UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ÁLESSON SCAPINELLO SELHORST

Real-time detection of traffic signs using onboard vehicular cameras

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Claudio Rosito Jung
Advisor

Porto Alegre, Março 2018

CIP – CATALOGING-IN-PUBLICATION

Selhorst, Álesson Scapinello

Real-time detection of traffic signs using onboard vehicular cameras / Álesson Scapinello Selhorst. – Porto Alegre: PPGC da UFRGS, 2018.

56 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Advisor: Claudio Rosito Jung.

1. Visão computacional, sistemas de apoio ao motorista, detecção de placas de trânsito. I. Jung, Claudio Rosito. II. Título.

An expert is a person
who has made all the mistakes that can be made
in a very narrow field.
— NIELS BOHR

ACKNOWLEDGMENTS

Firstly, I would like to thank my mom Odete and my father Ademir, who were present in all the difficulties encountered, guiding me to the correct decisions, giving themselves and giving up their dreams so that I can reach mine.

I would like to gratefully the entire university of UFRGS for transmitting their knowledge, such that I could be able to complete this work, the master's degree professors formed by Fernando Osório, Altamiro Susin and Marcelo Walter. Especially to professor Claudio Rosito Jung, for the trust, corrections, incentives and dedication of his time in the orientation of work.

I would like to express my thanks to all the colleagues, friends, my girlfriend who accompanied me during this walk and who were always believing in the success.

ABSTRACT

The application of new technologies has been profoundly affecting the automobile industry, especially when talking about autonomous cars. The self-driving scenario is close to becoming reality, however many challenges still need to be solved for this. Another aspect that is motivating the technological advances is the need to increase safety, in which much of the effort is being made to reduce the number of traffic accidents, especially those caused by driver errors. The reduction of accidents brings as a consequence a decrease in the resulting injuries and fatalities, as well as the related financial costs. Within this context, this thesis presents an approach for traffic sign detection and recognition using off-the-shelf onboard vehicular cameras. Assuming that the camera intrinsic parameters are obtained off-line, an on-line calibration scheme is used to estimate the extrinsic camera parameters, and Regions of Interest (ROIs) are created in the image domain based on the expected geometry and location of the traffic signs. Given the reduced size and background complexity of these ROIs, we developed a lightweight regional Convolutional Neural Network (CNN), called ScapNet. Our experimental results for Brazilian traffic signs indicate that the proposed approach presents classification accuracy comparable to state-of-the-art methods at much faster running times, with over 30 FPS on embedded devices.

Keywords: Traffic sign detection and recognition, Onboard vehicular cameras, Advanced driver assistance systems, Convolutional Neural Networks..

RESUMO

A aplicação de novas tecnologias tem afetado profundamente a indústria automobilística, especialmente quando se fala de carros autônomos. O cenário de auto-condução está próximo de se tornar realidade, entretanto, muitos desafios ainda precisam ser resolvidos. Outro aspecto que está motivando os avanços tecnológicos é a necessidade pelo aumento da segurança, no qual grande parte do esforço está sendo feito para reduzir o número de acidentes de trânsito, especialmente aqueles causados por erro do motorista. A redução de acidentes traz como conseqüência, uma diminuição nas mortes, lesões e nos custos financeiros associados aos acidentes. Dentro desse contexto, esta dissertação apresenta uma abordagem para detecção e reconhecimento de sinais de trânsito usando câmeras veiculares a bordo. Assumindo que os parâmetros intrínsecos da câmera são obtidos *off-line*, um esquema de calibração *on-line* é usado para estimar os parâmetros da câmera extrínseca, e as Regiões de Interesse (ROIs) são criadas no domínio da imagem com base na geometria e localização esperadas dos sinais de trânsito. Dado o tamanho reduzido e a complexidade de fundo desses ROIs, desenvolvemos uma Rede Neural Convolucional Regional (CNN), chamada ScapNet. Nossos resultados experimentais para os sinais de trânsito brasileiros indicam que a abordagem proposta apresenta uma precisão de classificação comparável a métodos de última geração em tempos de funcionamento muito mais rápidos.

Palavras-chave: Detecção e Reconhecimento de sinais de trânsito,Câmeras veiculares onboard, Sistemas avançados de assistência ao condutor, Redes neuras convolucionais.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACF	Aggregate Channel Features
BTSD	Belgium Traffic Sign Dataset
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CTSD	Chinese Traffic Sign Dataset
CPU	Central Process Unit
DAS	Driving Assistance Systems
DITS	Dataset of Italian Signs
FOV	Field of View
GPU	Graphics Processing Unit
GTSDB	German Traffic Sign Detection Benchmark
GTSRB	German Traffic Sign Recognition Benchmark
HoG	Histograms of Oriented Gradients
HSV	Hue, Saturation and Value
ICF	Integral Channel Features
ICS	Image Coordinate System
IoU	Intersect Over Union
MSER	Maximally Stable Extremal Regions
NMS	Non Maximum Suppression
RGB	Red, Green and Blue
R-CNN	Region Convolutional Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
SOM	Self Organizing Maps
SSD	Single Shot Multibox Detector
STSD	Swedish Traffic Sign Dataset

SVAPI	Street View Application Programming Interface
SVM	Support Vector Machines
TSR	Traffic Sign Recognition
WaDe	Wave Based Detector
WCS	World Coordinate System
YOLO	You Only Look Once

CONTENTS

1 INTRODUCTION

1.1 Motivation

Road traffic injuries are a major cause of death worldwide, with a toll of over 1.2 million of lives lost per year (WHO, 2013). Furthermore, these accidents cause 20 to 50 million non-fatal injuries (WHO, 2013), and many of the survivors develop post-traumatic stress symptoms that can become chronic, such as recurring nightmares and problems in concentration, among others (HERON-DELANEY et al., 2013).

According to the World Healyh Organization (WHO, 2013), many countries have successfully reduced the number of deaths on their roads in recent years, which does not occur in Brazil, where the number of deaths in traffic accidents remains high. As shown in Figure 1.1, data from DATASUS (SUS, 2017) indicate a number of approximately 40,000 deaths per year since 2010, with the lowest rate of 37,306 deaths in 2015.

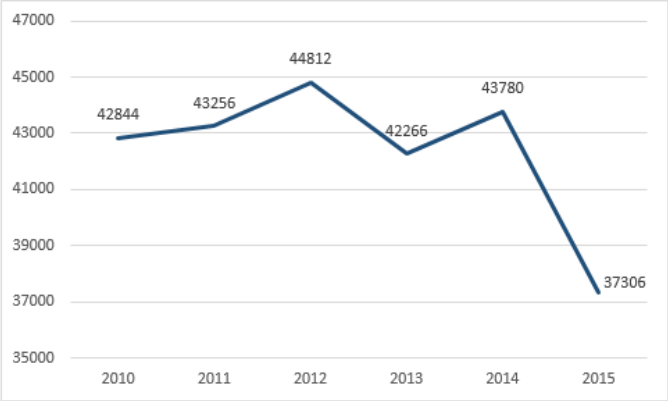


Figure 1.1 Number of deaths caused by traffic accidents in Brazil in recent years.

Source: The Author

Also, it is important consider the social, psychological and economical impacts of a traffic accident. First of all, many studies demonstrate that the vast majority of accidents occurs in low-income and middle-income countries. Likewise, individuals who have a low social status are more frequently involved in road accidents than individuals who have a high social status (ELVIK et al., 2007). At last, but not least, the financial impact of traffic injuries reaches exorbitant values due to the cost of the health care, discontinuation of professional activity, indemnities (WHO, 2013; ELVIK et al., 2007).

Governments and the automotive industry have been investing a great amount of

money and research efforts to reduce the number and severity of traffic accidents, and one research direction is toward the development of Driving Assistance Systems (DAS), which assist the driver with information about possible hazards ahead. From the many sensors typically explored in DAS, conventional video cameras are the most flexible and accessible. Smartphones with video cameras are widespread even in poorer/developing countries, which present a larger number of traffic accidents, and they can be attached to the windshield of a conventional vehicle turning it into a "smart car". In this context, the development of vision-based algorithms for DAS must take into account the limited memory and computational power of these devices (compared to modern desktop computers), keeping enough accuracy for being used in practical scenarios.

One particular problem of vision-based DAS is traffic sign recognition (TSR). Traffic signs are designed to inform driver about the local road conditions (speed limits, incoming sharp turns, forbidden overtaking, etc.), aiming to improve traffic safety. According to the Brazilian legislation (CONTRAN, 2007), vertical traffic signs that belong to either rural or urban areas of Brazil are divided into three categories, as shown in Figure 1.2. The first one is Regulation, whose purpose is to transmit to users the conditions, prohibitions or restrictions on the use of roads. The second one is warning, which is intended to alert users as potentially dangerous obstacles on the road or adjacent to it. The last one is indication, whose purpose is to identify routes and places of interest, as well as guide the routes, destinations, distances, auxiliary services and tourist attractions.

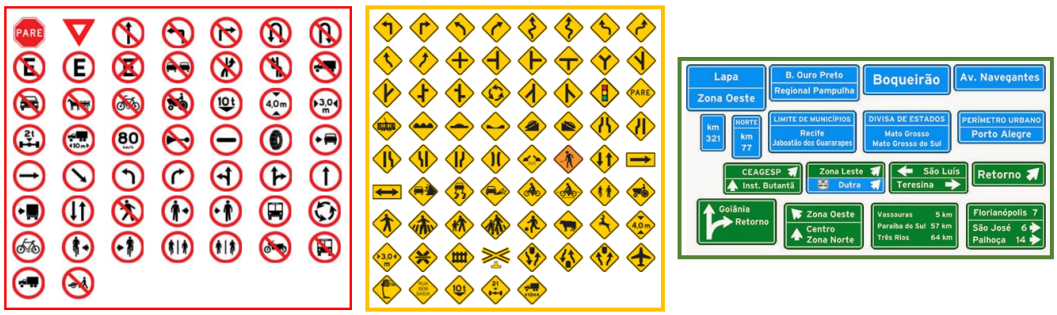


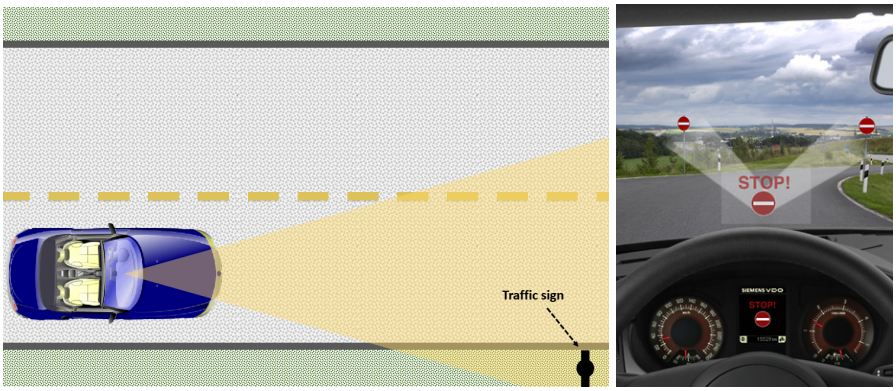
Figure 1.2 Brazilian traffic-sign. Signs in yellow, red and green boxes are warning, regulation and indication signs respectively.

Source: `http://www.detran.sc.gov.br/`

With the information provided by traffic signs, on-board TSR systems that use cameras basically take the single video stream to scan the road for traffic signs, as shown in Figure 1.3(a). When the system identifies a traffic sign, it must provide some kind of alert to the driver. For example, Figure 1.3(b) shows the system developed by Siemens, which uses a camera attached inside the car and an on-board computer to detect and recognize road signs, displaying the information to the driver. [1].

Despite the existence of several methods, there are still open problems regarding the flexibility of the camera setup, and compromise between accuracy and execution time. As shown in (MATHIAS et al., 2013; MOGELMOSE; LIU; TRIVEDI, 2015), machine-learning algorithms based on sliding windows and multiple scales, commonly used for pedestrian detection and face recognition, can reach very high accuracy rates in TSR

[1] Source: `https://www.siemens.com/press/en/presspicture/?press=/en/pp_sv/2007/sosv200703_01_(geisterfahrer)_1456407.htm`



(a) (b)

Figure 1.3 Example of a traffic sign recognition system using onboard camera: a) Schematic depiction of a vehicle approaching a traffic sign in a birds-eye view b) Example way of presenting the information about a detected sign to the driver.

Source: `https://www.siemens.com/press/en/presspicture/?press=/en/pp_sv/2007/sosv200703_01_(geisterfahrer)_1456407.htm`

without the need of encoding traffic sign specific information. Furthermore, the inclusion of additional information about the camera location can speed up the detection process and even improve the accuracy in the context of pedestrian detection (HOIEM; EFROS; HEBERT, 2008; FUHR; JUNG, 2015), so that the same behavior is expected to happen for TSR.

Recently, many authors have used Convolutional Neural Networks (CNNs) to achieve good detection and classification performance for TSR, as can be observed in benchmark results such as GTSDB (HOUBEN et al., 2013). Furthermore, new strategies such as CNN with region proposals (R-CNNs) (GIRSHICK, 2015; REN et al., 2015) allow both object detection and classification, eliminating the need of using sliding windows. However, detecting objects that are very small w.r.t. the image dimensions is still a challenge, as well as achieving real-time performance using CPUs (in particularly in hardwares with low processing power and energy requirements, which are desired for embedded applications).

1.2 Goals

1.2.1 Main Goals

The main goal of this work is to propose a new low-cost technique for detecting and classifying vertical traffic signs for Brazilian rural areas, in which they have well defined characteristics, such as two-way lanes where the signs are on the right side of the road, using images captured by a detachable onboard camera.

1.2.2 Specific Goals

To achieve the main goals, the following specific goals are defined:

- To identify Regions of Interest (ROIs) based on the expected location of the signs in the world and the camera parameters

- To develop a detection and recognition algorithm that achieves a good compromise between accuracy and running time
- To test the proposed method in hardwares suited for embedded applications
- To evaluate the obtained results and compare with the state of the art in traffic sign recognition.

1.2.3 Contributions

As consequence of our TSR approach, the major contributions of this work, which are closely related to the specific goals, are the following:

- To define a model using camera parameters to crop ROIs where the traffic sign appears in the image.
- To generate a dataset with Brazilian traffic sign, which represent conditions found on rural zone.
- To develop a regional CNN that presents a good compromise between accuracy and execution time.

1.3 Structure of the thesis

The remaining part of the text is organized as follows. Chapter 2 discusses the literature review related a TSR. In the Chapter 3 the created Brazilian traffic sign dataset is discussed. Chapter 4 presents the proposed TSR approach in Brazilian roads using a calibrated camera. Then, the results and comparisons with other state-of-art techniques are demonstrated in Chapter 5. Chapter 6 presents the final considerations and point out the future work. Finally, the references used throughout this work are listed.

2 LITERATURE REVIEW

The generic problem of traffic sign recognition (TSR) has been researched extensively by the computer vision and intelligent transportation systems communities, and a wide variety of methods have been proposed in the past years. In order to identify such initiatives, we conducted a review selecting works with expressive results on traffic signs detection and recognition systems.

According to Eichner e Breckon (EICHNER; BRECKON, 2008), the recognition of vertical traffic signs can be divided into two stages: detection and recognition. The detection process consists in locating the traffic signs that are present in the input images, while the recognition focuses on validating and identifying the exact kind of sign that was detected. The authors also mention the importance of dealing with video sequences, and the need of real-time processing for practical applications.

There are various approaches for both the detection and recognition steps. Generally, these tasks are usually accomplished by using color and shape information, with a classifier to define the correct class. This chapter presents the state of the art in traffic sign detection and classification systems. Firstly, we present the techniques that use color and shape information with a classifier, here called as conventional methods (Section 2.1). Then, we present a brief review on generic convolutional neural networks, exploring region-based approaches to image recognition (Section 2.2). Finally, the initiatives related to traffic sign recognition that explore neural networks are analyzed in Section 2.3.

2.1 Conventional Methods

As previously mentioned in this work, researchers have invested their efforts in both the traffic sign detection and recognition tasks. For the first phase (detection), the task is usually accomplished by using two main sources of information: color and shape. According to (MOGELMOSE; TRIVEDI; MOESLUND, 2012), color-based methods explore the fact that road signs are designed to be easily distinguished from its surroundings, and its visual identity defines strong and contrasting colors. Moreover, traffic signs also have different and characteristic shapes, which provide an additional cue for the detection phase. Gomez et al. (GOMEZ-MORENO et al., 2010) subdivided the detection task into segmentation (color information) and detection (use of geometric shapes). The notation and taxonomy vary considerably among authors, but the final purpose is the same and they often use the words detection and segmentation for the same finality.

Various methods exploit segmentation based on color cues, partitioning the image into regions. This class of techniques consists mostly of thresholding the input image in a given color space. One of the most commonly used color models is HSV, since it separates the component intensity of the color information (hue and saturation) in a color image,

and it is explored by several authors (KUO; LIN, 2007a; NGUWI; KOUZANI, 2008; REN et al., 2009; CHIANG et al., 2010; QINGSONG; JUAN; TIANTIAN, 2010; BERKAYA et al., 2016; LI et al., 2015). As noted in (MOGELMOSE; TRIVEDI; MOESLUND, 2012), some authors criticize the use of HSV for not handling adequately changes in color temperature due to different weather conditions, and explore the RGB color space (TIMOFTE; ZIMMERMANN; GOOL, 2009; PRISACARIU et al., 2010).

Some researchers choose to use shape-based segmentation, and edge cues are popular choices to find the boundaries of the traffic sign candidates. The Canny detector (CANNY, 1986) is present in various traffic sign systems such as (HOUBEN, 2011; TIMOFTE; ZIMMERMANN; GOOL, 2009; RUTA; LI; LIU, 2010; DEGUCHI et al., 2011). Another popular and most recent trend to explore shape information in a higher level is based on Histograms of Oriented Gradients (HOGs). The work of Dalal and Triggs (DALAL; TRIGGS, 2005) showed the potential of HOG in the context of pedestrian detection, and later other authors explored HOG-like features for traffic sign detection/recognition, such as (GAO et al., 2006; XIE et al., 2009; CREUSEN et al., 2010).

For the classification stage, a set of features computed within the detected sign are typically extracted and fed to a classifier. There is a great variety of possible features and classifiers, such as Haar wavelets (KUO; LIN, 2007b; PRISACARIU et al., 2010), Gabor filters (KONCAR; JANSSEN; HALGAMUGE, 2007), HOG (GREENHALGH; MIRMEHDI, 2012), Integral Channel Features (ICFs) (MATHIAS et al., 2013) and Aggregate Channel Features (ACFs) (MOGELMOSE; LIU; TRIVEDI, 2015). Among the classifiers, Support Vector Machines (SVMs) (BASCÓN et al., 2010; WANG et al., 2013), Self Organizing Maps (SOMs) (PRIETO; ALLEN, 2009), Random Forests (ZAKLOUTA; STANCIULESCU, 2014; ELLAHYANI; EL ANSARI; EL JAAFARI, 2016) and deep learning-based methods (STALLKAMP et al., 2012) have been used. A more comprehensive review can be found in recent survey papers (MOGELMOSE; TRIVEDI; MOESLUND, 2012; GUDIGAR; CHOKKADI; RAGHAVENDRA, 2016).

Although the typical pipeline for TSR involves first detection and then recognition, some machine-learning approaches may complete both tasks simultaneously. One possible approach is to characterize a particular traffic sign using a set of features (e.g. HOG, ACF, Haar-like, deep features, etc.), and explore a classifier based on sliding windows and multiple resolutions. Although this strategy achieves high accuracy rates in related problems, such as pedestrian detection and face recognition (MATHIAS et al., 2013; MOGELMOSE; LIU; TRIVEDI, 2015), the use of sliding windows tends to be computationally costly.

In this context, different from the majority of existing systems, the pipeline proposed in (SALTI et al., 2015) is based on interest regions extraction rather than sliding window detection. It uses Maximally Stable Extremal Regions (MSER) detector (MATAS et al., 2004) and Wavebased Detector (WaDe) (SALTI; LANZA; DI STEFANO, 2013) to detect candidate regions. Then it uses HoG and SVM to classify the traffic signals.

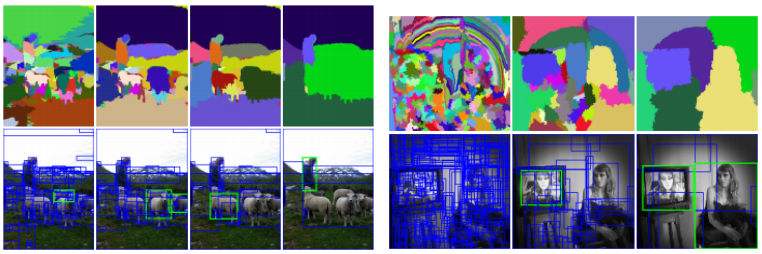
An attractive alternative approach is the use of end-to-end neural networks that are used for both detection and classification tasks, such as Fast-RCNN (REN et al., 2015), Faster-RCNN (GIRSHICK, 2015) and YOLO (REDMON; FARHADI, 2016). These methods use region proposals to improve the object detection performance and accuracy without using sliding windows. A brief revision of such CNNs for generic purpose object detection and recognition is presented next.

2.2 Convolutional Neural Networks

Despite the attractive qualities of deep Machine Learning algorithms, they have started to be widely used only in recent years, due to the advancement of parallel processing units (GPUs) and the increasing quantity of big datasets with high-resolution. In particular, Convolutional Neural Networks, which take advantage of a highly-optimized implementation of convolutions into current GPUs, have shown impressive results in several detection and classification problems. (CHEN et al., 2017)

Before this notorious advance, the networks had a fixed input size (almost always small because of computational cost, e.g. 32×32 or 48×48) and were used only for classification. Generally, the CNN used as classification was trained on a large set of examples with positive and negative samples, and applied in the whole image using sliding window approach. Besides sliding in all possible locations in the image, it is necessary to search at different scales, because the classifier was trained with fixed input size and objects could appear in different sizes in the same image. Therewith, the classifier typically generates multiple responses that subsequently need to be post-processed and also it is computationally very expensive when we search for multiple aspect ratios.

An alternative to cope with these problems is the Selective Search approach (UIJLINGS et al., 2013), proposed to be fast with a high recall rate. It is based on computing hierarchical grouping of similar regions based on color, texture, size and shape compatibility. The grouping algorithm starts by over-segmenting the image based on intensity of the pixels to get a set of small starting regions which ideally do not span multiple objects. To do this, they used the method proposed by Felzenszwalb and Huttenlocher (FELZENSZWALB; HUTTENLOCHER, 2004), which is an efficient segmentation algorithm based on measuring the evidence for a boundary between two regions using a graph-based representation of the image. Then, use a greedy algorithm that first calculate similarities between all neighboring regions, group together the two most similar regions, and calculate new similarities between the resulting region and its neighbors, until the whole image becomes a single region. Figure 2.1 shows examples of the selective search algorithm. The result is some region proposals, which could be classified using a CNN or other object recognition model, and the region proposals with the high probability scores can be considered locations of the object.



(a) (b)

Figure 2.1 Two examples of selective search algorithm showing the necessity of different scales

Source: (UIJLINGS et al., 2013)

In this context, Region Convolutional Neural Network (GIRSHICK et al., 2014) first gets the input image and uses Selective Search to generate approximately 2000 different

region of interest that have a high probability of containing an object, as show in Figure 2.2-2. Then, the region proposal are warped into an standard square image size which satisfies the input of a modified version of an CNN, to compute features vector for each region of interest (Figure 2.2-3). Finally, these features are categorized with an SVM classifier (Figure 2.2-4). Furthermore, a linear regression is used on the region proposal to generate tighter region of interest.

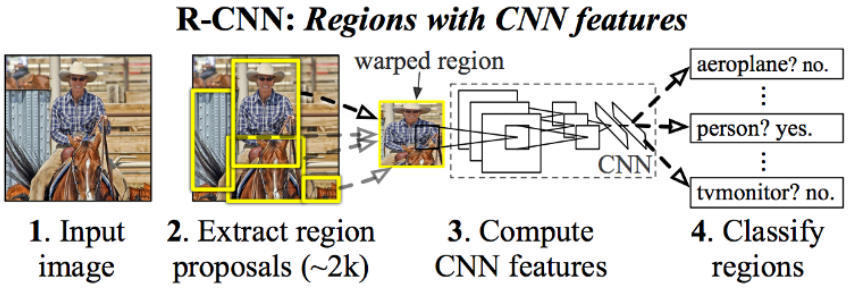


Figure 2.2 The R-CNN system overview

Source: (GIRSHICK et al., 2014)

The main limitation of the Region-CNN is the execution time. For each image, the system need around 2000 forward pass on the CNN (for every region box). Moreover, it is necessary to train one CNN (AlexNet in this case), one classifier and an regression model, making the pipeline very complex to train.

To cope with this issue, the authors created the second version, faster and easier to train, called Fast Region-based Convolutional Network (GIRSHICK, 2015). In this new version, instead of computing the region proposals and warp them into a ConvNet, the input image is passed through a convolutional network, and then the Region of Interest Pooling is used to share computed features into bounding boxes. Furthermore, the ConvNet that extract features, the classifier and the regressor used to tighten bounding boxes were joined in a single module, as shown in Figure 2.3. In this pipeline, the RoiPool is a layer of the CNN, and the SVM classifier was replaced by a fully-connected layer with sofmax at the end of the CNN. In addition, the regression model used in R-CNN was also changed by a linear regression layer parallel to the softmax layer.

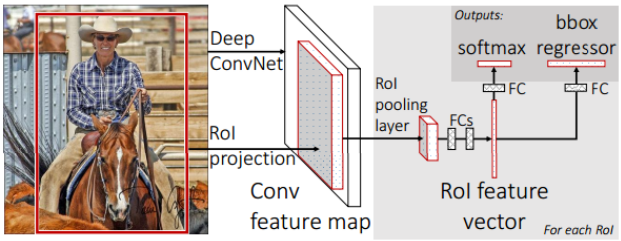


Figure 2.3 Fast R-CNN architecture

Source: (GIRSHICK, 2015)

In fact, the Fast R-CNN version is faster, cleaner and has higher detection quality, reaching a speedup of almost 9 times to train and 146 times to test, not including the time

to generate region proposals. Including the Selective Search method to generate region of interest proposals the speedup decreases to about 25 times for the test. Then, the Faster R-CNN (REN et al., 2015) version was proposed, and instead of using Selective Search they introduced the Region Proposal Network (RPN), which is a fully-convolutional network that simultaneously predicts object bounding boxes and "objectness" scores at each position.

Basically , the RPN added after the last convolutional layer produces region proposals from a feature map (Figure 2.4(a)). To generate region proposals, RPN explores a sliding window over the feature map, and at each sliding-window location, k region proposals are simultaneously predicted. The k proposals are parameterized relative to reference boxes, called anchors, that represent scale and aspect ratio from the dataset.

An anchor is labeled as positive if it presents the highest Intersection over Union (IoU) overlap w.r.t. the ground-truth box, also setting an IoU threshold of 0.7. Therefore, at each sliding-window location, the *reg* layer has $4k$ outputs encoding the coordinates, as shown in Figure 2.4(b). The *cls* layer outputs $2k$ scores that estimate probability of object or not-object for each proposal. The RPN loss function is given by

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{2.1}$$

where:

- i is the index of an anchor in a mini-batch
- p_i is the predicted probability of being an object for anchor i
- p_i^* is the ground-truth label (1 if positive, 0 if anchor is negative)
- t_i is the coordinates of the predicted bounding box for anchor i
- t_i^* is the ground-truth box associated with a positive anchor
- N_{cls} is the number of anchors in minibatch
- N_{reg} is the nuber of anchor locations
- L_{cls} is log loss over two classes (object vs. not object)
- L_{reg} is the smooth L1 loss function (GIRSHICK, 2015)
- λ is an costant to balance both terms

Another very widespread approach in object detection is You Only Look Once (YOLO) (REDMON et al., 2016). Different from the region-based techniques presented previously, the authors proposed a new approach to detecting objects. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation, without bounding box proposals and subsequent pixel or feature resampling stage.

The core idea in YOLO, shown in Figure 2.5 is to divide the input image into a grid with dimensions $S \times S$, and if the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Besides that, each grid cell predicts B bounding boxes and confidence scores for those boxes that represent how confident the model is to

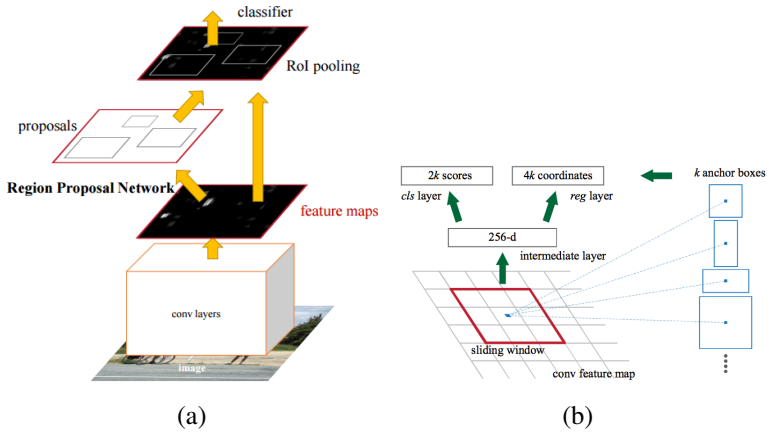


Figure 2.4 (a) The Faster R-CNN workflow (b) Region Proposal Network (RPN)

Source: (REN et al., 2015)

classify if each box contains or not an object, obtained as IOU of predicted box and any ground truth box.

Each grid cell also predicts C conditional class probabilities, that are multiplied by the individual box confidence predictions, which gives the class-specific confidence scores for each box. Finally, a non maximum suppression (NMS) step is applied in the final detections, as shown in Figure 2.5.

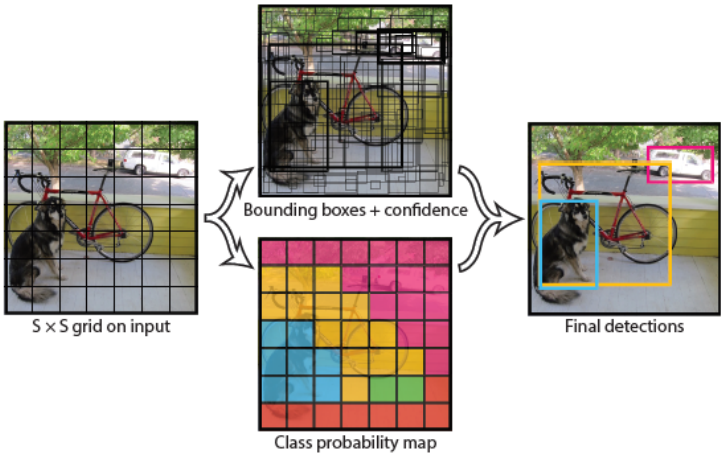


Figure 2.5 Yolo system model

Source: (REDMON et al., 2016)

In the second version of YOLO (REDMON; FARHADI, 2016), called YOLOv2, the authors improved training and increase performance making some modifications of the previous version. One of them was achieved by adding Batch Normalization on all of convolutional layers and removing dropout, without overfitting. To transform YOLOv2 robust to running on images of different sizes and since the model only uses convolutional and pooling layers, in which it can be resized on the fly, the authors introduced multi-

scale training. It consists in every 10 batches, randomly chooses a new image dimension according to the downsampling factor of the network, resize the network to that dimension and continue training.

Another major change was the addition of anchor boxes, as in R-CNN approach. Different from the first version of YOLO, which predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolutional feature extractor, in this approach they remove the fully connected layers from YOLO and use anchor boxes to predict bounding boxes. These anchors, are selected running *k-means* clustering on the training set to get good priors for the model and chose 5 centroids as a good tradeoff between complexity and high recall.

Besides these approaches in detecting objects, it is important to mention the Mask-RCNN (HE et al., 2017) and Single Shot MultiBox Detector (SSD) (LIU et al., 2016). However, even with the recent advances in object detection, runtime is still a problem (e.g. 7 fps with a good GPU using Faster-RCNN (GIRSHICK, 2015)), making it impossible to achieve higher framerates on low-power hardware.

2.3 Neural Network based TSR

Some of the architectures described in Section 2.2 have been explored in the context of TSR, as shown in (ZHU et al., 2016; MENG et al., 2017). Other studies with the neural networks were also used by Nguwi and Kouzani (NGUWI; KOUZANI, 2008), Fistrek and Loncaric (FISTREK; LONCARIC, 2011), Zhang Sheng and Li (ZHANG; SHENG; LI, 2012), Yang et al. (YANG et al., 2016), Stallkampa et al. (STALLKAMP et al., 2012), Cireşan et al. (CIREŞAN et al., 2012), Bruno D. and Osorio F. (BRUNO; OSÓRIO, 2017).

The method introduced by Meng et al. (MENG et al., 2017) exploits some concepts of convolutional networks to propose a approach that is capable of detecting small objects from large images (e.g. with a resolution of over 2000×2000). As shown in Figure 2.6, the first process is to break the image into patches with size 200×200 in a sliding window fashion. Since the Small Object Sensitive CNN (SOS-CNN) was designed to be sensitive to small objects, objects with larger sizes will not be detected in the original image. Thus, an image pyramid is constructed, where the larger objects that cannot be captured in the image with original resolution become detectable on images with smaller scales.

Figure 2.7 illustrates the proposed SOS-CNN. The network was designed for small object detection and was derived from an SSD model with a VGG-16 network, where only the first 4 convolutional stages are kept, combined with a set of convolutional layers with a kernel size of 3×3 in the end of the network. As in Faster-RCNN, a set of predefined default anchor boxes with different sizes and aspect ratios are introduced to assist producing the predictions for bounding boxes. Also, the network produces the confidence scores for each category.

The SOS-CNN predicts offsets relative to each of the default anchor boxes, rather predicting the location of bounding boxes for each object in an image, and also the corresponding confidence scores over the target classes simultaneously. A patch-level is considered as match if the IoU overlap is higher than a threshold. Then, all the patch-level predictions will be projected back onto the image at the original scale and a NMS is employed to generate the final image-level predictions, as illustrated in 2.6.

Yang et al. (YANG et al., 2016) proposed a real-time traffic sign recognition system consisting of detection and classification modules, although used CNNs just in the clas-

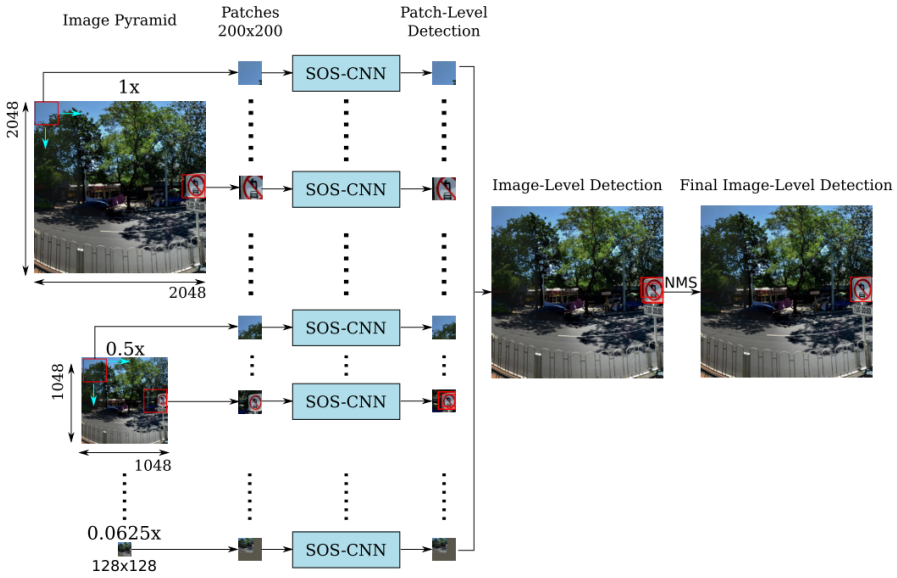


Figure 2.6 The proposed framework with an image pyramid, the SOS-CNN to produce patch-level detection and a Non Maximum Suppression (NMS) to generate the final predictions on the original image

Source: (MENG et al., 2017)

sification module. Essentially, they used a color probability model (YANG; WU, 2014) to transform the input color image into a traffic sign probability map. This probability map is a gray image that represents at which pixels the image contains a traffic sign. Afterwards, rather then using sliding windows, they extract traffic sign proposals by finding MSERs from probability maps and filter false positives of traffic sign proposals with a Support Vector Machine (SVM) based on a color HOG feature. Finally, in the classification module, they applied contrast limited adaptive histogram equalization (CLAHE) to adjust the contrast of the images and tested three CNNs in the proposals to classify into their sub-classes. Yang et al. also proposed a Chinese Traffic Sign Dataset (CTSD) with 1100 images in different sizes.

Zhu et al. (ZHU et al., 2016) showed the potential of regional CNNs for TSR, besides the difficulty of traditional solutions in detecting target objects that occupy a small part of the whole image as well (e.g. traffic signals), training two CNNs for just detect and simultaneously detect and classify traffic signs.

The difference between the two networks, is in the branches in the last layer. Inspired by the work of Huval et al. (HUVAL et al., 2015), which evaluates the performance of CNNs on lane and vehicle detection, the authors modified network architecture and adapted to detect and recognize traffic signals. Basically, they made the network branch after layer six into three streams, a bounding box layer, a pixel layer and a label layer which can output the probability to a specific class. Figure 2.8 illustrates the network architecture used.

Furthermore, the authors introduced a new large dataset with traffic signs. This dataset have been generated from 100000 Tencent Street View [1] panoramas, containing 30000 traffic-sign instances in different conditions of illumination and weather. Tencent maps is

[1]Link to access: map.qq.com

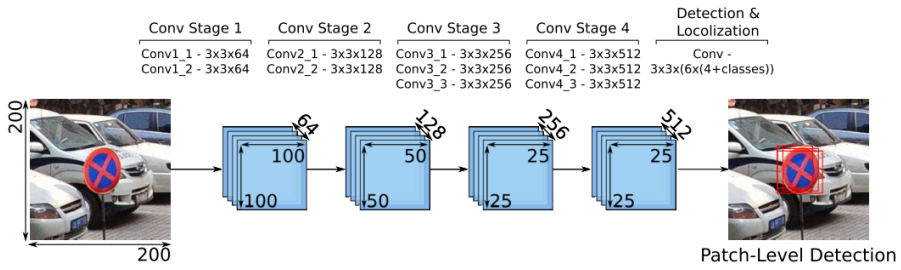


Figure 2.7 The proposed SOS-CNN.

Source: (MENG et al., 2017)

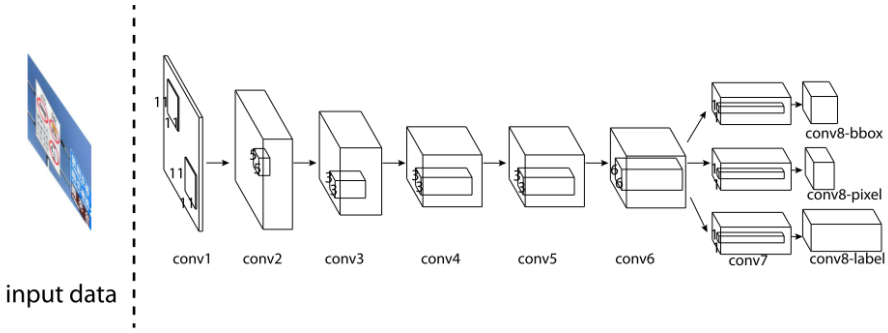


Figure 2.8 Architecture of multi-class network. The network is fully convolutional, and branches after the 6th layer.

Source: (ZHU et al., 2016)

a service application that offers satellite imagery, street maps, street view and historical view perspectives, from mainland China, Hong Kong and Taiwan. They chose 10 regions from 5 different cities in China (including both downtown regions and suburbs for each city) and downloaded 100000 panoramas from the Tencent Data Center.

Then, each Chinese traffic-sign in the data is annotated by hand, with a class label, its bounding box and pixel mask and divided into three categories, as shown in Figure 2.9. The first category represent warning signs (mostly yellow triangles with a black boundary and information), the second symbolize prohibitory signs (mostly white surrounded by a red circle and also possibly having a diagonal bar), and the last indicate mandatory signs (mostly blue circles with white information).

The final dataset contains 100,000 images with some of them only containing background. Of these, 10,000 contain 30000 traffic-signs in total, divided into three categories and 45 classes that have more than 100 instances. Classes with fewer than 100 instances were simply ignored.

Therefore, even though there are several techniques using convolutional neural networks, the challenge of recognizing traffic signs is still open, especially when the execution time is evaluated. The current techniques have satisfactory results, however they remain nonviable for embedded hardware.

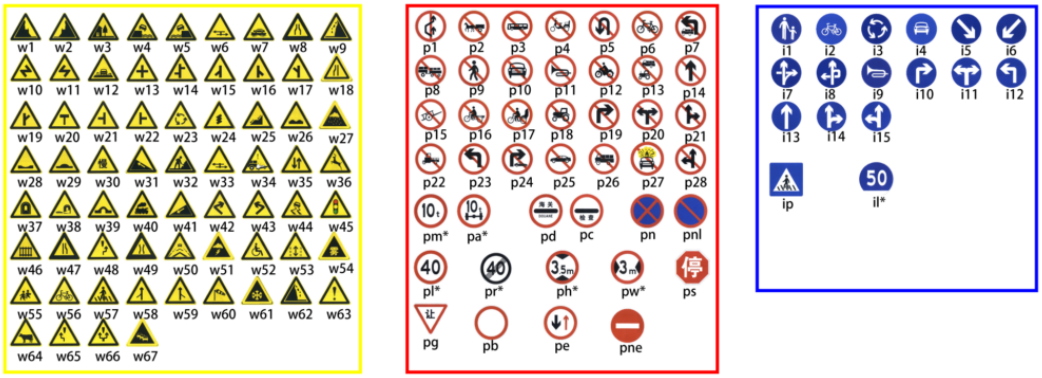


Figure 2.9 Chinese traffic-sign classes from Tencent Data Center. Signs in yellow, red and blue boxes are warning, prohibitory and mandatory signs respectively. Each traffic-sign has a unique label.

Source: (ZHU et al., 2016)

2.4 Conclusion

In these previous studies, the authors present their efforts to establish new techniques of traffic sign detection and recognition, more recently, migrating from conventional techniques to the use of Convolutional Neural Networks, obtaining satisfactory results. Nevertheless, the computational costs for good accuracy are still very high, making these solutions unfeasible for embedded environments with low-cost hardware, as shown in (ZHU et al., 2016), (REDMON; FARHADI, 2016), (REN et al., 2015), where a good quality GPU is required for execution on real time.

In this context, our approach is focused on maintaining a good trade-off between complexity and high recall, adapting state-of-the-art techniques in object recognition and using known information of location in the highway and norms where the transit signs appear. Thus, we propose a technique of traffic sign recognition that is applicable in scenarios with low cost of processing.

3 TRAFFIC SIGN DATASETS

Despite the success and popularity of deep CNNs for object detection and recogniton, they are inherently data hungry. They require several labeled images, and many datasets for object detection and recognition exist, as PASCAL VOC 2007-2012 dataset (EVERINGHAM et al., 2010), Common objects in context (COCO) (LIN et al., 2014), Salient Object Subitizing (SOS) (ZHANG; MA; SAMEKI, 2015), and the ImageNet Large Scale Visual Recognition Challenge (RUSSAKOVSKY et al., 2015).

However, labeled datasets for traffic sign detection and recognition are still emerging, and as far as we know there is no available dataset for Brazilian signs. In this chapter, we first revise some existing datasets, and then describe the proposed dataset for Brazilian traffic signs.

3.1 Publicly available traffic signs datasets

One of the first and most used traffic sign dataset is the German Traffic Sign Recognition Benchmark (GTSRB) (STALLKAMP et al., 2012) for classification and German Traffic Sign Detection Benchmark (GTSDB) (HOUBEN et al., 2013) for localization. Both datasets contain traffic sign images of German patterns that were taken from cars on real streets. The GTSRB contains images cropped for classification purposes, while the GTSDB has signs and labels into images for detection.

The GTSRB was proposed in 2011, containing 51,840 single-images in total to a multi-class classification problem. The dataset was collected from approximately 10 hours of video that were recorded on different road types in Germany during daytime, then the data collection, annotation and image extraction was performed using the NISYS Advanced Development and Analysis Framework (ADAF). The images was divided into 43 classes and some instances can be observed on Figure 3.1.

In order to increase the number of datasets freely available, the GTSDB was proposed in 2013, which comprises a large dataset of real-world images with different scenarios (e.g. urban, rural, highway). In total, the dataset contains 900 RGB images with size 1360×800 pixels, containing 1,206 traffic signs with sizes varying between 16 and 128 pixels. Every image was annotated with the rectangular regions of interest (ROIs) of the visible traffic signs and the specific traffic sign class (e.g., stop sign, speed limit 60, speed limit 80, etc.). They also divided classes into three competition-relevant categories (prohibitive signs, mandatory signs, and danger signs). Figure 3.2 illustrates some examples from the dataset.

As shown in Section 2.3, Zhe et al. (ZHU et al., 2016) proposed a new large dataset with traffic sign collected from Tencent Street View images. In this dataset, 30,000 traffic signs were divided into 45 classes, as shown in Figure 2.9. For this purpose, the authors

Figure 3.1 All classes from GTSRB dataset

Source: (STALLKAMP et al., 2012)

downloaded panoramas, as in Figure 3.3(a), and sliced vertically into 4 images. They also cropped sky and ground at top and bottom part of panoramas. Then, they manually mark the classes, the ROI that contains the respective traffic sign and the pixel mask, as shown in Figure 3.3(b).

Another large dataset is the Belgium Traffic Sign Dataset (BTSD) (TIMOFTE; ZIMMERMANN; VAN GOOL, 2014), which consists of 13,444 traffic sign annotations in 9,006 images. Most of them present signs visible at less than 50 meters from the camera. The images also are divided into three categories: mandatory, warning and prohibitory.

The Swedish Traffic Sign Dataset (STSD) was collected in 2011, presenting more than 20.000 images, from which approximately 20% are labeled. The dataset contains 3,488 traffic signs from highways and cities recorded from more that 350km of Swedish roads. Also, the Dataset of Italian Traffic Signs (DITS) provide challenging images captured at night and in presence of fog in 14 hours of videos recorded in different places around Italy.

Despite the existence of some (fully or partially) annotated datasets for traffic sign detection and/or recognition from several countries, it is not to our knowledge the existence of publicly available and labelled datasets with Brazilian traffic signs. We found a mention to a Brazilian dataset collected by Hoelscher, I. and Susin, A. (HOELSCHER, 2017), however were unable to use because it is not available yet. The proposed dataset is presented next.

3.2 The Proposed Brazilian Traffic Sign Dataset

Deep learning algorithms have shown superior performance for several tasks such as image detection, classification and speech recognition. For both tasks, the used data needs to be in accordance with some characteristics. The first one is that data must be

Figure 3.2 Some example images from the GTSDB dataset, representing variations in weather, lighting, and driving conditions.

Source: (HOUBEN et al., 2013)

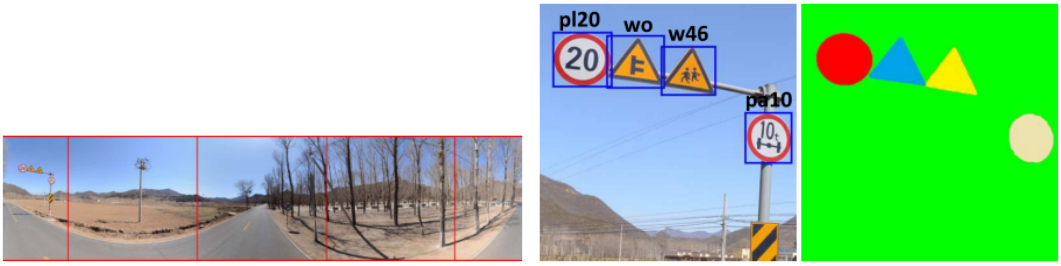
directly relevant to problem in which it is adapted to, in other words, the training data must resemble as much as possible the real-world data that will be processed.

The second characteristic is how the data is annotated. This process can be significantly more expensive and is very important for deep learning algorithms methods to learn with the training data, generalize, and to be applied to new unlabeled (and unseen) data. In some cases, it is necessary to perform a pre-processing, such as cutting, resizing, or applying some filters in the training data.

Other point to note is the minimum amount of data required for the algorithm to learn and generalize the problem. This quantity varies according to complexity of the problem, the number of classes or size of object to be detected. Generally, without considering over-training, the more amount for training can ensure better performance.

In this context, we propose a new Brazilian Traffic Sign Dataset (BrTSD) that represent real-world images in Brazilian rural roads. The aim is to create a dataset that allows algorithms to be applied with the patterns of Brazilian traffic signs and conditions of structures found in Brazilian highways, instead of other countries as found in the datasets discussed in Section 3.1.

Our dataset was generated with images provided by Street View Application Programming Interface (API) from Google Maps (ANGUELOV et al., 2010). This API provides communication with the pre-programmed functions defined by Google, which allows the



(a) (b)

Figure 3.3 (a) Panorama from Tencent StreetView, with size 8192×2048 and marks in red to slice vertically into 4 images. (b) Images from dataset annotated with class label, bounding box, and pixel mask.

Source: (ZHU et al., 2016)

creation of maps, route generation, access to the panoramas in Street View mode and download of information.

To capture the dataset images, the Google API[1] account with permissions was created. We then generated a map and put a marker (which can be dragged to other positions) in a Brazilian random highway, as shown Figure 3.4. The next step was to define a destination bookmark in the map, and finally the route was generated when the "Calculate Route!" button is clicked. Since the idea is to run the application on the roads of Brazilian rural areas, the routes were not defined within cities.

Jointly with the map and the route, we created a viewer with the Street View, as illustrated in the bottom of Figure 3.4. In this image, the current mark position was showed. When the "Start Navigation" button is clicked, the mark goes to the destination step by step and all images in this route were saved. We take care to update some parameter values when the marker advances one step, which is defined according to the car and camera used by google when capturing images, towards the destination, such as the Field of View (FOV) that determines the horizontal field of view of the image, the *pitch* that specifies the upper or lower angle of the camera relative to the Street View vehicle, and *heading* that indicates the compass direction of the camera. These values assist the system, to leave the panorama at an angle that resembles a camera in front of the car looking forward, rather than looking at the sides in curves when it was stepping automatically.

Several routes were generated on the map and about 100,000 images were saved with size 600×600. To label these images, we explored a semi-automatic process that consisted of: (i) firstly label approximately 200 images that contains signs; (ii) use our CNN presented in Section 4.2 to detect images with traffic signs in the rest of dataset, retrieving additional 600 images that contain traffic signs; (iii) re-train the network with all 800 images and run in the rest of the dataset.

To adapt our convolutional network to just detect images that present signs, we just do not consider the classes values in the last layer. In other words, we run a neural network in the dataset in order to capture all the images that contain signs, along with their location in the image. We also relax the acceptability values of the network confidence values, to not risk losing frames with signs. After this process, all captured images were manually

[1] `https://developers.google.com/maps/`

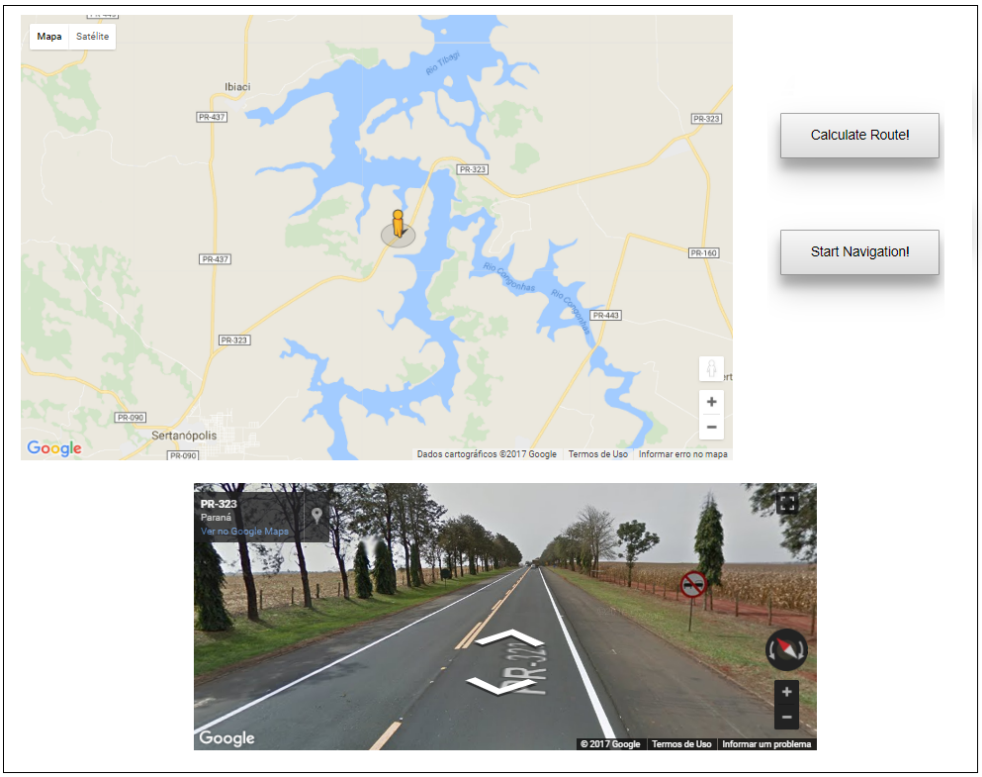


Figure 3.4 Overview of our system to generate the proposed dataset

Source: The author

validated.

These images that contains a sign were annotated with the class labels, as well as their corresponding bounding box. Images without a sign or containing signs that do not present enough samples (set empirically to 100) were discarded from the dataset. Figure 3.5 shows a bar plot related to the final dataset, which contains 3,798 images (and the same amount of traffic signs), divided into seven classes: No overtaking (1398 instances), left curve (733 instances), right curve (557 instances), 60Kmh speed limit (189 instances), 80Kmh speed limit (305 instances), trucks use right lane (433 instances) and bridge ahead (183 instances). The Brazilian traffic-sign classes are shown in Figure 3.6. In particular, traffic signs related to speed limit, no-overtaking and indication of curve ahead are very important in the context of DAS, particularly in two-way roads, where high speed and curves might be a potential combination for accidents.

One of the main difficulties encountered when generating the dataset with Google StreetView relates to images with blurred traffic signs, as shown in Figure 3.8. Visual inspection does not indicate motion or camera blur, and we believe that a possible cause could be the selective blur algorithm used by Google that is typically applied to faces, license plates, and other cues that might characterize individuals. In any event, these images were also discarded.

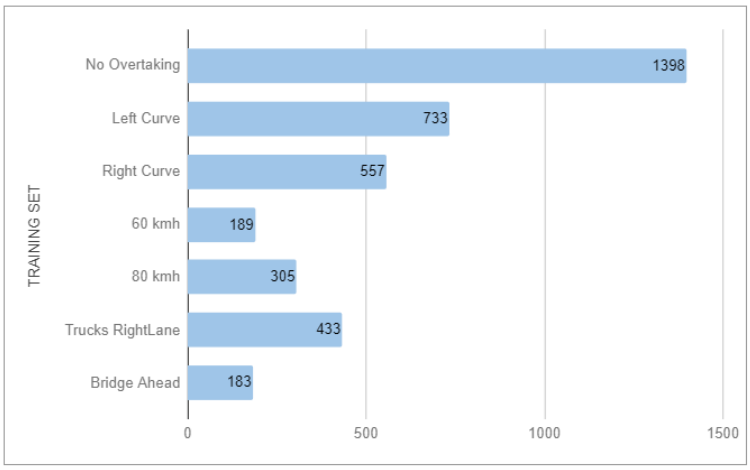


Figure 3.5 Amount of images, divided by class from our Brazilian Traffic Sign Dataset

Source: The author

Figure 3.6 All seven classes from our Brazilian Traffic Sign Dataset

Source: The author

Figure 3.7 Examples of images in the training dataset.

Source: The author

Figure 3.8 Examples of images in the training dataset with signs blurred.

Source: The author

4 THE PROPOSED APPROACH

There are different approaches in the literature, with methods for detecting and recognizing traffic signs, as shown in Chapter 2. Knowledge about the scenario, such as the expected location of road signs, as well as their size, shape, and color, can be used to simplify the problem. In this chapter, we present an approach for the recognition of traffic signals on Brazilian highways.

Figure 4.1 briefly illustrates the approach developed for the detection and recognition of traffic signs. Firstly, by using a camera coupled to the windshield of the car with known extrinsic and intrinsic parameters, as well as the information of traffic signs as position where it appear on the highway, ROIs are created to limit the search. Then, a CNN is trained with images that represent the Brazilian highways. Finally, the trained CNN trained searches for traffic signs inside the ROI projected onto the image (detection mode). When a sign is found, the search region is adjusted so that the sign is tracked every frame (tracking mode). These steps are detailed in the next sections.

Figure 4.1 Diagram illustrating the process for the recognition of traffic signals.

Source: The author

4.1 Definition of the ROIs

The core of the proposed approach is to explore a calibrated onboard camera (which can be done on-the-fly) to find Regions of Interest (ROIs) that may contain a traffic sign in the Image Coordinate System (ICS) based on the expected location of the signs in the World Coordinate System (WCS). The placement of vertical signs in each country is usually regulated by specific legislation, and this work is focused on Brazilian traffic signs.

According to the Brazilian legislation (CONTRAN, 2007), vertical traffic signs must be placed upright at an angle of 93 to 95 degrees about the direction of traffic flow and directed towards the external side of the road, or near the direction orthogonal to the central axis of the track. The height and lateral distance of placement depend on the type of road, urban or rural. For instance, for rural roads the height measured vertically from the bottom of the sign to the elevation of the near edge of the pavement, shall be 1.20m (CONTRAN, 2007), and the minimum lateral offset should be 1.20m from the shoulder.

The recommended dimensions for the sign vary according to the road type. For rural roads, circular traffic signs should have diameter of 1 meter. There are other special cases in the legislation, such as the suspended plates, which are not addressed in this paper. Figure 4.2 illustrates the main guidelines for vertical sign placement.

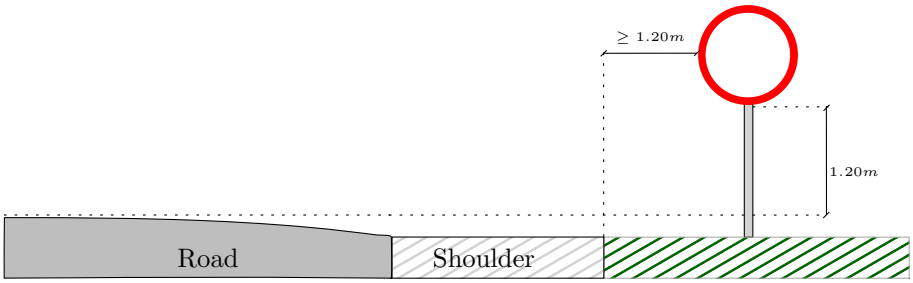


Figure 4.2 Height and lateral location of sign in rural area.

Source: The author

The proposed system is also aimed at low-cost solutions, in which a detachable camera (e.g. a smartphone) is placed at the top-central portion of the windshield, monitoring the road ahead, as illustrated in Figure 4.3. Hence, the extrinsic camera parameters are characterized by the camera height h and the yaw (α), pitch (β) and roll (γ) angles. In this work, we consider that there is no roll (since such movement is usually prevented by the windshield), and estimate the remaining parameters using the online self-calibration scheme presented in (DE PAULA; JUNG; DA SILVEIRA JR., 2014). This method assumes that the camera intrinsic parameters are known (they can be easily obtained using publicly available toolboxes such as (BOUGUET, 2008), and are always fixed if the focal length of the camera does not change), and explore the expected geometry of a flat planar road with dashed lane markings. Hence, it allows a very flexible setup, in which the user may attach the camera in a different way every time they enter the vehicle, and detach it when leaving.

Given a calibrated camera, the next step is to identify the region in the WCS where traffic signs are expected to lie on, and then project to the ICS using the known camera parameters. In the WCS, such regions are rectangles orthogonal to the ground plane and also to the central axis of the road. We define a rectangular region r, at a distance δ_z meters along the vertical axis in a birds-eye view, as shown in Figure 4.4 (left). The region is defined by its width δ_w (in meters) and height δ_h (in meters), as well as its central position $\boldsymbol{W}_z$ given by

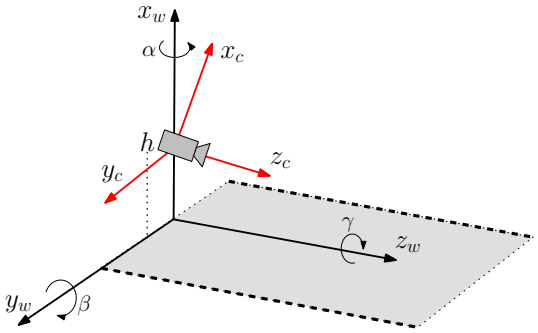


Figure 4.3 3D world and camera coordinate systems.

Source: (DE PAULA; JUNG; DA SILVEIRA JR., 2014)

$$\boldsymbol{W}_z = \begin{pmatrix} p_h + \frac{p_d}{2} \\ \delta_y \\ \delta_z \end{pmatrix}, \tag{4.1}$$

where δ_y is the approximate lateral distance from the vehicle location to the center of the sign, p_d and p_h are the expected diameter and height (computed from the ground to the bottom of the sign) of the vertical sign, respectively, and δ_z is the distance from the ROI to the camera. Figure 4.4 illustrates geometrically the parameters involved in Eq. (4.1), as well as the rectangular ROI reprojected to the ICS (in the right).

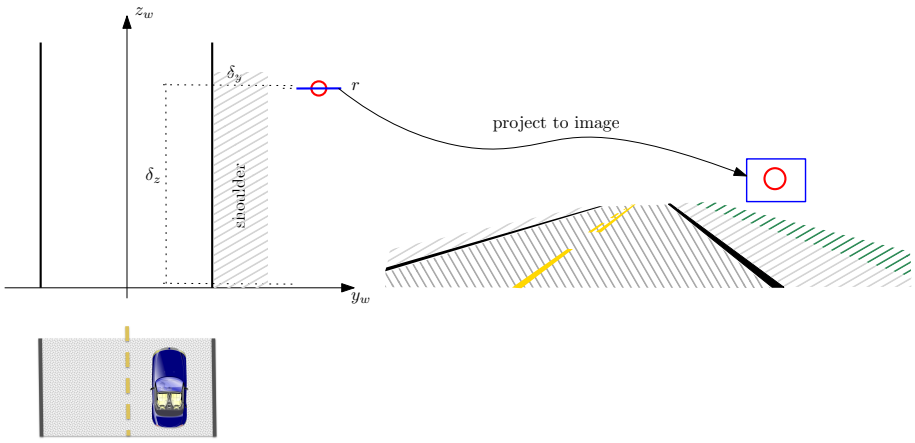


Figure 4.4 Region of interest in world coordinate system (left). Reprojection of the image regions (right).

Source: The author

The ROI clearly limits the spatial search, since just a fraction of the image is analyzed. This characteristic can be used to speed-up traditional classifiers based on sliding windows, such as (VIOLA; JONES; SNOW, 2005; DALAL; TRIGGS, 2005; DOLLAR et al., 2014), which have shown to present very good accuracy rates when trained in the context of TSR (MATHIAS et al., 2013; MOGELMOSE; LIU; TRIVEDI, 2015; CIREŞAN et al., 2012). Recognition CNNs, such as those discussed in (STALLKAMP et al., 2012), can also be employed in a sliding window fashion.

Although recent (and popular) deep CNNs based on region proposals (REN et al., 2015; GIRSHICK, 2015; REDMON et al., 2016; REDMON; FARHADI, 2016) do not rely on sliding windows, they can also benefit from the proposed ROIs. As noted in (ZHU et al., 2016), these approaches might face problems when the possible range of scales in which the object might appear (the traffic sign) is large. Within the proposed ROIs, the relative size of the traffic sign is roughly constant, alleviating the problem. Furthermore, the background variability is reduced when using the ROIs, which indicates that more shallow (and faster) CNNs can be used. The proposed CNN architecture is presented next.

4.2 The Proposed CNN

Given the image patch corresponding to the extracted ROI, the next step is to detect, localize and recognize traffic signs within the patch. Considering that the relative location and scale of the traffic sign w.r.t. to the ROI does not change significantly, we propose a simplified regional CNN that presents a good compromise between accuracy and complexity (i.e. running times), which is important for embedded applications. The chosen topology, called ScapNet, was inspired on the Fast-YOLO network, and it is described in Table 4.1. The main idea is to start from an already consolidated network (Fast-YOLO), and make changes in the network in order to reduce the cost of processing while maintaining a high performance. The input layer is fed with 104×104 RBG images (this input resolution presented a good compromise between running time and accuracy rates for traffic signs located at approximately 42 meters from the camera). The size of the proposed network ($\approx$ 4Mb) is almost sixteen times smaller than Fast-YOLO ($\approx$ 62Mb), which is one the fastest existing regional CNNs.

The detection layer locates and classifies the traffic sign, if present. Although there are dozens of vertical traffic signs according the Brazilian legislation, this work focuses mostly on roads and highways, where the most dangerous accidents occur. In this context of traffic safety, a subset of seven traffic signs related to speed limit, no-overtaking and indication of curve ahead were chosen, as discussed in Section 3.2.

It is also important to note that the relative size of the sign w.r.t. to the size of the ROI presents small variations, so that a smaller number of anchor boxes can be used. More precisely, we applied k-means clustering in our dataset and chose three anchors that can represent the ground truth boxes, the result values found with our dataset are respectively $(3.85, 6.59)$, $(4.33, 7.76)$ and $(4.80, 8.20)$.

Another key aspect when dealing with CNNs is the choice for the training dataset. However, it is not to our knowledge the existence of publicly available datasets with Brazilian traffic signs. To overcome this limitation, we collected a set of images provided by Street View API from Google Maps[1], as described in Chapter 3. Since the number of images in the training dataset is still small, training was actually performed in two stages. In the first stage, ScapNet was trained using the well-known Pascal VOC 2007-2012 dataset (EVERINGHAM et al., 2010), aiming to generalize the initial layers. Then, this pre-trained network was refined using the proposed dataset. More details about the training procedure are provided in Section 5.1.

[1] `https://developers.google.com/maps/documentation/javascript/streetview`

Type	Filters	Size/Stride	Output
Convolutional	8	3×3 / 1	104×104
Maxpool		2×2 / 2	52×52
Convolutional	16	3×3 / 1	52×52
Maxpool		2×2 / 2	26×26
Convolutional	32	3×3 / 1	26×26
Maxpool		2×2 / 2	13×13
Convolutional	64	3×3 / 1	13×13
Maxpool		2×2 / 1	13×13
Convolutional	128	3×3 / 1	13×13
Convolutional	128	3×3 / 1	13×13
Convolutional	65	1×1 / 1	13×13
Detection			

Table 4.1 The structure of ScapNet

4.3 Detection Mode

When no traffic sign was detected in the previous frames, the system operates in a "detection mode". In this mode, the ROI given by Eq. (4.1) is computed using $\delta_z = z_{max}$, where z_{max} is a fixed distance based on how far from the vehicle a traffic sign can be detected and recognized.

In other words, when dealing with video sequences, a traffic sign initially appears at the far field of the camera, and it approaches the vehicle as it moves. Based on this fact, and aiming to keep computational burden small, we decided to use a ROI placed in the far field of the camera, approximately at the limit for which a traffic sign can be detected. Figure 4.5 illustrates the detection mode running until sign is detected, then it changes to tracking mode, which is explained next.

Figure 4.5 Illustration of detection mode. ROI waits until signs appear

Source: The author

4.4 Tracking Mode

If a traffic sign is detected in a given frame, it is expected to appear in the following frames as well. Moreover, the path described by the center of the sign in the image domain is well-behaved: should be linear along straight portions of the road, and almost linear along curves (since the curvature of roads is typically small).

Although the predicted path of the sign could be estimated based on the current location, the camera parameters and the vehicle speed, we adopt a simpler approach in this work. Let $(u(t), v(t))$ denote the center of the sign detected at frame t, and $s(t)$ denote the diameter of the sign. Assuming spatial consistency between the location of the sign in adjacent frames, the search ROI at frame $t + 1$ is centered at $(u(t), v(t))$. The size of the ROI during the tracking mode is adjusted adaptively, based on the previous detected diameter. More precisely, it is given as a square region with dimensions $Ms(t) \times Ms(t)$, where M is a scaling factor (set experimentally to 3).

It is important to note that different traffic signs might appear in the same image, so that detection and tracking modes could co-exist. When a traffic sign being tracked generates a search ROI disjoint with the (fixed) ROI used in the detection mode, the proposed CNN (ScapNet) is applied to both ROIs. This situation is illustrated in Figure 4.6.

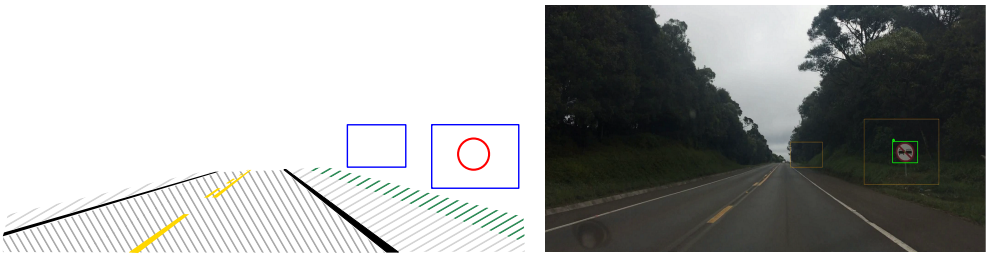


Figure 4.6 Illustration of the tracking mode: the search ROI is adjusted based on the previous detection (larger ROI in the right image). Since new traffic signs might also appear, the detection mode is also active (smaller ROI).

Source: The author

As in most problems involving object tracking, one drawback of assuming temporal continuity is that any mis-detection (false negative) in the process leads to failure in tracking. To cope with this issue, if a traffic sign is not detected at a given frame during the tracking mode, the system waits for detections in the next frames with the current ROI stagnated in its last position. This tolerance scheme is applied during md_{max} frames: after this limit, the tracking mode is aborted and the system resets to detection mode only.

5 EXPERIMENTAL RESULTS

This chapter presents the experimental results obtained with the proposed system. Here, we focus on three important aspects of practical TSR implementations: accuracy, flexibility to different cameras (resolution and location/pose), and running times.

In this chapter, we first present the protocol adpoted to train the proposed model, and then the test dataset. Next, we present the quantitative results of the proposed method and competitive techniques, as well as running times using different hardware platforms.

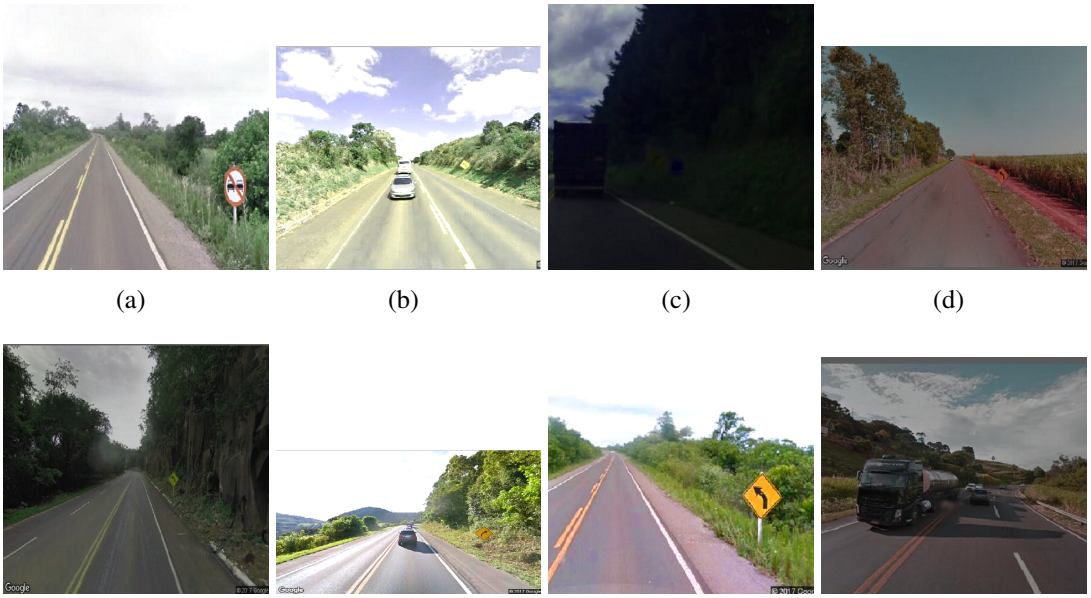
5.1 Training Details

As mentioned in Section 3.2, we have generated a dataset of Brazilian traffic signs containing 3,798 images. To cope with distortions that are not present in the training set but might appear in real scenarios (such as varying illumination conditions and motion blur), we have synthetically expanded the training set using data augmentation. As in (REDMON et al., 2016; REDMON; FARHADI, 2016), we used hue, saturation, and exposure shifts to expand the training set. Since the captured images might suffer from motion blur, we have also added smoothing with two different blur kernels (Gaussian smoothing kernels with dimensions 3×3 and 5×5). Figure 5.1 illustrates some examples of data augmentation used to train ScapNet.

Since the proposed classifier is applied to a ROI cropped from the full video frames, the training dataset must be adjusted accordingly. For that aim, we cropped the training images around the annotated signs, randomly generating regions 3 to 5 times larger than the traffic sign (so that the range of traffic scales learned is from $1/5$ to $1/3$ of the ROI size). These regions are selected centered at the annotated sign with a random artificial offset (horizontal and vertical), as illustrated by a blue rectangle in Figure 5.2.

It is also important to select image patches with no traffic sign, to avoid biasing the network. To cope this issue, we added hard negative samples by selecting patches in the rest of the image, that present pattern similar to a sign. Figure 5.3 shows images with regions selected to train the network which do not have class. With this step, we decreased the number of false positives. These regions, are selected automatically in the database, with regions similar to those shown above, however being careful that no signal was inserted.

Even with data augmentation, the training set is small for learning all the weights in ScapNet from scratch. To cope with this issue, we first trained the proposed network using the well known PASCAL VOC 2007-2012 dataset (EVERINGHAM et al., 2010), aiming to generalize the initial layers. Then, this pre-trained network was refined using the proposed dataset. We used the batch size initial learning rate of 0.001 and divided it by 10 at 100, 10 thousand and 30 thousand iterations. We also used the loss function



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.1 Examples of images with data augmentation used to train the proposed CNN.

Source: The author

defined in Fast-YOLO (REDMON; FARHADI, 2016), momentum 0.9 and batch size of 32.

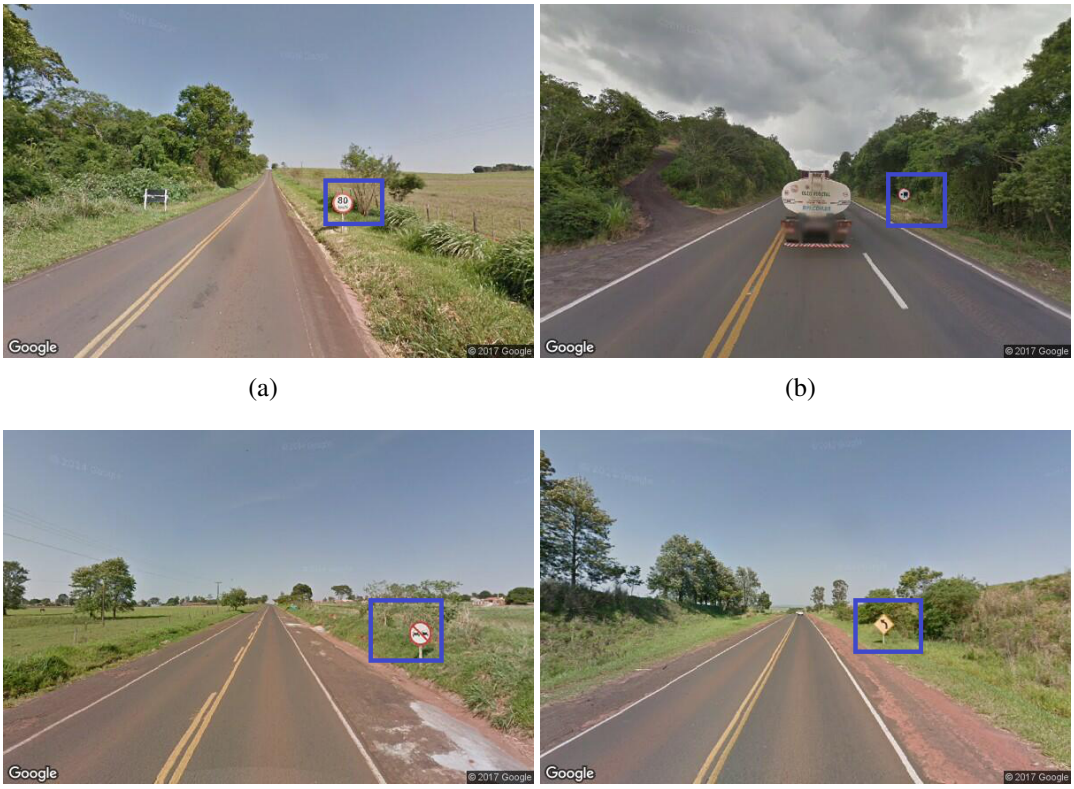
5.2 The Test Dataset

An important issue for validating any classification approach is a suitable choice for the test dataset. In this work, the goal is to perform fast and accurate detection and recognition of traffic signs in video sequences acquired with a detachable onboard camera.

The proposed Brazilian Dataset of Traffic Signs, which is based on "synthetic video sequences" acquired using the Google Maps API and used to train the model could be a possibility, but the analysis could be biased (since the test images would be similar to the training dataset). Furthermore, the intrinsic camera parameters are not known, so that the on-the-fly calibration scheme (DE PAULA; JUNG; DA SILVEIRA JR., 2014) used to obtain the extrinsic parameters and the ROIs cannot be applied.

In this work, we decided to use a set of video sequences captured by a smartphone, with different placement settings when attached to the windshield. More precisely, eleven full HD videos (with resolution of 1920×1080) were acquired using an iPhone 5s smartphone. For each video shooting session, the camera was manually attached to the windshield using a flexible smartphone holder, so that extrinsic parameters are different for each video (as previously explained, the intrinsic camera parameters were obtained offline). Table 5.1 shows the approximate vehicle speed, number of frames, number of annotated signs and also the estimated extrinsic camera parameters for each clip. The second and third sequences were acquired at different roads but at the same shooting session, which explains the same set of extrinsic parameters.

The eleven test videos captured represent the conditions found when traveling in Brazilian rural roads. The videos contain a set of frame sequences with all the classes



(a) (b)

(c) (d)

Figure 5.2 Examples of selected regions used to train ScapNet. Each region example has a labeled sign.

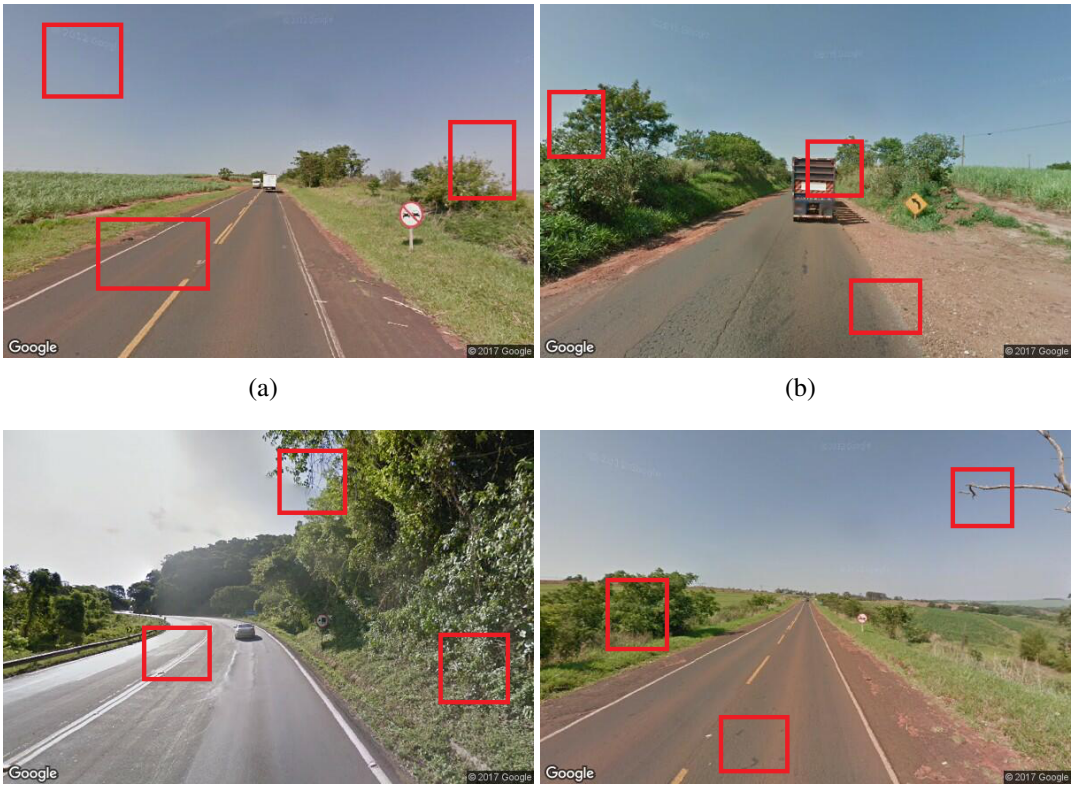
Source: The author

of traffic signs used in this work. Table 5.2 presents in detail the number of instances of each class in the test videos (one sign per frame). In fact, the "no overtaking" class presents more samples than all other classes, since it was the most frequent sign in the roads when the videos were recorded. Fortunately, this is possibly the most important sign to be detected, since it indicates portions of the road where overtaking is dangerous. In total, 2,588 signs were annotated in the test dataset. As in the train dataset, they were divided into seven classes, and each instance was labeled along with the corresponding bounding box.

5.3 Quantitative Evaluation

The proposed system was implemented in C++, using the Open source Computer Vision library (OpenCV) and the darknet (REDMON; FARHADI, 2016)/caffe (JIA et al., 2014) frameworks for the CNN classification. To evaluate running times, we used three different hardwares: i) a desktop computer with 3.40 GHz Intel Core i7-2600 CPU, 8GB RAM; ii) an embedded hardware with a 1.9 GHz Quad Core ARM Cortex-A15 CPU; iii) a Raspberry Pi 3 Model B hardware, with a 1.2 GHz Quad Core ARM BCM2837 CPU with 1GB RAM.

We tested our algorithm using all video sequences in our test dataset, evaluating the results in terms of precision and recall rates, as well as runtime. For the sake of com-



(a) (b)

(c) (d)

Figure 5.3 Examples of selected regions used to train ScapNet. Each image has three selected region without sign.

Source: The author

parison, we also ran Fast-YOLO (REDMON; FARHADI, 2016), Faster-RCNN (GIRSHICK, 2015) and the CNN-based traffic sign recognition approach proposed in (ZHU et al., 2016), called CNN-sign. Since these methods were not trained to detect Brazilian traffic signs, we only used the topology of these networks and re-trained them similarly to the procedure used for ScapNet: we pre-trained these networks with the PASCAL VOC 2007-2012 dataset from scratch, and then fine-tuned them with our training dataset. However, we used the full image frames when training these networks, instead of the proposed ROIs, since they were designed to process full frames.

In all experiments, the dimensions of the initial ROI (detection mode) were $\delta_h = 2.25$ meters high and $\delta_w = 3.25$ meters wide. Based on (CONTRAN, 2007), we selected $p_d = 1.0$ meters (diameter of circular traffic sign) and p_h = 1.2 meters (length of the sign stem), and the lateral offset from the vehicle to the sign was set to $\delta y = 3.75$ meters.

Based on the dimensions and center of the ROI, as given in Eq. (4.1), the ROI was projected onto the image plane using $z_{max} = 42$ meters, which is approximately the farthest detection distance. Furthermore, we used $md_{max} = 5$ frames as the temporal tracking tolerance.

To validate our experiments, a detection is considered correct if the Intersection over Union (IoU) given by

$$IoU(A, B) = \frac{\#(A \cap B)}{\#(A \cup B)} \tag{5.1}$$

Clip Name	Speed	Frames	Signs	h	α	β
clip_i5s_0789	20km/h	935	382	1.05m	-5.54°	1.48°
clip_i5s_0094	80km/h	2160	131	1.22m	6.00°	-0.85°
clip_i5s_0099	80km/h	628	90	1.22m	6.00°	-0.85°
img_5159	40km/h	449	167	1.09m	5.88°	0.69°
img_5160	60km/h	1534	196	1.04m	5.57°	1.44°
img_5164	50km/h	1641	408	1.14m	6.78°	-0.55°
img_5167	60km/h	1559	259	1.15m	4.36°	-0.47°
img_5169	40km/h	1353	290	1.08m	-3.98°	1.16°
img_5171	60km/h	336	107	1.20m	-6.21°	0.98°
img_5172	50km/h	473	276	1.24m	-5.19°	0.78°
img_5173	60km/h	1020	282	1.25m	-5.76°	0.57°

Table 5.1 Extrinsic parameters of the test dataset

	No Overtaking	Left Curve	Right Curve	60 kmh	80 kmh	Trucks Right	Bridge	Total
clip_i5s_0789	382							382
clip_i5s_0094	131							131
clip_i5s_0099	90							90
img_5159						167		167
img_5160	196							196
img_5164	186	222						408
img_5167	90						169	259
img_5169	186			104				290
img_5171			107					107
img_5172	88		188					276
img_5173				122	160			282
Total	1349	222	295	226	160	167	169	2588

Table 5.2 Instances of each class in the test videos

is greater than a threshold T_{IoU} (we used $T_{IoU} = 0.5$), where A and B denote the detection and ground truth bounding boxes, and $\#$ is the cardinality of a set.

Table 5.3 shows the precision-recall results and processing times (in different hardwares) for the proposed method (ScapNet) and competitive techniques considering all processed frames, while Table 5.5 shows individual results for each video clip. These values are computed per-class, meaning that if a sign of a given class "A" is correctly detected and localized in the image but recognized as class "B", it counts both as a false negative for class "A" and a false positive for class "B".

It can be observed that the runtime for our technique is significantly lower than the others in all tested hardwares (PC, Cortex-A15 and Raspberry Pi 3), achieving 20 FPS for the PC just using the CPU, with the second best precision (very low false positive rate), and recall rate of 96.32% considering all video sequences (with a minimum per-clip recall of 91.32%). The results using Faster-RCNN are the overall best in both precision and recall, but the execution time is very high, making it unfeasible to run on embedded devices with low processing power. We were unable to complete the training of Faster-RCNN (as well as CNN-sign) on the Raspberry 3 hardware due to excessive time and equipment overheating. Table 5.4 shows the individual time values (per frame, in seconds) obtained for each clip in the test dataset.

Figure 5.5 illustrates some (cropped) frames from the test video sequences with correct results of the proposed approach. In particular, Figure 5.5(a) shows a true negative

Figure 5.4 Examples of images in the test dataset.

Source: The author

	Precision	Recall	Processing speed (FPS)		
			PC	Cortex-A15	Rasp3
Faster-RCNN	**97.61%**	**97.36%**	0.054	0.044	–
CNN-sign	95.56%	93.84%	0.315	0.253	–
TinyYolo	95.55%	94.30%	0.397	0.323	0.027
ScapNet	97.18%	94.51%	**20.290**	**17.575**	**1.687**

Table 5.3 Precision-recall results and processing times.

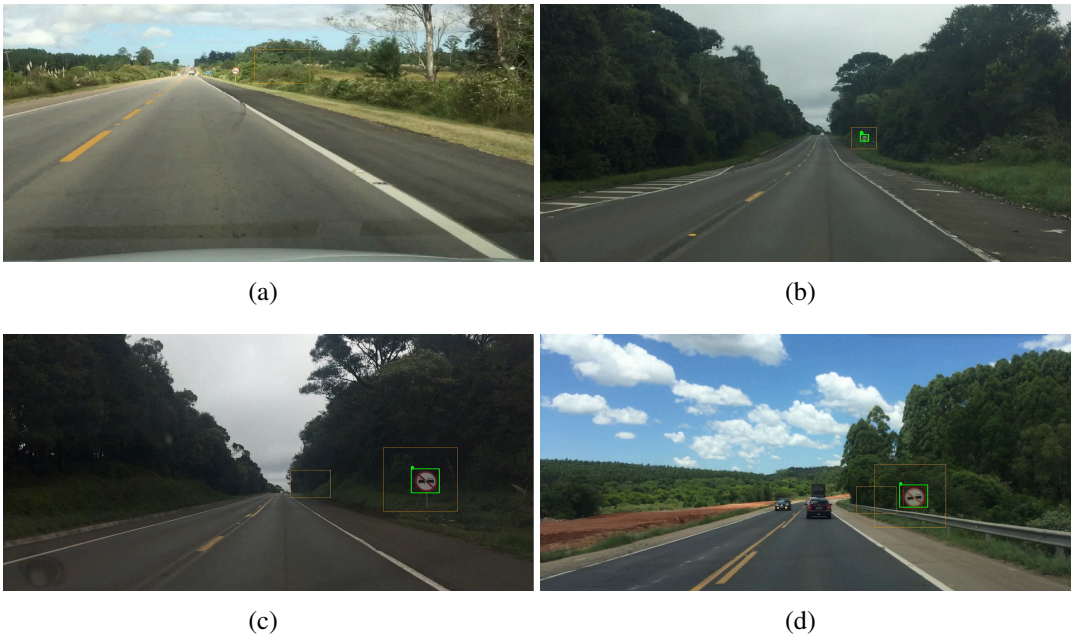
result, in which the detection ROI (yellow) is in the far field, and contains no sign (the no-overtaking sign present in the frame is farther from the ROI, and gets detected after a few frames). Figures 5.5(b)-(c) and (d) illustrate true positive results (note the second ROI in the far field waiting for others signs).

Figure 5.6 illustrates some (cropped) frames from the test video sequences examples when the proposed approach failed. In particular, Figure 5.6(a) shows a (soft) false negative, in which a 60km/h speed limit traffic sign was correctly detected, but misclassified as 80km/h. Figures 5.6(b)-(c) as the sign is too small, it is not located by CNN, although it was within the ROI. In both cases, the sign is detected by the ScapNet in the next frames, when the car gets closer to sign. Figure 5.6(d) presents a detection of a sign not present in the set of annotated classes that was classified as “No Overtaking”.

Finally, Table 5.6 shows the overall confusion matrix of the proposed method considering all videos together. It can be observed that 90% of the false positives relate to traffic signs that were correctly detected and localized, but misclassified. Considering all classes, the number of “actual” false positives (detection of any sign where there is none) is very small, as shown in the last line of the table. The number of actual false negatives (a traffic sign that is not even detected) considering all classes is smaller than 4%, as shown

	Faster-RCNN		CNN-Sign		TinyYolo			ScapNet		
	PC	Cortex-A15	PC	Cortex-A15	PC	Cortex-A15	Rasp3	PC	Cortex-A15	Rasp3
clip_i5s_0789	0.0542	0.0442	0.3132	0.2620	0.4051	0.3212	0.0263	20.6876	17.9521	1.5758
clip_i5s_0094	0.0539	0.0438	0.3156	0.2453	0.3950	0.3305	0.0261	20.8623	17.6875	1.5823
clip_i5s_0099	0.0538	0.0433	0.3211	0.2512	0.3972	0.3256	0.0262	20.3478	18.0563	1.5683
img_5159	0.0540	0.0430	0.3183	0.2497	0.3946	0.3346	0.0261	20.3564	17.2486	1.5767
img_5160	0.0535	0.0438	0.3164	0.2536	0.3894	0.3327	0.0263	19.8765	17.9531	1.5838
img_5164	0.0539	0.0438	0.3059	0.2574	0.4012	0.3275	0.0263	20.1953	17.1563	1.5825
img_5167	0.0528	0.0442	0.3170	0.2461	0.4035	0.3246	0.0263	20.5195	17.6023	1.5836
img_5169	0.0538	0.0430	0.3233	0.2563	0.3994	0.3312	0.0263	19.4587	17.5965	1.5832
img_5171	0.0548	0.0441	0.3165	0.2536	0.3980	0.3289	0.0263	19.8523	17.0932	1.9985
img_5172	0.0539	0.0440	0.3060	0.2515	0.3915	0.3324	0.0294	20.3598	17.5498	2.2695
img_5173	0.0547	0.0443	0.3163	0.2578	0.3975	0.3305	0.0270	20.6785	17.4325	1.6508

Table 5.4 Average processing framerate (frames per second) of our technique and competitive approaches in each of the test videos



(a) (b)

(c) (d)

Figure 5.5 Examples of correct classification results obtained by our method. The yellow rectangles illustrate the ROIs and green rectangle our detection.

in the last column of the table. The highest confusion rate occurs between the two speed limit signs, which is expected due to the visual similarity between them (in particular when they are in the far field, which leads to small regions in the image domain).

It is important to note that the most confusions occurs with similar signs, such as speed limit signs. In addition, most errors occur when the signs are in the far field. This occurs because the network is able to correctly detect, but not to recognize the corresponding class when the sign is too small in relation to the whole image. Figure 5.7 presents examples of traffic signs in far field, (cropped from the image - with a larger size, original approximately has 24×24 pixels). In this scenario, the network detects the sign but does not have enough information to recognize it.

Analyzing the errors of our approach, we can observe that approximately 90% of errors occur when detections of traffic signs are smaller than 48×48 pixels. The graph of

Clip Name	ScapNet		Fast-Yolo		Faster-RCNN		CVPR2016	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
clip_i5s_0789	**99.47%**	98.17%	98.95%	98.17%	98.96%	**98.96%**	97.13%	97.12%
clip_i5s_0094	98.44%	96.18%	98.47%	98.46%	**100.00%**	**100.00%**	98.46%	97.71%
clip_i5s_0099	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
img_5159	98.10%	92.81%	93.21%	90.42%	**98.77%**	**96.99%**	95.68%	92.81%
img_5160	98.96%	96.94%	99.48%	96.94%	**100.00%**	**100.00%**	98.47%	98.47%
img_5164	**99.24%**	96.32%	97.79%	97.06%	96.83%	**97.30%**	95.05%	94.12%
img_5167	91.80%	90.72%	91.83%	91.12%	**93.41%**	**93.05%**	87.76%	83.01%
img_5169	95.82%	94.82%	94.79%	94.14%	**96.88%**	**96.21%**	92.68%	91.10%
img_5171	**95.96%**	88.79%	94.06%	88.79%	95.33%	**96.32%**	95.15%	91.59%
img_5172	96.17%	90.94%	96.67%	94.57%	**97.79%**	**96.38%**	94.44%	92.39%
img_5173	94.98%	93.97%	85.76%	87.59%	95.74%	**95.74%**	**96.36%**	93.97%

Table 5.5 Results for the proposed approach and the baseline methods for all test videos.

(a) (b)

(c) (d)

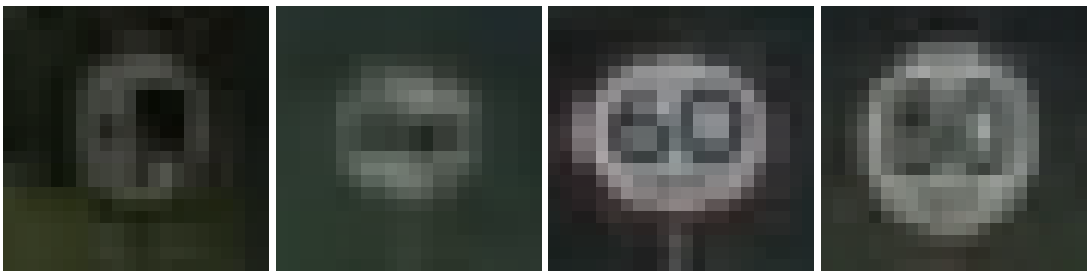
Figure 5.6 Incorrect examples of classification results obtained by our method. The yellow rectangles illustrate the ROIs and green rectangle our detection. (a) and (d) "Soft" false negative, meaning that a sign was detected but wrongly classified. (b)-(c) True false negative (sign not detected).

Source: The author

the Figure 5.8 presents the number of erroneous occurrences in relation to the size of the traffic sign detection, obtained by our approach. It shows that, as previously mentioned, errors occur exactly in the far field, where it is difficult to distinguish visually. However, as the sign approach the car, the errors stop happening.

	No Overtaking	Left Curve	Right Curve	60 kmh	80 kmh	Trucks Right	Bridge	Miss - FN
No Overtaking	1322	0	0	2	0	2	0	23
Left Curve	0	209	3	0	0	0	0	10
Right Curve	0	14	258	0	0	0	0	23
60 kmh	0	0	0	207	18	0	0	1
80 kmh	0	0	0	8	148	0	0	4
Trucks Right	0	0	0	2	1	155	0	9
Bridge	0	6	7	0	0	0	147	9
Nothing - FP	2	0	0	0	0	5	0	0

Table 5.6 Confusion matrix for our technique.



(a) (b) (c) (d)

Figure 5.7 Examples of signs in far field, showing the difficulty to recognize the corresponding class.

Source: The author

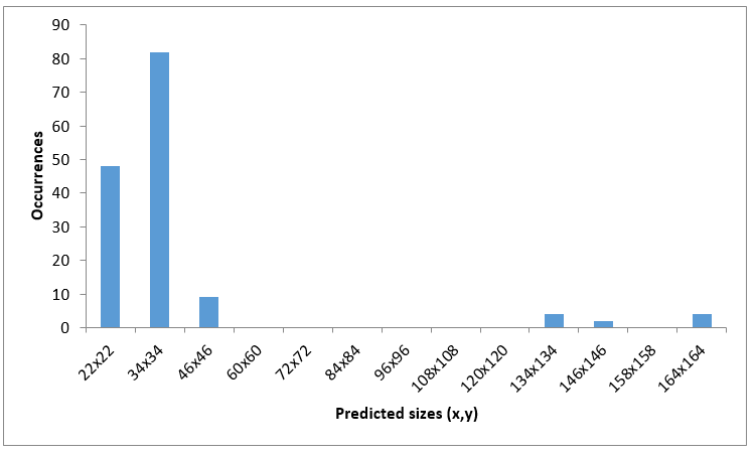


Figure 5.8 Predicted traffic sign size in cases of error

Source: The author

6 CONCLUSIONS

The research presented in this work proposed a framework for Brazilian TSR with a flexible camera setup (detachable camera installed on the interior of the windshield) that allows a good compromise between accuracy and running times. The core idea was to explore the given extrinsic and intrinsic camera parameters, together with the information of the Brazilian traffic signs based on their expected location in the world coordinate system, to define regions of interest (ROIs) in the image.

Within the proposed ROIs, the background complexity is reduced and the relative size of the sign present small variations, so that lighter (more shallow) CNNs might be adequate to detect and recognize signs. Based on this observation, a new and light CNN, called ScapNet, was proposed to locate and recognize Brazilian traffic signs within the extracted ROIs.

The proposed CNN was trained with captured data that represent Brazilian highways, as well as climate changes that influence illumination. In addition, the proposed CNN presents a much smaller number of parameters compared to state-of-the-art networks, while maintaining similar accuracy.

The proposed approach was tested on different hardware setups, and our results indicated that embedded harware with lower processing power can be used for processing. More precisely, the experimental results showed that the proposed TSR approach can run at over 17 FPS on an embedded Cortex-A15 processor for full HD video sequences, with precision and recall rates comparable to or better than more complex (and state-of-the-art) regional CNNs, which achieve less than 0.5 FPS on the same hardware when processinf full-frame images. These results indicate that the proposed approach presents potential for over 30 FPS on embedded devices that contain CPU and GPUs, such as NVIDIA's Jetson TX1.

As an additional but still important contribution of this work, we have collected and annotated two distinct datasets with Brazilian traffic signs: the first one is composed of 3,798 frames obtained from Google Street View, containing 3,798 traffic signs in the total (one per image), divided and annotated into seven different classes most common on Brazilian highways, which are: No overtaking, Left Curve, Right Curve, Limit Speed 60km/h, Limit Speed 80km/h, Trucks Right Lanea and Bridge Ahead. The second dataset is composed of 11 short full HD video clips, with 12,088 frames in the total and 2,588 annotated Brazilian signs.

6.1 Future Work

As future work, we plan to further expand our test database by collecting more video sequences, and then making it publicly available along with the vehicle speed and cam-

era parameters. Besides new images, we intend to collect new classes of traffic signs, expanding the coverage of Brazilian traffic signs. Moreover, the training dataset will be increased by running the application that captures images through the Google Street View API.

Together with new classes of acquired traffic signs, we also plan to expand ScapNet, until the network is able to generalize all the classes with the lowest computational cost, evaluating its performance using embedded CPU+GPU processors.

Another improvement that is planned, is to optimize the updating of Regions of Interest in the tracking mode, using the known car speed and the camera parameters. In addition, using this information, the traffic sign could be predicted in future frames and used to improve validations using temporal coherence.

REFERENCES

ANGUELOV, D. et al. Google Street View: capturing the world at street level. **Computer**, [S.l.: s.n.], v.43, 2010.

BASCÓN, S. M. et al. An optimization on pictogram identification for the road-sign recognition task using SVMs. **Computer Vision and Image Understanding**, [S.l.: s.n.], v.114, n.3, 2010, p.373–383.

BERKAYA, S. K. et al. On circular traffic sign detection and recognition. **Expert Systems with Applications**, [S.l.: s.n.], v.48, 2016, p.67–75.

BOUGUET, J. Y. **Camera calibration toolbox for Matlab**. 2008.

BRUNO, D. R.; OSÓRIO, F. S. Image classification system based on Deep Learning applied to the recognition of traffic signs for intelligent robotic vehicle navigation purposes. **Robotics Symposium**, [S.l.: s.n.], 2017.

CANNY, J. A Computational Approach to Edge Detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.: s.n.], v.PAMI-8, n.6, 1986, p.679–698.

CHEN, L.-C. et al. Rethinking atrous convolution for semantic image segmentation. **arXiv preprint arXiv:1706.05587**, [S.l.: s.n.], 2017.

CHIANG, H.-H. et al. Road speed sign recognition using edge-voting principle and learning vector quantization network. In: COMPUTER SYMPOSIUM (ICS), 2010 INTERNATIONAL. **Proceedings...** [S.l.: s.n.], 2010. p.246–251.

CIREŞAN, D. et al. Multi-column deep neural network for traffic sign classification. **Neural Networks**, [S.l.: s.n.], v.32, 2012, p.333–338.

CONTRAN. **Manual Brasileiro de Sinalização de Trânsito**. 2ª.ed. Brasília: [s.n.], 2007. v.I.

CREUSEN, I. et al. Color exploitation in hog-based traffic sign detection. In: IMAGE PROCESSING (ICIP), 2010 17TH IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. p.2669–2672.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: COMPUTER VISION AND PATTERN RECOGNITION, 2005. CVPR 2005. IEEE COMPUTER SOCIETY CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2005. v.1, p.886–893 vol. 1.

DE PAULA, M. B.; JUNG, C. R.; DA SILVEIRA JR., L. G. Automatic On-the-fly Extrinsic Camera Calibration of Onboard Vehicular Cameras. **Expert Syst. Appl.**, Tarrytown, NY, USA, v.41, n.4, Mar. 2014, p.1997–2007.

DEGUCHI, D. et al. Intelligent traffic sign detector: adaptive learning based on online gathering of training samples. In: INTELLIGENT VEHICLES SYMPOSIUM (IV), 2011 IEEE. **Proceedings...** [S.l.: s.n.], 2011. p.72–77.

DOLLAR, P. et al. Fast feature pyramids for object detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.: s.n.], v.36, n.8, 2014, p.1532–1545.

EICHNER, M.; BRECKON, T. Integrated speed limit detection and recognition from real-time video. In: INTELLIGENT VEHICLES SYMPOSIUM, 2008 IEEE. **Proceedings...** [S.l.: s.n.], 2008. p.626–631.

ELLAHYANI, A.; EL ANSARI, M.; EL JAAFARI, I. Traffic sign detection and recognition based on random forests. **Applied Soft Computing**, [S.l.: s.n.], v.46, 2016, p.805–815.

ELVIK, M. R. et al. Social and economic consequences of road traffic injury in Europe. **Brussels, Belgium**, [S.l.: s.n.], 2007.

EVERINGHAM, M. et al. The Pascal Visual Object Classes (VOC) Challenge. **Int. J. Comput. Vision**, Hingham, MA, USA, v.88, n.2, June 2010, p.303–338.

FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient graph-based image segmentation. **International journal of computer vision**, [S.l.: s.n.], v.59, n.2, 2004, p.167–181.

FISTREK, T.; LONCARIC, S. Traffic sign detection and recognition using neural networks and histogram based selection of segmentation method. In: ELMAR, 2011 PROCEEDINGS. **Proceedings...** [S.l.: s.n.], 2011. p.51–54.

FUHR, G.; JUNG, C. Camera Self-calibration Based on Non-Linear Optimization and Applications in Surveillance Systems. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.: s.n.], v.PP, n.99, 2015, p.1–1.

GAO, X. W. et al. Recognition of traffic signs based on their colour and shape features extracted using human vision models. **J. Visual Communication and Image Representation**, [S.l.: s.n.], v.17, n.4, 2006, p.675–685.

GIRSHICK, R. Fast r-cnn. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.: s.n.], 2015. p.1440–1448.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2014. p.580–587.

GOMEZ-MORENO, H. et al. Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition. **Intelligent Transportation Systems, IEEE Transactions on**, [S.l.: s.n.], v.11, n.4, 2010, p.917–930.

GREENHALGH, J.; MIRMEHDI, M. Real-time detection and recognition of road traffic signs. **Intelligent Transportation Systems, IEEE Transactions on**, [S.l.: s.n.], v.13, n.4, 2012, p.1498–1506.

GUDIGAR, A.; CHOKKADI, S.; RAGHAVENDRA, U. A review on automatic detection and recognition of traffic sign. **Multimedia Tools and Applications**, [S.l.: s.n.], v.75, n.1, 2016, p.333–364.

HE, K. et al. Mask r-cnn. **arXiv preprint arXiv:1703.06870**, [S.l.: s.n.], 2017.

HERON-DELANEY, M. et al. A systematic review of predictors of posttraumatic stress disorder (PTSD) for adult road traffic crash survivors. **Injury**, [S.l.: s.n.], v.44, n.11, 2013, p.1413 – 1422.

HOELSCHER, I. G. **Deteccao e Classificacao de sinalizacao vertical de transito em cenarios complexos**. 2017. dissertation — Universidade Federal do Rio Grande do Sul.

HOIEM, D.; EFROS, A. A.; HEBERT, M. Putting Objects in Perspective. **International Journal of Computer Vision**, Hingham, MA, USA, v.80, n.1, 2008, p.3–15.

HOUBEN, S. A single target voting scheme for traffic sign detection. In: INTELLIGENT VEHICLES SYMPOSIUM (IV), 2011 IEEE. **Proceedings...** [S.l.: s.n.], 2011. p.124–129.

HOUBEN, S. et al. Detection of Traffic Signs in Real-World Images: the German Traffic Sign Detection Benchmark. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS. **Proceedings...** [S.l.: s.n.], 2013. n.1288.

HUVAL, B. et al. An empirical evaluation of deep learning on highway driving. **arXiv preprint arXiv:1504.01716**, [S.l.: s.n.], 2015.

JIA, Y. et al. Caffe: convolutional architecture for fast feature embedding. , [S.l.: s.n.], 2014, p.675–678.

KONCAR, A.; JANSSEN, H.; HALGAMUGE, S. Gabor wavelet similarity maps for optimising hierarchical road sign classifiers. **Pattern Recognition Letters**, [S.l.: s.n.], v.28, n.2, 2007, p.260–267.

KUO, W.-J.; LIN, C.-C. Two-Stage Road Sign Detection and Recognition. In: MULTIMEDIA AND EXPO, 2007 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2007. p.1427–1430.

KUO, W.-J.; LIN, C.-C. Two-stage road sign detection and recognition. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO. **Proceedings...** [S.l.: s.n.], 2007. p.1427–1430.

LI, H. et al. A novel traffic sign detection method via color segmentation and robust shape matching. **Neurocomputing**, [S.l.: s.n.], v.169, 2015, p.77–88.

LIN, T.-Y. et al. Microsoft coco: common objects in context. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.: s.n.], 2014. p.740–755.

LIU, W. et al. Ssd: single shot multibox detector. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.: s.n.], 2016. p.21–37.

MATAS, J. et al. Robust wide-baseline stereo from maximally stable extremal regions. **Image and vision computing**, [S.l.: s.n.], v.22, n.10, 2004, p.761–767.

MATHIAS, M. et al. Traffic sign recognition – How far are we from the solution? In: NEURAL NETWORKS (IJCNN), THE 2013 INTERNATIONAL JOINT CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2013. p.1–8.

MENG, Z. et al. Detecting Small Signs from Large Images. **arXiv preprint arXiv:1706.08574**, [S.l.: s.n.], 2017.

MOGELMOSE, A.; LIU, D.; TRIVEDI, M. Detection of U.S. Traffic Signs. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.: s.n.], v.16, n.6, 2015, p.3116–3125.

MOGELMOSE, A.; TRIVEDI, M.; MOESLUND, T. Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: perspectives and survey. **Intelligent Transportation Systems, IEEE Transactions on**, [S.l.: s.n.], v.13, n.4, 2012, p.1484–1497.

NGUWI, Y.-Y.; KOUZANI, A. Z. Detection and classification of road signs in natural environments. **Neural Computing and Applications**, [S.l.: s.n.], v.17, n.3, 2008, p.265–289.

PRIETO, M. S.; ALLEN, A. R. Using self-organising maps in the detection and recognition of road signs. **Image and Vision Computing**, [S.l.: s.n.], v.27, n.6, 2009, p.673–683.

PRISACARIU, V. A. et al. Integrating Object Detection with 3D Tracking Towards a Better Driver Assistance System. In: ICPR. **Proceedings...** [S.l.: s.n.], 2010. p.3344–3347.

QINGSONG, X.; JUAN, S.; TIANTIAN, L. A detection and recognition method for prohibition traffic signs. In: INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND SIGNAL PROCESSING. **Proceedings...** [S.l.: s.n.], 2010.

REDMON, J. et al. You only look once: unified, real-time object detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2016. p.779–788.

REDMON, J.; FARHADI, A. YOLO9000: better, faster, stronger. **arXiv preprint arXiv:1612.08242**, [S.l.: s.n.], 2016.

REN, F. et al. General traffic sign recognition by feature matching. In: IMAGE AND VISION COMPUTING NEW ZEALAND. **Proceedings...** [S.l.: s.n.], 2009. p.409–414.

REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.: s.n.], 2015. p.91–99.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, [S.l.: s.n.], v.115, n.3, 2015, p.211–252.

RUTA, A.; LI, Y.; LIU, X. Real-time traffic sign recognition from video by class-specific discriminative features. **Pattern Recognition**, New York, NY, USA, v.43, n.1, Jan. 2010, p.416 – 430.

SALTI, S. et al. Traffic sign detection via interest region extraction. **Pattern Recognition**, [S.l.: s.n.], v.48, n.4, 2015, p.1039–1049.

SALTI, S.; LANZA, A.; DI STEFANO, L. Keypoints from symmetries by wave propagation. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2013. p.2898–2905.

STALLKAMP, J. et al. Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. **Neural networks**, [S.l.: s.n.], v.32, 2012, p.323–332.

SUS, D. de informática do. DATASUS. In: OF THE . **Proceedings...** [S.l.: s.n.], 2017.

TIMOFTE, R.; ZIMMERMANN, K.; GOOL, L. V. Multi-view traffic sign detection, recognition, and 3D localisation. In: APPLICATIONS OF COMPUTER VISION (WACV), 2009 WORKSHOP ON. **Proceedings...** [S.l.: s.n.], 2009. p.1–8.

TIMOFTE, R.; ZIMMERMANN, K.; VAN GOOL, L. Multi-view traffic sign detection, recognition, and 3d localisation. **Machine Vision and Applications**, [S.l.: s.n.], v.25, n.3, 2014, p.633–647.

UIJLINGS, J. R. et al. Selective search for object recognition. **International journal of computer vision**, [S.l.: s.n.], v.104, n.2, 2013, p.154–171.

VIOLA, P.; JONES, M. J.; SNOW, D. Detecting pedestrians using patterns of motion and appearance. **International Journal of Computer Vision**, [S.l.: s.n.], v.63, n.2, 2005, p.153–161.

WANG, G. et al. A hierarchical method for traffic sign classification with support vector machines. In: NEURAL NETWORKS (IJCNN), THE 2013 INTERNATIONAL JOINT CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2013. p.1–6.

WHO. **Global status report on road safety 2013**: supporting a decade of action. [S.l.]: World Health Organization, 2013.

XIE, Y. et al. Unifying visual saliency with HOG feature learning for traffic sign detection. In: INTELLIGENT VEHICLES SYMPOSIUM, 2009 IEEE. **Proceedings...** [S.l.: s.n.], 2009. p.24–29.

YANG, Y. et al. Towards real-time traffic sign detection and classification. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.: s.n.], v.17, n.7, 2016, p.2022–2031.

YANG, Y.; WU, F. Real-time traffic sign detection via color probability model and integral channel features. In: CHINESE CONFERENCE ON PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2014. p.545–554.

ZAKLOUTA, F.; STANCIULESCU, B. Real-time traffic sign recognition in three stages. **Robotics and autonomous systems**, [S.l.: s.n.], v.62, n.1, 2014, p.16–24.

ZHANG, J.; MA, S.; SAMEKI. Salient Object Subitizing. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.: s.n.], 2015.

ZHANG, K.; SHENG, Y.; LI, J. Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. **Intelligent Transport Systems, IET**, [S.l.: s.n.], v.6, n.3, 2012, p.282–291.

ZHU, Z. et al. Traffic-Sign Detection and Classification in the Wild. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.: s.n.], 2016.