


Article

Frequency maps as expert instructions to lessen data dependency on real-time traffic light recognition.

Hendrik Macedo ^{1,†,*} , Thiago Almeida ^{2,†}, Leonardo Matos ^{1,†}, Bruno Prado ¹

¹ Departamento de Computação (DCOMP), Universidade Federal de Sergipe (UFS)

² Programa de Pós-Graduação em Ciência da Computação (PROCC), Universidade Federal de Sergipe (UFS)

* Correspondence: hendrik@dcomp.ufs.br

† These authors contributed equally to this work.

Abstract: Research on Traffic Light Recognition (TLR) has grown in recent years, primarily driven by the growing interest in autonomous vehicles development. Machine Learning algorithms have been widely used to that purpose. Mainstream approaches, however, require large amount of data to properly work, and as a consequence, a lot of computational resources. In this paper we propose the use of Expert Instruction (EI) as a mechanism to reduce the amount of data required to provide accurate ML models for TLR. Given an image of the exterior scene taken from the inside of the vehicle, we stand the hypothesis that the picture of a traffic light is more likely to appear in the central and upper regions of the image. Frequency Maps of traffic light location were thus constructed to confirm this hypothesis. The frequency maps are the result of a manual effort of human experts in annotating each image with the coordinates of the region where the traffic light appears. Results show that EI increased the accuracy obtained by the classification algorithm in two different image datasets by at least 15%. Evaluation rates achieved by the inclusion of EI were also higher in further experiments, including traffic light detection followed by classification by the trained algorithm. The inclusion of EI in the PCANet achieved a precision of 83% and recall of 73% against 75.3% and 51.1%, respectively, of its counterpart. We finally presents a prototype of a TLR Device with that expert model embedded to assist drivers. The TLR uses a smartphone as a camera and processing unit. To show the feasibility of the apparatus, a dataset was obtained in real time usage and tested in an Adaptive Background Suppression Filter (AdaBSF) and Support Vector Machines (SVMs) algorithm to detect and recognize traffic lights. Results show precision of 100% and recall of 65%.

Keywords: Traffic Light Recognition (TLR); Machine Learning; Expert Instruction (EI); Frequency Maps; Computer Vision

1. Introduction

According to Brazilian's 2014 highway accident report, there were 150,000 accidents in that year, with 425 fatalities. 5% of these are related to the contempt to traffic signs [1]. Although traffic lights are worldwide used and known as a very intuitive traffic regulator, another study [2], from the city of São Paulo/Brazil, points out that in August 2016 the advance of the red light was the second most frequent violation associated with fatal accidents involving vehicles. Some particular situations can be considered as probable causes of the red light advance:

- Poorly located traffic lights that cannot be noticed in advance;
- Defective traffic lights or with very dim light;
- Ambient lighting that disturbs driver's vision;
- Driver's visual impairment;
- Hesitation at the timing to advance when the signal turns yellow;

- Number of traffic signaling to notice.

The first two aspects mentioned above can be solved by a preventive maintenance policy for traffic signs. A Traffic Light Recognition (TLR) device can help with the remaining aspects. According to [3], the main task of a DAS is to prevent accidents by warning the driver of the existence of a red or yellow traffic light ahead in a non-intrusive way. In addition, the TLR can detect the main traffic light for that road when there is more than one and how far the traffic light is.

There are, though, several issues to be overcome in order to reliably detect and recognize the state of a traffic light and implementing a robust TLR is not a trivial task [3]. Among the challenges, are the weather condition that alters the lighting of the environment, the existence of different types of traffic lights (horizontal or vertical, suspended or pole), the angle from which the traffic light is being observed which can distort its geometric characteristics and traffic lights that are partially hidden due to exterior aspects.

Current work with Image Processing (IP) techniques and Machine Learning (ML) from traffic lights image collections shows accuracy around 80%, except for the times when limitation occurs. A way to get around the limitations can be the use of knowledge about the problem and the environment that cannot be obtained directly from the image. We call such type of knowledge of *Expert Instruction* (EI) since knowledge may be implicit in the form the model is built and not in the data itself, a theory that this paper proposes to validate. Embedding expert instructions can also favor a smaller amount of data needed by ML algorithms to perform well, which is highly desired considering that collecting and categorizing traffic light data is indeed a extreme laborious task.

We are particularly interested in what is the impact of knowing the traffic light object location in a photograph obtained from a camera in the interior of the vehicle to the ML-based recognition process. The hypothesis is that there should be regions where the traffic light object image appear more frequently within the photograph (on the top? on the bottom? right? left?) and, for that reason, these regions may have higher likelihood. We believe that *frequency maps* properly sum up the result of a deep analysis performed by human experts over traffic light image datasets towards the object location discovering.

This work thus intends to assess whether the use of a traffic light detection and recognition model that explicitly embeds frequency maps as expert instructions achieves better results than a purely ML-based solution. Scientific contribution is two-fold: (1) provide a human-tailored mechanism to produce frequency maps as expert instructions from traffic light image datasets and (2) design an hybrid model to the automatic recognition of traffic lights that demands lower data.

In section 2 we list some of the main attempts to solve the problem. Three of them were chosen as the basis for our investigation and so we highlight enrolled techniques to expose the theoretical foundations of our approach and ease the understanding. Section 3 presents our approach to embed frequency maps as expert instruction in the machine learning model training. Results of model training as also shown. In section 4 we present the TLR device we developed to show the feasibility of our proposal. We conclude our work in section 5.

2. Current approaches for smart TLR device

An object recognition mechanism in an image consists of two parts: (1) a detection phase to identify target objects and (2) a posterior classification phase. For traffic lights, colors, shape and light are often the most important features to attend.

A sort of different techniques has been tried to deal with the detection phase. [4], [5], [6], [7], for instance, uses Convolutional Neural Network (CNN) to detect the presence of a semaphore in a image whereas [8] considers a PCAnet Neural Network to the same purpose. [9] uses a learning algorithm based on the image features channels and gradient histograms. Saliency maps are used by [10], [11], [12], [13] and [6]. [14], [15], [16] employ *Blob* detection algorithms. Some of the geometric transforms adopted in the detection phase are the Circular Hough Transform [17], [18] and [19] and the Radial Symmetry Fast Transform [20]. Other less common techniques have also been noticed such

as Adaptive Filters [21], *Template Matching* [22], Gaussian Distribution [23], Estimation of probability class associated with CNN [4] and *Top Hat* [24]. The use of image processing algorithms as one of the main techniques is also common; color or shape segmentation, for instance, is considered by [25], [26], [27] and [28].

In the recognition phase, most work employs machine learning algorithms such as Neural Networks or Support Vector Machines (SVM). [21], [14], [29], [30], [15], [8], [13], [31], [32], [33] employ SVMs as the main technique to recognize the semaphore. A non machine learning approach to recognition can be seen in the works of [11] and [34], where Fuzzy Logic has been successfully applied. Actually, many other techniques have been applied in this phase as an alternative to machine learning to improve false positive detection rates or to soften the connection between detection and recognition phases. [35], [14], [30], [13], [12], [36] use Histograms whereas [37], [38], [19] and [39] considers the use of Transforms for such purposes. The use of Template Matching [6], [40], [41], [42], Saliency Map [7], Normalized Cross Correlation [43] and Hidden Markov Models [44] should also be emphasized.

[15] proposes a new method that combines computer vision and machine learning techniques to recognize traffic lights. Color extraction and *blob* detection are used to perform detection. Color extraction is performed in the HSV color space. Blob detection is implemented by combining image processing algorithms such as flooding, contour tracking and closing. The combination of such techniques allows both circular traffic lights and traffic lights with arrows to be identified. After detecting the regions of interest, a Principal Component Analysis (PCA) classifier is applied. The proposed PCA classifier consists of a PCA network (PCAnet), a type of neural network that simulates the functioning of a CNN with fewer layers and a linear SVM. Principal Component Analysis (PCA) is a statistical method that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This allows PCAnet to simulate the behavior of a CNN with less computational cost. In general, convolution on PCAnet is performed using PCA filters as *kernel*; this filters are found by unsupervised learning during the training. The number of PCA filters can vary and, according to [15], the more PCA filters the better the network performance. In the work, 8 (eight) PCA filters were used within two layers. For each set of pixels in the input image there is a set of pixels of the same size in the filter. The average is calculated and the value is removed from the filter. Convolution is thus performed. The combination of averaging and convolution are seen as stages (or layers) of the network. The output of PCAnet contains positive and negative values, which are binarized and then converted to decimal values. A feature vector is formed with these values and is then used as an input to the SVM. Five PCA networks (number of groups of identified traffic lights) are created to estimate weights or features capable of identifying the types of traffic light. The weights then feed the SVM that perform the final classification. Tests are performed using a dataset produced and made publicly available by the authors.

[21] proposes an adaptive background suppression algorithm in order to highlight regions of interest, the AdaBSF. Normalized gradients of the input image are calculated. Combining the normalized gradients with the values of the R, G, and B layers of the original image gives a simple 4-layer feature map. Each W_i detection window in this feature map can be represented as a basic feature vector $x_i \in \mathbb{R}^{D \times 1}$. When searching for vertical and horizontal traffic lights, the window size is fixed at 16×8 pixels and 8×16 pixels, respectively. The size of each detection window is then $D = 16 \times 8 \times 4 = 512$. To solve the multi-scale problem in which the traffic light can appear in different sizes in the image, the original image is re-scaled while the detection window is kept to the same size. The purpose of the AdaBSF algorithm is to specify a Finite Impulse Response filter specified by the vector $w = [w_1, w_2, \dots, w_{256}]^T$ so that $y_i = w^T x_i$. The value y_i represents a *score* for each detection window that represents the probability of that window be part of a semaphore, x_i represents the weights learned by the algorithm. In the recognition phase, each candidate region is checked and classified into different traffic light classes using an SVM. The local color histogram and the gradient-oriented histogram of each candidate region are used as descriptors of linear SVM

characteristics and are used to train the model and to classify the regions found by AdaBSF. The tests are performed with a specific dataset, provided by the author and discussed in the following section.

3. Embedding expert instructions

According to [45], the main problem that machine learning algorithms face is the amount of data needed to learn correctly. In addition to the need for a reasonably large amount of data, these need to be well balanced and categorized. Building a data set with the aforementioned requirements requires time and human effort, and may be even unfeasible considering the natural shortage of data in some specific domain. Finally, more data implies more processing power needs, which can be another serious limiting factor.

One way to reduce the amount of data needed to train the algorithms is to introduce some sort of previously known information about the data (or the domain). This type of information is named as *prior knowledge* [45]. In this paper, we call this prior knowledge as "Expert Instruction" (EI).

An usual type of IE in regards to traffic aspects is geolocation, *Global Positioning System* (GPS). GPS as an IE can be used in different ways, associated with different techniques. Some works consider the exact position of the traffic light and compare it with the position of the vehicle; in such case, previous knowledge of all traffic light positions can be used, as in the work of [46], or a traffic light mapping algorithm can be defined as in [35], [47] and [48]. To considerably decreases the amount of processing required, it is also possible to check if the vehicle is positioned close to an intersection and, just in this case, activate the traffic light recognition mechanism [16]. [7] also use this type of information to assist in the recognition phase as a false positive eliminator at the end of processing.

The relationship between the geometric objects that make up the traffic light can also be considered as expert knowledge. It is a common understanding that a basic traffic light is mostly formed by 3 circles positioned horizontally or vertically within a dark rectangle. [49] and [42] made use of these characteristics.

[50], finally, combined these two types of expert knowledge in a system that requires little processing power and can be run on a smartphone.

In this section we propose an approach to validate the previously defined assumption that it is possible to reduce the amount of data needed to train a machine learning algorithm using rules created by an expert in the field. In the context of this research, these rules will be referred to as Expert Instruction (EI).

We implemented a model of traffic light detection and recognition that uses Frequency Maps as expert instructions and compared it with the works of [15] and [21] as baselines.

3.1. Description of the experiments scenarios

Two different experiments scenarios were set.

In the first scenario, data from two *datasets* were used; the first provided by the work [21] and the second from [15].

To elaborate the EI, test data for each dataset were firstly divided into 2 groups, one for testing and the other to generate the EI. From this second group, 650 images were randomly selected. A total of 1300 images divided into 2 sets of EI were thus provided.

Classification algorithm of [15] was implemented and used to compare the influence of EI on training results. Each experiment was carried out with and without the inclusion of EI and has the following tasks:

- Model training (for different dataset lengths:1000, 4000, 7000, 10000 and 13000 items);
- Model testing;
- Usage of EI;
- Accuracy evaluation.

All experiments were performed twice with data selected at random. Accuracy was then measured. Data used in the experiment are images of traffic lights taken from the inside a vehicle.

In the second scenario, the EI was considered on the complete traffic light detection and recognition flow. The traffic light detection algorithm of [15] was implemented and included in the experiment. The regions selected by the detection algorithm were submitted to classification by the [15] algorithm trained with and without EI. The regions identified as traffic lights were marked in the original test image for further evaluation by a human expert. Precision and recall were then calculated. The 17 test sequences of the [15] dataset were used in this scenario. The tasks performed on the test are listed below:

- Model training with 7000 data;
- Use of the detection algorithm from [15] to find ROIs in traffic scenes obtained from the inside of a vehicle;
- Model testing with 17 traffic scene sequences;
- Use of the original EI;
- Precision and recall analysis.

All experiments were performed on a Dell Inspiron 15 7000 notebook, 7th generation i7 processor and 16 Gigabytes of RAM. The experiments were organized as follows.

3.2. Frequency maps as Expert Instructions

If we consider an image taken from the inside of a vehicle towards a traffic light, it is more likely to note the traffic light in the central and upper regions of the image, since they are usually suspended on poles along the road. This knowledge is an important clue, indeed. We, thus, produced traffic light frequency maps to test this hypothesis.

The method consisted on the analysis of a series of images by a human expert. In each image, the region where the traffic light is located is demarcated. A matrix of the same dimensionality of the analyzed images is then created. The points of the matrix related to the demarcated regions receive an increment. In the end, the entire matrix is divided by the total number of images analyzed, thus generating the frequency of traffic lights for each pixel in the image.

The datasets from [21] and [15] were used to generate two frequency maps as shown in Figures 1 and 2; redder hues represent higher frequencies. A random sample of 650 images was generated from the test data for each dataset, totaling 1300 images. The frequency data was smoothed using a masked averaging algorithm.

To properly compare the results of two different machine learning models, it is necessary to train both models with the same data. However, the datasets do not provide the coordinates from where the traffic light events were obtained from the original images, making it impossible to directly combine the previously calculated frequency with the training samples.

To solve this problem, it was assumed that all training samples were found in regions with a non-zero value on the frequency map, so the combination could be made with random frequency values (in respect to the beta distribution). Figures 3 and 4 show the frequency distribution histograms of the frequency maps. The values of the parameters of the beta distribution found for the map of Figure 1 were ($\alpha = 0.01655, \beta = 1$), and for the map of Figure 2 were ($\alpha = 0.03420, \beta = 12.8985$).

3.3. Model training

The method used to evaluate the insertion of the expert instruction was that proposed by [15]. The authors proposed a PCANet to find feature vectors given the set of traffic lights informed for training. The feature vectors are then used to train SVMs that will classify pixels as belonging to (or not) a traffic light.

The addition of the expert instruction in this process can take several forms. It is possible, for instance, to obtain the coordinates of the original image of the region recognized as a traffic light and analyze the frequency map at the same coordinates by checking whether the values are above a certain

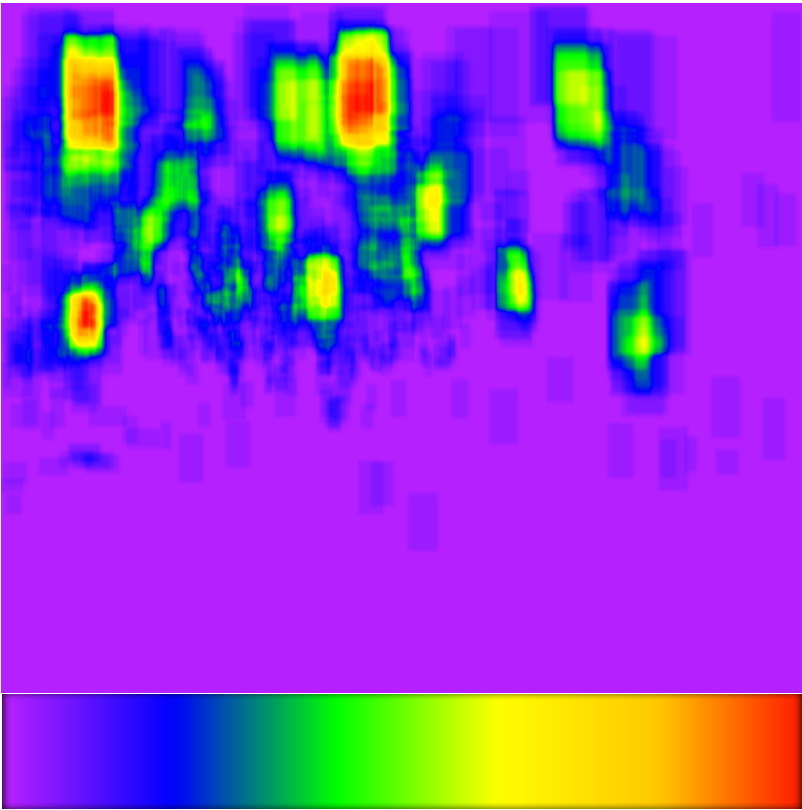


Figure 1. Frequency map obtained from the random dataset sample of [21]

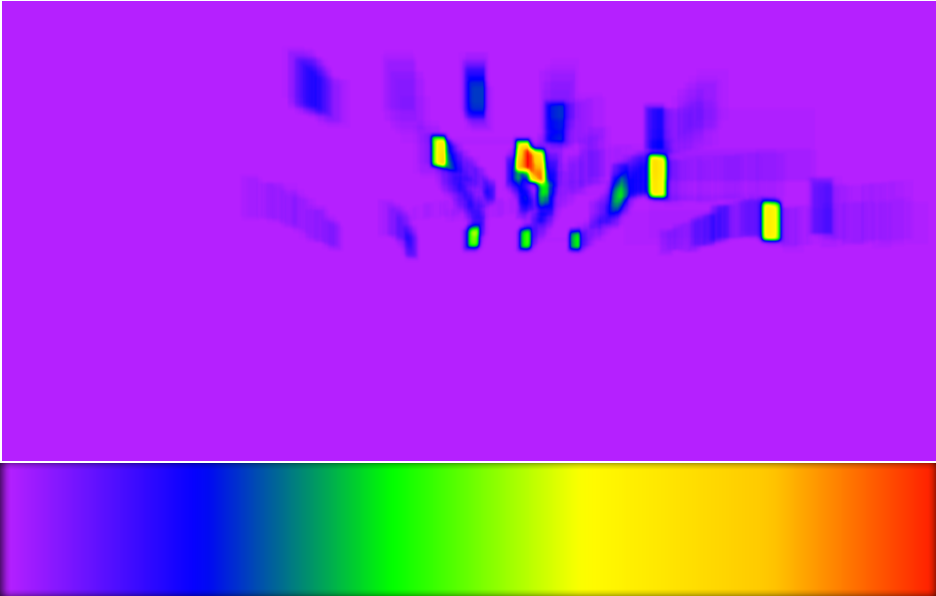


Figure 2. Frequency map obtained from the random dataset sample of [15]

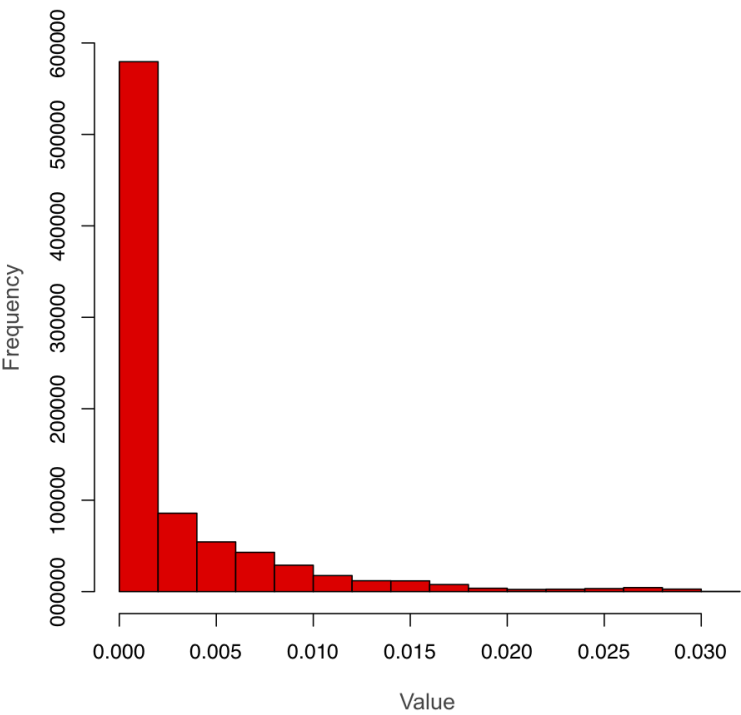


Figure 3. Histogram of the frequency distribution of the map in Figure 1

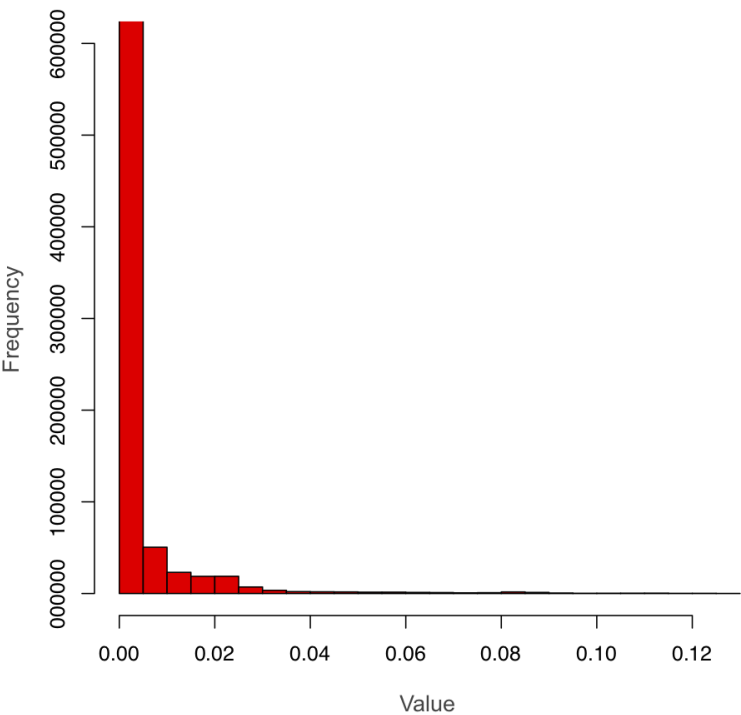


Figure 4. Histogram of the frequency distribution of the map in Figure 2

threshold. The problem with this alternative is that the algorithm becomes too deterministic, since only the static regions where the traffic lights appear on the frequency map would obtain positive results.

Another form of using the expert instruction is to perform a sum or multiplication operation in one of the internal phases of the recognition method. One of these phases could be between the PCANet layers, or at the PCANet exit before the data entry in the SVM. The problem of such combination is the size of the image that decreases as the convolutions inside the algorithm take place, making it difficult to combine directly with the value of the frequency map.

The most convenient form, thus, is to treat the expert instruction as the first layer of the recognition method. Soon after the detection phase, the Regions of Interest (ROI) are added or multiplied by the value of the frequency obtained for that region in the frequency map. Since the frequency map and the original scene have the same dimensionality, and knowing the ROI coordinates of the original scene, this combination will tend to highlight the ROI found for the next algorithm. The operation used in this work was multiplication. In Figure 5 we see the structure of a PCANet in the form of a process model. It is possible to notice the inclusion of the expert instruction soon after obtaining the input image for classification, in this case, the ROIs.

An important concern is to ensure that expert instruction does not make the method deterministic, as it would be if it was applied at the end of the method. It is possible that some part of the frequency map has a value of 0 (zero), which means that in the analyzed sample, no traffic lights appeared in that region. However, it is also possible for a traffic light to appear in an unusual region of the scene, and the DAS should be able to find it. It was necessary to define an increment *inc* that is added to every multiplication factor *fat*, which consists of the frequency value found in the frequency map. All the experiments depicted in this section assumed *inc* = 0.1.

The [21] training dataset is composed of 2 main classes: GREEN and RED. Each of these classes has data that can be divided into 4 classes according to the type of traffic light: round, with an up arrow, left arrow and right arrow. In this phase of the work, only the main division between GREEN and RED was used. The dataset has 9,977 units of green traffic lights and 10,975 units of red traffic lights, summing up 20,952 units of training.

The recognition method proposed by [15] was trained with different amounts of data obtained from the training dataset from [21]. Firstly, no expert instruction was considered. Next, we embedded the frequency maps. After each training episode, an accuracy test was performed. Final accuracy values can be seen in Figure 6. It is possible to notice the great positive effect of expert instruction on the accuracy for a smaller amount of data (orange line vs. blue line in the figure). For greater data, behavior becomes similar until over-fitting occurs.

Similar experiment was carried out with the dataset from [15], using the same amounts of training data. As in the previous experiment, data is distributed along 4 classes according to the type of traffic light: round, with an up arrow, left arrow, and right arrow. However, in this case, data are divided into more specific groups depending on how the traffic light appears: GreenRoi1, GreenRoi3 and GreenRoi4 for variations of green traffic lights; and RedRoi1 and RedRoi3 for red traffic light variations. Final accuracy values can be seen in Figure 7. As before, it is possible to notice the great positive effect of expert instruction on the accuracy for smaller amount of data. Also, results seem to indicate that *over-fitting* for EI version happens earlier.

It is worth noting that the behavior of applying EI to different data may vary, even within the same problem domain. Figures 8, 9, 10, 11 and 12 further detail the accuracy for each traffic light set. Although the EI has improved the rates for all sets, the groups concerning the red traffic lights got minor improvements if compared to the green ones. This suggests a lower sensitivity of the red traffic lights to the addition of EI. This hypothesis is further supported by the sharper fall in the accuracy rate for the green groups of Figures 8 and 9 (over-fitting), which not happens with that intensity in the red groups.

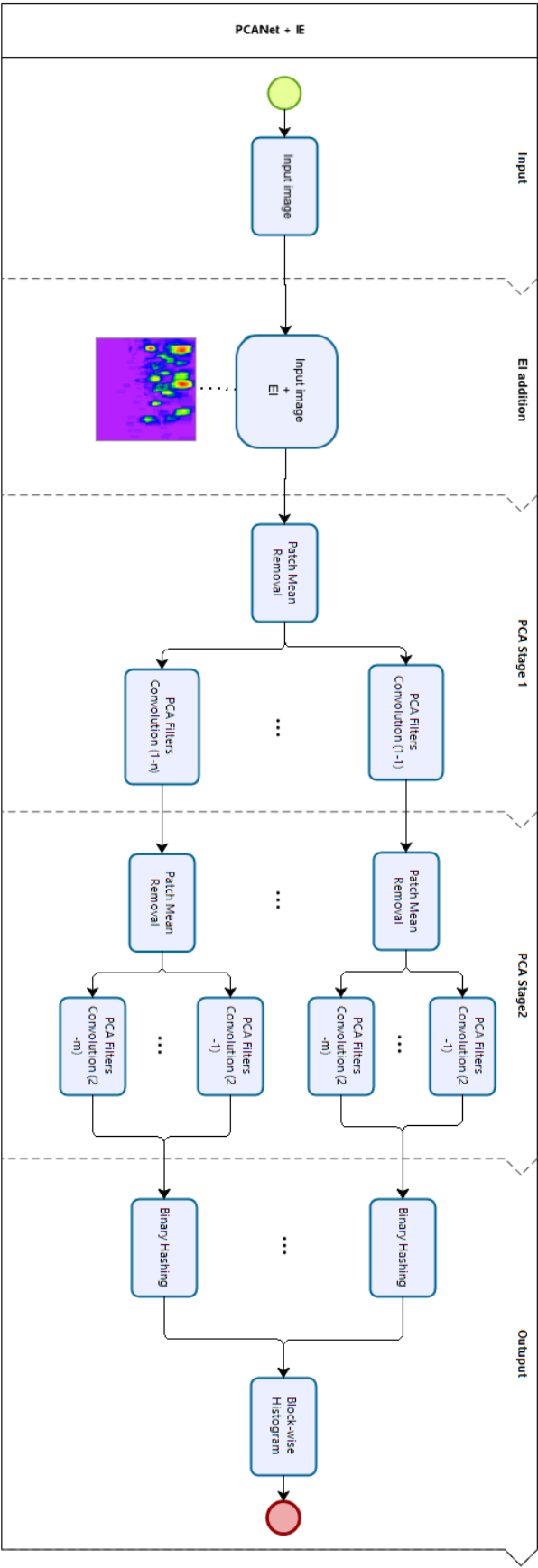


Figure 5. Structure of a two-phase PCANet (atages) including EI.

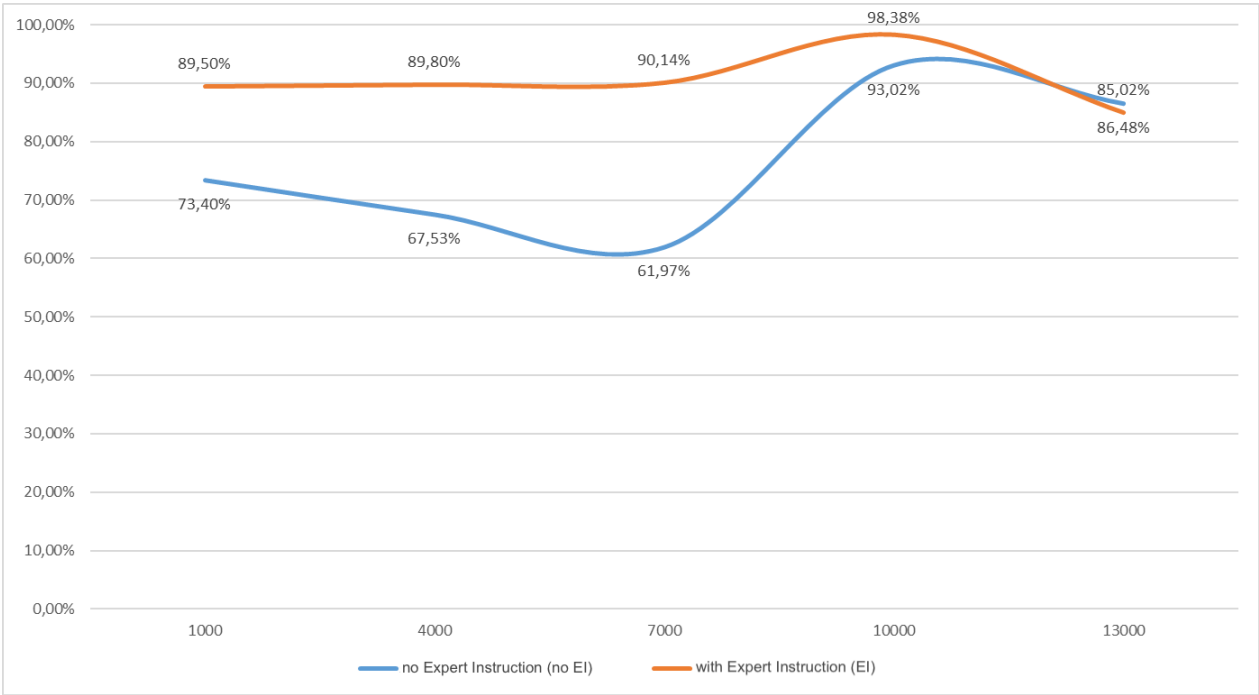


Figure 6. Test accuracy levels with Expert Instruction (orange line) and without Expert instruction (blue line) for the dataset [21]

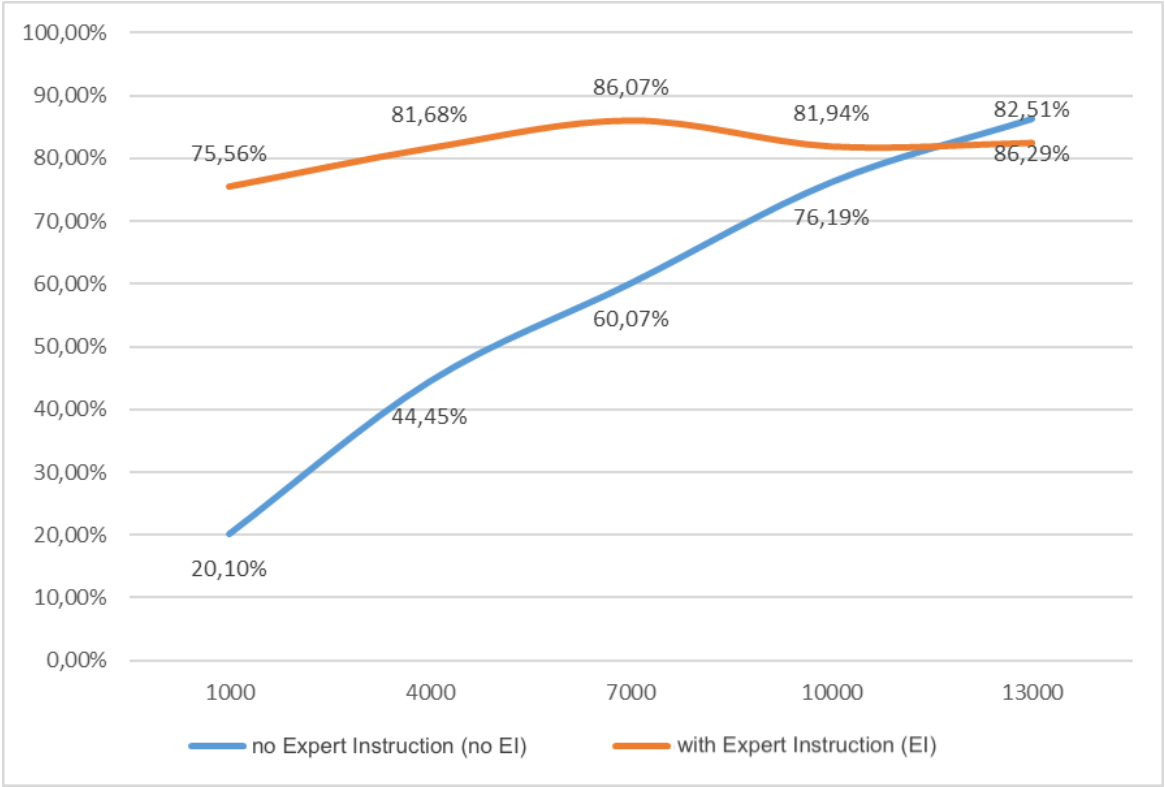


Figure 7. Test accuracy levels with Expert Instruction (orange line) and without Expert instruction (blue line) for the dataset [15].

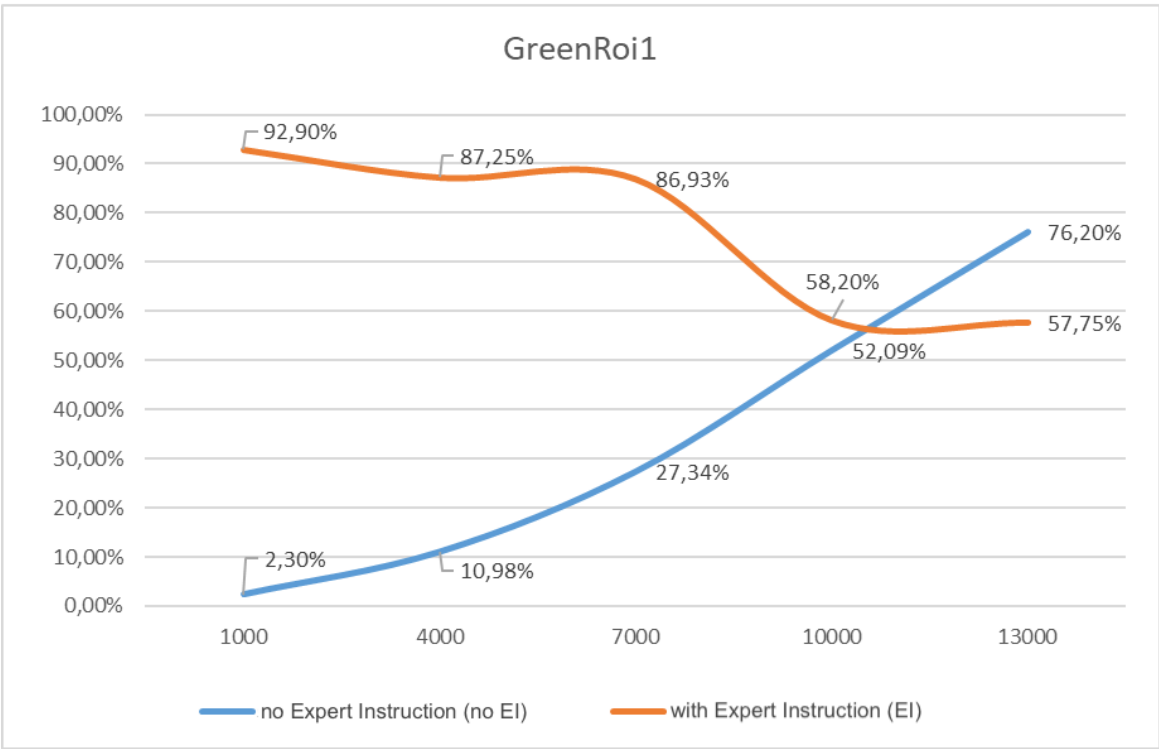


Figure 8. Test accuracies for the GREENROI1 data group of [15] after training with EI (orange line) and without EI (blue line).

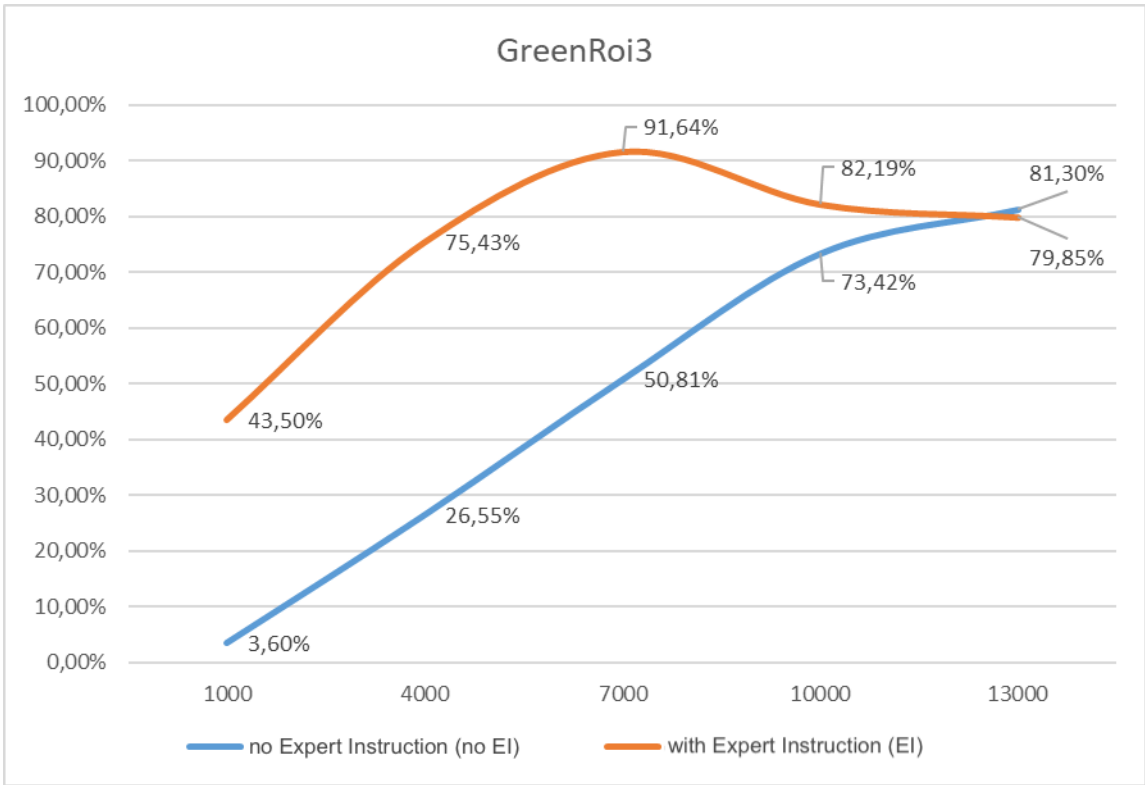


Figure 9. Test accuracies for the GREENROI3 data group of [15] after training with EI (orange line) and without EI (blue line).

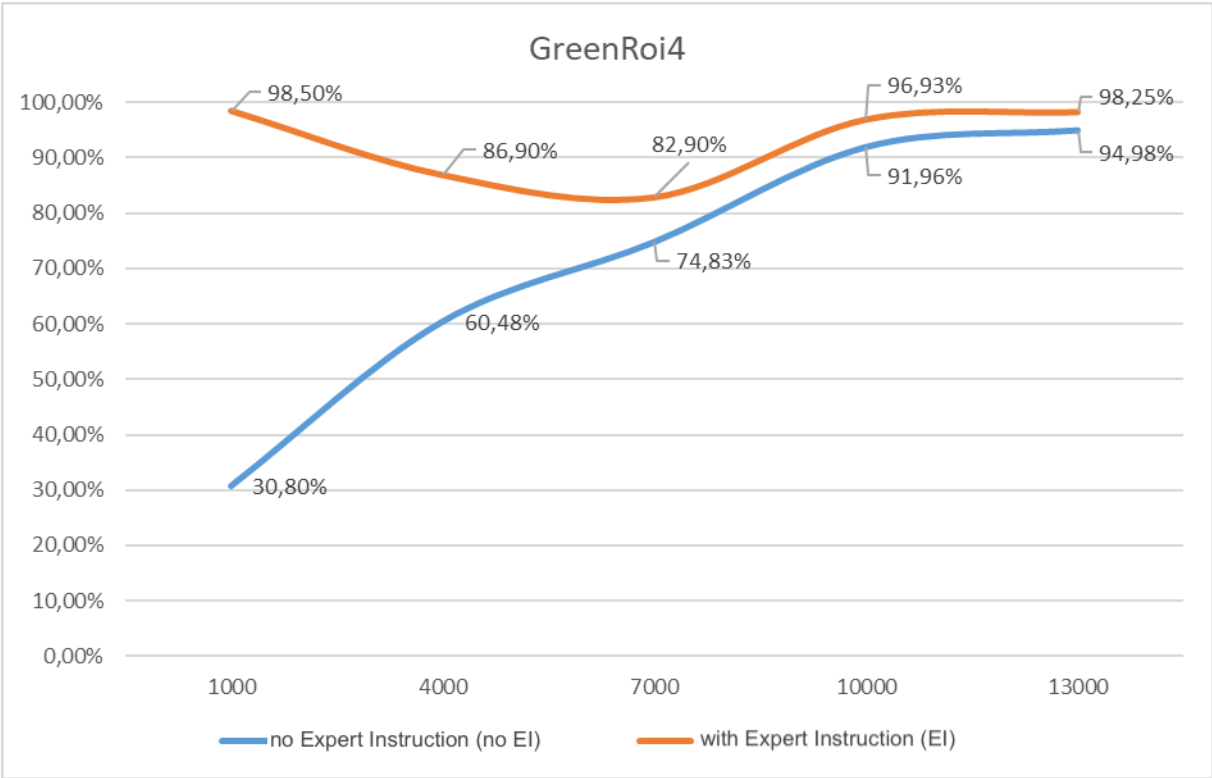


Figure 10. Test accuracies for the GREENROI4 data group of [15] after training with EI (orange line) and without EI (blue line).

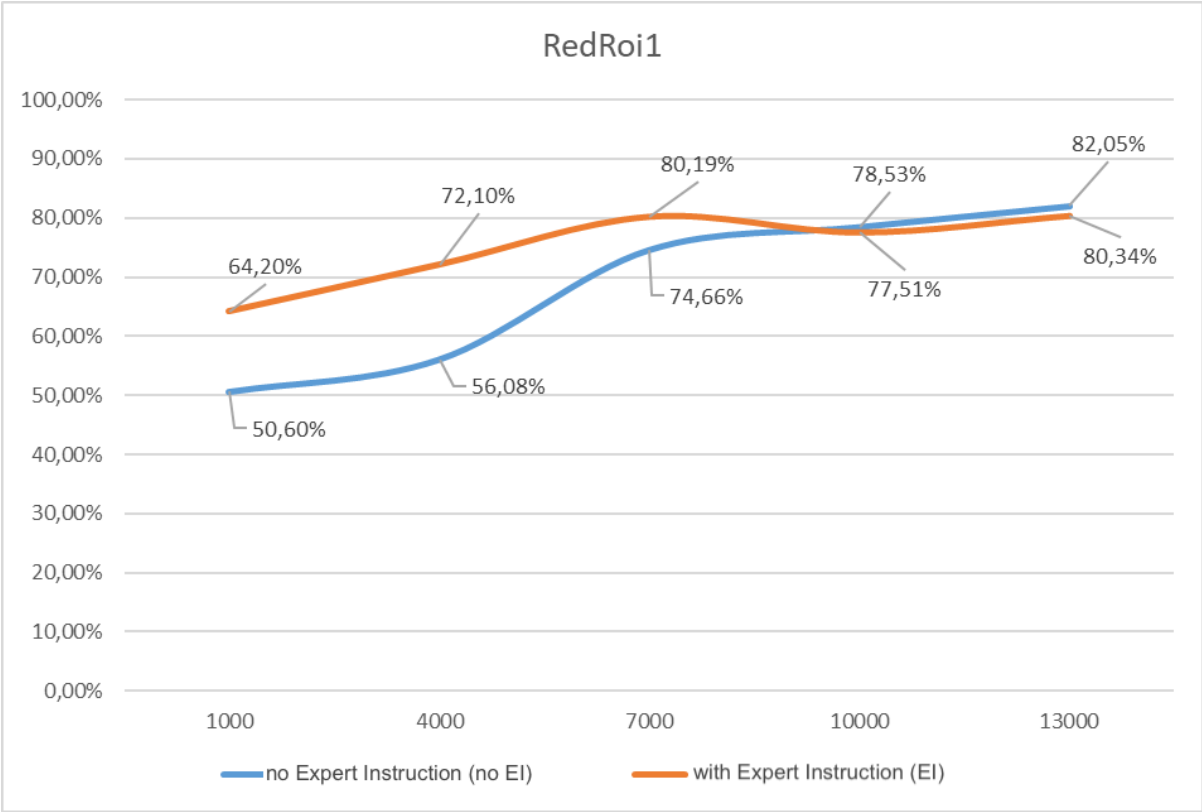


Figure 11. Test accuracies for the REDROI1 data group of [15] after training with EI (orange line) and without EI (blue line).

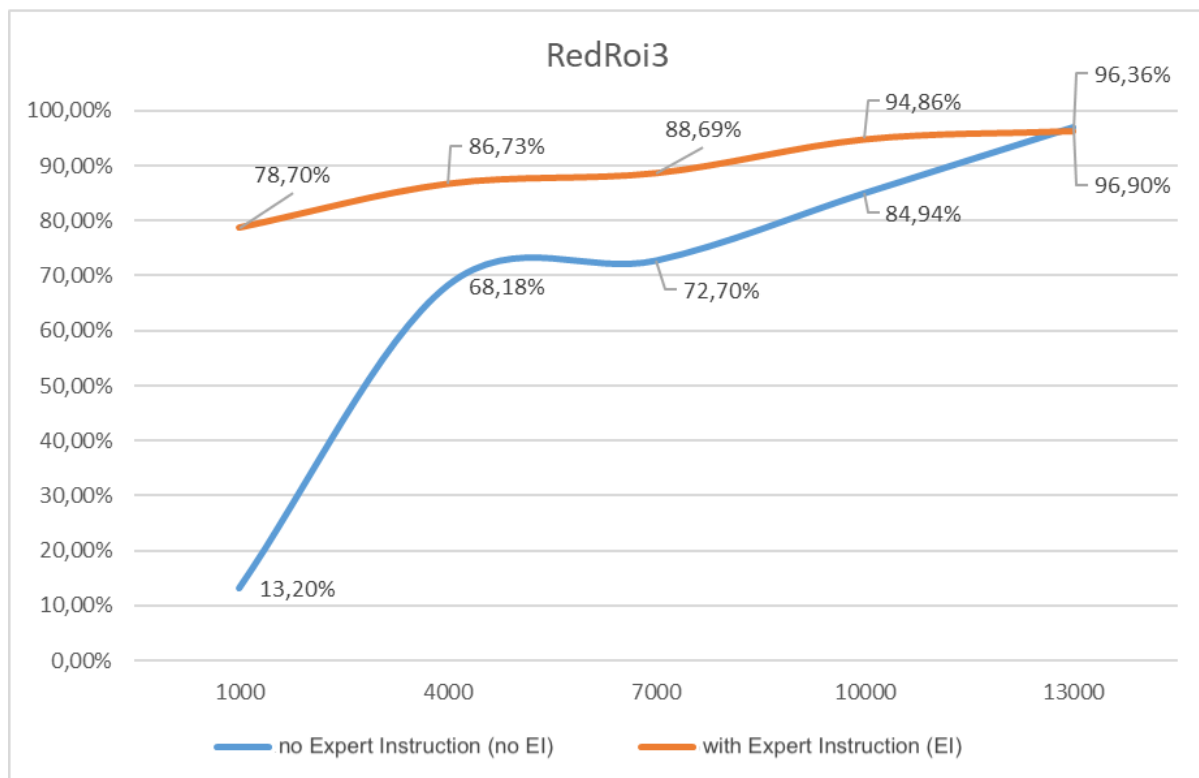


Figure 12. Test accuracies for the REDROI3 data group of [15] after training with EI (orange line) and without EI (blue line).

3.4. Detection and recognition

To validate the inclusion of EI in the complete flow of a DAS, an experiment was carried out performing the ROIs detection phase followed by the submission of the ROIs found to the trained model. The detection algorithm of [15] was used in conjunction with the PCANet for classification. Unlike the previous experiment, the frequency value obtained from the original frequency map is gets an increment of $inc = 0.1$, since the ROIs are now identified by the detection algorithm and keep the region's origin coordinates. In the previous experiment, a frequency estimate was generated based on the beta distribution of the original frequencies.

Two tests was performed with the images test dataset provided by [15]: with and without EI. This test dataset consists of 17 sequences of 2142 images (in which there is a traffic light), the shortest with 56 frames, and the largest with 518 frames.

A sample of length 7,000 for the training set of both tests was chosen. This value is the average of the minimum and maximum amounts of data used in the previous experiment.

Accuracy measure is manually done. Indeed, automating the counting of hits and errors in such kind of problem is quite complicated because it is not possible to associate an image to a specific class, since it is not a matter of classifying images, but embedded objects. In addition, the number of objects that can be found in the image may also vary in each image of the dataset, and could even be 0. A metadata file for each image containing a list of expected objects with their classes could be created, but, without a manual review, an unexpected object could be erroneously classified in one of the classes.

Table 1 shows the precision and recall rates obtained by the test with and without EI. The use of EI presents significantly better results, achieving accuracy of 83% and recall of 73%, in contrast to 75.3% and 51.1% for the version without EI.

Observing the results for each sequence, individually, in the Table 1, we notice better precision rates for the use of EI in 12 of the 17 sequences, worse rates for 3 sequences (10, 11 and 14), and similar

rates for the rest. In regards to recall, the use of EI performed better in 11 of 12 sequences; sequence number 12 shows similar behavior. It is worth noting that although the use of EI performed consistently better, improving recall is clearly critical in the domain of traffic light detection and recognition.

Table 1. Precision and recall rates using the full images.

Sequence	Prec. Expert	Recall Expert	Prec. PCA	Recall PCA
1	66%	52%	56,7%	45,8%
2	100%	92%	92,0%	41,0%
3	95%	95%	78,3%	42,0%
4	95%	64%	95,0%	77,8%
5	100%	100%	89%	77,1%
6	60,4%	45%	60,1%	14,0%
7	100%	92%	89,7%	80,0%
8	98%	98%	79,7%	81,3%
9	84%	43%	76,3%	42,0%
10	79%	90%	84,3%	67,5%
11	51%	37%	62,5%	18,1%
12	89%	60%	81,8%	61,3%
13	91%	97%	63,6%	35,0%
14	92%	85%	95,0%	76,0%
15	96%	72%	84,0%	41,1%
16	87%	78%	73,5%	38,0%
17	53%	48%	48,0%	31,2%
Average	83%	73%	75,3%	51,1%

4. TLR prototype deployment

We developed a car device to interact with the driver and continuously capture images of the environment from the inside of the vehicle. In this section we depict the device and its usage in the creation of a local dataset and the ensuing employment of the developed algorithm.

One of the main concerns when prototyping a TLR device is where it will be positioned in the vehicle, once it has to provide a clear view of the exterior scene and at the same time do not compromise the vision of the driver. Another critical point is that the device shall be protected from adverse meteorological conditions like rain, or be waterproof. Heat also might cause problems, so the sunlight incidence may be considered as well. Also, trepidation of vehicle motion might have critical influence in the device vision and, so, stabilization is a huge requirement. Finally, the device should be able to provide warning sound to advise the distracted driver. Smartphones accomplish these requirements and, so, are an affordable alternative to TLR device.

A smartphone was positioned inside a vehicle to capture actual traffic scenes with and without the presence of traffic lights. To meet the requirements depicted previously, we designed a stable device support using a two-sided tape and part of a windshield suction cup support. We removed the support portion that holds the device from the cable with suction cup that is attached to the windshield. Next we fixed the first part centralized at the vehicle panel with the two-sided tape. This design allow the device to capture the traffic scene without a bias to the left or to the right. The proposed layout forces the device to use the camera in landscape mode, reducing the portion of sky captured and maximizing the traffic scene size obtained (Figure 13).

Three different smartphones were used to capture traffic videos containing traffic lights: a Motorola G second generation, an iPhone 6, and a Galaxy S8+. All devices were configured to capture video with HD resolution. Figure 14 shows an example of images obtained with these devices. The images were extracted from videos at 5 frames per second (fps) rate.

4.1. Results and discussion

The images were subjected to classification using the method of [21].



Figure 13. TLR device support holding an iPhone 6.

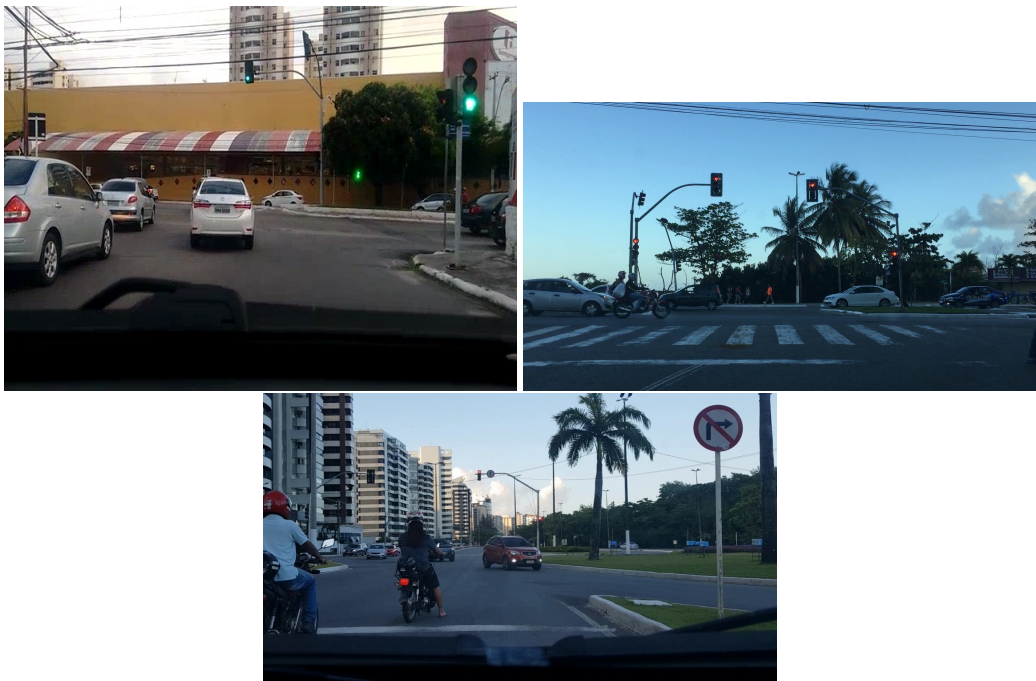


Figure 14. Images obtained using the TLR device support prototype with different smartphones. From top to bottom (left to right): (a) Motorola G 2nd Generation, (b) iPhone 6, and (c) Galaxy S8+

First images set, obtained with a Motorola G2, did not present good results and was discarded. Second set, with images obtained by the iPhone 6, contains 682 images, being 209 without traffic lights and 473 with traffic lights. The third images set, from Galaxy S8 +, consists of 247 images, being 165 with traffic lights and 82 without traffic lights. The final dataset contains 929 traffic images, being 638 images with green or red traffic lights and 291 images without traffic lights, the negative group.

It was also found that just categorizing the dataset images is not enough to properly evaluate the results. Suppose the label of image 01 says it is a red traffic light and the TLR also say that there is a red traffic light in such image. Suppose further that the red traffic light found by the TLR is actually a stop sign. If the comparison were made only between the labels found, it would result in a false positive. Hence, when working with the recognition of isolated objects in the image, it is crucial that the TLR highlights the exact region identified as a traffic light within the image, and when computing the results, the human expert must check each image one by one.

Considering that most of the time there are two equal traffic lights for the same route, only one positive result was counted for the type of traffic light found when one or both traffic lights are recognized in the image. So for each image we have 01 error or 01 hit. This reflects the daily behavior of observing a single traffic light to make a choice.

Figure 15 shows the detailed results per images set. Both sets achieved high precision rates, but the "iPhone 6"'s images set had poor recall of only 60%. This can be explained by the fact that the samples of traffic lights used in training are very different from some of the examples found in the test set, as shown in Figure 16. Indeed, when the training set do not fully represent the real world, some traffic lights are not recognized.

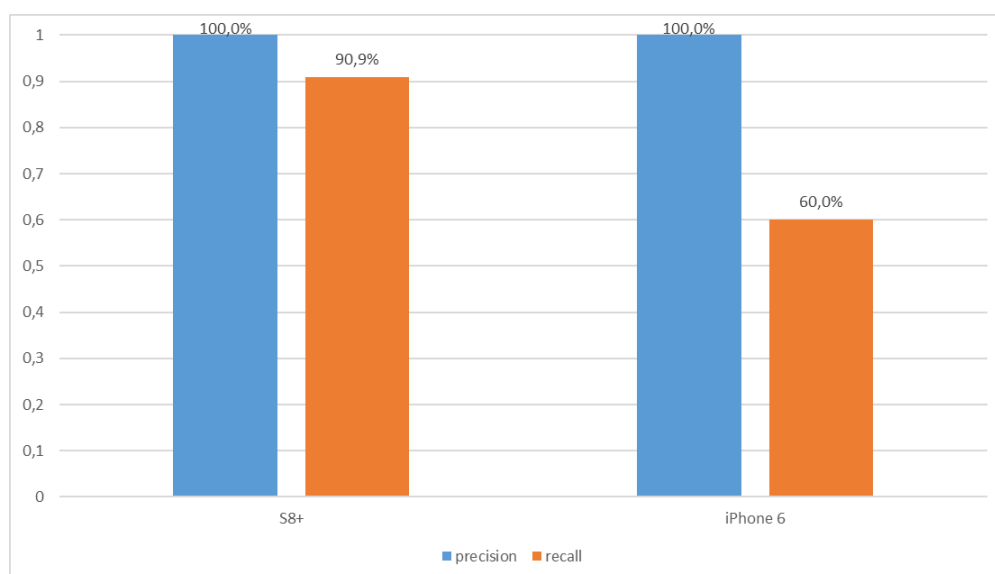


Figure 15. Precision and recall rates per images set.

Lighting conditions in the training dataset also vary from those under which the test dataset was created; indeed, the geography, weather, and the device used to take the images have a major influence on lighting.

It is worth noting that the distance between the TLR and the traffic light is crucial for recognition. The greater the distance from which the TLR is able to correctly classify the traffic light, the greater its robustness. A threshold has been empirically defined in respect to the size of the traffic light in the image: only traffic lights with at least 20 pixels on their diagonal were taken into account.

Low recall for the iPhone's set has adversely affected the final results of the tests with the TLR, as observed in Figure 17. However, compared to original results of [21], results validate the TLR and justify its use in future research.



Figure 16. From left to right: examples of green and red traffic lights used in training, examples of green and red traffic lights obtained by the TLR device and used in tests.

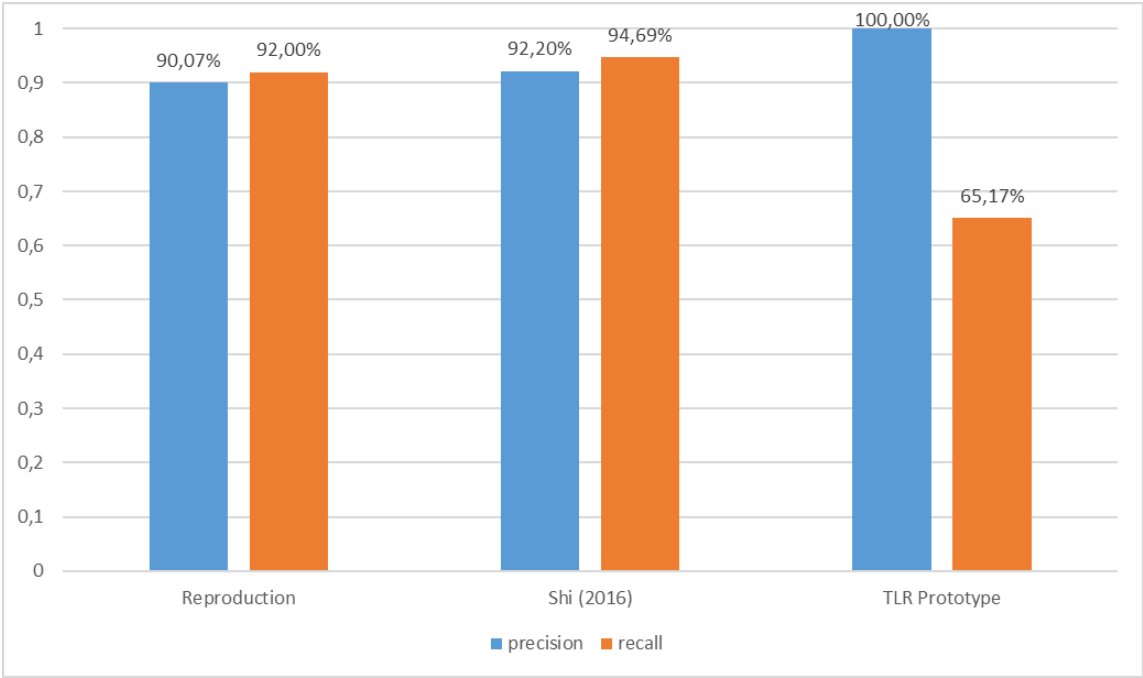


Figure 17. Precision and recall: our reproduction of the work of [21], the original work of [21], and the tests using the images obtained by the TLR, respectively.

5. Conclusions

The main goal of this work was to evaluate whether the inclusion of Expert Instruction (IE) could decrease the amount of data needed to train a machine learning algorithm in the Traffic Light Recognition (TLR) problem. For this, a PCANet with and without EI was trained. The EI consisted of the frequency of occurrence of semaphores per pixel of a sequence of images. This was chosen based on the driver's implicit knowledge of what is the usual location of a traffic light in his/her field of view from the inside of the vehicle: central and upper regions of the scene has higher likelihood. Frequency maps were built to confirm such hypothesis.

Results have shown that the use of EI helps to decrease the amount of data for model training, for both datasets of [21] and [15]. Test results with EI model training performed better than the conservative approach. Even for little data (around 1,000 elements) EI-based approach achieved 75% of accuracy.

To evaluate the use of this training in a complete TLR process, an experiment was carried out in which an ROI detection algorithm was applied and then the ROIs were subjected to classification by PCANet using EI and without EI. EI-based TLR achieved 83% of precision and 73% of recall against 75.3% of accuracy and 51.1% of recall of its counterpart.

A TLR device was also deployed. The goal was to show the actual feasibility of such a device for driver assistance. The images obtained with the TLR device were submitted to the TLR algorithm. Precision of 100% and recall of 65.17% validate the use of this TLR device *layout*.

Further research include handling of parameters and values. One of these parameters is the number of images used to generate the frequency map, which can be expanded to be even more representative. Other important parameters are: the mask used for smoothing the frequency map and the increment *inc* added to every frequency value *fat* for the combination with the algorithm's input image.

Author Contributions: Conceptualization, Hendrik Macedo, Leonardo Matos and Bruno Prado; methodology, Hendrik Macedo and Leonardo Matos; software, Thiago Almeida; validation, Thiago Almeida; investigation, Thiago Almeida, Hendrik Macedo and Leonardo Matos; resources, Thiago Almeida, Hendrik Macedo and Leonardo Matos; data curation, Thiago Almeida; writing—original draft preparation, Hendrik Macedo; writing—review and editing, Hendrik Macedo, Leonardo Matos and Bruno Prado; supervision, Hendrik Macedo and Leonardo Matos; project administration, Hendrik Macedo. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank CAPES and FAPITEC-SE for the financial support [Edital CAPES/FAPITEC/SE No 11/2016 - PROEF, Processo 88887.160994/2017-00]. The authors also thank FAPITEC-SE for granting a graduate scholarship to Thiago Almeida and CNPq for granting a productivity scholarship to Hendrik Macedo [DT-II, Processo 310446/2014-7].

References

1. Gonçalves, Jéssica.. Desrespeito à sinalização causa 7,5 mil acidentes no trânsito em 2014. <http://www.ebc.com.br/noticias/2015/07/desrespeito-sinalizacao-causa-75-mil-acidentes-no-transito-em-2014>, 2015. Acesso em: 09 ago. 2017.
2. CET - Companhia de Engenharia de Tráfego. Análise da mortalidade ocorrida por acidentes de trânsito em agosto e setembro de 2016. <http://www.cetsp.com.br/media/523410/parcial2016.pdf>, 2016. Acesso em: 09 ago. 2017.
3. Jensen, M.B.; Philipsen, M.P.; Møgelmoose, A.; Moeslund, T.B.; Trivedi, M.M. Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives. *IEEE Transactions on Intelligent Transportation Systems* **2016**, *17*, 1800–1815. doi:10.1109/TITS.2015.2509509.
4. Weber, M.; Wolf, P.; Zöllner, J.M. DeepTLR: A single deep convolutional network for detection and classification of traffic lights. 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 342–348. doi:10.1109/IVS.2016.7535408.

5. John, V.; Zheming, L.; Mita, S. Robust traffic light and arrow detection using optimal camera parameters and GPS-based priors. 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), 2016, pp. 204–208. doi:10.1109/ACIRS.2016.7556213.
6. John, V.; Yoneda, K.; Liu, Z.; Mita, S. Saliency Map Generation by the Convolutional Neural Network for Real-Time Traffic Light Detection Using Template Matching. *IEEE Transactions on Computational Imaging* **2015**, *1*, 159–173. doi:10.1109/TCL.2015.2480006.
7. John, V.; Yoneda, K.; Qi, B.; Liu, Z.; Mita, S. Traffic light recognition in varying illumination using deep learning and saliency map. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 2286–2291. doi:10.1109/ITSC.2014.6958056.
8. Wang, Z.; Bi, Z.; Wang, C.; Lin, L.; Wang, H. Traffic lights recognition based on PCANet. 2015 Chinese Automation Congress (CAC), 2015, pp. 559–564. doi:10.1109/CAC.2015.7382563.
9. Philipsen, M.P.; Jensen, M.B.; Møgelmoose, A.; Moeslund, T.B.; Trivedi, M.M. Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset. 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 2341–2345. doi:10.1109/ITSC.2015.378.
10. Kim, S.; Jang, E.; Hyun, S.; Han, D.S. Real time object detection based on saliency map. 2016 IEEE International Conference on Consumer Electronics (ICCE), 2016, pp. 534–535. doi:10.1109/ICCE.2016.7430719.
11. Almeida, T.; Vasconcelos, N.; Benicasa, A.; Macedo, H. Fuzzy model applied to the recognition of traffic lights signals. 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), 2016, pp. 1–4. doi:10.1109/EATIS.2016.7520130.
12. Almeida, T.; Vasconcelos, N.; Benicasa, A. Framework para Detecção de Semáforos Baseado em Atenção Visual. Conference on Graphics, Patterns and Images, 28.(SIBGRAPI), Salvador, 2015.
13. Ji, Y.; Yang, M.; Lu, Z.; Wang, C. Integrating visual selective attention model with HOG features for traffic light detection and recognition. 2015 IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 280–285. doi:10.1109/IVS.2015.7225699.
14. Zhou, Y.; Chen, Z.; Huang, X. A system-on-chip FPGA design for real-time traffic signal recognition system. 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp. 1778–1781. doi:10.1109/ISCAS.2016.7538913.
15. Chen, Z.; Huang, X. Accurate and Reliable Detection of Traffic Lights Using Multiclass Learning and Multiobject Tracking. *IEEE Intelligent Transportation Systems Magazine* **2016**, *8*, 28–42. doi:10.1109/MITS.2016.2605381.
16. Zhang, Y.; Xue, J.; Zhang, G.; Zhang, Y.; Zheng, N. A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles. Proceedings of the 33rd Chinese Control Conference, 2014, pp. 4924–4929. doi:10.1109/ChiCC.2014.6895775.
17. Widiantoro, D.H.; Saputra, K.I. Traffic lights detection and recognition based on color segmentation and circle hough transform. 2015 International Conference on Data and Software Engineering (ICoDSE), 2015, pp. 237–240. doi:10.1109/ICoDSE.2015.7437004.
18. Chiu, Y.T.; Chen, D.Y.; Hsieh, J.W. Real-time traffic light detection on resource-limited mobile platform. 2014 IEEE International Conference on Consumer Electronics - Taiwan, 2014, pp. 211–212. doi:10.1109/ICCE-TW.2014.6904063.
19. Omachi, M.; Omachi, S. Detection of traffic light using structural information. IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS, 2010, pp. 809–812. doi:10.1109/ICOSP.2010.5655932.
20. Sooksatra, S.; Kondo, T. Red traffic light detection using fast radial symmetry transform. 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014, pp. 1–6. doi:10.1109/ECTICon.2014.6839767.
21. Shi, Z.; Zou, Z.; Zhang, C. Real-Time Traffic Light Detection With Adaptive Background Suppression Filter. *IEEE Transactions on Intelligent Transportation Systems* **2016**, *17*, 690–700. doi:10.1109/TITS.2015.2481459.
22. Trehard, G.; Pollard, E.; Bradai, B.; Nashashibi, F. Tracking both pose and status of a traffic light via an Interacting Multiple Model filter. 17th International Conference on Information Fusion (FUSION), 2014, pp. 1–7.

23. Oniga, F.; Prodan, S.; Nedevschi, S. Traffic light detection on mobile devices. 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2015, pp. 287–292. doi:10.1109/ICCP.2015.7312645.
24. Jie, Y.; Xiaomin, C.; Pengfei, G.; Zhonglong, X. A new traffic light detection and recognition algorithm for electronic travel aid. 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), 2013, pp. 644–648. doi:10.1109/ICICIP.2013.6568153.
25. Borrmann, J.M.; Haxel, F.; Nienhüser, D.; Viehl, A.; Zöllner, J.M.; Bringmann, O.; Rosenstiel, W. STELLaR - A case-study on SysTEmaticaLLy embedding a Traffic Light Recognition. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 1258–1265. doi:10.1109/ITSC.2014.6957860.
26. Zong, W.; Chen, Q. Traffic Light Detection Based on Multi-feature Segmentation and Online Selecting Scheme. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2014, pp. 204–209. doi:10.1109/SMC.2014.6973908.
27. Sathiya, S.; Balasubramanian, M.; Priya, D.V. Real time recognition of traffic light and their signal count-down timings. International Conference on Information Communication and Embedded Systems (ICICES2014), 2014, pp. 1–6. doi:10.1109/ICICES.2014.7033965.
28. 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. 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 431–435. doi:10.1109/IVS.2010.5548083.
29. Shi, X.; Zhao, N.; Xia, Y. Detection and Classification of Traffic Lights for Automated Setup of Road Surveillance Systems. *Multimedia Tools Appl.* **2016**, *75*, 12547–12562. doi:10.1007/s11042-014-2343-1.
30. Michael, M.; Schlipfing, M. Extending traffic light recognition: Efficient classification of phase and pictogram. 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–8. doi:10.1109/IJCNN.2015.7280499.
31. Salarian, M.; Manavella, A.; Ansari, R. A vision based system for traffic lights recognition. 2015 SAI Intelligent Systems Conference (IntelliSys), 2015, pp. 747–753. doi:10.1109/IntelliSys.2015.7361224.
32. Jang, C.; Kim, C.; Kim, D.; Lee, M.; Sunwoo, M. Multiple exposure images based traffic light recognition. 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014, pp. 1313–1318. doi:10.1109/IVS.2014.6856541.
33. Chen, Q.; Shi, Z.; Zou, Z. Robust and real-time traffic light recognition based on hierarchical vision architecture. 2014 7th International Congress on Image and Signal Processing, 2014, pp. 114–119. doi:10.1109/CISP.2014.7003760.
34. Chung, Y.C.; Wang, J.M.; Chen, S.W. A vision-based traffic light detection system at intersections. *Journal of National Taiwan Normal University: Mathematics, Science & Technology* **2002**, *47*, 67–86.
35. Levinson, J.; Askeland, J.; Dolson, J.; Thrun, S. Traffic light mapping, localization, and state detection for autonomous vehicles. 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 5784–5791. doi:10.1109/ICRA.2011.5979714.
36. Almagambetov, A.; Velipasalar, S.; Baitassova, A. Mobile Standards-Based Traffic Light Detection in Assistive Devices for Individuals with Color-Vision Deficiency. *IEEE Transactions on Intelligent Transportation Systems* **2015**, *16*, 1305–1320. doi:10.1109/TITS.2014.2361139.
37. Balcerek, J.; Konieczka, A.; Marciniak, T.; Dąbrowski, A.; Maćkowiak, K.; Piniarski, K. Automatic detection of traffic lights changes from red to green and car turn signals in order to improve urban traffic. 2014 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2014, pp. 110–115.
38. Cai, Z.; Li, Y.; Gu, M. Real-time recognition system of traffic light in urban environment. 2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications, 2012, pp. 1–6. doi:10.1109/CISDA.2012.6291516.
39. Omachi, M.; Omachi, S. Traffic light detection with color and edge information. 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009, pp. 284–287. doi:10.1109/ICCSIT.2009.5234518.
40. Choi, J.; Ahn, B.T.; Kweon, I.S. Crosswalk and traffic light detection via integral framework. The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision, 2013, pp. 309–312. doi:10.1109/FCV.2013.6485511.
41. Fan, B.; Lin, W.; Yang, X. An efficient framework for recognizing traffic lights in night traffic images. 2012 5th International Congress on Image and Signal Processing, 2012, pp. 832–835. doi:10.1109/CISP.2012.6469638.

42. de Charette, R.; Nashashibi, F. Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates. 2009 IEEE Intelligent Vehicles Symposium, 2009, pp. 358–363. doi:10.1109/IVS.2009.5164304.
43. Mascetti, S.; Ahmetovic, D.; Gerino, A.; Bernareggi, C.; Busso, M.; Rizzi, A. Robust Traffic Lights Detection on Mobile Devices for Pedestrians with Visual Impairment. *Comput. Vis. Image Underst.* **2016**, *148*, 123–135. doi:10.1016/j.cviu.2015.11.017.
44. Gomez, A.E.; Alencar, F.A.R.; Prado, P.V.; Osório, F.S.; Wolf, D.F. Traffic lights detection and state estimation using Hidden Markov Models. 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014, pp. 750–755. doi:10.1109/IVS.2014.6856486.
45. Haykin, S.S.; Haykin, S.S.; Haykin, S.S.; Haykin, S.S. *Neural networks and learning machines*; Vol. 3, Pearson Upper Saddle River, 2009.
46. Yoneda, K.; Suganuma, N.; Aldibaja, M.A. Simultaneous state recognition for multiple traffic signals on urban road. 2016 11th France-Japan 9th Europe-Asia Congress on Mechatronics (MECATRONICS) / 17th International Conference on Research and Education in Mechatronics (REM), 2016, pp. 135–140. doi:10.1109/MECATRONICS.2016.7547129.
47. Li-tian, Z.; Meng-yin, F.; Yi, Y.; Mei-ling, W. A framework of traffic lights detection, tracking and recognition based on motion models. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 2298–2303. doi:10.1109/ITSC.2014.6958058.
48. Fairfield, N.; Urmson, C. Traffic Light Mapping and Detection. Proceedings of ICRA 2011, 2011.
49. Ying, J.; Tian, J.; Lei, L. Traffic light detection based on similar shapes searching for visually impaired person. 2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP), 2015, pp. 376–380. doi:10.1109/ICICIP.2015.7388200.
50. Sung, T.P.; Tsai, H.M. Real-time traffic light recognition on mobile devices with geometry-based filtering. 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC), 2013, pp. 1–7. doi:10.1109/ICDSC.2013.6778222.