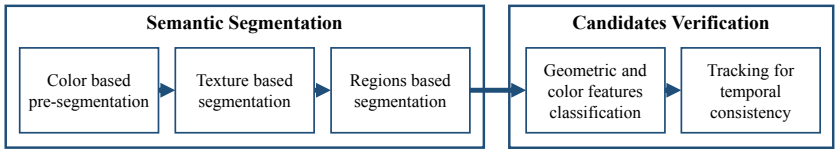


Fig. 2. Overview of the method pipeline.

The general method pipeline is illustrated in Fig. 2. A semantic segmentation algorithm is first used to label each image pixel and find potential candidate regions in the image. Those regions are then verified by a classifier based on several color and geometric features, which are also used to determine the state of the traffic light. The verification stage also includes a tracking step, which helps to enforce temporal consistency on the traffic lights.

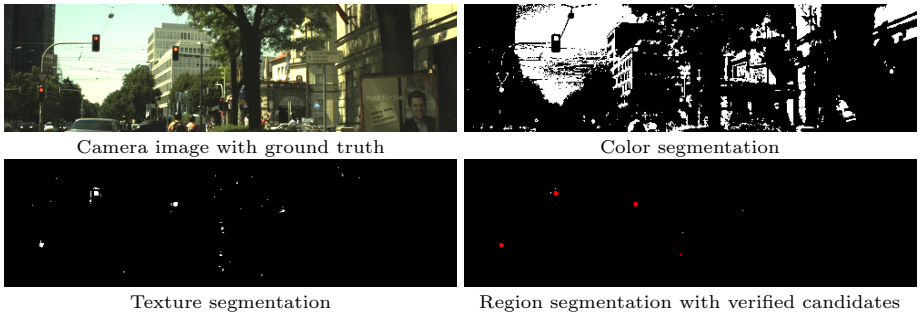


Fig. 3. Intermediate results of our method at the different stages of the pipeline. At every step the number of traffic light candidates (in white) is reduced. The candidate regions confirmed by the classifier in the last stage are marked in red.

3.1 Semantic segmentation based candidates

The goal of this stage is to find regions in the image that are potential traffic light objects. In this stage, having false positives (e.g. candidates that are not traffic lights) is not critical, since the subsequent verification stage is designed to filter them out. The number of false negatives, on the other hand, needs to be low, because missed traffic lights will not be evaluated in the next steps. Nevertheless, a segmentation method that has few false positives is desirable since the verification stage will be both more accurate and more efficient. It is also important to note that our goal is to design a method that will be applicable both at day and at nighttime.

The semantic segmentation problem aims to divide the image into semantically meaningful regions. This is usually done by classifying each image pixel x_i with a label y_i from a predefined set of labels $\mathcal{L}$. In our case, we need only two labels: BACKGROUND or TRAFFIC LIGHT CANDIDATE. Our goal here is to label only the light spot of the traffic light and not the whole box, because in most cases the box is not visible in the camera image at night.

We employ a three-step semantic segmentation method based on the method of [13]. Each step follows the same approach: for every pixel we compute features from the image and from the result of the previous steps. Each pixel is then classified based on those features by a JointBoost [25] classifier. The three steps are described in detail below, while in Section 4.3 we show how they contribute to the final detection performance.

Color segmentation The first step is a simple color segmentation used to improve the runtime of the method. Instead of tuning the color thresholds by hand, we employ a classifier that uses only the color of the pixel in the Lab color space as input and is biased to have few false negatives on traffic light candidates by giving the traffic light pixels very high weight (see Fig. 3). Formally, the color classifier models the conditional probability distribution $P(y_i|x_i)$ of the pixel

label y_i given the pixel intensities x_i. The subsequent steps ignore all pixels that were labeled as background.

Texture segmentation In the second step, the pixels are classified based on the texture in their surrounding area. For this we compute a feature vector $f(x_i)$ based on the 2D Walsh-Hadamard Transform [14], which is a discrete and computationally efficient approximation of the cosine transform and has successfully been used for template matching [14] and semantic segmentation [13, 27]. Similarly to the other works, we compute the first 16 coefficients of the transform separately for each Lab color channel at five scales around the pixel of interest. We also add the 2D coordinates of the pixel to the feature vector to encode spatial context. The classifier operating on those features can be seen as modeling distribution $P(y_i|f(x_i))$.

While the classifier trained on texture features is already able to provide good results, the shape of the regions may not be very robust due to small pixel errors around the borders (see Fig. 3). This happens because the classifier takes the decision about the class of each pixel individually and independently of the labels of the neighboring pixels. This problem is addressed in the next step.

Region segmentation This step is equivalent to the neighborhood classification stage from [13]. The region classifier considers not only the pixel of interest itself, but also a set of related pixels called a neighborhood. While in [13] several alternatives are proposed that deliver highly accurate results, they are based on geodesic distance which is slow to compute. We define the neighborhood N_i of pixel i to contain all pixels in a circle of radius 3 around each pixel, because it is much more computationally efficient.

Every pixel j in the neighborhood N_i votes for its most probable class v_j based on the output of the classifier in the texture segmentation step $P(y_j = v_j|f(x_j))$. Those votes are then summarized in a normalized histogram h_i over the possible labels $c \in \mathcal{L}$. Formally, we write:

$$h_i(c) = \frac{\sum_{j \in N_i} [c = v_j]}{|N_i|}. \tag{1}$$

The normalized histogram computed in this way is used as a feature vector for the region segmentation classifier together with the response of the pixel itself, which means that the region classifier models the distribution $P(y_i|h_i, P(y_i|f(x_i))$. This formulation allows the classifier to model local context relations, which leads to better segmentation performance and better candidate regions (see Fig. 3).

3.2 Candidates verification

The semantic segmentation method introduced in the previous section learns texture features and label interactions that are characteristic for traffic lights. However, since the classifiers from the segmentation stage classify each pixel individually, it is difficult to model geometric features that describe whole regions,

like for example, if the region has a circular shape. Therefore, in the verification stage we train another classifier based on the region geometry and color features. The classifier does not take decisions on the pixel level anymore, but on the region level. Furthermore, we introduce a simple tracking by detection algorithm in order to enforce temporal consistency of the detections.

Traffic lights classifier Each candidate region coming from the semantic segmentation method is classified in the classes BACKGROUND, GREEN, YELLOW or RED traffic light. The input to the classifier is a set of 21 geometric and color features described in Table 1. Here, we again make use of the JointBoost classifier, which now operates on regions instead of pixels. The result of the classification is shown visually in Fig. 3, where only the candidates that were classified correctly are painted in the corresponding color, while the white candidates are rejected.

Table 1. Geometric and color features used for the classification of candidate regions.

Feature	Values	Description
Mean (RGB) Mean (Lab)	3 3	Mean of the region pixels computed separately for each color channel.
Std. deviation (RGB) Std. deviation (Lab)	3 3	Standard deviation of the region pixels computed separately for each color channel.
Image position	2	The pixel coordinates of the center of the region.
Area	1	Area of the region.
Orientation	1	Angle between the x-axis and the major axis of the region.
Aspect ratio	1	Aspect ratio of the two sides of the region's bounding box.
Ratio of areas	1	Ratio of the areas of the region and its bounding box.
Y-coordinate-area ratio	1	Ratio between the region's center y-coordinate and its area.
Solidity	1	Ratio between the areas of the region and its convex hull.
Eccentricity	1	Ratio of the distance between the foci and the major axis length of an ellipse that has the same second moment as the region.

Tracking Since the verification method described above operates on individual frames, one can often observe sporadic false detections that last only one or two frames or detected traffic lights can be missed for several frames mainly due to motion blur or due to LED traffic lights appearing too dark in some frames.

To deal with this problem we introduce a simple tracking by detection algorithm to enforce temporal consistency. The traffic lights are detected separately in each frame and then the detections from two subsequent frames are matched based on the distance between them. This allows us to determine the number of frames each traffic lights has been tracked and only traffic lights that were already seen in at least three frames are counted as detected.

4 Results

We evaluate our method on three challenging datasets in both day and night situations and present a comparison with two related works. Furthermore, we evaluate the influence of the different steps of our method and its runtime.

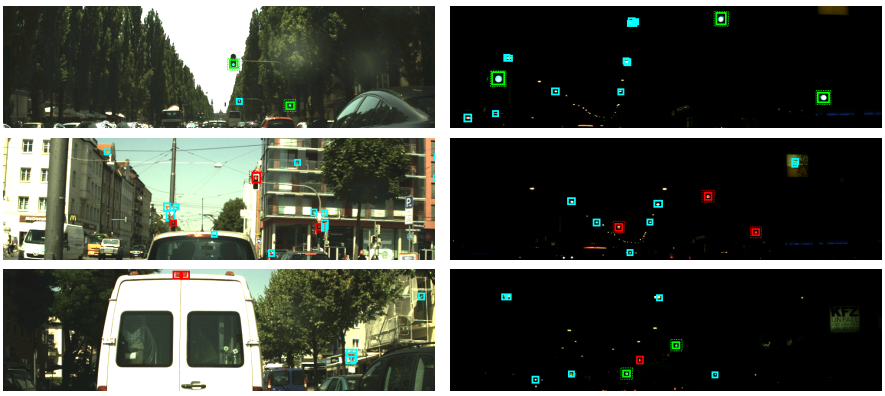


Fig. 4. Results from *Germany Day* and *Germany Night*. The candidates are shown in cyan, the confirmed detections in red or green and the ground truth with a dashed bounding box. The last row shows typical false positive detections.

4.1 Datasets

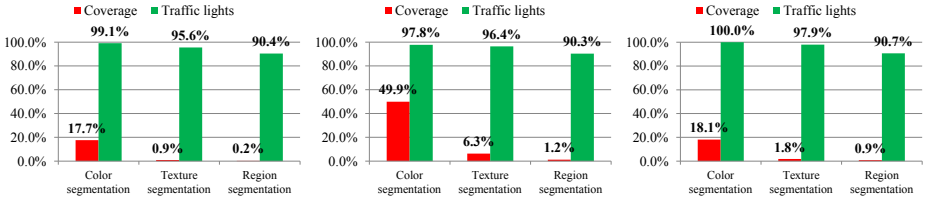
We use the publicly available dataset of [4] which is recorded at day in Paris and has a length of around 17 minutes and manually labeled bounding boxes in each frame. Since there is no fixed training and testing split we perform a 3-fold cross-validation. In the rest of the section we refer to this dataset as *France Day*.

We also created two additional datasets in order to analyze the performance of our method at day and nighttime. We used a 1 megapixel camera taking images at 16 frames per second mounted behind the windscreen of a vehicle. We defined a city route in Germany with a length of around 17 km, which contains 57 intersections with traffic lights ranging from side streets to big multi-lane streets. The recordings were done both on a sunny day and at night. All traffic lights have been labeled with bounding boxes around the 3 lights for the day scenes and around the illuminated light only for the night scenes, if the light source is bigger than 5 pixels in the camera image. We refer to these datasets as *Germany Day* and *Germany Night* respectively. Because of the small number of yellow traffic lights in all of the datasets, they are ignored during evaluation.

4.2 Comparison to related methods

Two of the related methods [4, 23] have published results on the complete *France Day* dataset so that we can perform a quantitative comparison.

Fig. 5. Results from *France Day* [4]. The candidates are shown in cyan, the confirmed detections in red or green and the ground truth with a dashed bounding box.



France Day Germany Day Germany Night

Fig. 6. Results of the 3 steps of the segmentation stage showing the percentage of all image pixels labeled as candidates and the correctly classified traffic light pixels.

Although the authors of [4] use precision and recall as benchmark measure, they define a computation rule based on temporal tracks instead of frames. This means that one physical traffic light is counted as correctly detected if it is detected in at least one frame during its lifetime. Since our method is able to detect 33 of the 34 traffic lights in at least one of the frames (we also consider the partially occluded ones), the recall of our method is 97.1%, while the authors of [4, 23] report 97.7% and 93.8% respectively. Unfortunately, the authors of [4] do not give precise description of how they compute the precision measure. For details about the false positive detections of our method we refer to our frame based precision in the next section.

4.3 Method analysis

Semantic Segmentation For the training of the semantic segmentation method all bounding boxes are first converted into pixel-wise labels on the active light spot of the traffic light. The performance of the three segmentation steps is measured according to the percentage of pixels labeled correctly as BACKGROUND or CANDIDATE. While this does not directly translate to detection rate for the traffic lights, because some traffic light regions could be only partially segmented, it is a very good indicator.

From the quantitative results shown in Fig. 6 we see that with every step in the pipeline the number of pixels labeled as traffic lights ("Coverage") decreases significantly, while our semantic segmentation method is able to retain almost all of the real traffic lights ("Traffic lights"). While the *Germany Day* dataset is more challenging for the simple color segmentation due to the big variety

of traffic lights and illumination conditions because of the sunny weather, the region segmentation step achieves results similar to those of the other datasets.

Table 2. Quantitative results based on frame-wise recall and precision.

Stage	France Day		Germany Day		Germany Night	
	Recall	Precision	Recall	Precision	Recall	Precision
Without tracking	76.1%	63.3%	91.6%	61.3%	91.5%	57.4%
With tracking	71.7%	73.2%	84.3%	71.5%	84.4%	73.8%

Candidates Verification The tracks based recall measure used by the authors of the *France Day* dataset [4] is not suitable for many functions that need a stable tracking of the traffic lights while approaching the intersection, like for example red light warning or autonomous breaking. Therefore, we employ a frame based measure of recall and accuracy, which are more natural for the mentioned functions.

The quantitative results on all three datasets are summarized in Table 2 with our method achieving similar performance in all scenarios. Fig. 4 and Fig. 5 show some example detections. The tracker is an essential step to reduce the amount of false positives both at day and at night, because they tend to appear only for short periods of time.

System runtime Semantic segmentation methods can be slow in general, since they need to classify each image pixel. Our three-step semantic segmentation approach, however, filters out many of the pixels in the first step, so that the more expensive texture analysis is performed only on the relevant image parts.

The total runtime of our method is 65ms per frame, with the semantic segmentation accounting for 92% of it. All experiments were performed on a machine with 2 Intel Xeon X5690 processors running at 3.5 GHz. The code is written in C++ without the use of SSE instructions and is only partially parallelized.

5 Conclusion

In this paper, we presented a unified machine learning framework for traffic light detection at different lighting conditions. The used powerful semantic segmentation method is able to provide robust candidates both at day and at night by analyzing the image structure. We also describe several geometric and color features that are used to reject false candidates and to classify the color of the traffic light. An additional tracking by detection step is important for enforcing consistency of the results over time and reducing the amount of false positives.

We showed that our method runs in real-time and delivers good results on three challenging datasets recorded in different illumination conditions and containing data from more than 100 intersections with multiple traffic lights.

References

1. Traffic Safety Facts 2008. National Highway Traffic Safety Administration (2008)
2. Fachserie. 8, Verkehr. 7, Verkehrsunfälle. Statistisches Bundesamt Wiesbaden (2013)
3. Cai, Z., Li, Y., Gu, M.: Real-time recognition system of traffic light in urban environment. In: CISDA (2012)
4. de Charette, R., Nashashibi, F.: Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In: IV (2009)
5. Chiang, C.C., Ho, M.C., Liao, H.S., Pratama, A., Syu, W.C.: Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts. In: International Journal of Innovative Computing, Information and Control (2011)
6. Diaz-Cabrera, M., Cerri, P.: Traffic light recognition during the night based on fuzzy logic clustering. In: Computer Aided Systems Theory - EUROCAST 2013 (2013)
7. Diaz-Cabrera, M., Cerri, P., Sanchez-Medina, J.: Suspended traffic lights detection and distance estimation using color features. In: ITS (2012)
8. Fairfield, N., Urmson, C.: Traffic light mapping and detection. In: ICRA (2011)
9. Fan, B., Lin, W., Yang, X.: An efficient framework for recognizing traffic lights in night traffic images. In: CISP (2012)
10. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.G.: Making bertha see. In: ICCV Workshop on Computer Vision for Autonomous Driving (2013)
11. Gomez, A., Alencar, F., Prado, P., Osorio, F., Wolf, D.: Traffic lights detection and state estimation using hidden markov models. In: IV (2014)
12. 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: IV (2010)
13. Haltakov, V., Unger, C., Ilic, S.: Geodesic pixel neighborhoods for multi-class image segmentation. In: BMVC (2014)
14. Hel-Or, Y., Hel-Or, H.: Real time pattern matching using projection kernels. ICCV (2003)
15. Insurance Institute for Highway Safety (IIHS): Status Report, Vol. 42, No. 1. Rep. IIHS (2007), `http://www.iihs.org/externaldata/srdata/docs/sr4201.pdf`
16. Jang, C., Kim, C., Kim, D., Lee, M., Sunwoo, M.: Multiple exposure images based traffic light recognition. In: IV (2014)
17. Kim, H.K., Shin, Y.N., gong Kuk, S., Park, J.H., Jung, H.Y.: Night-time traffic light detection based on svm with geometric moment features. In: World Academy of Science, Engineering and Technology (2013)
18. Kim, Y., Kim, K., Yang, X.: Real time traffic light recognition system for color vision deficiencies. In: ICMA (2007)
19. Li, J.: An efficient night traffic light recognition method. In: Journal of Information & Computational Science (2013)
20. Lindner, F., Kressel, U., Kaelberer, S.: Robust recognition of traffic signals. In: IV (2004)
21. Omachi, M., Omachi, S.: Traffic light detection with color and edge information. In: ICCSIT (2009)

22. Shen, Y., Ozguner, U., Redmill, K., Liu, J.: A robust video based traffic light detection algorithm for intelligent vehicles. In: IV (2009)
23. Siogkas, G., Skodras, E., Dermatas, E.: Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information. In: VISAPP 2012 (2012)
24. Tae-Hyun, H., In-Hak, J., Seong-Ik, C.: Detection of traffic lights for vision-based car navigation system. In: Advances in Image and Video Technology (2006)
25. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. In: PAMI (2007)
26. Wang, C., Jin, T., Yang, M., Wang, B.: Robust and real-time traffic lights recognition in complex urban environments. In: International Journal of Computational Intelligence Systems (2011)
27. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: ECCV (2008)