Rafael Horimoto de Freitas

Relevant Traffic Light Recognition with Deep Learning Approaches

Vitória, ES

2019

Rafael Horimoto de Freitas

Relevant Traffic Light Recognition with Deep Learning Approaches

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Título de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Supervisor: Prof. Dr. Thiago Oliveira dos Santos

Vitória, ES

2019

Rafael Horimoto de Freitas

Relevant Traffic Light Recognition with Deep Learning Approaches

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Título de Mestre em Informática.

Trabalho aprovado. Vitória, ES, 22 de outubro de 2019:

Prof. Dr. Thiago Oliveira dos Santos
Orientador

Prof^{a}. Dr^{a}. Claudine Santos Badue Gonçalves
Membro interno

Prof. Dr. Patrick Marques Ciarelli
Membro externo

Vitória, ES
2019

Acknowledgements

I would like to acknowledge a master student scholarship supported by Conselho Nacional de Desenvolvimento Científico e Tecnologico (CNPq, Brazil); as well as to thank the NVIDIA Corporation for their donation of GPUs.

Resumo

Carros autônomos têm a importante tarefa de reconhecer o estado (e.g., vermelho, verde, ou amarelo) dos semáforos que são relevantes, i.e., que definem orientação para o carro. Abordagens comuns consistem em usar a imagem capturada de uma câmera prospectiva para detectar semáforos na cena e classificar os estados dos respectivos semáforos. Essas abordagens têm duas limitações principais: (i) além de computacionalmente demorada, detecção geralmente requer anotações caras, como caixas delimitadoras dos objetos alvos; e (ii) ainda há necessidade de um processo de tomada de decisão no qual semáforos relevantes devem ser diferenciados dos demais. Este trabalho trata essas limitações investigando duas abordagens baseadas em aprendizado profundo para reconhecer o estado dos semáforos relevantes: classificação-direta e detecção-com-classificação. Na primeira, ambas as limitações são tratadas treinando o sistema para classificar diretamente o estado dos semáforos relevantes na imagem. Na segunda, o reconhecimento do estado é realizado detectando semáforos na imagem com seus respectivos estados classificados; então a segunda limitação é tratada com heurísticas diferentes para selecionar um exemplo relevante. Além disso, um sistema de regressão profundo com uma nova função de perda resiliente a outliers é proposto para prever as coordenadas de um semáforo relevante no plano da imagem, de modo que uma das heurísticas consiste na seleção da detecção mais próxima dessas coordenadas. Ambas as abordagens foram avaliadas com diferentes conjuntos de dados do mundo real. As conclusões gerais são de que a abordagem de classificação-direta pode alcançar desempenho comparável à detecção-com-classificação com maior número de imagens de treino facilmente anotadas; e que simples heurísticas baseadas em regra têm resultados comparáveis à heurística do sistema de regressão. Além disso, avaliação qualitativa com instâncias desafiadoras revelou que ambas as abordagens têm nível de desempenho semelhante em captar a informação contextual necessária para inferir o semáforo relevante. O sistema de regressão também é avaliado sozinho. Os resultados são promissores e indicam que as coordenadas previstas também podem ser usadas para ajudar um classificador mais barato a trabalhar em uma região de interesse.

Palavras-chave: Reconhecimento de semáforo. Aprendizado Profundo. Carros autônomos. Sistemas avançados de assistência ao motorista.

Abstract

Self-driving cars have the important task of recognizing the state (e.g., red, green, or yellow) of the traffic lights that are relevant, i.e., that define guidance to the car. Common approaches consist of using the image captured from a forward-looking camera to detect traffic lights in the scene and classify the respective traffic lights' states. These approaches have two main limitations: (i) besides computationally time-consuming, detection usually requires expensive annotations, such as target objects' bounding boxes; and (ii) there is still need for a decision-making process in which relevant traffic lights should be distinguished from the others. This work address these limitations by investigating two deep learning-based approaches to recognize the relevant traffic lights' state: direct-classification and detection-with-classification. In the first, both limitations are addressed by training the system to direct classify the state of the relevant traffic lights in the image. In the second, the state recognition is accomplished by detecting traffic lights in the image with their respective states classified; then the second limitation is addressed with different heuristics to select a relevant exemplar. Also, a deep regression system with an novel outliers resilient loss is proposed to predict the coordinates of a relevant traffic light in the image plane, such that one of the heuristics consists in selecting the closest detection to these coordinates. Both approaches were evaluated with different real-world datasets. The overall conclusions are that the direct-classification approach can achieve comparable performance to detection-with-classification with higher number of easily annotated training images; and that simple rule-based heuristics have comparable results to the regression system's heuristic. Additionally, qualitative assessment with challenging instances revealed both approaches have similar performance level on grasping the contextual information required to infer the relevant traffic light. The regression system is also evaluated alone. The results are promising and indicates that the predicted coordinates can also be used to assist a cheaper classifier to work on a region of interest.

keywords: Traffic Light Recognition. Deep Learning. Self-driving cars. Advanced Driver Assistance Systems.

List of Figures

List of Tables

List of abbreviations and acronyms

ADAS	Advanced Driver Assistance Systems
CNN	Convolutional Neural Network
DNN	Deep Neural Network
HoG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
SVM	Support Vector Machines
TLR	Traffic Light Recognition

Contents

1 Introduction

Several tasks that were only performed by humans in the past are being automatized with the current advances on artificial intelligence. With the large number of cars circulating around the world, common driving tasks (e.g., keeping the vehicle in the lane, avoiding collisions, recognizing the traffic lights' state, etc.) turn into very attractive targets of automatic systems.

1.1 Motivation

The more people use visual cues (e.g., signalization) to guide driving, the more valuable are assistive technology capable of visually perceiving and interpreting traffic signals to autonomous cars and advanced driver assistance systems (ADAS). One of the main tasks related to intelligent vehicles is the recognition of traffic lights' state. A more specific and perhaps more directly useful problem is to recognize the state of the relevant traffic lights, i.e., to identify the state of the traffic light (or traffic lights, in case of redundant ones) that define guidance to the car. Although it seems trivial for humans, there are some challenges: the light bulb might be dimming (i.e., not as bright as usual), traffic lights can be partially occluded, a scene may depict several traffic lights in different states, there is not a unique position for them in the scene, etc. In addition, even though traffic lights usually have a well-defined shape, they may vary in different countries.

Several methods have addressed the problem of detecting traffic lights and recognizing their state. Some methods exploit different sensors in their perception systems, but some of these sensors (e.g., SONAR, RADAR, and LIDAR) are not fully capable of identifying the state of a traffic light (FAIRFIELD; URMSON, 2011). Even when the position of a traffic light is known, cameras are still required to classify its state. To tackle this issue, a common approach (PHILIPSEN et al., 2015; JENSEN et al., 2016) leverages the image captured from a forward-looking camera to detect all traffic light instances in the scene and classify the state of each detected traffic light.

Even though this approach has been shown applicable over the years (CHARETTE; NASHASHIBI, 2009b; LEVINSON et al., 2011; JENSEN et al., 2015), it presents two main limitations: (i) the first step (detection) can be computationally costly and usually demands expensive annotations (e.g., bounding-boxes of every traffic light). In this scenario, every traffic light present in the scene must be annotated using the expensive annotation; and (ii) there is still need for a decision-making process in which relevant traffic lights should be distinguished from the others. This critical step is almost always disregarded in literature since publications are focused on detecting/recognizing any traffic light in a scene. However, it is of great importance for the practical use of such systems. Some works in the literature use expensive localization systems to map pre-annotated positions of the traffic lights back to the current image (JOHN et al., 2014; MU et al., 2015; JANG et al., 2017; POSSATI et al., 2019). With this information, the system can correctly choose the relevant traffic light in the scene. Nevertheless, such solution requires a much more complex setup with, for example, maps, annotation of relevant traffic lights per lane, and a good online localization system.

1.2 Proposal

This work address the detection aproach's limitations by investigating two deep learning-based approaches to recognize the state of relevant traffic lights: direct-classification and detection-with-classification.

Both approaches receive as input images taken from a forward-looking camera installed on a vehicle and output, for each image, the state of the relevant traffic light, which can be either *red-or-yellow*, *green*, or *none*. The red and yellow states are grouped together to cope with the lack of yellow samples.

1.2.1 direct-classification

In real-traffic context drivers are not required to locate every traffic light in the field-of-view to further decide whether to proceed or to stop. In fact, the driver attention is somehow "captured" by few traffic lights, usually those considered relevant to the driver's route. Intelligent vehicles could imitate this behavior by learning the recognition task without need of prior traffic lights detection. This perspective presents three main advantages: (i) the whole process becomes less time-consuming, mainly because there is no detection step; (ii) it requires a much cheaper annotation: a single label per image indicating the state of the relevant traffic light instead of bounding boxes; and (iii) there is no need for a further decision-making process to decide which of the detected traffic lights is the one of interest.

In this approach, classification networks were leveraged to recognize their state without any prior detection. The challenge is the end-to-end learning of the state of the relevant traffic lights when each training image is only assigned a simple label indicating its relevant state.

1.2.2 detection-with-classification

This aproach is based on state-of-the-art detection networks to locate traffic lights, and different strategies were evaluated to select a relevant traffic light among the detections (if any). Also, a deep regression system with a novel outliers resilient regression loss is proposed to predict the coordinates of a relevant traffic light in the image plane, such that one of the heuristics consists in selecting the closest detection to these coordinates. This approach requires extensive bounding box annotation to enable the detection of traffic lights.

1.3 Contributions

This work is mainly composed by two papers' contributions: A paper submitted to Computer Vision and Image Understanding jounal, that is being reviwed, and a paper accepted on XVI Encontro Nacional de Inteligência Artificial e Computacional conference.

The contribution include two local datasets (IARA-TL and VITÓRIA) and; the classification annotation of these datasets and also of two publicly available and well-known data sets: LISA (JENSEN et al., 2016; PHILIPSEN et al., 2015) and LaRA (CHARETTE; NASHASHIBI, 2009b; CHARETTE; NASHASHIBI, 2009a; PARISTECH, 2015); a modification of ResNet-50 architecture (HE et al., 2016) for regression and a novel outliers resilient regression loss; and the evaluation results of the investigated aproaches on IARA-TL, VITÓRIA, LaRA and LISA.

The overall conclusions are that the classification-based system can achieve comparable performance to the detection-based system with higher number of training images; and that simple rule-based heuristics have comparable results to the regression system's heuristic. Nevertheless, annotating images for detection is significantly more expensive, and such models usually require more processing time to be trained and evaluated. Additionally, qualitative assessment with challenging instances revealed both approaches have similar performance level on grasping the contextual information required to infer the relevant traffic light. The regression system is also evaluated alone. The results are promising and indicates that the predicted coordinates can also be used to assist a cheaper classifier to work on a region of interest.

1.4 Structure

The rest of this document is organized as following: Sect. 2 presents fundamental concepts and the related work; in Sect. 3 the systems are proposed; Sect. 4 defines the experimental methodology adopted; Sect. 5 shows the results and discuss then; and Sect. 6 shows the conclusions and future work.

2 Theoretical background

2.1 Convolution Neural Networks

Convolutional Neural Network (CNN) is a type of Deep Neural Network (DNN) that presents convolutional layers. Convolutional layers have as inputs the pixel values of an image of two spatial dimensions and one channels dimension and as output (before aplying the activation function) pixel values of an image whose each channel is the result of applying a convolution filter over the input image. This way, the convolutional layer's connections are a tiny subset of the connections of a fully connected layer between the input and the output, with still an extra constraint: the neurons that output values from the same output's channel share the same parameters, the values of the respective convolution filter, but connecting with different neurons of the input, i.e., aplying the filter in different positions of the input image.

Convolutional layers have as advantages: (i) being less prone to overfitting than a much more parameterized fully connected layers; (ii) they generalize features spatially, since the same filters are applyied over different positions; and (iii) they can learn known important filters that extract features related to close pixels relations, like edges, without the need to manually define these filters.

2.1.1 Classification

CNNs can be used for solving classification problems by defining on the last layer one output logit value for each class of the problem that is desired to be the greater possible when feeding the network with images of the correspondent class, and the lower possible otherwise. For reaching this goal, it is defined a loss function that measures how distant the logits values are from the goal with respect to the ground truth class. Commonly this loss is the cross entropy applyied over a softmax of the logits. With the loss value its possible to calculate the gradients of the network's parameters with the backpropagation algorithm. And with the gradients its possible to apply training algorithms like stochastic gradient descent, that will hopefully bring the network closse to the desirable behavior.

SqueezeNet (IANDOLA et al., 2016) (v1.1) and ResNet-50 (HE et al., 2016) are examples of classification CNNs used in this work. SqueezeNet has an architecture that aims to have a very low number of parameters while still having good performance. ResNet-50 has an relativelly deep architecture which is more prone to learn identity functions, due to it residual connections, wich helps preventing overfitting by allowing less complex features to propagate it values on deeper layes.

2.1.2 Regression

Alternativelly, CNNs can be used for solving regression problems by defining on the last layer outputs that represent the regressed values and a loss that measures how much that values are distant from the ground truth values.

An example of a modified ResNet-50 architecture (HE et al., 2016) for regression and appropriate loss are proposed in this work.

2.1.3 Detection

Detection CNNs are more complex arquitectures that involves different strategies to infer a variable number of spatial delimitations in the image, if any, and possible classificating them. Faster R-CNN (REN et al., 2015) and YOLOv3 (REDMON; FARHADI, 2018) are example of detection CNNs used in this work that infer bounding-boxes, defined by four regressed coordinates, all associated with an classification.

YOLOv3 is simplier and faster than Faster R-CNN, but has lower perfomance level.

2.2 Related work

The Traffic Light Recognition (TLR) problem is important to provide a smooth driving experience and safety, but is challenging since they appear as tiny light spots on complex landscape. In a recent survey (JENSEN et al., 2016), the authors illustrated a general vision-based TLR pipeline in three main stages: detection, (state) classification, and tracking. While detection intends to locate possible targets, in classification each detected traffic light is assigned a unique label (e.g., red, green, or yellow) with the purpose of identifying its state. A posterior tracking procedure can be applied to improve temporal coherence, handle occluded traffic lights, etc, but it is out of the scope of this work.

The related works usually include a detection procedure mainly to reduce the processing time of further classification stage. However, the detection itself may be computationally costly to ensure real-time performance when using conventional image processing or feature extraction procedures (GONG et al., 2010; DIAZ-CABRERA; CERRI; SANCHEZ-MEDINA, 2012; MU et al., 2015; BARNES; MADDERN; POSNER, 2015; JENSEN et al., 2015). Also, approaches to locate the relevant traffic lights very often use prior location and/or maps information (JOHN et al., 2014; MU et al., 2015; JANG et al., 2017; POSSATI et al., 2019) whose availability depends on expensive annotation processes. To a better view of the TLR literature, detection and state classification are briefly discussed.

Detection can be categorized as model or learning-based (JENSEN et al., 2016). Model-based approaches rely on color (GONG et al., 2010; DIAZ-CABRERA; CERRI; SANCHEZ-MEDINA, 2012; LI et al., 2018), shape (CHIANG et al., 2011; GóMEZ et al., 2014), and/or structural (CHARETTE; NASHASHIBI, 2009a; CHARETTE; NASHASHIBI, 2009b; DIAZ-CABRERA; PIETROCERRI; PAOLOMEDICI, 2015) assumptions about the traffic lights appearance (e.g., lamp colors, bulbs shape, bulbs arrangement on the black box). Since these assumptions rarely hold in real-world scenes, literature has turned to learn traffic lights from large amounts of data in a variety of situations. Most of the learning-based approaches combine popular hand-crafted features with well-known classifiers. The works (LINDNER; KRESSEL; KAELBERER, 2004) and (FRANKE et al., 2013) used Haar-like features together with a cascade of classifiers, whereas (MU et al., 2015; BARNES; MADDERN; POSNER, 2015; JENSEN et al., 2015) applied Histogram of Oriented Gradients (HoG) with Support Vector Machines (SVM).

Hand-crafted features have also been widely used to classify the state of traffic lights. The works (GONG et al., 2010; KIM; PARK; JUNG, 2011) used Haar-like features with cascade classifiers to determine the state of previously detected traffic lights, whereas (CAI; LI; GU, 2012) applied a simple nearest neighbor approach on Gabor features. In (CHIANG et al., 2011), the authors adopted Local Binary Patterns (LBP) and SVM for state classification. The use SVMs with HoG features was also explored in TLR (JANG et al., 2014; MU et al., 2015; MICHAEL; SCHLIPSING, 2015), including the recent work of (LIU et al., 2017), which combines HoG with LBP features.

Despite the relative success of hand-crafted features, late remarkable advances in deep learning have encouraged its application in TLR, particularly with the use of Convolutional Neural Networks (CNNs). In 2014, (JOHN et al., 2014) used CNNs to classify traffic lights detected by means of GPS-based information and some image processing. In (SAINI et al., 2017), the authors adapted (and reduced) the Lecun's CNN for digits recognition in order to classify the state of traffic lights. Nevertheless, traffic lights are detected by applying color segmentation and HoG/SVM-based detection. In (WANG; ZHOU, 2018), the authors adapted the CaffeNet for state classification. Their system, however, demands an even more expensive annotation, which includes competitors objects (e.g., luminous artifacts or vehicle lights) and specific traffic lights type (e.g., vertical/horizontal, three/four bulbs). Detection relies on color-based saliency map computed over low-exposure image.

Deep neural networks have been also used to detect the traffic lights before the state classification. In (WEBER; WOLF; ZÖLLNER, 2016), the authors adapted the AlexNet to detect and classify the state of traffic lights. However, their system is not end-to-end since a post-processing stage is performed outside the network to keep a single region proposal for each traffic light. Following a similar idea, the work (BACH; REUTER; DIETMAYER, 2017) proposed a CNN model on top of the GoogLeNet for detection and state classification. They focus on fusing data in a multi-camera setup for long-range traffic light detection. With a different architecture, the authors, in (BEHRENDT; NOVAK; BOTROS, 2017), customized YOLO in order to locate traffic lights. State classification (based on a custom CNN) and tracking can be enabled to improve the detection. A more comprehensive investigation on YOLO-based traffic light detection can be seen in (JENSEN; NASROLLAHI; MOESLUND, 2017). More recently, in (PON et al., 2018), the authors modified the Faster R-CNN architecture to implicitly explore the object hierarchy in the detection of traffic lights and signs and, in (MÜLLER; DIETMAYER, 2018), the authors extended the Single Shot Detector to also classify the state of traffic lights.

In addition to these works, there are also challenges and benchmarks for the problems of traffic light detection and recognition. The Vision for Intelligent Vehicles and Applications (VIVA) Challenge, for instance, comprises several benchmarks including the VIVA Traffic Light Detection Challenge (INTELLIGENT; DIEGO, 2015). In our work, one of the datasets of the experimentation came from this challenge: the LISA dataset. There are some initiatives from the industry side as well. In the 2016 Nexar Challenge (NEXAR, 2016), for instance, competitors had to develop a model to recognize the state of the traffic light in the car driving direction (i.e., the relevant traffic light). The goal of this challenge is closely related to ours, but they also wanted to have a small model size in addition to high accuracy. A downside of this challenge, however, is that the Nexar dataset, studies and the proposed works are not publicly available due to industry privacy.

Although recent works have demonstrated the tendency of using deep learning to solve TLR, literature still lacks a more comprehensive evaluation on different publicly available datasets. In addition, the deep learning-based methods do not address the state classification of the relevant traffic light. Therefore, no baseline performance is currently available for this task. Finally, deep learning detection methods require bounding-box annotation of the traffic lights. In other words, there is no investigation on state classification based on weak annotation (i.e., image-level labels). These three limitations are addressed in this work.

3 Proposed system

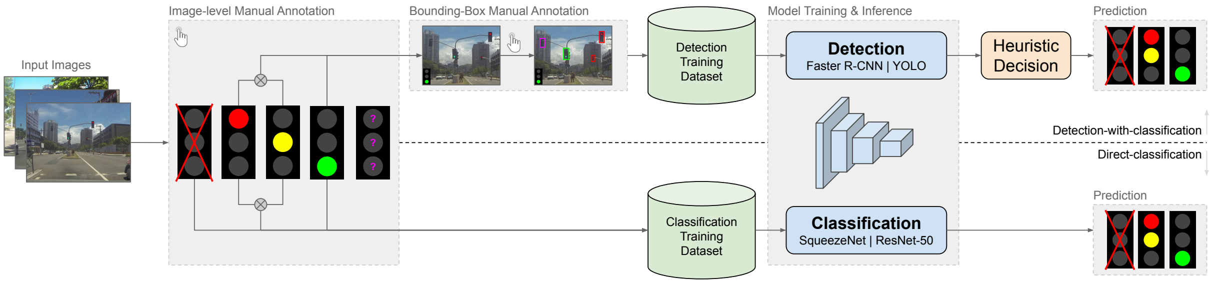
Input Images
Image-level Manual Annotation
?
?
?
Bounding-Box Manual Annotation
Detection Training Dataset
Classification Training Dataset
Model Training & Inference
Detection
Faster R-CNN | YOLO
Classification
SqueezeNet | ResNet-50
Heuristic Decision
Prediction
Detection-with-classification
Direct-classification
Prediction

Figure 1 – Overview of the proposed system. First, the training images are undergone manual labeling in which each image is assigned either *red-or-yellow*, *green*, *none*, or *unknown*. While *none* indicates the absence of traffic lights, *unknown* implies the state of the relevant traffic lights could not be determined. Images with this label are discarded, and a subset of the images labeled as *red-or-yellow* or *green* are used to annotate bounding boxes enclosing traffic lights. Two approaches were investigated. In the detection-with-classification approach (top-flow), a deep network is trained to detect all traffic lights in a scene. A heuristic decision is further performed to choose the relevant traffic light among the detections. Direct-classification (bottom-flow), in turn, leverages a classification network to directly predict the relevant state of the scene. The training labels (*red-or-yellow*, *green*, *none*) match the possible network predictions.

This work investigates two CNN-based approaches for state recognition of relevant traffic lights in a scene (Figure 1): (i) direct-classification (bottom-flow) and (ii) detection-with-classification (top-flow).

In the first case, were investigated how direct-classification networks addresses the relevant state recognition problem when trained with just a single label per scene, i.e., *red-or-yellow*, *green*, or *none*. This approach eliminates the need for detection and address the problem directly. In other words, the perception of what/where is a relevant traffic light should be solved only by the CNN-learned filters instead of rule-based heuristics.

In the second case, a detection network propose regions (bounding boxes) of the input image containing traffic lights and predict (classify) their respective states. Then, a heuristic is employed to choose the relevant traffic light among the detections based on positioning assumptions. The selected traffic light's label is taken as the system output. A particular drawback of this approach is the tedious and expensive process of annotating traffic lights' bounding boxes. Also, the success of the task strongly depends on the performance of the detection and coherence of the heuristic.

The next sections describe the manual annotation process and the approaches investigated in this work.

3.1 Manual annotation for classification and detection networks

The training of the proposed system leverages a proprietary collection comprising several traffic scenes with size 640×480 captured from a fixed forward-looking camera on top of a car. These images were manually labeled in two steps. First, each image was labeled (image-level annotation) as either *none* (i.e., no traffic lights), *red-or-yellow*, *green*, or *unknown*, the latter indicating broken traffic lights, or situations where the annotators were not able to clearly decide the traffic light state (e.g., discordant traffic lights associated to the same lane). Images labeled as *unknown* were discarded for further stages. Red and yellow traffic lights were grouped together into a single class due to the scarcity of yellow samples in the dataset, as usually observed in traffic lights datasets. Although this direction differs from some works in the literature (WEBER; WOLF; ZÖLLNER, 2016; NEXAR, 2016) that restrict themselves to recognize only red and green states, it is more appropriate for the traffic rules that usually requires the cars to reduce speed and stop with the yellow light. The transition case from green to yellow, if necessary (e.g., when car is already near to the traffic light), could be treated with temporal information (not focus of this work).

In a second level, traffic lights' bounding boxes and their respective states were annotated for a subset of the remaining images not labeled as *unknown*. This annotation is required only for the detection-with-classification approach, and is notably more expensive in terms of time and effort than the image-level annotation because a single image may contain multiple traffic lights. Each traffic light object also received one of the labels *red-or-yellow*, *green*, or *unknown*, and those labeled as *unknown* were discarded before the training begins.

3.2 Direct-classification approach

Although detection networks work well in locating traffic lights, it does not consider holistic information and cannot determine the relevant traffic light. To circumvent this problem, were investigated whether the recognition task can be solved end-to-end as a multi-class classification problem. In this context, two well-known classification CNNs were evaluated: (i) SqueezeNet (IANDOLA et al., 2016) (v1.1) and (ii) ResNet-50 (HE et al., 2016). Briefly, SqueezeNet was chosen mainly due to it compactness, and ResNet CNN family is one of the state-of-the-start architectures in classification tasks.

3.3 Detection-with-classification approach

This approach works by first detecting the regions of interest in the scene modeled as traffic lights' bounding boxes. To this purpose, two state-of-the-art object detection networks were investigated: (i) Faster R-CNN (REN et al., 2015) and (ii) YOLOv3 (REDMON; FARHADI, 2018). These networks were trained to detect traffic lights in the wild, where detecting means jointly regressing the objects' location (bounding box) and predicting their respective class (state) rather than sequential processing, as in (BEHRENDT; NOVAK; BOTROS, 2017). In this context, deciding a relevant traffic light naturally implies determining the system output. The decision is made heuristically based on traffic lights positions and/or dimensions properties.

3.3.1 Heuristics for deciding the relevant traffic lights

After the training, a detection network can be used to predict traffic lights in a scene, but this is not enough. Among all the traffic lights, there is usually one (or more) that is actually relevant for the driver, one that defines his/her instant decision (e.g., slowing the speed, start to move). Ambiguous cases (even for humans) may arise when different decisions are possible for the same lane. This type of decision would require information of the planned path, but this is beyond the scope of this work.

In this work, five heuristics to select a traffic light were investigated: (i) select the closest traffic light to the top-center point (1H); (ii) select the largest traffic light (1L); (iii) apply 1H after selecting the two largest traffic lights (1H-2L); (iv) select the closest traffic light to the coordinates outputted by a regression system, proposed in the next section (1C); and (v) select randomly a traffic light (1R). The last criteria will be used as a baseline performance for the detection-with-classification approach, as further described in the experiments. Figure 2 illustrates the decisions made by the different heuristics.

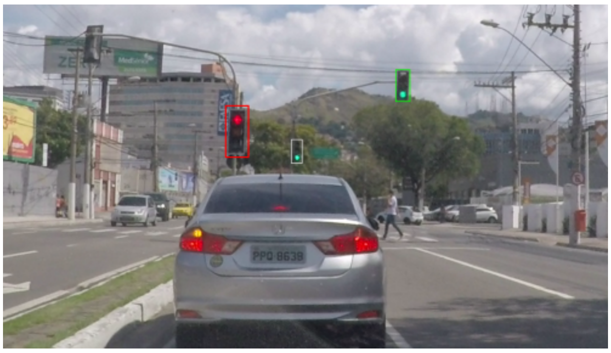

Figure 2 – When applying the heuristics in the image shown, the green box would be selected for 1H and 1H-2L, but the red box would be selected for 1L. The white box could only be selected by 1C or (randomly) by 1R.

3.3.2 1C heuristic's regression system

1C heuristic requires a deep regression model to predict the 2-d coordinates (a single point per image) of a relevant traffic light in the image plane. 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 to the top-center point (thereafter referred to as target).

3.3.2.1 Relevant Traffic Light Localization

The pipeline of the proposed regression system (illustrated in Figure 3) is broadly divided into the learning and test stages. The learning stage (left part of Figure 3) 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.

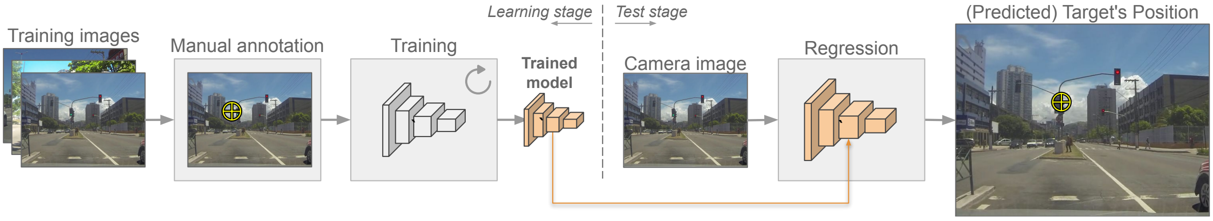

Figure 3 – Overview of the proposed method for relevant traffic light localization. The yellow circle with a cross marks the position of the relevant traffic light in the image.

3.3.2.2 Deep Regression Model

The regression model (illustrated in Figure 4) 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:

$$\hat{x}_t = [(\hat{x}_m + 1)/2]w \tag{3.1}$$

$$\hat{y}_t = \hat{y}_m h. \tag{3.2}$$

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.

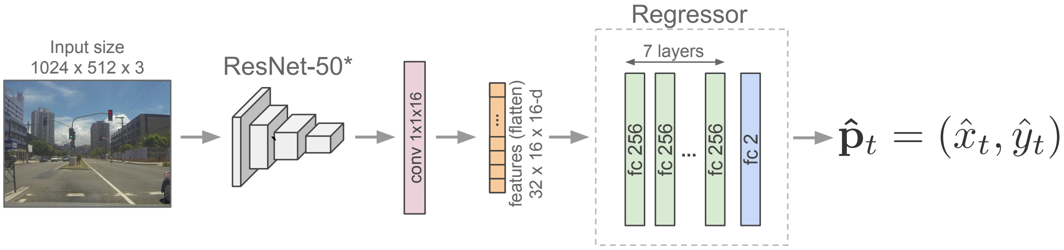

Figure 4 – Deep regression model.

3.3.2.3 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

$$\mathcal{L}(z) = \begin{cases} z^2, & \text{if } z \leq 1 \\ log(z^2) + 1, & \text{otherwise.} \end{cases} \tag{3.3}$$

The function in (3.3) 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 5 together with 2×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 $z > 1$.

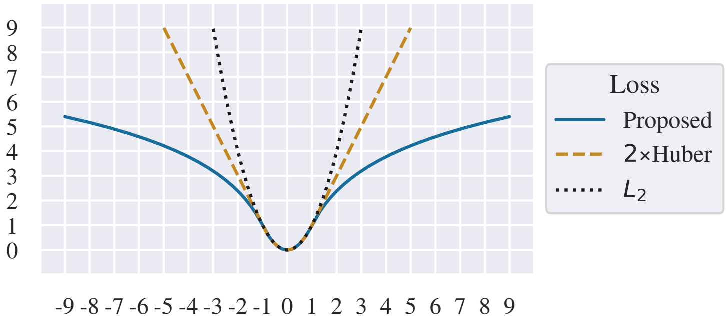

Figure 5 – The proposed loss function together with 2×Huber and L_2 losses.

4 Experimental methodology

Table 1 – Class distribution across datasets. All images from VITÓRIA are hard-to-decide.

	full				hard-to-decide			
	none	red	green	total	none	red	green	total
PROP-TRAIN	5,052	3,091	13,672	21,815	-	-	-	-
PROP-VAL	599	406	2,222	3,227	-	-	-	-
PROP-BBOX-TRAIN	5,052	1,286	5,918	12,256	-	-	-	-
PROP-BBOX-VAL	599	383	2,099	3,081	-	-	-	-
IARA-TL	2,954	3,904	3,695	10,553	5	0	138	143
LaRA	2,149	4,208	3,208	9,565	83	481	597	1,161
LISA-DayTest	328	1,330	1,396	3,054	0	0	0	0
LISA-DayTrain	618	9,628	2,426	12,672	0	3,097	295	3,392
VITÓRIA	-	-	-	-	3,548	19,604	8,717	31,869

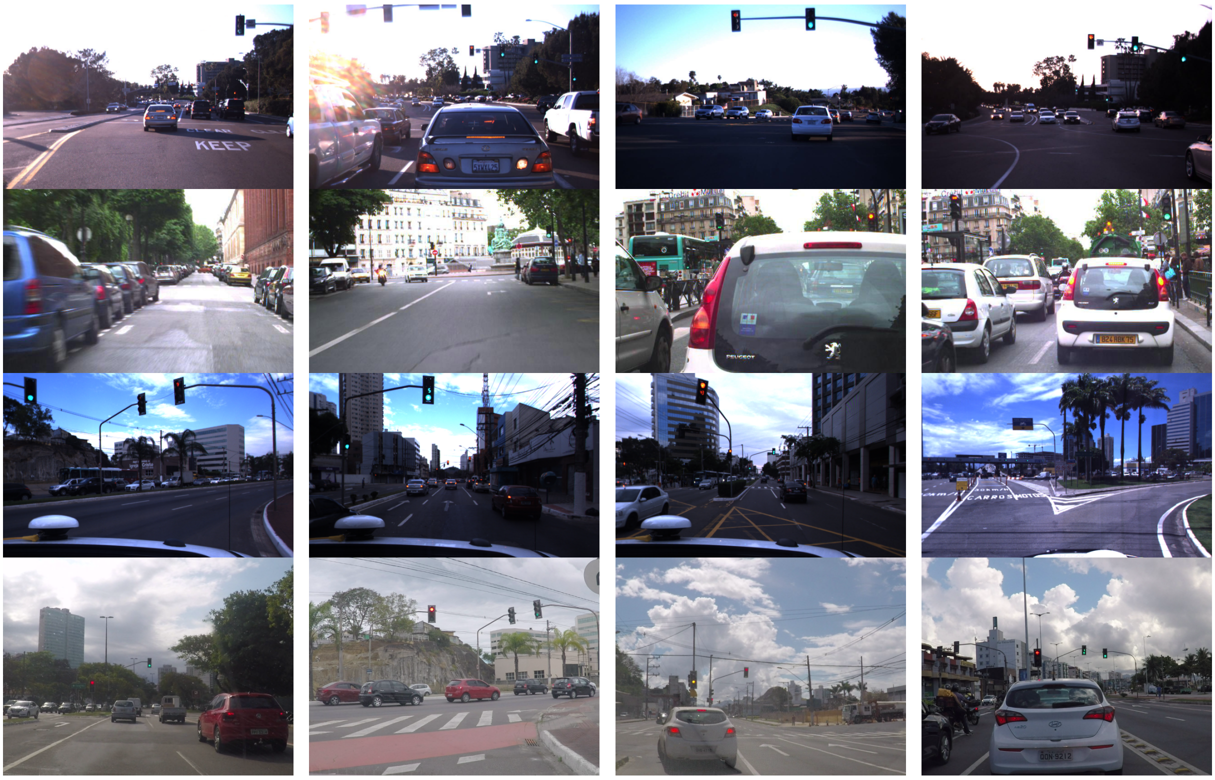

Figure 6 – Sample images of evaluation datasets. The rows, from top to bottom, represent the datasets LISA, LaRA, IARA-TL, and VITÓRIA, respectively.

This section details the experimental methodology for evaluation of the proposed system under the detection-with- and direct- classification approaches. The following topics are addressed: datasets, performance metric of the recognition system, the experiments themselves, and, finally, the hardware/software setup.

4.1 Datasets

The training and validation of the detection and classifiction networks were conducted on a proprietary dataset, whereas four collections were used to evaluate the systems' performance: LaRA and LISA, which are publicly available benchmarks commonly used in the intelligent vehicle literature; IARA-TL refers to a local dataset produced with the IARA (Intelligent Autonomous Robotic Automobile) platform, an autonomous vehicle developed by our research group; and VITÓRIA, a local dataset comprising video sequences captured with a GoPro camera mounted in the windshield of a regular vehicle focusing on recording challenging instances for recognition. Also DTLD, another publicly available benchmark, is used for training and testing the regression network for the 1C heuristic. The IARA-TL and VITÓRIA datasets are contributions of this work. All test images were annotated following the same protocol described in Sect. 3.1. Table 1 presents the class distribution across datasets, whereas Figure 6 depicts samples of the evaluation datasets.

4.1.1 PROPRIETARY

The detection and classification networks were trained using only the PROPRIETARY dataset. The images were acquired by a camera mounted on the roof of a car in a resolution of 640×480 pixels in RGB across 24 Brazilian cities during day-time. Unlike the datasets used in the evaluation, this one is not sequential, presenting a larger variety of scenes and traffic light objects.

The total of 25,042 images was partitioned into training (PROP-TRAIN) and validation (PROP-VAL) sets. The partitioning was performed randomly and location-wise, which means that instead of arbitrarily choosing individual samples for each set, it was ensured that all images from the same location (city, or region, in case of too big cities) were placed in the same set. In addition to the image-level annotation, 15,337 images ($\approx$61% of the entire set) from PROP-TRAIN and PROP-VAL received additional annotation of traffic lights' bounding boxes and their respective states to enable the training of the detection networks, giving rise to PROP-BBOX-TRAIN and PROP-BBOX-VAL (Table 1). Despite the reduced quantity, the annotation of these images requires much effort since it is performed at object level instead of image level, and due to the images with multiple (traffic lights) objects. The positioning distribution of the traffic lights can be seen in Figure 7.

4.1.2 LISA

The LISA Traffic Light dataset (JENSEN et al., 2016; PHILIPSEN et al., 2015) was made available as a benchmark for the VIVA challenge (INTELLIGENT; DIEGO, 2015). The video sequences were acquired in San Diego, USA, with a stereo camera centered on the vehicle roof, of which the left camera was considered for frame extraction. Video was captured at 16 frames

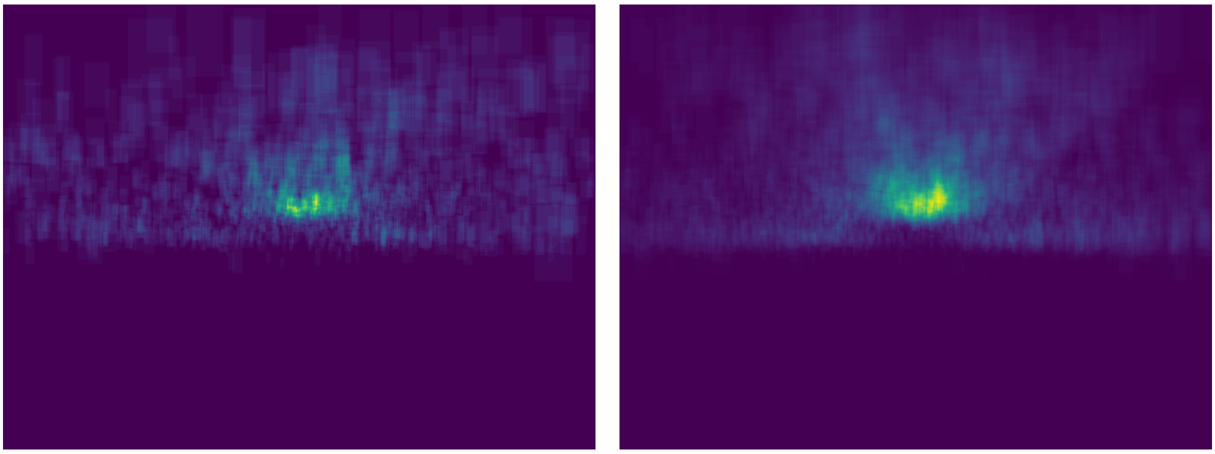

Figure 7 – Distribution of the (a) red-or-yellow traffic lights, and (b) green traffic lights in the training dataset (PROPRIETARY). The distributions were normalized separately.

per second (FPS), and uncompressed in a total of 43,007 images with 1280×960 pixels in RGB format. The whole dataset was originally split into day- and night-time generated data, and these sets were individually partitioned into train and test sets. For this work, the images were scaled down to 640×480 pixels, being only the sets recorded during the day (i.e., LISA-DayTrain and LISA-DayTest) considered for experimental analysis.

4.1.3 LaRA

LaRA (CHARETTE; NASHASHIBI, 2009b; CHARETTE; NASHASHIBI, 2009a; PARISTECH, 2015) was generated from video sequences of the urban traffic in Paris, France. The acquisition was performed at 25 FPS with a single camera inside the car, on the back of the rear-view mirror. Uncompressed data comprise 11,179 RGB images with 640×480 pixels. Unlike LISA, no train/test partition is officially provided, therefore the whole dataset was used during evaluation.

4.1.4 IARA-TL

As mentioned, IARA-TL is one of the two novel datasets that our research group – High Performance Computing Lab (LCAD[1]) – produced for traffic-related applications. This local collection comprises day-time traffic scenes recorded in the city of Vitória, Brazil, being its name borrowed from the IARA autonomous vehicle that has been developed by LCAD (BADUE et al., 2019). This car has a Bumblebee XB3 stereo camera on top of it that captures images in RGB format with 1280×960 pixels. Only the right camera was considered for frame extraction, and all frames were scaled down to 640×480 pixels. The uncompressed data comprise a total of 10,329 labeled frames, of which only traffic lights with at least six pixels wide (criterion similar to the adopted in the LaRA dataset) were considered for labeling.

[1] A research group from the Universidade Federal Espírito Santo (UFES).

4.1.5 VITÓRIA

The VITÓRIA dataset was specifically produced for this work and consists in day-time video sequences of the usual traffic of the city of Vitória, Brazil. This dataset focuses on hard-to-decide cases, which are defined as scenes with one or more non-redundant (i.e., different state) traffic lights close to the relevant traffic light. The driver was given a set of pre-defined routes where hard-to-decide instances are likely to be found. A GoPro camera was mounted on the windshield of a regular car and the acquisition was performed at 29.97 FPS (RGB format), producing 1920×1200-size images. These images were later cropped (160 pixels from each side, changing the ratio from 16:10 to 4:3) and resized to 640×480 pixels. Unlike the other test sets, VITÓRIA only keeps those scenes which are considered hard-to-decide cases.

4.1.6 DTLD

The DriveU Traffic Light Dataset (DTLD) (FREGIN et al., 2018), the largest publicly available dataset of traffic lights, 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 Sect. 3.3.2.1, 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. The original DTLD train and test splits were leveraged, 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.

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.

4.1.6.1 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 (4.1) 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.

$$f_{lum}(x) = \begin{cases} 0.5, & \text{if } x > 128 \\ 64/x, & \text{if } 64 \leq x \leq 128 \\ 1.0, & \text{otherwise.} \end{cases} , \tag{4.1}$$

4.1.6.2 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.

4.1.6.3 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.

4.1.6.4 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.

4.2 Performance metrics

The performance of the direct-classification and detection-with-classification systems was measured in terms of the macro average of the accuracy (or simply macro-accuracy). The macro average was used to account for the unbalance of the test sets. The macro-accuracy metric for a multi-class problem is defined as

$$\text{ACC}_{macro} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_{c_i}}{\text{P}_{c_i}}, \tag{4.2}$$

where $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ is the set of classes; TP_{c_i} and P_{c_i} denote the number of True Positives and Positives (from ground-truth) of the i-th class, respectively. In our case $\mathcal{C} = \{$*none*, *red-or-yellow*, *green*$\}$, thus, $N = 3$.

For the regression system alone, 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.

4.3 Experiments

Quantitative experiments were conducted to verify whether the direct-classification approach can achieve comparable performance to detection-with-classification when analyzed in their best configurations. In addition, qualitative experiments performed on challenging instances were carried out for better understanding the approaches' behavior. Also, experiments are done to evaluate the regression system alone and to compare the proposed loss with the Huber loss.

4.3.1 Quantitative experiments

For detection-with-classification assessment, the deep detectors Faster R-CNN and YOLOv3 were combined with the heuristics for relevant traffic light selection (i.e., 1H, 1L, 1H-2L, 1C, 1R), introduced in Sect. 3.3. The training was conducted on PROP-BBOX-TRAIN using PROP-BBOX-VAL as validation, and evaluation on IARA-TL, LISA-DayTest, LISA-DayTrain, and LaRA. To account for non-determinism, the training-test cycle was repeated 10 times with different seeds, and the resulting macro-accuracy was recorded.

In the direct-classification approach, both SqueezeNet and ResNet-50 were trained in two different ways. In the first, the models were trained on PROP-TRAIN, whereas, in the second, the training was restricted to PROP-BBOX-TRAIN only. While the latter aims to compare the different approaches using the same amount of images, the former allows to verify the impact of additional data on the direct-classification's performance. The evaluation followed the same protocol of the detection-with-classification approach described in the previous paragraph.

4.3.2 Qualitative experiment

This experiment focuses on qualitatively evaluating the system's behavior (under the two recognition approaches) on the hard-to-decide samples. In particular, this experiment provides insights of what the networks are "seeing" by observing the detection and selection (with different heuristics) of the relevant traffic light obtained with Faster R-CNN, as well as by analyzing the activation maps of the SqueezeNet's last convolutional layer. The models already trained for the quantitative experiments were leveraged for qualitative evaluation. The Faster R-CNN and SqueezeNet architectures were used because they achieved the best performance in their respective approaches, as discussed in the next section. A video summarizing the qualitative results can be found at https://www.dropbox.com/sh/dhjqjq3bsj9h5ug/AADjSD2oYqfQzxpE4aj5jAe1a?dl=0.

4.3.3 Regression system alone experiments

The conducted experiments aim to assess the ability of the proposed method in selecting relevant traffic lights, when used in conjunction with an ideal detector (i.e., the ground-truth bounding boxes), 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 (Sect. 3.3.2.3) and with the Huber loss as a performance baseline. A single training-test section was conducted for each loss. 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.

A video summarizing the qualitative results of the system trained with the proposed loss can be found at https://www.dropbox.com/sh/dhjqjq3bsj9h5ug/AADjSD2oYqfQzxpE4aj5jAe1a?dl=0.

4.3.4 Training the detection networks

The detection models (Faster R-CNN (REN et al., 2015) and YOLOv3 (REDMON; FARHADI, 2018)) should learn to locate traffic lights and recognize their states. This step relies on the previously annotated data, which consists of bounding boxes and traffic lights' states. The training procedure follows the same protocol described in their corresponding original works.

4.3.5 Training the classification networks

SqueezeNet and ResNet followed a similar training scheme where only image-level annotated data were used. These images were kept at their original size of 640×480, which is larger than the input size of the pre-trained models on ImageNet (224×224). Since SqueezeNet is fully-convolutional, the larger input size does not increase the number of the network parameters, but it implies more processing due to the larger feature maps. The same holds for ResNet because it applies a global average pooling operation right before the fully connected layer.

Moreover, an off-line data augmentation process was performed to balance the number of samples per class and increase the variability of the images. Basically, the augmentation consists in applying horizontal flipping and pixel-wise arithmetic operations (addition and multiplication) on randomly selected images in order to increase the number of samples of the two classes. The models were initialized with pre-trained weights on ImageNet, except the last convolutional layer of Squeezenet, and last (only) fully connected layer of ResNet, whose number of filters/neurons was reduced to match the number of classes of the new task (i.e., *red-or-yellow*, *green*, *none*). In these cases, the weights were initialized using the Xavier (GLOROT; BENGIO, 2010) algorithm. The models were trained with mini-batches of 15 images (5 images per class) during 8 epochs.

The learning rate was initially set to 2^{-9} and 2^{-16} for SqueezeNet and ResNet, respectively, and was further decreased to half of its current value every two epochs. The traditional stochastic gradient descent (SGD) algorithm was used for SqueezeNet, while Adam (KINGMA; BA, 2014) was adopted for ResNet. Every half epoch the current model was evaluated on a validation set, and the best model was saved for inferences.

4.3.6 Training the regression networks

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; SUTSKEVER; HINTON, 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.

4.4 Experimental platform

When measuring inference times, only the time for the network forward was considered, thus ignoring image loading burden.

4.4.1 Detection and classification networks

The experiments were conducted on an Intel® Core™ i7-4770 PC with 16 GB of RAM and a Titan Xp GPU. The original Darknet[2] implementation was used for YOLOv3, while a Tensorflow implementation[3] was adopted for Faster R-CNN.

4.4.2 Regression network

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

[2] <https://github.com/pjreddie/darknet>

[3] <https://github.com/endernewton/tf-faster-rcnn>

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).

5 Results and discussion

This section starts by discussing separately the results obtained with the detection-with-classification and direct-classification approaches. Next, a comparative analysis is done in order to verify the feasibility of performing recognition relying only on classification models. Next, the results on hard-to-decide scenarios are described. Next, the results of the regression system alone are showed in different conditions comparing the proposed loss with the Huber loss; and finally results in some scenarios are described.

5.1 Detection-with-classification quantitative results

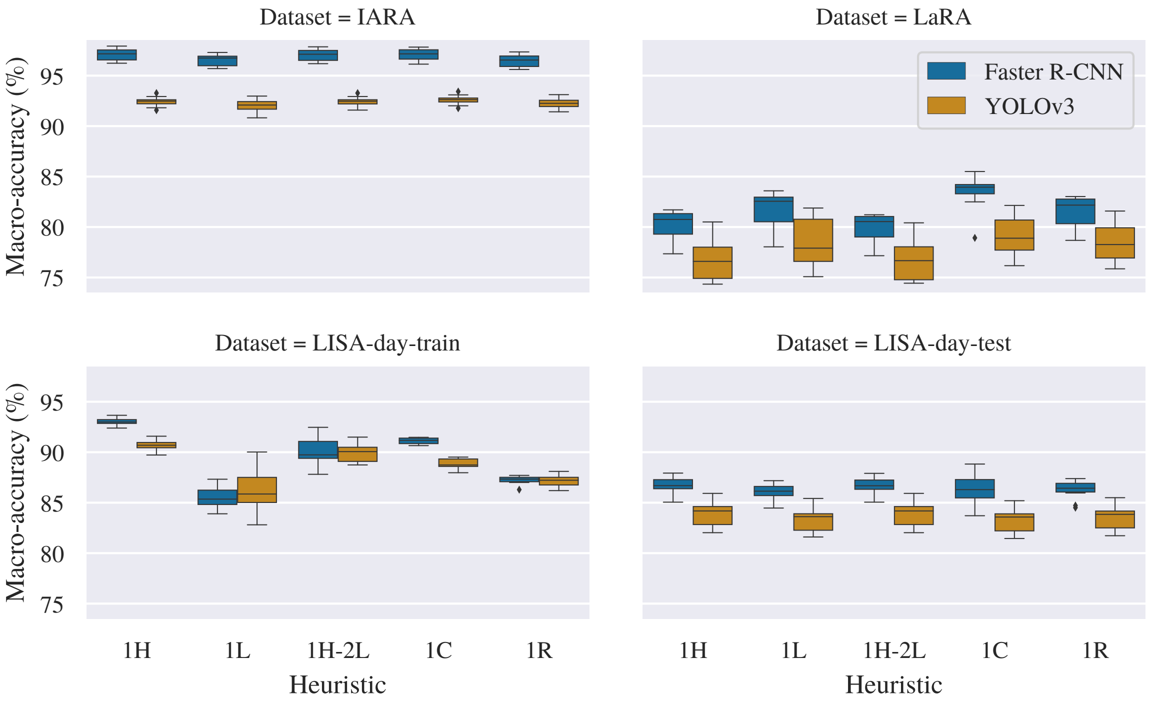

Figure 8 – Performance of the detection-with-classification approach based on 10 runs.

The results for the detection-with-classification approach were arranged in the four box-plots (one per dataset) shown in Figure 8. Each box represents a detection model (i.e., Faster R-CNN and YOLOv3), and each pair of boxes are associated with a heuristic (i.e., 1H, 1L, 1H-2L, 1C, 1R). Results account for the statistics of 10 runs in terms of the macro-accuracy.

Clearly, Faster R-CNN consistently outperformed YOLOv3 for all datasets. In a total of 20 dataset/heuristic combinations, Faster R-CNN achieved higher accuracy than YOLOv3 in 17 cases. For the remaining three cases, which are related to the LISA-DayTrain dataset, the difference was not significant.

Reducing the analysis to the Faster R-CNN model, the heuristic 1H achieved the highest average macro-accuracy on IARA-TL, LISA-DayTrain, and LISA-DayTest, however only on LISA-DayTrain the difference in performance was significant. LaRA was the most challenging dataset for the task of the relevant traffic light recognition since it yielded the lowest accuracy for both models, as showed in Figure 8. For this dataset, Faster R-CNN performed better with the 1C heuristic, but with no significant difference to the random selection strategy (1R), the baseline result. Moreover, the heuristics relying on traffic light position (i.e., 1H and 1H-2L) yielded the poorest results for LaRA. This fact can be explained by the frequent occlusion of the top-center pixels of LaRA's images. In this context, the redundant traffic lights (usually located on the side of the road) are barely selected as the relevant one since they are far from the top-center position.

The overall results also showed that the random heuristic (1R) performed similar to the best-performed heuristic of each dataset, except for LISA-DayTrain. By analyzing Figure 8, it is difficult to notice any improvement from the heuristic 1H over 1R for the datasets IARA-TL and LISA-DayTest. For a better evaluation, the system performance were verified for instances where the heuristics' influence is emphasized over the detectors'. This was accomplished by restricting the test to images that contain any traffic light (based on the ground-truth annotation) for which Faster R-CNN detected at least three bounding boxes. In this context, the difference in average accuracy between 1H and 1R, respectively, increases from 0.62 to 4.72 percentage points (pp) on IARA-TL, from -1.27 (the negative value means 1H performed worse than 1R) to 1.62 pp on LaRA, and from 0.36 to 0.86 pp on LISA-DayTest.

5.2 Direct-classification quantitative results

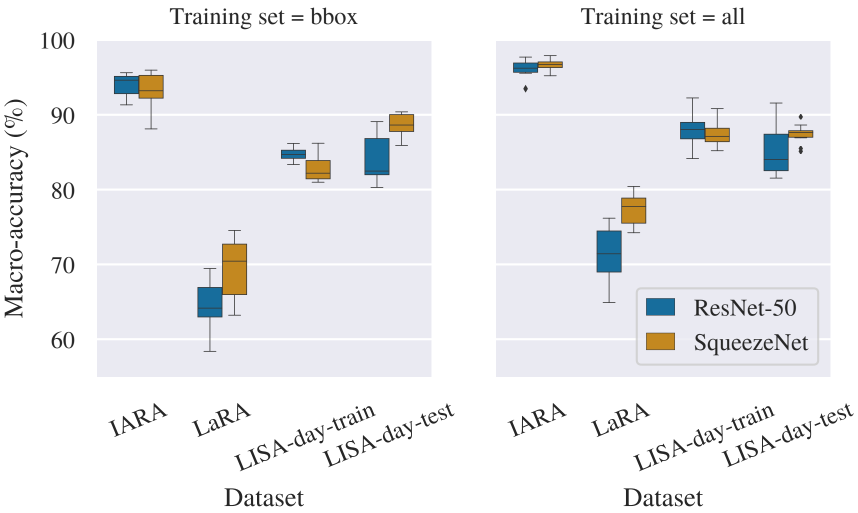

Figure 9 – Results of the classification-based approach using all training set and restricted to images with bounding boxes annotation.

The box-plots in Figure 9 show the performance of the classification models across the test datasets (both full and reduced training sets are considered, as discussed in Sect. 4.3.1). Each box of a pair represents a classification model (i.e., SqueezeNet or ResNet-50), and accounts for 10 macro-accuracy values. Considering the reduced set (PROP-BBOX-TRAIN), SqueezeNet was significantly better than ResNet-50 for LaRA and LISA-DayTest, whereas ResNet-50 was superior in LISA-DayTrain. For IARA-TL there was no significant difference in performance. Using all available images, the two models performed very similar for IARA-TL and LISA-DayTrain. For LaRA, however, SqueezeNet significantly outperformed ResNet-50 in nearly 6 pp.

Overall, the most remarkable performance discrepancy was observed for LaRA, where SqueezeNet was significantly better in both training scenarios. This is likely caused by insufficient data to effectively train ResNet, even performing augmentation to increase the variability to the training set. As result, the model tends to quickly overfit the training images and is not able to generalize for unknown data, which is even more serious due to the high variability of the LaRA scenes. Furthermore, LaRA yielded the lowest average accuracy, as also observed for the detection approach. A plausible reason for this is the structural differences between the training and test scenes. Besides, this dataset has several distractors (e.g., vehicle lights, luminous panels, reflection distortions), and traffic lights are often blurred due to the poor image quality.

5.2.1 Detection-with- vs. direct-classification

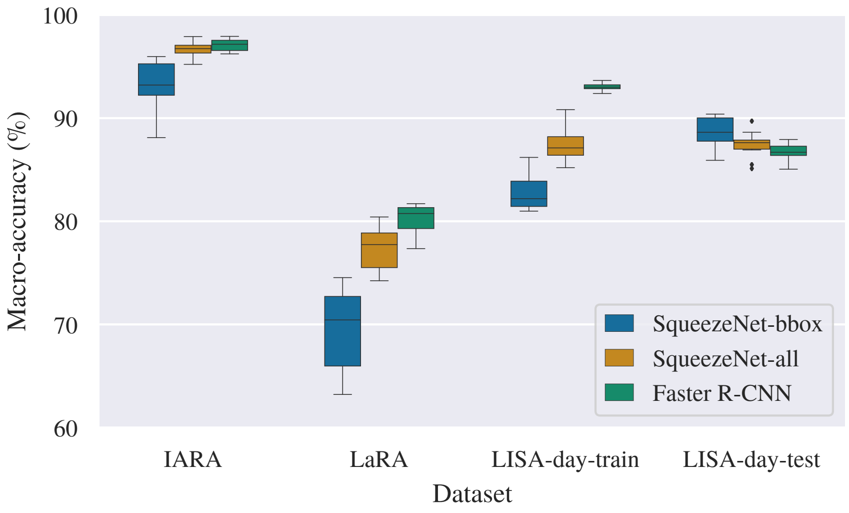

Figure 10 – Comparison of the best performing models of the detection-with-classification and direct-classification approaches. For SqueezeNet, results were reported for the reduced (SqueezeNet-bbox) and entire training set (SqueezeNet-all).

For comparative evaluation, the best performed models of each approach were selected: the Faster R-CNN with 1H heuristic (detection model) and SqueezeNet (classification model). Figure 10 shows the performance results across the datasets. For SqueezeNet, results training with both the reduced (PROP-BBOX) and full training set (PROP-TRAIN) were reported, respectively, as SqueezeNet-bbox and SqueezeNet-all.

Training with the reduced dataset, SqueezeNet was outperformed by Faster R-CNN on all datasets but LISA-DayTest, where SqueezeNet's accuracy was significantly better. As seen in Figure 10, the general rule is that SqueezeNet demands larger training sets (i.e., more images) in order to approximate the Faster R-CNN's performance. Therefore, the detection-with-classification approach would be preferred in terms of accuracy performance if one has access to a fully annotated training set, i.e., a collection where all the images have bounding-box and image-level annotation.

On the other hand, if the application context implies continuous updating (increasing) of the training images, the simple annotation process of image-level labels can lead to a higher amount of labeled images when compared to annotation of bounding-boxes. In other words, the effort in assembling a larger training set to improve the classification model could be balanced by the cheaper annotation process. With additional training data, as shown in Figure 10, Faster R-CNN and SqueezeNet performed similarly on IARA-TL, whereas for LaRA the performance difference drops dramatically from 10.66 to 2.85 pp, and from 10.12 to 5.71 pp for LISA-DayTrain. In this scenario, SqueezeNet becomes attractive since it runs at more than 240 FPS, while Faster R-CNN can operate at nearly 10 FPS.

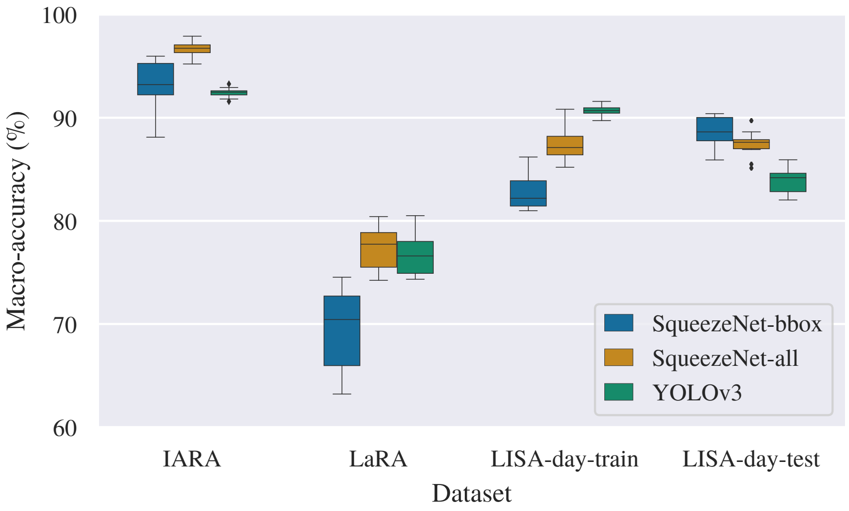

Figure 11 – Performance comparison between YOLOv3 (with 1H heuristic) and SqueezeNet.

To address time efficiency, also was analyzed YOLOv3 with 1H heuristic as an alternative to Faster R-CNN (Figure 11) since YOLOv3 is able to perform real-time (≈34 FPS). Training on the reduced set, SqueezeNet was outperformed by YOLOv3 on LaRA and LISA-DayTrain, while they were similar on IARA-TL. SqueezeNet performed better only on LISA-DayTest. With the full training set, SqueezeNet outperformed YOLOv3 on IARA-TL, and matched the YOLO's performance on LaRA. For LISA-DayTrain, YOLOv3 was still better, however the difference in accuracy dropped from 7.75 to 3.34%.

5.3 Qualitative experiments

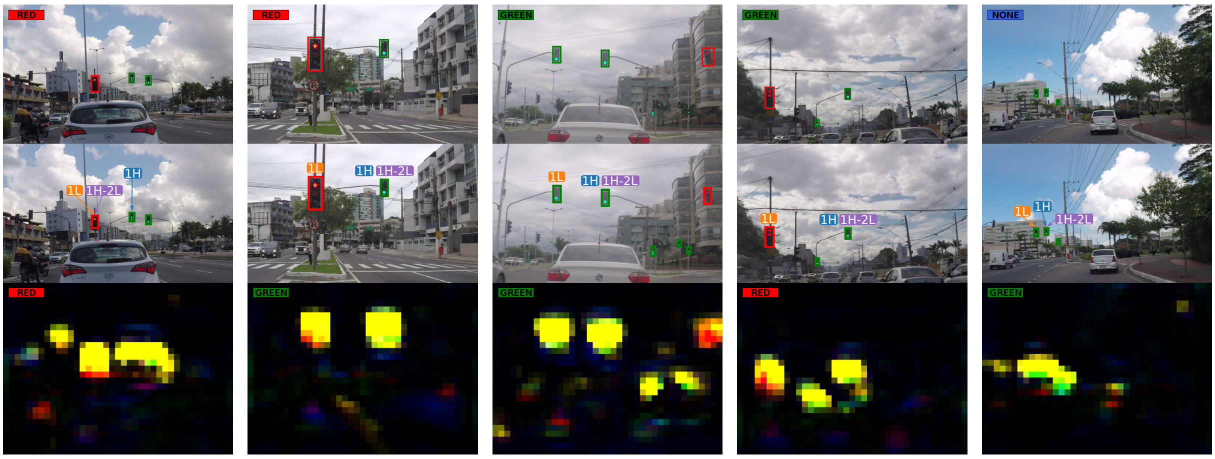

Figure 12 – Samples of hard scenarios (top row) and qualitative results for Faster R-CNN (middle row) and SqueezeNet (bottom row). The ground-truth traffic lights are enclosed by boxes in the top row, and the relevant traffic light state is labeled on the upper-left corner. Detections outputted by Faster R-CNN are represented by boxes, and the heuristic selection output is indicated by the labels 1H, 1L, 1H-2L. In the bottom row, the activation in red and green reflects, respectively, the spatial probability distribution for the red-or-yellow and green traffic lights, the network's final prediction is labeled on the upper-left corner of each image.

Figure 12 shows samples of challenging instances with the respective detections outputted by Faster R-CNN (including the relevant traffic light selection) and the SqueezeNet's activation maps. The red, green, and blue colors in the activation maps are, respectively, evidence of the *red-or-yellow*, *green*, and *none* classes. Mixed colors indicated activation of two or three classes (e.g., yellow indicates *red-or-yellow* and *green* together).

The leftmost column depicts a situation where the car should obey the red traffic light that rules the turn-left maneuver. While the heuristics 1L and 1H-2L yielded to the correct prediction, 1H interpreted a green traffic as the relevant one. SqueezeNet decided correctly the relevant traffic light state, however it was influenced by the rear light (red spot) of a motorbike. Note the predominance of yellow regions in the activation maps, which evidences the presence of traffic lights. The red/green spots indicates the specific traffic light class. A similar situation is depicted in the second scene, but only two traffic lights are present in the scene. Note that for two or less traffic lights, 1H-2L and 1H always output the same result. In the shown case, both heuristics were wrong, while 1L yielded the correct traffic light.

In the third scenario, the car should obey a green traffic light, however there is a red traffic light at nearly the same distance from the car's view. All the heuristics yielded to the correct state since the red traffic light is far from the center position and its size is smaller than the others. Note that pedestrians traffic lights were also detected, nevertheless they were disregarded by the selection heuristics. Although SqueezeNet led to the correct result, the activation maps show the influence of pedestrians traffic lights', which are more distant from the car. In the fourth scenario, the red traffic light is larger and closer than the relevant (green) one for the current lane. In this case, the selection heuristic had an essential role for the final result: while 1H and 1H-2L return the correct traffic light, 1L assigns the wrong (red) traffic light, i.e., the largest one. The SqueezeNet's activation maps responded to both traffic lights, but the red activation was decisive for the (wrong) final decision.

The last situation (rightmost column) evidences a limitation of both approaches. The car is traveling in the rightmost lane, for which no traffic light is assigned. Nevertheless, there are traffic lights in the scene, and they led detection-with-classification based assign one of them as relevant for the current lane. This is a design issue that could be better handled if the system was aware of the intended vehicle's route before deciding the relevant traffic lights. The direct-classification was also fooled by the green traffic lights in its visual field.

5.4 Regression system alone results

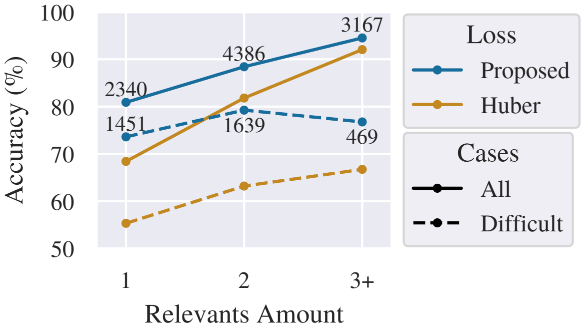

Figure 13 – Selection of a relevant traffic light. The numbers inside the graph indicates the total number of images of each subcase.

Figure 13 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.

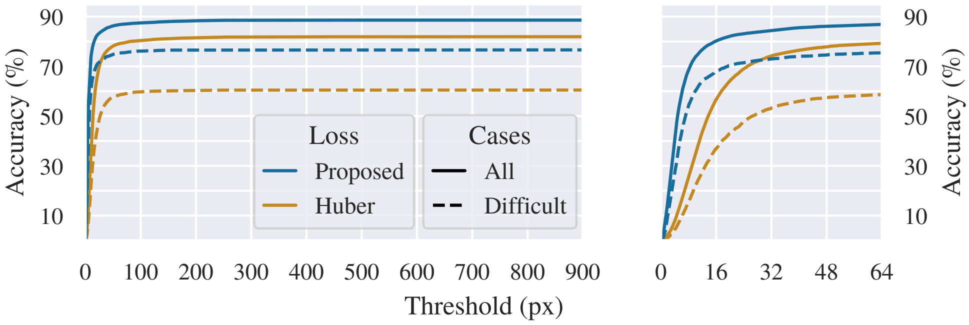

Figure 14 – Selection of a relevant traffic light with threshold distance.

Figure 14 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 13. 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 15 shows a more strict scenario where accuracy is related to the ability of correctly selecting the target traffic light. Comparing to Figure 14, the curves in Figure 15 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 16 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

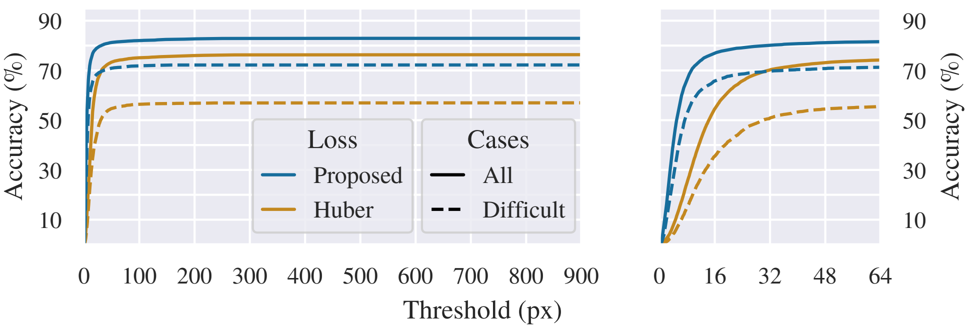

Figure 15 – Selection of the target traffic light with threshold distance.

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.

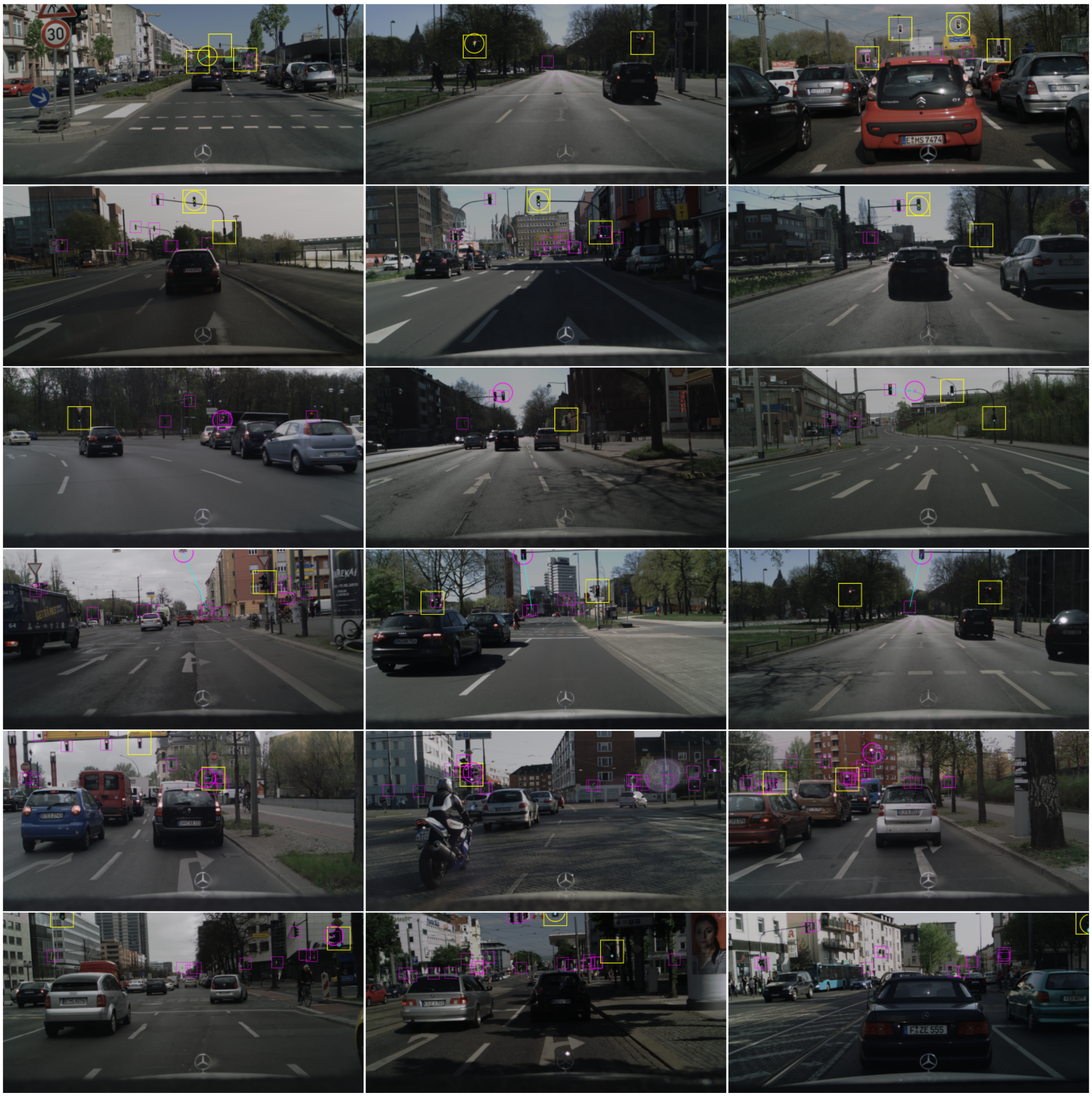

Figure 16 – Results on several images from DTLD test partition. Relevant and irrelevant traffic lights are marked in the images with yellow and lower magenta squares, respectively. The predicted coordinates are marked with a circle of the same color of the selected traffic light square. A cyan line connects 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.

6 Conclusions and future work

This work addressed the problem of recognizing the state of the relevant traffic lights in a scene as an application for intelligent vehicles. In particular, it investigates the ability of two deep learning-based approaches to solve the problem: detection-with-classification and direct-classification.

Quantitative experiments were conducted to evaluate the proposed approaches on local and publicly available datasets. Results showed that Faster R-CNN and SqueezeNet achieved the best accuracy performance for detection-with-classification and direct-classification approaches, respectively; and that simple rule-based heuristics have comparable results to the regression system's heuristic Although SqueezeNet requires additional training data to achieve accuracy comparable to the system based on Faster R-CNN, SqueezeNet runs real-time ($\approx$24 times faster than Faster R-CNN), and yielded better accuracy when compared to the detection-with-classification system with the real-time YOLOv3.

The qualitative experiments showed different situations where the models/heuristics potentially struggle with. Since traffic scenes are inherently diverse, the different heuristics for relevant traffic light selection will perform better under certain situations. Furthermore, the analysis showed some limitations of both approaches in understanding the contextual information which defines the relevant traffic light. Although they work well on standard scenarios, they might get confused in some dubious situations. To deal with this issue, future work will address the incorporation of navigation information (e.g., intended route, prior maps) to these deep learning models.

The regression system is also evaluated alone. 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 classifying system.

Bibliography

BACH, M.; REUTER, S.; DIETMAYER, K. Multi-camera traffic light recognition using a classifying Labeled Multi-Bernoulli filter. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2017. p. 1045–1051. Citado na página 25.

BADUE, C. et al. Self-driving cars: A survey. *arXiv preprint arXiv:1901.04407*, 2019. Citado na página 35.

BARNES, D.; MADDERN, W.; POSNER, I. Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2015. p. 573–578. Citado 2 vezes nas páginas 24 and 25.

BEHRENDT, K.; NOVAK, L.; BOTROS, R. A deep learning approach to traffic lights: Detection, tracking, and classification. In: *International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2017. p. 1370–1377. Citado 2 vezes nas páginas 25 and 29.

CAI, Z.; LI, Y.; GU, M. Real-time Recognition System of Traffic Light in Urban Environment. In: *Symposium on Computational Intelligence for Security and Defence Applications (CISDA)*. [S.l.: s.n.], 2012. p. 1–6. Citado na página 25.

CHARETTE, R. de; NASHASHIBI, F. Real Time Visual Traffic Lights Recognition Based on Spot Light Detection and Adaptive Traffic Lights Templates. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2009. p. 358–363. Citado 3 vezes nas páginas 21, 25, and 35.

CHARETTE, R. de; NASHASHIBI, F. Traffic Light Recognition using Image Processing Compared to Learning Processes. In: *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2009. p. 333–338. Citado 4 vezes nas páginas 19, 21, 25, and 35.

CHIANG, C.-C. et al. Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts. *International Journal of Innovative Computing, Information and Control (IJICIC)*, v. 7, n. 12, p. 6919–6934, 2011. Citado 2 vezes nas páginas 24 and 25.

DIAZ-CABRERA, M.; CERRI, P.; SANCHEZ-MEDINA, J. Suspended Traffic Lights Detection and Distance. Estimation Using Color Features. In: *International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2012. p. 1315–1320. Citado na página 24.

DIAZ-CABRERA, M.; PIETROCERRI; PAOLOMEDICI. Robust real-time traffic light detection and distance estimation using a single camera. *Expert Systems with Applications*, v. 42, n. 8, p. 3911–3923, 2015. Citado na página 25.

FAIRFIELD, N.; URMSON, C. Traffic Light Mapping and Detection. In: *International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2011. p. 5421–5426. Citado na página 19.

FRANKE, U. et al. Making Bertha See. In: *International Conference on Computer Vision Workshops (ICCVW)*. [S.l.: s.n.], 2013. p. 214–221. Citado na página 25.

FREGIN, A. et al. The driveu traffic light dataset: Introduction and comparison with existing datasets. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 3376–3383. Citado na página 36.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. [S.l.: s.n.], 2010. p. 249–256. Citado na página 39.

GONG, J. et al. The Recognition and Tracking of Traffic Lights Based on Color Segmentation and CAMSHIFT for Intelligent Vehicles. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2010. p. 431–435. Citado 2 vezes nas páginas 24 and 25.

GóMEZ, A. E. et al. Traffic Lights Detection and State Estimation Using Hidden Markov. Models. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2014. p. 750–755. Citado na página 24.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado 5 vezes nas páginas 21, 23, 24, 28, and 30.

HUBER, P. J. Robust estimation of a location parameter. In: *Breakthroughs in statistics*. [S.l.]: Springer, 1992. p. 492–518. Citado na página 31.

IANDOLA, F. N. et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv preprint arXiv:1602.07360*, 2016. Citado 2 vezes nas páginas 23 and 28.

INTELLIGENT, L. for; DIEGO, S. A. L. U. S. *Vision for Intelligent Vehicles and Applications (VIVA) Challenge*. 2015. Http://cvrr.ucsd.edu/vivachallenge/. [Online; accessed 23-April-2018]. Citado 2 vezes nas páginas 26 and 34.

JANG, C. et al. Traffic light recognition exploiting map and localization at every stage. *Expert Systems with Applications*, v. 88, p. 290–304, 2017. Citado 2 vezes nas páginas 20 and 24.

JANG, C. et al. Multiple Exposure Images Based Traffic Light Recognition. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2014. p. 1313–1318. Citado na página 25.

JENSEN, M. B.; NASROLLAHI, K.; MOESLUND, T. B. Evaluating State-of-the-art Object Detector on Challenging Traffic Light Data. In: *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 882–888. Citado na página 26.

JENSEN, M. B. et al. Traffic Light Detection at Night: Comparison of a Learning-Based Detector and Three Model-Based Detectors. In: *International Symposium on Visual Computing (IVSC)*. [S.l.: s.n.], 2015. p. 774–783. Citado 3 vezes nas páginas 19, 24, and 25.

JENSEN, M. B. et al. Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives. *Transactions on Intelligent Transportation Systems*, v. 17, n. 7, p. 1800–1815, 2016. Citado 4 vezes nas páginas 19, 21, 24, and 34.

JOHN, V. et al. Traffic Light Recognition in Varying Illumination using Deep Learning and Saliency Map. In: *International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2014. p. 2286–2291. Citado 3 vezes nas páginas 20, 24, and 25.

KIM, H.-K.; PARK, J. H.; JUNG, H.-Y. Effective Traffic Lights Recognition Method for Real Time Driving Assistance System in the Daytime. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, v. 5, n. 11, p. 1429–1432, 2011. Citado na página 25.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 40.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems (NIPS)*. [S.l.: s.n.], 2012. p. 1097–1105. Citado na página 40.

LEVINSON, J. et al. Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles. In: *International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2011. p. 5784–5791. Citado na página 19.

LI, X. et al. Traffic Light Recognition for Complex Scene With Fusion Detections. *Transactions on Intelligent Transportation Systems*, v. 19, n. 1, p. 199–208, 2018. Citado na página 24.

LINDNER, F.; KRESSEL, U.; KAELBERER, S. Robust recognition of traffic signals. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2004. p. 49–53. Citado na página 25.

LIU, W. et al. Real-Time Traffic Light Recognition Based on Smartphone Platforms. *Transactions on Circuits and Systems for Video Technology*, v. 27, n. 5, p. 1118–1131, 2017. Citado na página 25.

MICHAEL, M.; SCHLIPSING, M. Extending traffic light recognition: Efficient classification of phase and pictogram. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2015. p. 1–8. Citado na página 25.

MU, G. et al. Traffic light detection and recognition for autonomous vehicles. *The Journal of China Universities of Posts and Telecommunications*, v. 22, n. 1, p. 50–56, 2015. Citado 3 vezes nas páginas 20, 24, and 25.

MÜLLER, J.; DIETMAYER, K. Detecting traffic lights by single shot detection. *arXiv preprint arXiv:1805.02523*, 2018. Citado na página 26.

NEXAR. *Nexar Challenge #1*. 2016. Https://www.getnexar.com/challenge-1. [Online; accessed 30-April-2018]. Citado 2 vezes nas páginas 26 and 28.

PARISTECH, R. C. of M. *LaRA traffic lights dataset*. 2015. <http://www.lara.prd.fr/benchmarks/trafficlightsrecognition/>. [Online; accessed 24-April-2018]. Citado 2 vezes nas páginas 21 and 35.

PASZKE, A. et al. Automatic differentiation in pytorch. 2017. Citado na página 40.

PHILIPSEN, M. P. et al. Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset. In: *International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2015. p. 2341–2345. Citado 3 vezes nas páginas 19, 21, and 34.

PON, A. D. et al. A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. *arXiv preprint arXiv:1806.07987*, 2018. Citado na página 26.

POSSATI, L. C. et al. Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars. *International Joint Conference on Neural Networks (IJCNN)*, 2019. To appear (preprint available at <http://www.possatti.com.br/paper-tl-ijcnn-2019/ijcnn2019.pdf>). Citado 2 vezes nas páginas 20 and 24.

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv preprint* *arXiv:1804.02767*, 2018. Citado 3 vezes nas páginas 24, 29, and 39.

REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2015. p. 91–99. Citado 3 vezes nas páginas 24, 29, and 39.

SAINI, S. et al. An Efficient Vision-Based Traffic Light Detection and State Recognition for Autonomous Vehicles. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2017. p. 606–611. Citado na página 25.

WANG, J.; ZHOU, L. Traffic Light Recognition With High Dynamic Range Imaging and Deep Learning. *Transactions on Intelligent Transportation Systems*, p. 1–12, 2018. ISSN 1524-9050. Citado na página 25.

WEBER, M.; WOLF, P.; ZÖLLNER, J. M. DeepTLR: A single deep convolutional network for detection and classification of traffic lights. In: *Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2016. p. 342–348. Citado 2 vezes nas páginas 25 and 28.