information

MDPI

Article

Prototyping a Traffic Light Recognition Device with Expert Knowledge

Thiago Almeida *, Hendrik Macedo, Leonardo Matos and Nathanael Vasconcelos

Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Sergipe,
São Cristóvão 49100000, Brazil; hendrik@dcomp.ufs.br (H.M.); leonardo@dcomp.ufs.br (L.M.);
novasconcelos@dcomp.ufs.br (N.V.)

* Correspondence: thiagosa@dcomp.ufs.br

Received: 27 September 2018; Accepted: 9 November 2018; Published: 13 November 2018

Abstract: Traffic light detection and recognition (TLR) research has grown every year. In addition, Machine Learning (ML) has been largely used not only in traffic light research but in every field where it is useful and possible to generalize data and automatize human behavior. ML algorithms require a large amount of data to work properly and, thus, a lot of computational power is required to analyze the data. We argue that expert knowledge should be used to decrease the burden of collecting a huge amount of data for ML tasks. In this paper, we show how such kind of knowledge was used to reduce the amount of data and improve the accuracy rate for traffic light detection and recognition. Results show an improvement in the accuracy rate around 15%. The paper also proposes a TLR device prototype using both camera and processing unit of a smartphone which can be used as a driver assistance. To validate such layout prototype, a dataset was built and used to test an ML model based on adaptive background suppression filter (AdaBSF) and Support Vector Machines (SVMs). Results show 100% precision rate and recall of 65%.

Keywords: traffic light detection and recognition, computer vision, expert systems, machine learning, support vector machines, adaptive background suppression filter, pcanet

1. Introduction

According to the Brazilian Federal Highway Police [1], more than 150 thousand traffic accidents were reported in 2014. Traffic lights are clear and intuitive guidance devices for traffic control. However, they are systematically disrespected by Brazilian drivers. It was reported that 7500 of these accidents were related to such disrespect, causing 425 fatal victims.

Some hypotheses are raised as possible motivations for such infractions: (1) poorly located traffic lights; (2) faulty/off traffic lights or in very dim light; (3) ambient light that disturbs the vision of the driver; (4) visual impairment of the driver; (5) yellow traffic light advance; and (6) number of traffic regulator items to be observed.

The first two items can be easily solved with the effort of the traffic regulator in arranging and maintaining traffic lights optimally on the streets. A Traffic Light Recognition Device (TLR) to assist the driver could deal with the remaining items.

The main task of a TLR is to avoid accidents and save lives by informing the presence of a red or green traffic light to the driver in a non-intrusive way. More complex TLR can bring richer information, such as pointing out the main traffic light for a specific route (when there is more than one) and how far from the driver the traffic light is. Smarter insights could also be provided, as indicating the speed the driver should keep so he/she could advance the largest number of green signals in a row, for instance. A smart TLR is also useful for pedestrians who are visually impaired, especially those with sound signaling.

The electronic device used to build a smart TLR and how it should be positioned inside the vehicle influence the TLR success. If the device has a faulty camera or lack stability in the attachment to the vehicle, for example, images can be blurred or may not accurately reflect the scene.

Devices that are capable of automatically detect and classify objects, whether these are images or not, usually require rules-driven programming; the rules guide the device in its perception of the world. Machine learning methods automate the burden of having to find the proper set of rules to specific problems, as discussed in Section 2.

A major problem with ML techniques is the need of large datasets to find patterns, i.e., the rules. Large and representative datasets are difficult to build. Another important problem concerns processing costs: the more data, the greater the need for processing power and memory capacity. Machines with high computational power run into costs of acquisition and maintenance.

In Section 3, we propose a TLR layout prototype using a detection and recognition method from [2]. The proposed TLR uses the camera and processing unit of an ordinary smartphone. Furthermore, Section 4 seeks to evaluate if the use of Expert Knowledge (EK) within ML models reduce the burden of building up huge datasets. In Section 5, concluding remarks are reported.

2. Related Works

An object recognition mechanism works in two phases in order to recognize objects from an image: (1) detection of possible targets and (2) classification of the targets.

When working with object detection/recognition, we need to define what object features shall be used to guide the algorithm. In traffic light recognition, features such as light, shape and color are commonly used.

Artificial Neural Networks (NN), Saliency Map (SM), and Blob Detection (BD) are the most common techniques used to detect traffic lights. In [3–6], Convolutional Neural Network (CNN) was used to detect possible traffic lights whereas in [7] PCAnet was used with the same goal. In [8], the authors used a learning algorithm based on image feature channels and Histogram of Oriented Gradient (HOG) to detection and recognition. Saliency maps were used as a detection tool by [5,9–12]. We also observed fine examples of blob detection use in [13–15]. Geometric transforms were used in the detection phase by [16–18], which applied the Hough Circular Transform and by [19], which used the Radial Symmetry Fast Transform. Some less common techniques used alone or in association with the ones cited before are Adaptive Filters [2], Template Matching [20], Gaussian Distribution [21], Probability Estimation with CNN [3], and Top Hat [22]. Processing image algorithms are also commonly used to detect traffic lights: color or shape segmentation was used by [23,24] whereas threshold was used by [25,26].

For traffic light recognition, most works consider CNN and variants, Support Vector Machines (SVMs) and Fuzzy systems. In [27], the authors used a CNN whereas in [3,14] the authors used a PCAnetwork, an NN that simulates a CNN using less layers. SVMs were used by [2,7,12–14,28–32] to recognize traffic lights, sometimes along with a NN. Fuzzy systems were also used in [10,33]. Other techniques were used as ML substitutes, to improve false positives detection or to make the connection between detection output and recognition input. In [11–13,29,34], the authors used Histograms. In [18,35–37], the authors used Transforms. In [5,38–40], the authors used Template Matching. In [6], the authors used Saliency Map and in [41] probability histograms were used. Normalized Cross Correlation was observed in [42] to recognize pedestrian traffic lights. Hidden Markov Models (HMM) were used in [43] to recognize common traffic lights.

Table 1 relates the rates obtained by the works previously mentioned and its main techniques. It is difficult to make an analysis and compare the results to see which one was better once different metrics have been used and, more importantly, few papers worked with the same datasets. However, it is notable that many papers presented more than a 90% precision rate, or above an 80% accuracy rate. When the paper presented more than one result, we included the lowest and the highest value in the referred column at Table 1. The papers non-cited in Table 1 used only computational time as metric, which is out of scope for this paper.

The lowest accuracy rates were achieved by the use of Geometric Transforms, Template Matching and Color or Shape Segmentation Combined with Histograms, while the highest one was achieved by PCAnet/SVM technique, and also Color or Shape Segmentation Combined with Histograms. Regarding the Color or Shape Segmentation Combined with Histograms, this variety of results is understandable once each paper can use different approaches in this task.

The lowest precision rate was achieved by Template Matching, while all the other approaches have obtained above an 80% precision rate, including Template Matching in other tests in the same paper where the worst result was accounted.

Table 1. Rates achieved by related works and techniques.

Author	Technique (s)	Recall (%)	Precision (%)	Accuracy (%)
[13]	Blob Detection/SVM/Histograms	93.00	98.70	-
[9]	Saliency Map	97.66	93.20	-
[14]	Blob Detection/PCAnetwork/SVM	93.10	93.20	-
[2]	Adaptive Filters/SVM	94.69	92.20	-
[4]	CNN	99.83	99.70	-
[10]	Saliency Map/Fuzzy	-	-	80.00
[5]	CNN/Saliency Map/Template Matching	99.70	98.20	-
[29]	Color or Shape Segmentation/HOG/SVM	-	-	89.90
[30]	Color or Shape Segmentation/SVM	86.20	95.50	-
[21]	Gaussian Distribution	-	-	80.00–85.00
[16]	Geometric Transforms	-	-	56.00–93.00
[34]	Color or Shape Segmentation/Histograms	-	-	97.50
[7]	PCAnet/SVM	-	-	97.50
[6]	CNN/Saliency Map	-	-	96.25
[24]	Color or Shape Segmentation	-	-	92.00–96.00
[19]	Geometric Transforms	87.32	84.93	-
[17]	Geometric Transforms	-	-	70.00
[25]	Color or Shape Segmentation/Threshold	-	-	88.00–96.00
[35]	Color or Shape Segmentation/Histograms	-	-	50.00–83.33
[32]	Color or Shape Segmentation/SVM	98.96	99.18	-
[20]	Template Matching	98.00	97.00	-
[43]	Hidden Markov Models	-	-	90.55
[38]	Template Matching	-	-	90.50
[22]	Top Hat	-	-	97.00
[39]	Template Matching	-	-	69.23
[36]	Histograms	-	-	91.00
[41]	Probability Histograms	-	-	94.00
[18]	Geometric Transforms/Histograms	-	-	89.00
[40]	Template Matching	98.41	95.38	-
[44]	Template Matching	44.00–63.00	75.00–94.00	-

Data used in the related works are not always made available by the authors, and, when available, only a few are complete, i.e., contains separate traffic light images and whole traffic scene images. In Table 2, the Type column refers to what kind of traffic light the dataset contains, the Traffic light samples column shows how many images containing only a traffic light exists, these images are very useful to train Machine Learning algorithms and are obtained from whole frames containing traffic scenes. The Traffic frame samples column accounts how many traffic scenes exists in the dataset which are not the same used to extract the traffic lights images.

One of the most used datasets in academy research is the LARA dataset [40]. The LARA dataset provides video and image of vehicle traffic lights in day light.

A pedestrian traffic light dataset is the PRIA dataset provided by [45]. A larger dataset with both vehicle and pedestrian traffic light is the Worcester Polytechnic Institute (WPI) Dataset [14]. Another vehicle dataset is provided by [2]. All of these datasets provide only daytime samples of traffic lights.

Recently, the LISA dataset was provided as part of the Visions for Intelligent Vehicles and Applications project—VIVA [46]. LISA consists of a vehicle traffic light dataset, a traffic sign dataset, hand detection/gesture/tracking dataset and a face dataset, with samples of traffic lights in both daytime and nighttime.

It is important to notice that the datasets were built up in different regions of the planet, with differences concerning the shape and the type of the traffic lights. All experiments of this paper uses the dataset created by [2].

Considering the availability, balance, quality and the amount of data, we chose two datasets to work with. In this article, we show the results using the dataset presented by [2]. In future work, we will present new results using the dataset from [14]. Looking at Table 2, we can see that these two datasets are the only ones having traffic lights' exclusive images for machine learning training and whole traffic images.

Table 2. Public traffic light datasets.

Author	Type	Traffic Light Samples	Traffic Frame Samples
[46]	vehicle	-	43,007
[14]	vehicle and pedestrian	85,444	8,112
[2]	vehicle	20,952	5654
[8]	vehicle	-	800
[45]	pedestrian	-	501
[40]	vehicle	-	11,180

3. Traffic Light Recognition Device Prototype

A main question 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 observation is that the device shall be protected from adverse meteorological conditions like rain, or be waterproof. The heat also might cause problems in some electronic devices, so the sunlight incidence at the device location may be considered as well. In addition, trepidation of vehicle motion might have a critical influence in the device vision and, so, stabilization is a huge requirement. The TLR also should be able to provide warning sound to advise the distracted driver. Smartphones accomplish these requirements and, so, are an affordable alternative to TLR devices.

In this work, a smartphone was positioned inside a vehicle to capture actual traffic scenes with and without the presence of traffic lights. Two different types of support are commonly used to attach a smartphone in the car panel: air conditioning supports and windshield suction cups. Air conditioning supports can not be used to position a TLR device because it has no outside view from the vehicle. Windshield suction cups supports are a possible choice; however, the support may fall down or become very shaking if low quality suckers are used.

To overcome the smartphone support issue and to meet the requirements specified previously, we designed a stable device support using a two-sided tape and part of a windshield suction cup support. We remove the support portion that holds the device from the cable with a suction cup that is attached to the windshield. Then, we fixed the first part centralized at the vehicle panel with the two-sided tape. This design allows 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 1).

Three different smartphones were used to capture traffic videos containing traffic lights: a Motorola G second generation (Motorola Mobility LLC, Chaoyang District, China), an iPhone 6 (Foxconn CMMSG Indústria de Eletrônicos LTDA, Jundiaí-SP, Brazil), and a Galaxy S8+ (Samsung

Eletrônica da Amazônia LTDA., Manaus-AM, Brazil). All devices were configured to capture video with HD resolution. Figure 2 shows an example of images obtained with these devices. The images were extracted from videos at 5 frames per second (fps) rate.

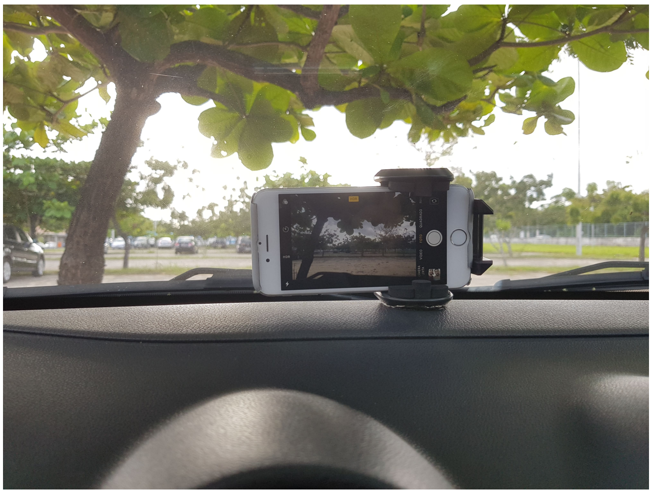
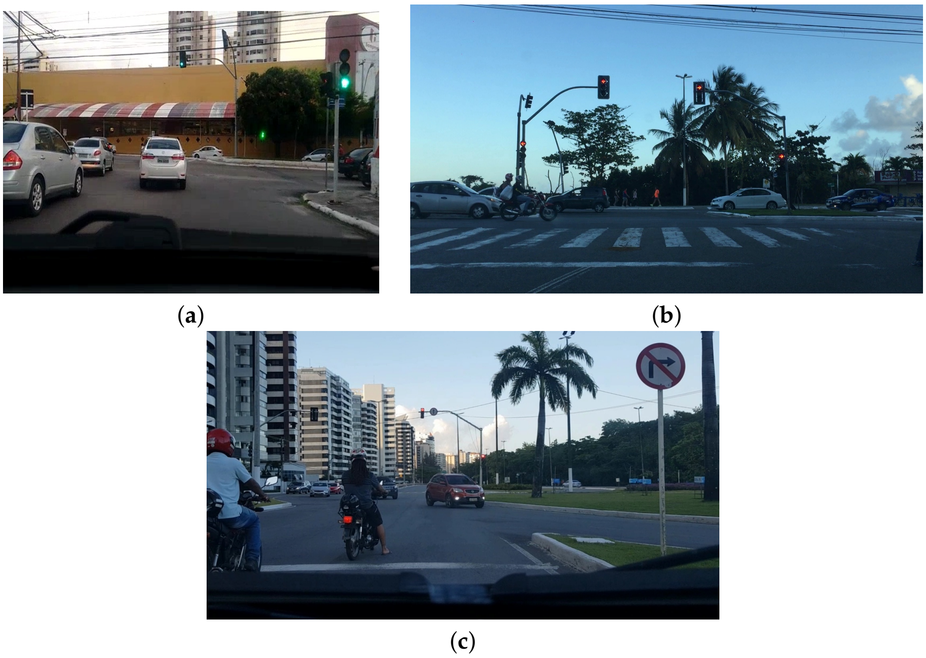


Figure 1. Traffic Light Recognition device support holding an iPhone 6.

A main concern when using smartphones in tasks like this is if such devices has enough processing power. It is important to notice that nowadays most smartphones have as much processing power and memory as some modern notebooks. Besides that, other researchers have been investigating and testing the use of mobile devices to recognize traffic lights throughout the last years. For example, Roters et al. [45] were able to recognize traffic lights in live video streams using a Nokia N95 mobile phone (manufacturer not specified by the author). The prototype could run 5 to 10 frames/s and used only a few seconds to give the user a feedback when testing in real field tests. In addition, Sung and Tsai [47] used an Android 4.04 smartphone equipped with a 1.5 GHz dual-core processor and 1 GB of RAM memory, with this configuration was able to process each image in 15.7 ms, which is half of the time required for real time processing on a mobile device, according to the author.



(**a**)

(**b**)

(**c**)

Figure 2. 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+.

More recently, Oniga et al. [21] used a smart mobile device equipped with a quad-core processor at 2.3 GHz. The results were in the range of 50–60 ms computation time for images with resolution of 1280 × 720 pixels, and 30 ms for images with resolution of 640 × 480. The image resolution used by [42] was 1413 × 1884, and the experiments with a Nexus 5 device (manufacturer not specified by the author) with Android 5 obtained an average computation time of 115.5 ms. This papers shows that, although it can be improved in some cases, today's smartphones have enough processing power to accomplish such a task.

3.1. Adaptive Background Suppression Filter

To highlight regions of interest (ROI) in the image, Oniga et al. [2] proposed an Adaptive Background Suppression Filter—AdaBSF. In the algorithm, a 4-channel feature map W_i, where i represent the 4-channel feature map index, is generated extracting R, G and B channels and calculating the normalized gradients of the input image.

To search for vertical and horizontal traffic lights, the window size for W_i is fixed at 16 × 8 pixels and 8 × 16 pixels, respectively. Since each window is four-dimensional, the pixel amount is $D = 16 \times 8 \times 4$ per window. Each window is represented by a feature vector x of size $D = 512$. The multi-scale problem was solved by down-sampling the original image to different scales while the window detection remains with fixed size.

The aim of AdaBSF algorithm is to design an Finite Impulse Response (FIR) filter specified by the vector $w = [w_1, w_2, ..., w_D]^T$ in a way that $y = w^T x$. The output y assigns a score to each detection window, which represents how likely the detection window covers a traffic light.

To classify the ROI found by AdaBSF, ref. [2] used Support Vector Machines (SVM). The author created a cascade of SVM classifiers that begins classifying the ROI, whether it is a traffic light or not. If it is a traffic light, the next SVM classifies the ROI into "red type" or "green type". The traffic light is further classified by the next SVM, observing whether it has an arrow and its direction, using an '1-vs.-1' voting method.

In this paper, the method proposed by [2] was applied to images obtained by a smartphone as a TLR device. The TLR follows a specific layout suitable for real time use. This layout is specified in the following sections. SVMs and the AdaBSF algorithm were trained with traffic light samples provided by the author. Negative samples, i.e., background samples, were extracted from four random test sequences also provided. Since we had no access to author's code, the algorithm was implemented in Python language and results were compared to the original's as a way to ensure coding correctness.

3.2. Prototype Results

Images obtained by the TLR device using this support prototype were submitted to classification in a personal computer using the method applied in [2].

The images were obtained using three different smartphone models, as cited before. Images from Motorola G 2nd Generation did not present good results and were discarded. Images obtained by the iPhone 6 contains 682 images, 209 negative samples and 473 traffic light samples. The third group is formed by 247 images obtained with a Galaxy S8+, being 165 traffic light samples and 82 negative samples. A total of 929 traffic images were analyzed: 638 images for green or red traffic lights and 291 negative samples.

Most images in [2]'s dataset present two equal traffic lights for the same road. In these cases, we account only one result by pair of classifications. If both traffic lights were classified correctly, or one was classified correctly and the other was missed, one true positive is accounted. If both traffic lights were missed or at least one of them is classified in the wrong class, one false negative is accounted. This reflects the real-life behavior when we just need to look at one traffic light to make a choice.

In Figure 3, it is possible to see detailed results from each image group. The two groups achieved high precision rates, but the iPhone 6 group presented a low recall (60%). This can be explained by the fact that the traffic light samples used in the training dataset are too different from some traffic lights of the iPhone group dataset as shown in Figure 4. If the training samples do not properly represent the real world, some traffic lights can not be recognized.

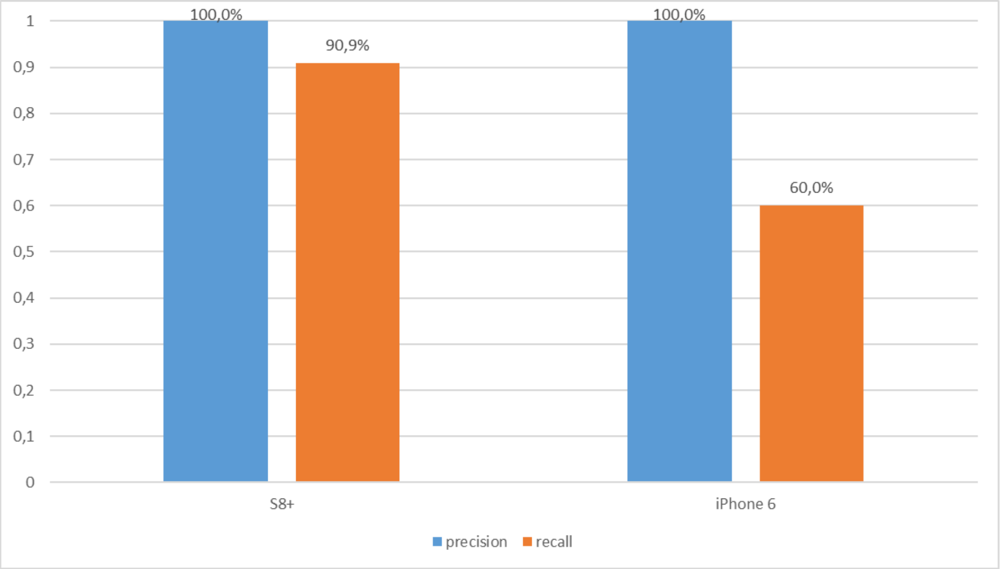


Figure 3. Precision and recall rates by the smartphone used to obtain the images.

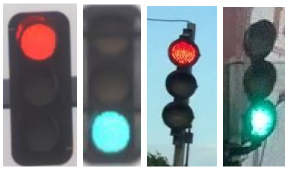


Figure 4. From left to right: the red and green traffic light samples used in training, red and green traffic light samples from the test dataset obtained with the TLR device prototype

In addition, due to geographic/meteorological issues (and possibly, the device used to obtain the images), the lighting aspect of the training set images is very different from that of the test set.

The low recall rates of the iPhone images group influenced the final rates of TLR tests, as observed in Figure 5. In comparison to the results obtained by [2], the TLR results are valid to justify its use in future research.

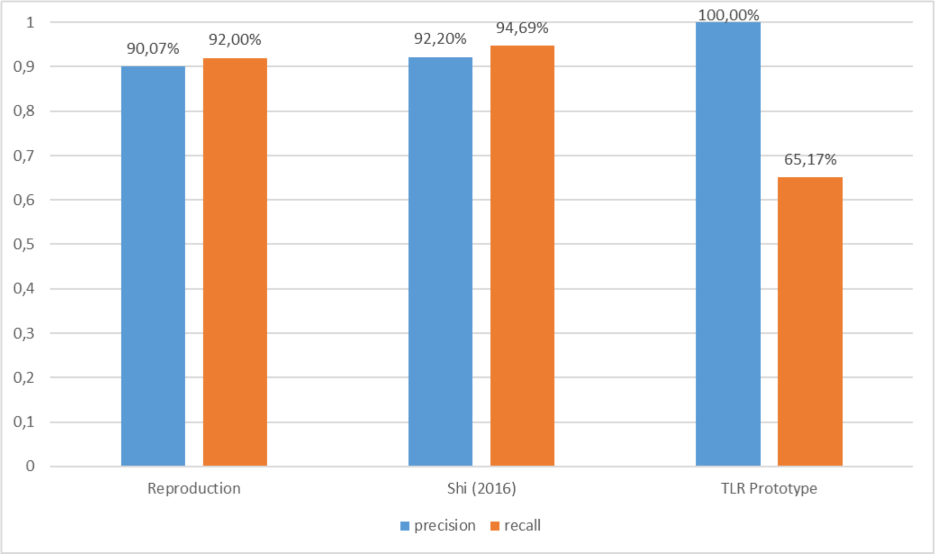


Figure 5. Precision and recall rates on our reproduction of [2], original work from [2], and tests using the images obtained with the TLR device prototype, respectively.

4. Expert Knowledge

The main problem with ML algorithms is that a reasonable amount of data is required, and the data needs to be balanced and categorized. Building up such database from scratch requires time and great effort and, in most cases, data may not even exist in large number.

We attempt to reduce the amount of required data to train a ML algorithm taking into account Expert Knowledge (EK). The EK used consists of the traffic light location on a given image obtained from the vehicle interior. The idea behind the chosen EK is that the traffic light is more likely to appear in certain locations, for example in the central and upper portions of the image, since they are always suspended on poles and must be visible to the driver from a reasonable distance.

To corroborate the idea, traffic light frequency maps were constructed. The method consists of hand tagging a set of images with the region in which the traffic light(s) appear. As a result, we have a frequency map of the regions where traffic lights most appear.

Datasets from [2] were used to generate a frequency map. A random sample of 650 images was generated from the test data and was analyzed. In Figure 6, we can observe the graphical representation of the frequency maps produced. Frequency data were smoothed using an averaged filter with size mask = 7. In the bottom, the scale goes from the lowest value (left) to the highest value (right).

To compare the results of the EK approach with the original or classic approach (without EK), the same data were used in training. Unfortunately, the datasets found in the literature do not provide the coordinates from which the traffic light samples were obtained from the original images, making it impossible to combine the previously calculated frequency with the training samples. To solve this problem, it was assumed that all training samples were found in regions of nonzero value in the frequency map, so the combination could be made with random frequency values. It was found that the data follow the beta distribution, which was used to generate the random frequency values so that they still keep a relation with the previous calculated frequency map.

In Figure 7, we observe the frequency distribution histogram of the map shown in Figure 6. The x-axis represents the previously calculated traffic light frequency values, and the y-axis represents the frequency of those values. The parameters of the beta distribution found for the map of Figure 6 were $(\alpha = 0.0342, \beta = 12.8985)$.

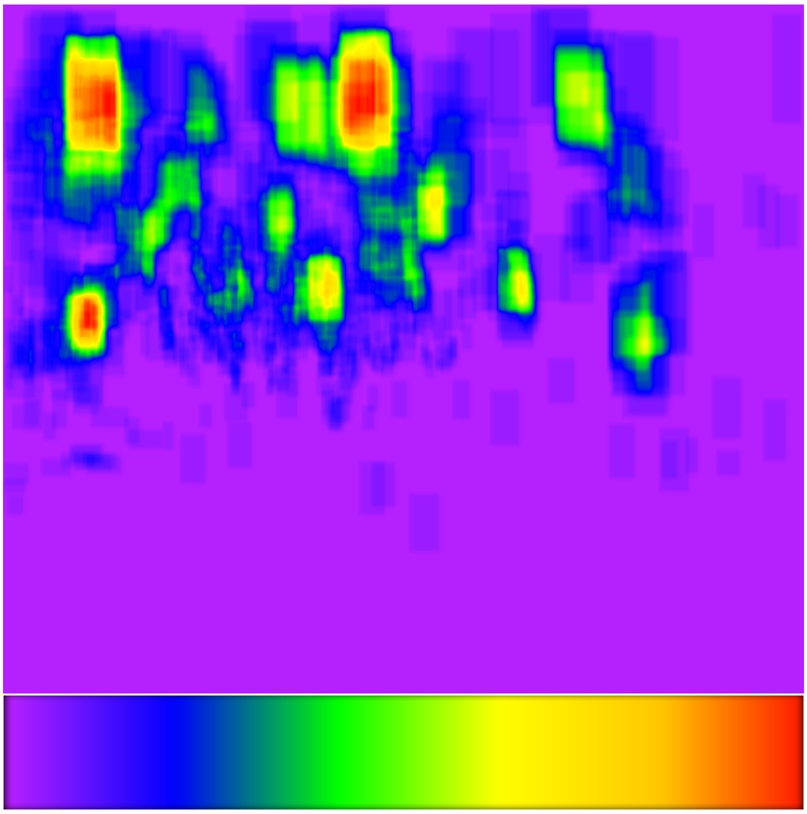


Figure 6. Frequency map of traffic light appearances in a sample of [2] dataset. Scale on the bottom.

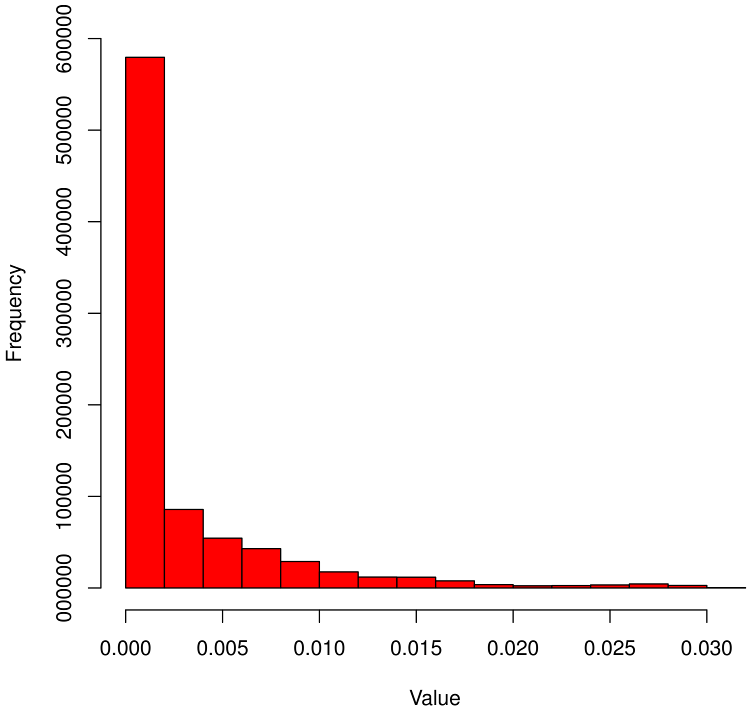


Figure 7. Precision and recall rates on our reproduction of [2], original work from [2], and tests using the images obtained with the TLR device prototype, respectively.

4.1. PCANet/SVM Classifier

Color extraction and blob detection are used to detect possible traffic lights or ROI. The color extraction is performed in the HSV color space. Blob detection is implemented by combining image processing algorithms such as flooding, contour tracking, and closure. The combination of these techniques allows the identification of both circular traffic lights and arrow lights.

The classification phase consists of a PCA network (PCANet). The PCANet is based on Principal Component Analysis—PCA. According to [48], the PCA is a statistics approach that can be used to analyze relations between a large number of variables to find a way to condensate the information from the set of original variables into a smaller set of variables with a minimal loss of information. This allows the PCANet to simulate the behavior of a Convolutional Neural Network (CNN).

The creators of PCAnet, [49], wanted to design a simple deep learning network to easily train and to adapt to different data and tasks. The convolution in a PCAnet is made using PCA filters as a kernel, the filters are found by an unsupervised learning method during the training process. The number of filters can vary and, accordingly to [14], more filters means better network performance. Eight (8) PCA filters were used in the two-layer PCAnet proposed.

The use of PCA filters in PCAnet is given as follows: for each patch of pixels in the input image, there is a filter of the same size. The mean of the pixels are calculated and its value is removed from the filter, operation that is called the Patch Mean Removal Phase. After that, the filters are used to convolve over the image. The combination of Patch Mean Removal Phase and PCA Filters Convolution Phase are called stages or layers of the network.

The output of the network is a binary hashing that is converted to a block-wise histogram in decimal values. The block-wise histogram represents the output features used to feed the SVM which performs the final ranking. For further details, see [49].

4.2. Training with Expert Knowledge

The traffic light recognition method used to test the EK insertion has been proposed by [14]. The author proposed a PCANet to find feature vectors given a set of traffic lights informed for training. The feature vectors are then used to train SVMs that will classify the traffic lights (see Section 4.1 for details).

The inclusion of EK in this process can take many different forms. It is possible, for instance, to obtain the coordinates (in the original scene) of the region classified by the method as a traffic light, analyze the frequency map in the same coordinates and submit these values to a threshold operation that will classify the traffic light. The problem of this approach is that the algorithm becomes too deterministic since only the exact regions in which the traffic light appear in the frequency map would achieve positive results.

A different approach is to perform a sum or multiplication operation on one of the recognition method internal phases. This could be done between PCANet layers, or after the PCANet output just before the SVM layer. The problem in this case is that the dimensionality of data is reduced by the convolution inside the algorithm, making it difficult the direct combination with the value of the frequency map.

We decided to use the EK as the first layer of the recognition method. In such a way, after the detection phase, Regions Of Interest (ROI) are multiplied by the frequency value obtained for the equivalent region in the frequency map. Since the frequency map and the original scene have the same dimensionality, and if one is in possession of the ROI coordinates in the original scene, this combination will tend to highlight the ROI found for the next algorithm.

An important concern at this point is to ensure that the expert instruction does not turn the method too deterministic, as it would if it were 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, although the odds are small, it is also possible that a traffic light appears in a non-common region of the scene, and TLR needs to be able to find it.

To avoid the combination of the expert instruction with the data completely canceling some data, it was necessary to define an increment inc that is added to the multiplication factor fat, which consists of the frequency value found in the frequency map. In all the experiments, we set $inc = 0.1$.

We have used same training set defined by [14], consisting of 9977 samples of green traffic lights and 10,975 samples of red traffic lights, amounting to 20,952 training samples in total. In our case, EK is added though.

After each training, an accuracy test was performed. The amounts of data used in the training were: 1000, 4000, 7000, 10,000 and 13,000.

In Figure 8, we can observe the accuracy rates obtained by successive tests. We can notice that the training with EK (orange line) using 1000, 4000 and 7000 samples increased in at least 15% the accuracy rate in comparison with the training without EK (blue line).

The difference between the two rates were close only when using 10,000 samples to train the model. In this case, the training with EK reached 98.38% accuracy and the method without EK obtained 93.02% accuracy. With 13,000 samples, the rates were even closer, with the EK training reaching 85.02% while the no-EK training obtained 86.48%. In this last test, we can see that, although the test without EK had a higher accuracy, the difference is only 1.46%, which is not statistically significant (p-value of the test for difference of proportions equal to 0.8092).

Other interesting information observed in Figure 8 is the decrease in the accuracy at the end of both lines, when the tests are performed with more data. This behavior—known as overfitting—is common when the model becomes too specialized in the data, which means that the model can classify correctly only the training data, but has not generalized the data to correctly classify new data.

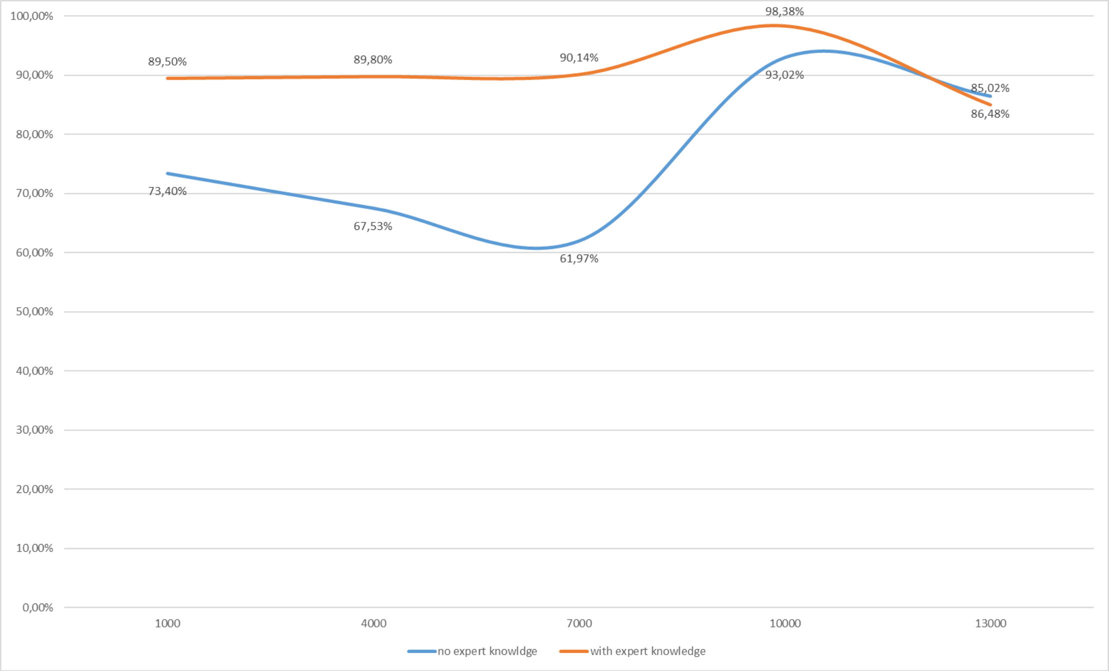


Figure 8. Test accuracies after training with expert knowledge (orange line) and without expert knowledge (blue line).

5. Conclusions

This work presented a TLR device layout prototype used to capture road scenes in order to validate the usage of a smartphone as a TLR in a real environment. This work also shows a method of EK inclusion to improve accuracy rates of a machine learning algorithm used to classify traffic light images.

The tests with the TLR device prototype achieved a 100% precision rate and 65% recall rate. The results demonstrate the prototype feasibility. The recall rate can be improved by training the applied algorithm with more representative samples, which will be done in the future along with cross-validation tests. The results also show that Galaxy S8+ and iPhone 6, two different mobile platforms, can be successfully used as TLR devices.

The use of Expert Knowledge explored in this paper also showed promising results. Training with 1000, 4000 and 7000 samples with Expert Knowledge always achieved test accuracy rates at least 15% higher than the training without Expert Knowledge. This result proves that the Expert Knowledge can reduce the amount of data required to train an algorithm, reducing at same time the computational effort needed. This experiment also stimulates the research of similar methods in other areas.

Future work includes testing the Expert Knowledge method in other datasets, tests with the complete flow of TLR, i.e., using a automatic traffic light detection method along with the Expert Knowledge classification algorithm, and real-time tests using the prototype presented in this paper with our classification method.

Author Contributions: Conceptualization, T.A. and H.M.; Methodology, T.A., H.M. and L.M.; Software, T.A.; Validation, T.A., H.M. and L.M.; Formal Analysis, T.A., H.M. and L.M.; Investigation, T.A., H.M.; Data Curation, T.A. and N.V.; Writing-Original Draft Preparation, T.A., H.M. and N.V.; Writing-Review & Editing, T.A. and H.M.; Visualization, H.M.; Supervision, H.M. and L.M.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001.

Acknowledgments: 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).

Conflicts of Interest: The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AdaBSF	Adaptive Background Suppression Filter
AI	Artificial Intelligence
BD	Blob Detection
CNN	Convolutional Neural Network
EK	Expert Knowledge
HD	High Definition
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradient
HSV	Hue, Saturation, Value
ML	Machine Learning
NN	Neural Network
PCANet	Principal Component Analysis Network
ROI	Region Of Interest
SM	Saliency Map
SVM	Support Vector Machine
TLR	Traffic Light Recognition
VIVA	Visions for Intelligent Vehicles and Applications
WPI	Worcester Polytechnic Institute

References

1. Gonçalves, J. Desrespeito à sinalização causa 7,5 mil acidentes no trânsito em 2014. Available online: http://radioagencianacional.ebc.com.br/educacao/audio/2015-07/desobediencia-sinalizacao-causa-75-mil-acidentes-no-transito-em-2014 (accessed on 9 July 2015).
2. Shi, Z.; Zou, Z.; Zhang, C. Real-Time Traffic Light Detection With Adaptative Background Suppression Filter. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 690–700. [CrossRef]

3. Weber, M.; Wolf, P.; Zöllner, J.M. DeepTLR: A single deep convolutional network for detection and classification of traffic lights. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 342–348. [CrossRef]
4. John, V.; Zheming, L.; Mita, S. Robust traffic light and arrow detection using optimal camera parameters and GPS-based priors. In Proceedings of the 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tokyo, Japan, 20-22 July 2016; pp. 204–208. [CrossRef]
5. 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 Trans. Comput. Imaging* **2015**, *1*, 159–173. [CrossRef]
6. John, V.; Yoneda, K.; Qi, B.; Liu, Z.; Mita, S. Traffic light recognition in varying illumination using deep learning and saliency map. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 2286–2291. [CrossRef]
7. Wang, Z.; Bi, Z.; Wang, C.; Lin, L.; Wang, H. Traffic lights recognition based on PCANet. In Proceedings of the 2015 Chinese Automation Congress (CAC), Wuhan, China, 27–29 November 2015; pp. 559–564. [CrossRef]
8. 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 Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 2341–2345. [CrossRef]
9. Kim, S.; Jang, E.; Hyun, S.; Han, D.S. Real time object detection based on saliency map. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 7–11 January 2016; pp. 534–535. [CrossRef]
10. Almeida, T.; Vasconcelos, N.; Benicasa, A.; Macedo, H. Fuzzy model applied to the recognition of traffic lights signals. In Proceedings of the 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, Colombia, 28–29 April 2016; pp. 1–4. [CrossRef]
11. Almeida, T.; Vasconcelos, N.; Benicasa, A. Framework para Detecção de Semáforos Baseado em Atenção Visual. In Proceedings of the Conference on Graphics, Patterns and Images (SIBGRAPI), Salvador, Brazil, 26–29 August 2015.
12. Ji, Y.; Yang, M.; Lu, Z.; Wang, C. Integrating visual selective attention model with HOG features for traffic light detection and recognition. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 280–285. [CrossRef]
13. Zhou, Y.; Chen, Z.; Huang, X. A system-on-chip FPGA design for real-time traffic signal recognition system. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 1778–1781. [CrossRef]
14. Chen, Z.; Huang, X. Accurate and Reliable Detection of Traffic Lights Using Multiclass Learning and Multiobject Tracking. *IEEE Intell. Transp. Syst. Mag.* **2016**, *8*, 28–42. [CrossRef]
15. Zhang, Y.; Xue, J.; Zhang, G.; Zhang, Y.; Zheng, N. A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; pp. 4924–4929. [CrossRef]
16. Widyantoro, D.H.; Saputra, K.I. Traffic lights detection and recognition based on color segmentation and circle hough transform. In Proceedings of the 2015 International Conference on Data and Software Engineering (ICoDSE), Yogyakarta, Indonesia, 25–26 November 2015; pp. 237–240. [CrossRef]
17. Chiu, Y.T.; Chen, D.Y.; Hsieh, J.W. Real-time traffic light detection on resource-limited mobile platform. In Proceedings of the 2014 IEEE International Conference on Consumer Electronics, Taipei, Taiwan, 26–28 May 2014; pp. 211–212. [CrossRef]
18. Omachi, M.; Omachi, S. Detection of traffic light using structural information. In Proceedings of the IEEE 10th International Conference on Signal Processing Proceedings, Beijing, China, 24–28 October 2010; pp. 809–812. [CrossRef]
19. Sooksatra, S.; Kondo, T. Red traffic light detection using fast radial symmetry transform. In Proceedings of the 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Ratchasima, Thailand, 14–17 May 2014; pp. 1–6. [CrossRef]

20. Trehard, G.; Pollard, E.; Bradai, B.; Nashashibi, F. Tracking both pose and status of a traffic light via an Interacting Multiple Model filter. In Proceedings of the 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–7.
21. Oniga, F.; Prodan, S.; Nedevschi, S. Traffic light detection on mobile devices. In Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 3–5 September 2015; pp. 287–292. [CrossRef]
22. Jie, Y.; Xiaomin, C.; Pengfei, G.; Zhonglong, X. A new traffic light detection and recognition algorithm for electronic travel aid. In Proceedings of the 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), Beijing, China, 9–11 June 2013; pp. 644–648. [CrossRef]
23. 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. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1258–1265. [CrossRef]
24. Zong, W.; Chen, Q. Traffic Light Detection Based on Multi-feature Segmentation and Online Selecting Scheme. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 204–209. [CrossRef]
25. Sathiya, S.; Balasubramanian, M.; Priya, D.V. Real time recognition of traffic light and their signal count-down timings. In Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 27–28 February 2014; pp. 1–6. [CrossRef]
26. 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 Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 431–435. [CrossRef]
27. Chen, Z.; Shi, Q.; Huang, X. Automatic detection of traffic lights using support vector machine. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 37–40. [CrossRef]
28. 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. [CrossRef]
29. Michael, M.; Schlipsing, M. Extending traffic light recognition: Efficient classification of phase and pictogram. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8. [CrossRef]
30. Salarian, M.; Manavella, A.; Ansari, R. A vision based system for traffic lights recognition. In Proceedings of the 2015 SAI Intelligent Systems Conference (IntelliSys), London, UK, 10–11 November 2015, pp. 747–753. [CrossRef]
31. Jang, C.; Kim, C.; Kim, D.; Lee, M.; Sunwoo, M. Multiple exposure images based traffic light recognition. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 1313–1318. [CrossRef]
32. Chen, Q.; Shi, Z.; Zou, Z. Robust and real-time traffic light recognition based on hierarchical vision architecture. In Proceedings of the 2014 7th International Congress on Image and Signal Processing, Dalian, China, 14–16 October 2014; pp. 114–119. [CrossRef]
33. Chung, Y.C.; Wang, J.M.; Chen, S.W. A vision-based traffic light detection system at intersections. *J. Natl. Taiwan Normal Univ. Math. Sci. Technol.* **2002**, *47*, 67–86.
34. Almagambetov, A.; Velipasalar, S.; Baitassova, A. Mobile Standards-Based Traffic Light Detection in Assistive Devices for Individuals with Color-Vision Deficiency. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1305–1320. [CrossRef]
35. 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. In Proceedings of the 2014 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, Poland, 22–24 September 2014; pp. 110–115.
36. Cai, Z.; Li, Y.; Gu, M. Real-time recognition system of traffic light in urban environment. In Proceedings of the 2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, ON, Canada, 11–13 July 2012; pp. 1–6. [CrossRef]

37. Omachi, M.; Omachi, S. Traffic light detection with color and edge information. In Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 8–11 August 2009; pp. 284–287. [CrossRef]
38. Choi, J.; Ahn, B.T.; Kweon, I.S. Crosswalk and traffic light detection via integral framework. In Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision, Incheon, Korea, 30 January–1 February 2013; pp. 309–312. [CrossRef]
39. Fan, B.; Lin, W.; Yang, X. An efficient framework for recognizing traffic lights in night traffic images. In Proceedings of the 2012 5th International Congress on Image and Signal Processing, Dindigul, India, 22–24 February 2012; pp. 832–835. [CrossRef]
40. de Charette, R.; Nashashibi, F. Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 358–363. [CrossRef]
41. Levinson, J.; Askeland, J.; Dolson, J.; Thrun, S. Traffic light mapping, localization, and state detection for autonomous vehicles. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5784–5791. [CrossRef]
42. 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. [CrossRef]
43. 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. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 750–755. [CrossRef]
44. de Charette, R.; Nashashibi, F. Traffic light recognition using image processing compared to learning processes. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 333–338. [CrossRef]
45. Roters, J.; Jiang, X.; Rothaus, K. Recognition of Traffic Lights in Live Video Streams on Mobile Devices. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1497–1511. [CrossRef]
46. 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.* **2016**, *17*, 1800–1815. [CrossRef]
47. Sung, T.P.; Tsai, H.M. Real-time traffic light recognition on mobile devices with geometry-based filtering. In Proceedings of the 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC), Palm Springs, CA, USA, 29 October–1 November 2013; pp. 1–7. [CrossRef]
48. Hair, J.; Anderson, R.; Tatham, R.; Black, W. *Multivariate Data Analysis*, 6th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2006.
49. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [CrossRef] [PubMed]