Relevant Traffic Light Localization via Deep Regression

Rafael Horimoto de Freitas[1], **Thiago M. Paixão**[1,2], **Rodrigo F. Berriel**[1], **Alberto F. De Souza**[1], **Claudine Badue**[1], **Thiago Oliveira-Santos**[1]

[1]Universidade Federal do Espírito Santo, Brazil

[2]Instituto Federal do Espírito Santo, Brazil

rafael.hdefreitas@gmail.com

Abstract. *Artificial intelligence advances have an important role on self-driving cars development, such as assisting the recognition of traffic lights. However, when relying on images of the scene alone, little progress was observed on selecting the traffic lights defining guidance to the car. Common detection approaches rely on additional high-level decision-making process to select a relevant traffic light. This work address the problem by proposing a deep regression system with an outliers resilient loss to predict the coordinates of a relevant traffic light in the image plane. The prediction can be used as a high-level decision-maker or as an assistant to a cheaper classifier to work on a region of interest. Results for European scenes show success in about* 88% *of the cases.*

1. Introduction

Advances in artificial intelligence have an important role on enabling the automation of several tasks, including the development of self-driving cars [Badue et al. 2019]. This topic has received great attention in the recent years, and a great part of the related research addresses machine learning techniques to provide perception and understanding of the traffic environment.

Besides keep tracking of moving objects (e.g., other vehicles and pedestrians), self-driving cars should be capable of adjusting their behavior according to the visual signals that are presented along the road. In particular, for safety reasons, it is essential that these vehicles perceive traffic lights along the road, i.e., identify (recognize) their states (i.e., red, yellow, green). This general problem is referred in literature as Traffic Light Recognition (TLR) [Philipsen et al. 2015, Jensen et al. 2016].

To enable TLR, self-driving vehicles are usually equipped with one or more forward-looking cameras that capture traffic scenes from the driver point of view. The images are later processed in order to detect (i.e., locate) traffic lights based on structural and/or appearance models [Gómez et al. 2014, Diaz-Cabrera et al. 2015, Li et al. 2018], or using learning-based techniques [Lindner et al. 2004, Franke et al. 2013, Barnes et al. 2015, Jensen et al. 2015].

More recently, state-of-the-art deep networks were used to jointly perform the detection and state classification of traffic lights [Behrendt et al. 2017, Jensen et al. 2017, Pon et al. 2018, Müller and Dietmayer 2018]. Despite the significant advances in solving TLR, when relying on images of the scene alone, little progress was observed for a more specific and interesting (considering real world applications) problem: the recognition of

the state focusing on the relevant traffic lights [Fregin et al. 2018], i.e., a subset of the visible traffic lights that define whether or not the vehicle can continue on its way.

The common strategy to address this problem comprises two major steps: (i) the detection of traffic lights in a scene and (ii) a decision-making process to select an exemplar of relevant traffic light. For the second step, several works [John et al. 2014, Mu et al. 2015, Jang et al. 2017, Possati et al. 2019] propose to decide the relevant traffic light by combining complex localization systems with priorly mapped traffic-lights which are relevant for specific routes. The localization is the process of estimating the current car's pose in the world and is performed with the aid of several expensive sensors like Global Positioning System (GPS) and Light Detection And Ranging (LiDAR). Although this approach achieves high accuracy, it demands expensive annotations and restricts the TLR system to work only with expensive sensors and on previously mapped areas. These problems could be avoided by heuristically deciding the relevant traffic light based on position and size assumptions [Li et al. 2018]. Such approach, however, comes at the cost of losing some accuracy since the selection is blindly performed without considering the context of the scene.

To tackle the aforementioned issues, this work uses a deep regression model to predict the 2-d coordinates (a single point per image) of a relevant traffic light in the image plane. This information can be incorporated into a TLR system in order to decide which traffic light the car should obey. The regression model – a convolutional neural network (CNN) – is trained specifically to regress the coordinates of a particular traffic light: the relevant traffic light closest (in euclidean distance sense) to the top-center position of the input image (thereafter referred to as target). The training is driven by an outliers resilient loss function proposed in the scope of this work.

The conducted experiments evaluated the feasibility of using the predicted coordinates to recover relevant traffic lights when used in conjunction with an ideal detector (i.e., the ground-truth bounding boxes). Results for European traffic scenes show that our method was able to recover a relevant traffic light in approx. 88% of the cases.

The remainder of the text is organized as follows. The next section describes the proposed localization method. In Section 3, the experimental methodology is described. Results and discussing are in Section 4, whereas conclusions are withdrawn in Section 5.

2. Relevant Traffic Light Localization

The pipeline of the proposed method (illustrated in Figure 1) is broadly divided into the learning and test stages. The learning stage (left part of Figure 1) requires a collection of images depicting traffic scenes which are annotated with the respective target's position (2D coordinates of the relevant traffic lights in the scene, indicated by the yellow cross marker). Then, given an input image and its annotation, a deep convolutional neural network (CNN) is trained as a regression model in order to predict the target position of the traffic light. In the test stage, the current image captured with the car's onboard camera is passed to the trained model in order to regress the current target position. The remainder of this section focuses on describing the deep regression model and the loss function proposed to guide the model training. Details of the training procedure are presented in the next section.

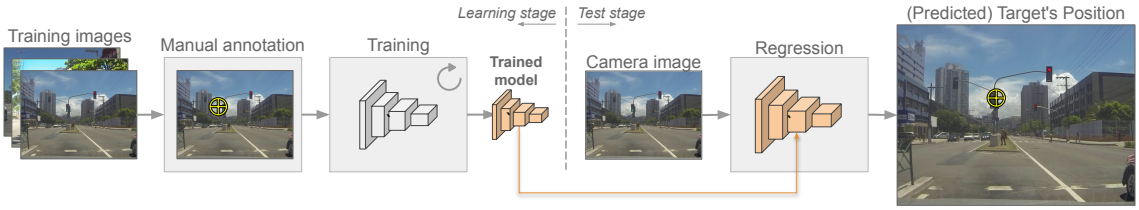

Figure 1. Overview of the proposed method for relevant traffic light localization. The yellow circle with a cross marks the position of the relevant traffic light in the image.

2.1. Deep Regression Model

The regression model (illustrated in Figure 2) is a deep neural network with input size $1024 \times 512 \times 3$ comprising a backbone for feature extraction, and some fully connected layers appended in the end. The backbone is a modified ResNet-50 architecture [He et al. 2016] (referred to as ResNet-50*) resulting from removing the average pooling layer (*avg pool*) and the subsequent fully connected layer (*fc 1000*). Instead, the final features are obtained by convolving the $32 \times 16 \times 2048$ volume outputted by the last convolutional block of ResNet* (*conv5_x*) with a $1 \times 1 \times 16$ filter, with ReLU activation, and then flattening the resulting volume.

The regression part comprises a stack of 7 fully connected layers (*fc 256*) – each one outputting 256-d features – followed by a single fully connected layer (*fc 2*) that outputs a 2-d vector. ReLU is used as the activation function of the *fc 256* layers, whereas an identity function is applied in *fc 2*. Instead of directly predicting the target's position $\hat{\mathbf{p}}_t = (\hat{x}_t, \hat{y}_t) \in [0, w] \times [0, h]$, with $w = 1024$ and $h = 512$, the model regresses normalized coordinates $\hat{\mathbf{p}}_m = (\hat{x}_m, \hat{y}_m)$ such that:

$$\hat{x}_t = [(\hat{x}_m + 1)/2]w \tag{1}$$

$$\hat{y}_t = \hat{y}_m h. \tag{2}$$

This normalization preserves the aspect ratio, and maps the top-center position of the input image onto $(0, 0)$ in the (normalized) regression domain. The $\hat{x}_m$, $\hat{y}_m$ values are expected to be (most of the time) within the ranges $[-1, 1]$ and $[0, 1]$, respectively. However, this is not ensured since the image of the identity activation function (in *fc 2*) is unbounded. Therefore, the final prediction $\hat{\mathbf{p}}_t$ is not restricted to the image frame, making it possible to predict the position of traffic lights that are cut by the image boards with it middle point outside the image.

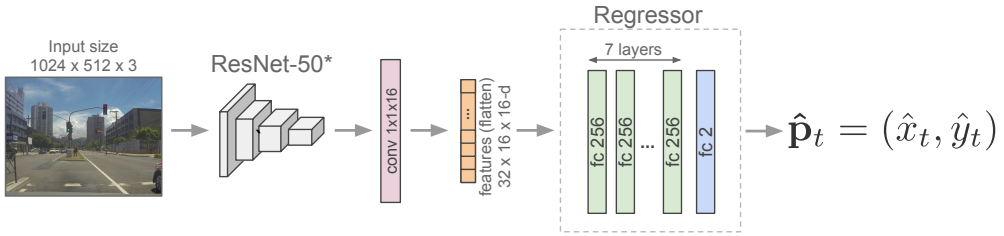

Figure 2. Deep regression model.

2.2. Loss Function

The loss function used to train the regression model was adapted from the Huber loss function [Huber 1992] in order to be still less sensitive to outliers, i.e., to have less influence of those predictions too far from ground-truth positions. Given a ground-truth position $\mathbf{p}_t$ in the image domain, a prediction $\hat{\mathbf{p}}_t$ is considered an outlier *iff* $||\hat{\mathbf{p}}_t - \mathbf{p}_t||_2 > 16$. Therefore, in the regression domain, $\hat{\mathbf{p}}_m$ is an outlier *iff* $||\hat{\mathbf{p}}_m - \mathbf{p}_m||_2 > 1/32$, or, analogously, *iff* $z = 32||\hat{\mathbf{p}}_m - \mathbf{p}_m||_2 > 1$. Based on this last relation, the loss function was piecewise defined as

$$\mathcal{L}(z) = \begin{cases} z^2, & \text{if } z \leq 1 \\ log(z^2) + 1, & \text{otherwise.} \end{cases} . \tag{3}$$

The function in Equation 3 is also continuous and differentiable for $z = 1$, since $z^2 = log(z^2)+1 = 1$ and $\frac{d}{dz}(z^2) = \frac{d}{dz}(log(z^2)+1) = 2$. Our loss function is depicted in Figure 3 together with $2\times$Huber and L_2 losses for comparison (Huber loss is doubled for better visualization and comparison). Note the smoother behavior of the proposed function for $z > 1$.

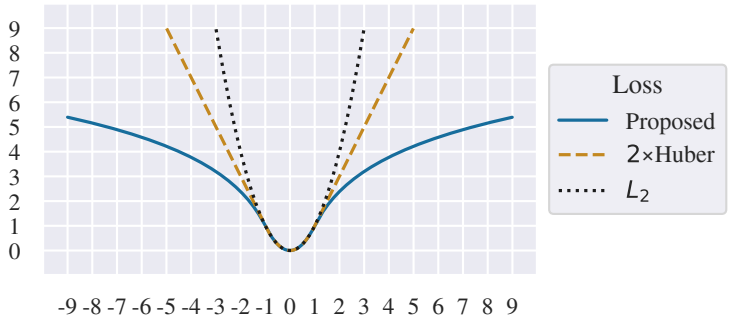

Figure 3. The proposed loss function together with $2\times$Huber and L_2 losses.

3. Experimental Methodology

This section describes the experimental evaluation of the proposed method, which includes: the training and test datasets, the data augmentation process, the experiments, and the software/hardware platform.

3.1. Training and Test Datasets

The experiments were run on the DriveU Traffic Light Dataset (DTLD) [Fregin et al. 2018], the largest publicly available dataset of traffic lights. DTLD was assembled based on daytime records of 11 German cities in different weather conditions. Scenes were originally captured by two cameras (stereo), being the left camera data used to annotate traffic lights, resulting in more than 40,000 frames of 2048×1024 pixels with more than 230,000 hand-labeled bound-boxes.

For our purposes, as discussed in Section 2, images without relevant traffic lights were discarded since the proposed method assumes the car is already in a place where a

decision should be made, i.e., there is a relevant traffic light in the scene. Such information could come, for example, from navigation systems based on inexpensive GPSs. The remaining images were resized to 1024×512 in order to fit the network's input. The ground-truth annotation (i.e., traffic light positions) was derived from the bounding boxes annotation by computing the middle point of the boxes. We leveraged the original DTLD train and test splits, both including images from the 11 cities. The images from Bremen and Fulda cities in the training split were used only for validation. Trivial scenes with only relevant traffic lights were discarded from the test partition.

3.2. Data Augmentation

To increase variability in the training data, the images of the training partition were submitted to an off-line data augmentation process. Two new instances were produced from each training image. In some cases, a third additional instance was generated to reach the total of 65,536 (2^{16}) instances. The augmentation process comprises four sequential operations: (i) luminosity transformation, (ii) affine transformation, (iii) blur and (iv) horizontal flip. The parameters of the operations were picked randomly for each image.

Luminosity Transformation The luminosity transformation consists in multiplying the image pixels by a factor in $[f_{lum}(m), 2f_{lum}(m)]$, where f_{lum} is the function defined in Equation 4 and m is the mean value of the luminosity image (taken as the channel-wise maximum for each pixel). In summary, the transformation was designed to avoid over/underflow for high/low luminosity images.

$$f_{lum}(x) = \begin{cases} 0.5, & \text{if } x > 128 \\ 64/x, & \text{if } 64 \leq x \leq 128 \\ 1.0, & \text{otherwise.} \end{cases}, \tag{4}$$

Affine Transformation The affine transformation comprises uniform scaling by a factor in $[31/32, 33/32]$, rotating by an angle in $[-\pi/64, \pi/64]$, and finally translating the image in both axis (independently) restricted to the interval $[-16, 16]$. Background pixels (i.e., those not defined by the original image) were assigned 128 for the three channels. This transformation is not applied only when target is closer than 64 pixels from some of the image's borders.

Blur The blur is one between a gaussian blur with σ in $[0, 1]$ or, with same chances, a median blur with a 3×3 kernel.

Horizontal Flip In the end, some generated images are horizontally flipped. If two instances were generated from a training image, one of them was flipped. In the cases where three instances were generated, one of them was flipped in half of the cases and two instances were flipped in the other cases.

3.3. Experiments

The conducted experiments aim to assess the ability of the proposed method in selecting relevant traffic lights in scenes containing (simultaneously) relevant and irrelevant exemplars. Since the method outputs coordinates, a traffic light is said to be selected if it is the closest traffic light with respect to the regressed point and (optionally) if the respective distance is not above a predefined threshold.

The model was trained and tested with the proposed loss (Section 2.2) and with the Huber loss as a performance baseline. A single training-test section was conducted for each loss. For quantitative evaluation, the accuracy was defined as the ratio between correct choices (for all the test images) and the number of tested images. The "correct" choice means (i) the selected traffic light is among the relevant ones or, more strictly, (ii) it is exactly the target traffic light when using the same criteria as in the training phase. Both scenarios were investigated in the experiments. Additionally, the method's performance was also investigated for a subset of difficult instances, here defined as those scenes whose the traffic light closest to the top-center position is not a relevant one.

Training Details The model was trained during 8 epochs with the Stochastic Gradient Descent (SGD) algorithm (0.9 of momentum) using 16-size mini-batches (in fact, for hardware limitations, images were passed in 8-size batches and the gradients for every 2 subsequent batches were accumulated). The loss considered for each batch was the sum of the losses for it images. If the mean loss was considered, it would be necessary to take the mean instead of accumulating gradients. The training images were shuffled off-line and the resulting order was kept throughout the epochs. A validation step was performed every $\frac{1}{8}$ of epoch to determine the best model, defined as that with lower average loss on the validation set. The initial learning rate was 2^{-14} for the Huber loss and 2^{-16} for the proposed loss. In both cases, the learning rate was halved every 2 epochs. The model was initialized with pretrained weights for ImageNet [Krizhevsky et al. 2012], except for the altered layers, which were randomly initialized. For compatibility with the pretrained model, the input images' channels were normalized to values in $[\frac{-\mu}{\sigma}, \frac{1-\mu}{\sigma}]$, being μ and σ the mean and standard deviation (normalized to values in $[0, 1]$) of the respective channel averaged across the ImageNet instances.

3.4. Experimental Platform

The experiments were conducted in an Intel® Core™ i7-4770 CPU @ 3.40GHz with 16GB of RAM equipped with Linux Ubuntu 16.04 and 1 TITAN X (Pascal) GPU with 12GB of memory. Python 3.5 was used to implement the experiments. Training and inference were performed using PyTorch 1.1 deep learning framework [Paszke et al. 2017] configured with CUDA 9.0 and cuDNN 7.3 for low-level computations. The average time (approximate value) for training was 7 hours and 20 minutes, and the inference time per image was, on average, less than 20 ms (more than 50 FPS). The implementation will be made available at https://github.com/LCAD-UFES/publications-horimoto-eniac2019/blob/master/README.md.

4. Results and Discussion

Figure 4 shows the method's accuracy in selecting relevant traffic lights without set any distance threshold. The curves were plotted considering the increasing number of relevant exemplars present in the image (3+ indicates three or more exemplars). The numbers inside the graph (i.e., 2340, 4386, etc.) represent the amount of images for each subcase (the quantities are the same for both losses).

Clearly, the proposed loss yielded better accuracy than Huber loss, notably for the challenging instances (Difficult). It can be noticed, nevertheless, that the losses tend to perform more similarly (and better) with the increasing of relevant traffic lights in the scene. As expected, the lower accuracy is observed for scenes with only one relevant traffic light (this exemplar is also the target). Interestingly, for this case, the accuracy achieved with proposed loss on difficult cases (73.60%) also surpassed the Huber loss accuracy on the entire dataset (68.42%). Moreover, grouping all the three subcases (i.e., 1+ relevant images), the Huber loss yielded accuracies of 81.92% and 60.47% for the entire dataset and the difficult instances, respectively, whereas the proposed loss yielded 88.59% and 76.62%. This shows a great improvement, mainly in the difficult cases.

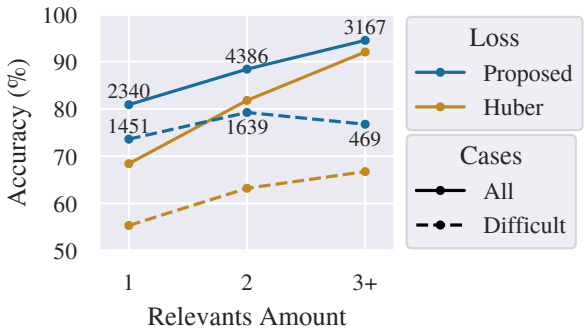

Figure 4. Selection of a relevant traffic light. The numbers inside the graph indicates the total number of images of each subcase.

Figure 5 shows the accuracy of selecting relevant traffic lights, but now with the additional distance threshold constraint. Different accuracies were obtained by varying the distance threshold (horizontal axis). The right graph depicts the same information of the left graph but restricted to the interval $[0, 64]$.

As can be seen in the left graph, the four curves increases very sharply, reaching close to the maximum for low distance thresholds. Note that the accuracy converge to the values obtained by grouping the three subcases in Figure 4. This means that applying a relatively small threshold does not affect significantly the performance. Based on this observation, the predicted position could be used to restrict an area of interest surrounding a relevant traffic light. Therefore, instead of using our method in conjunction with detectors, it could be leveraged to crop traffic lights whose state could be determined by a classifier. The right graph shows more detailed the same curves for smaller thresholds. Considering the entire test set, the Huber loss yielded accuracies of 56.66% and 74.24% for the threshold values 16 and 32, respectively, whereas proposed loss yielded accuracies of 80.22% and 84.49%.

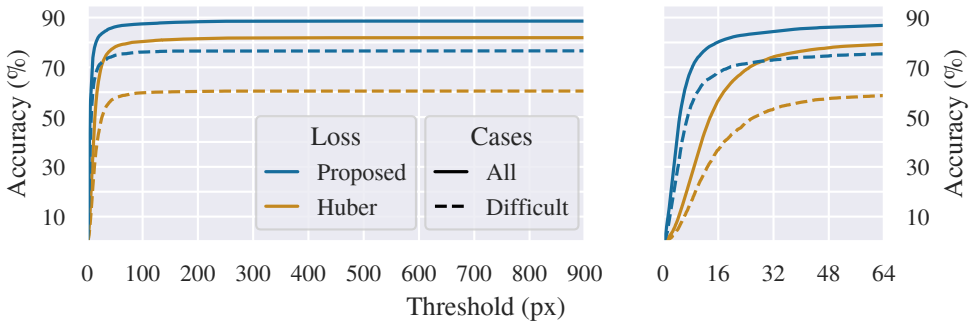

Figure 5. Selection of a relevant traffic light with threshold distance.

Figure 6 shows a more strict scenario where accuracy is related to the ability of correctly selecting the target traffic light. Comparing to Figure 5, the curves in Figure 6 present a similar behavior (i.e., they quickly increase towards the maximum) but achieving slightly lower accuracies. The maximum accuracies for the Huber loss were 76.35% and 56.98% for the entire test set and the difficult cases, respectively, while the proposed loss yielded 82.91% and 72.21%.

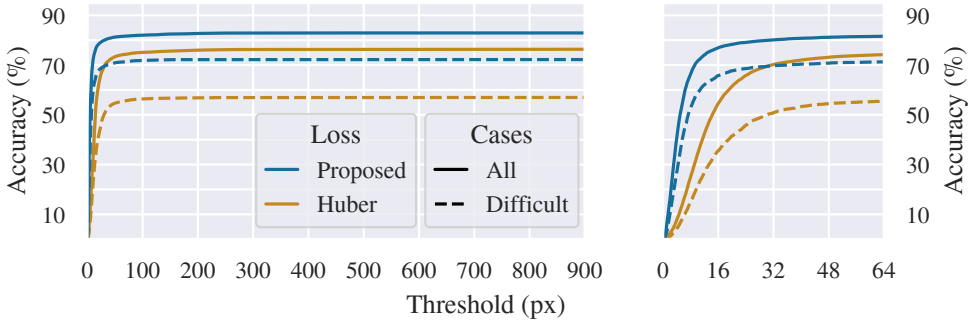

Figure 6. Selection of the target traffic light with threshold distance.

Figure 7 shows results on several images from DTLD test partition. First row shows very easy cases with multiple relevant. Second row shows successful cases with a relatively great irrelevant traffic light next to the target. Third row shows difficult cases failures, mostly occur when the irrelevant that is closer to the top-center position is also relatively great. Fourth row shows false failures due to wrong annotated data. Fifth row shows failures where a relevant was surrounded by many smaller irrelevant. Sixth row shows cases where the target traffic light is cut by the image's boards, in some of these cases the middle point of the traffic light is outside the image and the model was also capable to predict a close position outside.

As can be seen the system delivery high quality predictions in most of the scenes and for most of it failuers it is possible to identify reasons for difficulty, giving direction on what need to be enhanced in future works.

Figure 7. Results on several images from DTLD test partition. Relevant and irrelevant traffic lights are marked in the images with yellow and lower magenta squares, respectively. The predicted coordinates are marked with a circle of the same color of the selected traffic light square. A cyan line connects the prediction to its selection. First row shows very easy cases. Second row shows successful cases with a great irrelevant traffic light next to the target. Third row shows difficult cases failures. Fourth row shows false failures due to wrong annotated data. Fifth row shows failures where a relevant was surrounded by many irrelevant. Sixth row shows cases where the target traffic light is cut by the image's boards

5. Conclusion

This work explored the Traffic Lights Recognition problem, more specifically the localization of the, or one of the, relevant traffic lights of a scene. A particularly challeging task, considering the small size of traffic lights compared to the image and considering the presence of other traffic lights rather than the relevant ones. A deep regression model and also a novel regression loss that is less sensitive to outliers were proposed. The pro-

posed model training rely just on the position of the relevant traffic light (the closer to the top-center position of the image in case of multiple options), a relatively cheap annotation compared to others detection systems that need bound-boxes of all the traffic lights of the scene.

The proposed model was trained/tested over the DTLD dataset training/test partitions, with the proposed loss and also the Huber loss for comparison. The training with the Huber loss led to 81.92% of success rate in selecting a relevant traffic light from test annotations, 74.24% when imposing a maximum threshold distance of 32 pixels between the regression and the selection, and 56.66% further decreasing this threshold to 16 pixels. The training with the proposed loss led to the improvement of these rates to 88.59%, 84.49% and 80.22%, respectively.

The results are promising and show that the system can assist other detecting systems selecting a relevant from it detections. In addition, they also show that the successful regressions are, mostly, very close to the selected relevant, which makes it possible to define a region of interest to assist a cheaper traffic light state classifying system.

Future work include integrating the proposed system into a complete system to predict the relevant state of traffic lights in the road.

Acknowledgement

We would like to acknowledge the scholarships of Productivity on Research (grants 311120/2016-4 and 311504/2017-5) and a master student scholarship supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil); the financial support provided in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior 'Brasil (CAPES)' Finance Code 001 and by the Programa de Apoio a Núcleos Emergentes (Pronem, grant 594/2018) from Fundação de Amparo à Pesquisa e Inovação do Espírito Santo (FAPES, Brazil); as well as to thank the NVIDIA Corporation for their donation of GPUs.

References

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L. F. R., Berriel, R. F., Paixão, T. M., Mutz, F., et al. (2019). Self-driving cars: A survey. *arXiv preprint arXiv:1901.04407*.

Barnes, D., Maddern, W., and Posner, I. (2015). Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation. In *Intelligent Vehicles Symposium (IV)*, pages 573–578.

Behrendt, K., Novak, L., and Botros, R. (2017). A deep learning approach to traffic lights: Detection, tracking, and classification. In *International Conference on Robotics and Automation (ICRA)*, pages 1370–1377.

Diaz-Cabrera, M., PietroCerri, and PaoloMedici (2015). Robust real-time traffic light detection and distance estimation using a single camera. *Expert Systems with Applications*, 42(8):3911–3923.

Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., and Herrtwich, R. G. (2013). Making Bertha See. In *International Conference on Computer Vision Workshops (ICCVW)*, pages 214–221.

Fregin, A., Müller, J., Kreβel, U., and Dietmayer, K. (2018). The driveu traffic light dataset: Introduction and comparison with existing datasets. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3376–3383. IEEE.

Gómez, A. E., Alencar, F. A. R., Prado, P. V., Osório, F. S., and Wolf, D. F. (2014). Traffic Lights Detection and State Estimation Using Hidden Markov Models. In *Intelligent Vehicles Symposium (IV)*, pages 750–755.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

Jang, C., Cho, S., Jeong, S., Suhr, J. K., Jung, H. G., and Sunwoo, M. (2017). Traffic light recognition exploiting map and localization at every stage. *Expert Systems with Applications*, 88:290–304.

Jensen, M. B., Nasrollahi, K., and Moeslund, T. B. (2017). Evaluating State-of-the-art Object Detector on Challenging Traffic Light Data. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 882–888.

Jensen, M. B., Philipsen, M. P., Bahnsen, C., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. (2015). Traffic Light Detection at Night: Comparison of a Learning-Based Detector and Three Model-Based Detectors. In *International Symposium on Visual Computing (IVSC)*, pages 774–783.

Jensen, M. B., Philipsen, M. P., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. (2016). Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives. *Transactions on Intelligent Transportation Systems*, 17(7):1800–1815.

John, V., Yoneda, K., Qi, B., Liu, Z., and Mita, S. (2014). Traffic Light Recognition in Varying Illumination using Deep Learning and Saliency Map. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2286–2291.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105.

Li, X., Ma, H., Wang, X., and Zhang, X. (2018). Traffic Light Recognition for Complex Scene With Fusion Detections. *Transactions on Intelligent Transportation Systems*, 19(1):199–208.

Lindner, F., Kressel, U., and Kaelberer, S. (2004). Robust recognition of traffic signals. In *Intelligent Vehicles Symposium (IV)*, pages 49–53.

Mu, G., Xinyu, Z., Deyi, L., Tianlei, Z., and Lifeng, A. (2015). Traffic light detection and recognition for autonomous vehicles. *The Journal of China Universities of Posts and Telecommunications*, 22(1):50–56.

Müller, J. and Dietmayer, K. (2018). Detecting traffic lights by single shot detection. *arXiv preprint arXiv:1805.02523*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Philipsen, M. P., Jensen, M. B., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. (2015). Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2341–2345.

Pon, A. D., Andrienko, O., Harakeh, A., and Waslander, S. L. (2018). A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. *arXiv preprint arXiv:1806.07987*.

Possati, L. C., Guidolini, R., Cardoso, V. B., Berriel, R. F., Paixão, T. M., Badue, C., Souza, A. F. D., and Oliveira-Santos, T. (2019). Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars. *arXiv preprint arXiv:1906.11886*.