



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Monhel Maudoonny Pierre

**Recognition of Brazilian vertical traffic signs and lights from a
car using Single Shot Multi box Detector**

Uberlândia

2023

Monhel Maudoony Pierre

**Recognition of Brazilian vertical traffic signs and lights from a car using
Single Shot Multi box Detector**

Final thesis for the post-graduation course of Computer Science of the Universidade Federal de Uberlândia for the degree of Master in Computer Science.
Supervisor: Prof. Dr. Henrique Coelho Fernandes

Uberlândia
2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

P622 Pierre, Monhel Maudoony, 1994-
2023 Recognition of Brazilian vertical traffic signs and
lights from a car using Single Shot Multi box Detector
[recurso eletrônico] / Monhel Maudoony Pierre. - 2023.

Orientador: Henrique Coelho Fernandes.
Dissertação (Mestrado) - Universidade Federal de
Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: <http://doi.org/10.14393/ufu.di.2023.525>
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Fernandes, Henrique Coelho, 1986-,
(Orient.). II. Universidade Federal de Uberlândia. Pós-
graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
 Coordenação do Programa de Pós-Graduação em Ciência da Computação
 Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
 Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpqfacom@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

| | | | | | |
|------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------|-------|-----------------------|-------|
| Programa de Pós-Graduação em: | Ciência da Computação | | | | |
| Defesa de: | Dissertação de Mestrado 24/2023, PPGCO | | | | |
| Data: | 19 de setembro de 2023 | Hora de início: | 13:00 | Hora de encerramento: | 14:30 |
| Matrícula do Discente: | 12122CCP036 | | | | |
| Nome do Discente: | Monhel Maudoony Pierre | | | | |
| Título do Trabalho: | Recognition of Brazilian vertical traffic signs and lights from a car using Single Shot Multi box Detector | | | | |
| Área de concentração: | Ciência da Computação | | | | |
| Linha de pesquisa: | Ciência de Dados | | | | |
| Projeto de Pesquisa de vinculação: | - | | | | |

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Jefferson Rodrigo de Souza - FACOM/UFU, Ahmad Osman - Hochschule für Technik und Wirtschaft des Saarlandes - Alemanha e Henrique Coelho Fernandes - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Ahmad Osman - Saarbrücken/Alemanha, Jefferson Rodrigo de Souza e Henrique Coelho Fernandes - Uberlândia/MG. A discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos, o presidente da mesa, Prof. Dr. Henrique Coelho Fernandes, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

Ressalta-se que o examinador Ahmad Osman por ser estrangeiro, residente em outro país e não possuir CPF registrado no Brasil não assinará a ata de defesa.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Henrique Coelho Fernandes, Professor(a) do Magistério Superior**, em 19/09/2023, às 16:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Jefferson Rodrigo de Souza, Professor(a) do Magistério Superior**, em 19/09/2023, às 17:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4832645** e o código CRC **79CE1D2A**.

This work is dedicated to my parents Monine Destin and
Ismael Pierre, for their supports and confidence at all
times ever since my childhood until now.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Professor Dr. Henrique Coelho Fernandes for his invaluable support and guidance throughout this work. His expertise and knowledge have been instrumental in shaping the outcome of this research.

I extend my sincere appreciation to my friends from Haiti who have been with me during my Master's journey. Their unwavering friendship, encouragement, and companionship have been a source of strength and inspiration. Together, we have overcome challenges and embarked on numerous projects, fostering growth and mutual success.

I am also grateful to my friends and colleagues at UFU (Federal University of Uberlândia) for their camaraderie and enriching interactions over the years. Their contributions and collaborations have greatly enhanced my academic experience.

I would like to acknowledge the professors of the Master's program for sharing their knowledge and experiences, which have been pivotal in shaping my understanding and academic growth.

Lastly, I express my profound gratitude to God for granting me the opportunity and blessing to achieve this significant milestone in my academic journey.

Gostaria de expressar minha sincera gratidão ao Professor Dr. Henrique Coelho Fernandes por seu inestimável apoio e orientação ao longo deste trabalho. Sua experiência e conhecimento foram fundamentais para moldar o resultado desta pesquisa.

Estendo minha sincera gratidão aos meus amigos do Haiti que estiveram comigo durante minha jornada de Mestre. Sua amizade inabalável, incentivo e companheirismo têm sido uma fonte de força e inspiração. Juntos, superamos desafios e embarcamos em inúmeros projetos, fomentando o crescimento e o sucesso mútuo.

Agradeço também aos meus amigos e colegas da UFU (Universidade Federal de Uberlândia) pela camaradagem e interações enriquecedoras ao longo dos anos. Suas contribuições e colaborações melhoraram muito minha experiência acadêmica.

Gostaria de agradecer aos professores do programa de mestrado por compartilharem seus conhecimentos e experiências, que foram fundamentais para moldar minha compreensão e crescimento acadêmico.

Por fim, expresso minha profunda gratidão a Deus por me conceder a oportunidade e a bênção de alcançar este marco significativo em minha jornada acadêmica.

“Imagination is more important than knowledge. Knowledge is limited while imagination encompasses the whole world, stimulates progress, stimulates evolution.”
(Albert Einstein)

ABSTRACT

This document presents a system for recognizing Brazilian traffic signs and lights using artificial intelligence. The main objective of the system is to contribute to road safety by alerting drivers to potential risks such as speeding, alcohol consumption, and cell phone use, which could lead to severe accidents and jeopardize lives. The system's core contribution lies in its ability to accurately detect and classify various traffic signs and lights, providing crucial warnings to drivers to prevent potential hazards. To achieve this, the system used the light version of the Single Shot Multibox Detector called SSD-Lite using Mobilenet version 2 and Mobilenet version 3 as base networks for feature extraction. The optimal Mobilenet version was selected based on performance evaluations to ensure a Mean Average Precision (mAP) higher than 80%, which guarantees reliable detection results. The dataset used for training and evaluation comprises images extracted from YouTube traffic videos, each meticulously annotated to create the necessary labels for model training. Through this extensive experimentation, the system demonstrates its efficacy in achieving accurate and efficient traffic sign and light detection. The results of the experiments are compared with other existing approaches that focus on detecting only one type of traffic sign or employ different network types. The proposed system outperforms these comparative works, showcasing its superiority in handling various traffic sign and light classes by providing a dedicated dataset for Brazilian traffic sign and light

Keywords: Artificial Intelligence. MobileNet. SSD. Traffic Signs

LIST OF FIGURES

| | |
|----------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1 – Brazilian vertical traffic signs. Extracted from (PALMIERI, 2021). | 17 |
| Figure 2 – Brazilian traffic lights. Extracted from (AUTOMOTIVO, 2022). | 17 |
| Figure 3 – Example of a regulatory sign. Extracted from (PALMIERI, 2021). | 22 |
| Figure 4 – Example of a warning sign. Extracted from (PALMIERI, 2021). | 22 |
| Figure 5 – Example of a guide sign. Extracted from (PALMIERI, 2021). | 23 |
| Figure 6 – Example of a information sign. Extracted from (PALMIERI, 2021). | 24 |
| Figure 7 – Example of a temporary sign. Extracted from (PALMIERI, 2021). | 25 |
| Figure 8 – R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEMB- HURNE, 2023). | 26 |
| Figure 9 – R-FCN architecture. Extracted from (DAI et al., 2016). | 27 |
| Figure 10 – SPP-Net architecture. Extracted from (HE et al., 2014). | 27 |
| Figure 11 – Fast R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEM- BHURNE, 2023). | 28 |
| Figure 12 – Faster R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEM- BHURNE, 2023). | 28 |
| Figure 13 – YOLO architecture. Extracted from (DIWAN; ANIRUDH; TEMB- HURNE, 2023). | 29 |
| Figure 14 – SSD architecture. Extracted from (JEE et al., 2021). | 30 |
| Figure 15 – CNN architecture. Source: (The authors, 2023). | 31 |
| Figure 16 – Mobilenet architecture. Extracted from (PALMIERI, 2021). | 33 |
| Figure 17 – Sample of the traffic signs after the pre-processing. Extracted From (ALGHMUGHAM et al., 2019). | 38 |
| Figure 18 – Results of the hierarchical model on images from Los Angeles, United States. Extracted From (PON et al., 2018). | 39 |
| Figure 19 – The system overview of the author. Extracted From (HOELSCHER, 2017). | 39 |
| Figure 20 – The classes used by the authors. Extracted From (SILVA, F. A. d. et al., 2020). | 40 |
| Figure 21 – Indian traffic sign dataset used by the Authors. Extracted From (BHATT; LALDAS; LOBO, 2022). | 40 |
| Figure 22 – Block diagram of the traffic sign recognition system. Extracted From (WALI et al., 2019). | 41 |
| Figure 23 – Testing architecture. Extracted From (WILLIAM et al., 2019). | 42 |
| Figure 24 – Image annotation. Extracted From (DALBORGO et al., 2023). | 42 |
| Figure 25 – Detection examples on the TT100K testing set by the authors. Ex- tracted From (CHEN et al., 2021). | 43 |
| Figure 26 – Classes used by the authors. Extracted From (ZHU; YAN, 2022). | 44 |

| | |
|---------------------------------------------------------------------------------------------------------|----|
| Figure 27 – Confusion matrix on the GTSRB dataset. Extracted From (ZHU; YAN, 2022). | 45 |
| Figure 28 – Proposed method of the authors. Extracted From (FREDJ et al., 2023). | 46 |
| Figure 29 – Methodology used in our work. Source: (The authors, 2023). | 49 |
| Figure 30 – The classes of the created dataset. Adapted from (ALGHMGMHAM et al., 2019). | 50 |
| Figure 31 – Labeling tool for image labeling. Source: (The authors, 2023). | 51 |
| Figure 32 – Dataset class distribution. Source: (The authors, 2023). | 52 |
| Figure 33 – Example of the augmentation methods on one image. Source: (The authors, 2023). | 53 |
| Figure 34 – Intersection over union formula. Extracted from (PADILLA; NETTO; SILVA, E. A. B. da, 2020). | 58 |
| Figure 35 – System overview. Adapted from (ALGHMGMHAM et al., 2019). | 59 |
| Figure 36 – Loss for input size 128. Source: (The Authors, 2023). | 64 |
| Figure 37 – Accuracy for input size 128. Source: (The Authors, 2023). | 65 |
| Figure 38 – Loss for input size 320. Source: (The authors, 2023). | 65 |
| Figure 39 – Accuracy for input size 320. Source: (The authors, 2023). | 65 |
| Figure 40 – Loss for input size 512. Source: (The authors, 2023). | 66 |
| Figure 41 – Loss for input size 512. Source: (The authors, 2023). | 66 |
| Figure 42 – Second and third experiment loss for input size 320. Source: (The authors, 2023). | 66 |
| Figure 43 – Second and third experiment accuracy for input size 320. Source: (The authors, 2023). | 67 |
| Figure 44 – Example of detection from video. Source: (The authors, 2023). | 69 |
| Figure 45 – Example of detection from video. Source: (The authors, 2023). | 70 |
| Figure 46 – Example of detection from video. Source: (The authors, 2023). | 70 |
| Figure 47 – Example of detection from video. Source: (The authors, 2023). | 71 |
| Figure 48 – Example of detection from video. Source: (The authors, 2023). | 71 |
| Figure 49 – Traffic light with a new color in Brazil. Extracted from (RIBAS, 2023). | 75 |

LIST OF TABLES

| | |
|----------------------------------------------------------------------------------------------------------|----|
| Table 1 – SSD-Lite regression/classification heads. | 55 |
| Table 2 – SSD-Lite extra layers. | 55 |
| Table 3 – SSD-Lite extraction feature layers from mobilenet v2. | 61 |
| Table 4 – Hyperparameter list. | 63 |
| Table 5 – Number of parameters for each model. | 63 |
| Table 6 – Training results for every input size where the threshold value for the IoU is 0.5. | 64 |
| Table 7 – Accuracy of every class using the best model with the test data. | 67 |

LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|--------------|---------------------------------------------------|
| ABRAMET | Associação Brasileira de Medicina de Tráfego |
| AdaGrad | Adaptive Gradient Algorithm |
| AI | Artificial Intelligence |
| AP | Average Precision |
| AUC-PR | Area Under the Precision-Recall Curve |
| BN | Batch Normalization |
| BTSC | Belgian Traffic Sign Dataset |
| CIFAR | Canadian Institute for Advanced Research |
| CM | Confusion Matrix |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| Faster R-CNN | Faster Region-based Convolutional Neural Networks |
| Fast R-CNN | Fast Region-based Convolutional Neural Networks |
| FC | Fully Connected |
| FN | False Negative |
| FP | False Positive |
| FPS | Frames Per Second |
| GB | Giga Byte |
| GiB | Giga Byte |
| GTSRB | German Traffic Sign Recognition Benchmark |
| IoU | Intersection Over Union |
| mAP | Mean Average Precision |
| ML | Machine Learning |
| ONNX | Open Neural Network Exchange |
| OpenCV | Open Source Computer Vision Library |
| PASCAL VOC | PASCAL Visual Object Classes Challenge |
| RAM | Random Access Memory |
| R-CNN | Region-based Convolutional Neural Networks |
| ReLU | Rectified Linear Unit |
| ResNet | Residual Neural Network |
| R-FCN | Region-based Fully Convolutional Network |
| RGB | Red Green Blue |
| RMSProp | Root Mean Squared Propagation |
| RoI | Region of Interest |
| RPN | Region Proposal Network |
| RPPROP | Resilient backpropagation |
| SGD | Stochastic Gradient Descent |

| | |
|---------|-----------------------------------------------|
| SPP-Net | Spatial Pyramid Pooling Network |
| SSD | Single Shot MultiBox Detector |
| TGL | The Graphics Library |
| TL | Transfer Learning |
| TN | True Negative |
| TP | True Positive |
| TSDR | Traffic Sign Detection and Recognition system |
| TT100K | Tsinghua-Tencent 100k |
| UHD | Ultra High Definition |
| VGG | Visual Geometry Group |
| WHO | World Health Organization |
| YOLO | You Only Look Once |

CONTENTS

| | | |
|----------------|-----------------------------------------------|-----------|
| 1 | INTRODUCTION | 16 |
| 1.1 | MOTIVATION | 17 |
| 1.2 | RESEARCH OBJECTIVES AND CHALLENGES | 18 |
| 1.2.1 | General Objective | 18 |
| 1.2.2 | Specific objectives | 18 |
| 1.3 | HYPOTHESIS | 19 |
| 1.4 | CONTRIBUTIONS | 19 |
| 1.5 | GENERAL OUTLINE OF THE DOCUMENT | 19 |
| 1.6 | PARTIAL CONCLUSION | 20 |
| 2 | FUNDAMENTALS | 21 |
| 2.1 | TRAFFIC SIGN | 21 |
| 2.1.1 | Category | 21 |
| 2.2 | ARTIFICIAL INTELLIGENCE | 25 |
| 2.3 | COMPUTER VISION | 25 |
| 2.3.1 | Image recognition | 26 |
| 2.3.2 | Object recognition | 26 |
| 2.3.3 | Object detection types | 26 |
| 2.3.4 | Machine learning | 30 |
| <i>2.3.4.1</i> | <i>Artificial Neural Network</i> | <i>30</i> |
| 2.3.5 | Deep learning | 30 |
| <i>2.3.5.1</i> | <i>Convolutional neural network</i> | <i>30</i> |
| <i>2.3.5.2</i> | <i>Transfer learning</i> | <i>31</i> |
| <i>2.3.5.3</i> | <i>Dropout</i> | <i>31</i> |
| <i>2.3.5.4</i> | <i>Learning rate</i> | <i>31</i> |
| <i>2.3.5.5</i> | <i>Layers</i> | <i>31</i> |
| <i>2.3.5.6</i> | <i>ReLU</i> | <i>32</i> |
| <i>2.3.5.7</i> | <i>Pooling</i> | <i>32</i> |
| <i>2.3.5.8</i> | <i>Residual layer</i> | <i>32</i> |
| 2.3.6 | Mobilenet | 33 |
| <i>2.3.6.1</i> | <i>Mobilenet Architecture</i> | <i>33</i> |
| 2.4 | DATASET | 33 |
| 2.4.1 | Data augmentation technique | 33 |
| 2.5 | TRAINING | 34 |
| 2.6 | PERFORMANCE INDICATOR | 36 |
| 2.6.1 | SSD components | 36 |
| 2.7 | PARTIAL CONCLUSION | 37 |
| 3 | RELATED WORKS | 38 |

| | | |
|-------|----------------------------------------------|-----------|
| 3.1 | FINAL REMARKS | 46 |
| 4 | METHODOLOGY | 48 |
| 4.1 | DATA COLLECTION AND ANNOTATION | 49 |
| 4.1.1 | Data collection | 49 |
| 4.1.2 | Data annotation | 50 |
| 4.1.3 | Data augmentation | 51 |
| 4.2 | MODEL SELECTION | 53 |
| 4.2.1 | SSD-Lite Base network | 54 |
| 4.3 | TRAINING AND VALIDATION | 55 |
| 4.3.1 | Hardware and software Setup | 56 |
| 4.3.2 | Evaluation metrics | 56 |
| 4.3.3 | Intersection over Union | 57 |
| 4.3.4 | Mean Average Precision | 58 |
| 4.3.5 | Proposed Approach | 58 |
| 4.3.6 | Data Preparation | 59 |
| 4.4 | PARTIAL CONCLUSION | 60 |
| 5 | EXPERIMENTS | 62 |
| 5.1 | TRAINING PROCESS | 62 |
| 5.2 | RESULTS EVALUATION | 63 |
| 5.3 | DISCUSSION | 68 |
| 5.4 | PARTIAL CONCLUSION | 72 |
| 6 | CONCLUSIONS | 73 |
| 6.1 | CONTRIBUTIONS | 73 |
| 6.2 | FUTURE WORK | 74 |
| | REFERENCES | 76 |

1 INTRODUCTION

Vision is a very important ability for humans, allowing us to interact with and learn much from our environment, the reason why scientists are trying to replicate it for computer systems using computer vision. According to (KARN, 2021) computer vision is about teaching computers to recognize the objects in our lives using a camera as the input for images that will then be processed.

With the development of society, in the case of road traffic, we have some rules, and these rules are largely indicated by road signs. Looking at these signs correctly is very important, because if we do not interpret them correctly, it can lead to some life-threatening facts in the case of car traffic and that's why there are traffic signs that help drivers navigate in traffic with other cars and people (WONTORCZYK; GACA, 2021).

In recent years, driver assistance systems with image sensors, such as monocular and stereo cameras, have gained importance and contributed to pedestrian and driver safety. In particular, the use of a vehicle-mounted rear-view camera is gradually increasing. A camera with a wide field of view that overcomes the driver's limited vision can contribute to this goal, and the use of a fish-eye camera is becoming more common (MIN SU KIM et al., 2016).

To achieve the goal of establishing an assistance system, object detection is used by identifying and locating objects in images or videos, and there are a variety of use cases for object recognition models in different industries and sectors such as medicine, retail, and agriculture (SINGH, 2021). For example, according to (PA, 2020) , MobileNet, which is a family of mobile-first computer vision models, is a simplification of neural networks to enable their use in web applications and mobile devices that allow us to quickly build an image recognition application using very less memory.

With the possibility of mounting a special camera in a car and a classification model for image recognition, an algorithm can be developed to recognize traffic signs whose images are provided by a camera positioned in front of a car, and this work consists precisely in recognizing vertical traffic signs and traffic lights along with other work to reduce the number of accidents and help computers interact better with their environment. Figure 1 and Figure 2 illustrate the Brazilian vertical traffic signs from which we will work and an example of the three states of the traffic lights found in the images of our dataset.



Figure 1 – Brazilian vertical traffic signs. Extracted from (PALMIERI, 2021).



Figure 2 – Brazilian traffic lights. Extracted from (AUTOMOTIVO, 2022).

1.1 MOTIVATION

According to (WHO, 2022), traffic accident results in the death or disability of approximately 1.3 million people every year in the world. Speeding, driving under influence of alcohol or any other psychoactive substances, distracted drivers using mobile phones are some of the main reasons why those accidents occur. As a result, between 20 and 50 million more people are injured, with many of them suffers from injuries that lead to disabilities.

In addition to putting the driver's life at risk, the recklessness of the driver can affect other people, inside and outside the vehicle. After all, pedestrians and other vehicles can be involved in the accident resulting in the death of innocent people (ELAINA, 2021).

Brazil is the third country with the most traffic deaths in the world, according to data from the Global Status Report on Road Safety, by the World Health Organization. Deaths resulting from traffic accidents are the one of the leading cause of death in the country. In 2020, 32,716 people died in traffic accidents in Brazil, which means that, on average, three people die in traffic every hour (GIOVANNA, 2022).

The most common factors that caused accidents in 2022 are lack of attention, driver drowsiness, ingestion of alcohol and/or drugs and sudden illness at the wheel, in that order. The Brazilian Association for Traffic Medicine (ABRAMET) counts for 567 deaths on federal highways from January to July 2022 related to those causes. In the same period of 2021, there were 544. The increase is just over 4%. Only in the 9th position of the ranking appears a factor related to the condition of the road, which is the accumulation of water on the pavement (LAFORE, 2022).

One of the important causes for this high number of accidents could be related to the increasing dependence of humans to cell phones and their use behind the wheel, mortality rate from road traffic accidents in Brazil was 19.7 deaths per 100,000 inhabitants in 2016 (BAZILIO et al., 2022).

According to the mentioned motivations many accidents take place because of the lack of focus from the drivers. But, is it possible to reduce traffic car accidents by creating a detection algorithm whose images come from a camera placed inside of a car considering both vertical traffic signs and traffic lights? To help with answering that question below is the objective of the present work.

1.2 RESEARCH OBJECTIVES AND CHALLENGES

This section presents the objective of the work separated into general and specific objectives in the sections below.

1.2.1 General Objective

The objective of this work is to develop an approach for the recognition of vertical traffic signs and lights, in order to develop a driver assistance system for an autonomous vehicle, by adapting it to the federal regulation of colors and shapes in Brazil.

1.2.2 Specific objectives

- Create a dataset of typical Brazilian vertical traffic signs and traffic lights explained in section 4;

- Apply an algorithm that can be used on a mobile system using a model that consumes less resources with a good performance and an acceptable accuracy;
- Achieve high accuracy and low false positive and false negative rates;
- Bring an increased understanding of real-world driving conditions by collecting and analyzing data from a vehicle driving in various real-world conditions that will help researchers to better understand the challenges and opportunities presented by these conditions, and how they may affect the performance of a detection system.

1.3 HYPOTHESIS

Our hypothesis is that by implementing a combination of image processing techniques and machine learning algorithms with a small convolutional neural network, it is possible to develop a system for accurately detecting Brazilian vertical traffic signs and lights in real-time from a camera mounted on a moving vehicle and one could assist in reducing the number of traffic accidents on Brazilian roads by alerting the driver.

1.4 CONTRIBUTIONS

This section addresses the contributions of this work, which are as follows:

- The assembly of a Brazilian dataset from the vertical traffic signs and traffic lights with images available on the Internet;
- An increased understanding of real-world driving conditions by collecting and analyzing data from a vehicle driving in various real-world conditions, which is how this work could contribute to our understanding of these conditions and how they may affect the performance of a detection system;
- The present work will help reinforce the computer vision field by giving the opportunity to take in consideration new images in the learning detection process and more specifically in the Brazilian context;
- The development of a light and fast algorithm for embedded systems for the recognition of traffic signs and lights.

1.5 GENERAL OUTLINE OF THE DOCUMENT

The remainder of this document is organized as follows:

- In chapter 2 we present the concepts of object detection, traffic signs and lights and artificial intelligence as well as some techniques important to the understanding of the work;
- In chapter 3 we present some works related to our work;

- Chapter 4 states the methodology of this project, including the image acquisition process and the dataset used with the metrics used for model evaluation;
- Chapter 5 describes the experimental results obtained, discussions and comparisons between the results obtained;
- Finally, in chapter 6 we present the conclusions, the contributions of the work, as well as some suggestions for future work.

1.6 PARTIAL CONCLUSION

This chapter serves as an introduction to the research problem addressed in our study, which primarily stems from driver inattention. It highlights the significance of developing an algorithm to address this issue and outlines the key motivations driving our research. The ultimate goal is to create a driver assistance system for autonomous vehicles that can effectively recognize vertical traffic signs and traffic lights while adhering to the color and shape regulations mandated by the Brazilian federal regulations.

In addition to providing an overview of the research problem and objectives, this chapter also emphasizes the main contributions of our work. Furthermore, it presents the research hypothesis that guides our investigation. Lastly, the chapter provides a brief overview of the remaining five chapters that comprise the complete document, outlining the structure and content of each chapter.

2 FUNDAMENTALS

In this chapter, we delve into the fundamental concepts necessary for a comprehensive understanding of this work. We begin by providing definitions and explanations of key concepts related to Artificial Intelligence (AI). Subsequently, we explore the structure and functioning of neural networks, which serve as the backbone of many AI applications. Furthermore, we touch upon the history and technological advancements in the field of deep learning, with a specific focus on object detection. By covering these topics, we aim to lay the groundwork for the subsequent chapters, where we delve deeper into the specific application of object detection for our research purposes.

2.1 TRAFFIC SIGN

As per the grammar definition provided by (ENGLISH GRAMMAR HERE, 2020), traffic signs are regulatory guidelines that dictate where, when, and at what speed one can drive. These signs play a crucial role in ensuring a smooth flow of traffic and ensuring the safety of drivers and their passengers. They provide instructions on various aspects, such as lane usage, right-of-way, and parking. Traffic signs are categorized into different types, as outlined below.

2.1.1 Category

According to the Road Traffic Regulations (GOVERNMENT, 2022), traffic signs can be classified into various categories, each serving a specific purpose. These categories include regulatory signs, warning signs, guide signs, and information signs. Below is a brief overview of each category:

- **Regulatory signs:** Regulatory signs are an essential component of traffic control systems as they communicate specific traffic laws and regulations to road users. These signs serve to inform drivers of important instructions and restrictions that must be followed for safe and orderly traffic flow. Common examples of regulatory signs include speed limit signs, stop signs, yield signs, and various other signs indicating specific actions or prohibitions.

Regulatory signs are typically designed in a rectangular or square shape, allowing for clear visibility and easy recognition on the road. The standardized shape and color schemes of these signs contribute to their effectiveness in conveying the intended messages to drivers. By displaying concise and universally understood symbols or text, regulatory signs help ensure compliance with traffic laws and promote overall road safety.



Figure 3 – Example of a regulatory sign. Extracted from (PALMIERI, 2021).

- Warning signs: Warning signs play a crucial role in ensuring road safety by notifying drivers of potential hazards or specific conditions that they should be aware of while traveling. These signs serve as early warnings, allowing road users to take appropriate precautions and adjust their driving behavior accordingly.

Warning signs cover a wide range of situations and can include signs indicating construction zones, detours, sharp curves, steep grades, narrow bridges, or any other factors that may pose a potential risk. To enhance visibility and distinguish them from other types of signs, warning signs are typically shaped like diamonds. This distinct shape, combined with bold colors and clear symbols or text, ensures that these signs stand out and capture the attention of drivers, enabling them to anticipate and respond to potential dangers effectively. By providing advance warning and promoting cautious driving, warning signs contribute to minimizing accidents and enhancing overall road safety.



Figure 4 – Example of a warning sign. Extracted from (PALMIERI, 2021).

- Guide signs: Guide signs serve as essential navigational aids, providing road users

with valuable information to help them navigate the road network effectively. These signs offer guidance by indicating the names of roads, directions to major destinations, and the distances to those destinations. By providing clear and concise information, guide signs assist drivers in making informed decisions about their route and reaching their desired destinations.

Guide signs come in various shapes, depending on the type of information they convey. For instance, signs indicating the name of a road or highway typically have a rectangular shape, while signs providing directional information may have an arrow-like shape. The use of different shapes helps road users quickly identify the purpose and relevance of each sign, enhancing their ability to follow the intended route and make informed decisions while driving.

With their informative nature and distinct shapes, guide signs play a crucial role in assisting road users with navigation, ensuring smoother and more efficient travel experiences.



Figure 5 – Example of a guide sign. Extracted from (PALMIERI, 2021).

- Information signs: Information signs play a vital role in providing road users with valuable information about various services and facilities available near the road. These signs serve as guides, helping drivers locate and access essential amenities and services during their journey. Examples of information signs include those indicating rest areas, gas stations, parking areas, food establishments, lodging facilities, and more.

By displaying clear and recognizable symbols or icons, these signs enable drivers to quickly identify the availability of specific services or facilities along their route. This information is particularly helpful during long trips or when drivers are in unfamiliar areas, as it allows them to plan their stops and make necessary arrangements conveniently.

Information signs are designed to be easily noticeable and readable, ensuring that road users can quickly interpret the displayed information while maintaining their focus on the road. These signs typically feature universally recognized symbols or concise textual descriptions to convey the type of service or facility being advertised. This standardized approach enhances the effectiveness of these signs and promotes safe and efficient travel by providing drivers with the necessary information to meet their needs during their journey.



Figure 6 – Example of a information sign. Extracted from (PALMIERI, 2021).

- Temporary signs: Temporary traffic control signs play a crucial role in informing road users about construction activities or temporary changes in traffic patterns. These signs are specifically designed to ensure the safety and efficiency of traffic flow in and around construction zones. They serve as visual cues to alert drivers of detours, lane closures, or other temporary traffic control measures that may be in place.

Examples of temporary traffic control signs include detour signs, which guide drivers along alternative routes when their regular route is temporarily closed or impassable. Lane closure signs inform drivers of reduced or restricted lanes due to construction work. Signs indicating flagger-controlled traffic notify drivers that traffic flow is being managed by a flagger at a specific point, requiring them to follow their directions.

To enhance their visibility and recognition, temporary traffic control signs often feature bold and vibrant colors, such as orange or yellow, and are equipped with reflective materials to ensure visibility during both daytime and nighttime conditions. These signs typically have a standardized design, incorporating clear symbols, arrows, and text to convey the necessary information quickly and effectively.

By providing clear and concise information about temporary traffic control or construction activities, these signs contribute to the overall safety of both drivers and construction workers, minimizing confusion and potential hazards in the vicinity of work zones.



Figure 7 – Example of a temporary sign. Extracted from (PALMIERI, 2021).

2.2 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) encompasses the ability of computer programs or machines to demonstrate intelligent behavior, acquire knowledge from data, and carry out tasks that traditionally necessitate human intelligence. It is an expansive discipline within computer science dedicated to the advancement and construction of intelligent systems.

According to (AMISHA et al., 2019), AI involves the construction of intelligent machines capable of thinking and learning. It encompasses a range of techniques and approaches designed to empower computers to execute intricate tasks and make intelligent choices.

According to (HAENLEIN et al., 2019), AI is characterized as the capacity of a system to effectively comprehend external data, acquire knowledge from it, and utilize that knowledge to accomplish predefined objectives and tasks. This definition emphasizes the significance of data-driven learning, which serves as a fundamental element in our project.

In the context of this study, the definition provided by (HAENLEIN et al., 2019) is particularly relevant due to its emphasis on learning from data, which aligns with the specific objectives of our project. However, other definitions, such as the one by Science and Intelligence (2019), also emphasize the goal of making computers intelligent and highlight the role of computer vision as a subset of AI, which we will explore further in the following section.

2.3 COMPUTER VISION

Computer vision refers to the capability of computers to recognize patterns in images, enabling them to perceive and understand visual information similar to humans. (XU et al., 2021) describe it as the ability of computers to see and detect objects in images. This field of study is an integral part of artificial intelligence.

According to (BROWNLEE, 2019), computer vision is a specific area that focuses on teaching computers to see and learn from digital images. By analyzing and interpreting visual data, computers can gain insights and extract meaningful information.

Both definitions are relevant to the objectives of this study. Our goal is to utilize computer vision techniques and algorithms to analyze digital images, specifically traffic

signs, in order to detect and classify them accurately. The subsequent section will provide further details about the specific aspects of our research related to traffic sign recognition.

2.3.1 Image recognition

Image recognition, in the context of machine vision, is the ability of software to identify objects, places, people in images and assign a single, high-level label to that image by analyzing and interpreting the image's pixel patterns (EWAN, 2019).

2.3.2 Object recognition

According to (TAN; LE, 2019), object recognition is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos. Below we'll see some different types of object detection.

2.3.3 Object detection types

CNN-based object detection methods can be grouped into two genres: one-stage and two-stage. The two-stage methods first extract region proposals and then classify and regress each proposal to achieve detection results. The mainstream two-stage methods include R-CNN, SPPNet, Fast R-CNN, Faster R-CNN, etc. But the two-stage approaches incur a lot of computational costs. One-stage methods discard the stage of generating region proposals, in order to accelerate the inference speed and achieve real-time detection. The representative of one-stage methods includes YOLO, SSD, and RetinaNet. According to (CHOUDHURY, 2020), below are some object detection used in computer vision.

- Region Based Convolutional Neural Networks (R-CNN): R-CNN was proposed by (GIRSHICK et al., 2013) and obtained a mean average precision (mAP) of 53.3 % with more than 30 % improvement over the previous best result on PASCAL VOC 2012. It improves the quality of candidate bounding boxes and uses deep architecture to extract high-level features.

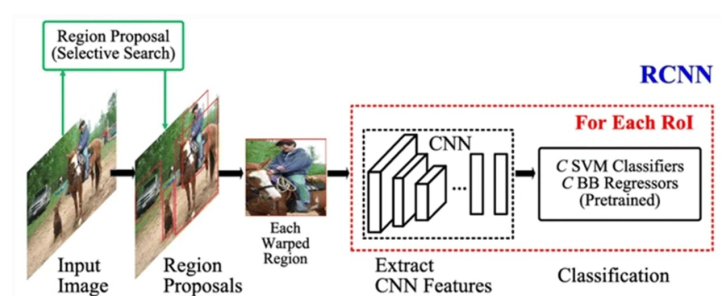


Figure 8 – R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEMBHURNE, 2023).

- Region-based Fully Convolutional Networks (R-FCN): R-FCN is a region-based detector for object detection. Unlike other region-based detectors that apply a costly per-region sub-network such as Fast R-CNN or Faster R-CNN, this region-based detector is fully convolutional with almost all computation shared on the entire image. R-FCN consists of shared, fully convolutional architectures as is the case of FCN that is known to yield a better result than the Faster R-CNN. In this algorithm, all learnable weight layers are convolutional and are designed to classify the ROIs into object categories and backgrounds.

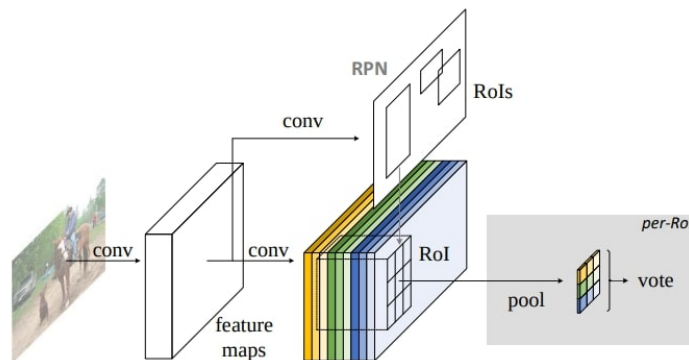


Figure 9 – R-FCN architecture. Extracted from (DAI et al., 2016).

- Spatial Pyramid Pooling in Deep Convolutional Networks (SPP-Net): In R-CNN, due to the existence of FC layers, CNN requires a fixed size input, and due to this R-CNN crops each region proposal into the same size. It may happen that objects may partially appear in the wrapped region and also unwanted geometric distortion may be produced due to wrapping operation. These content losses or distortions will reduce recognition accuracy, especially when the scales of objects vary (HE et al., 2014).

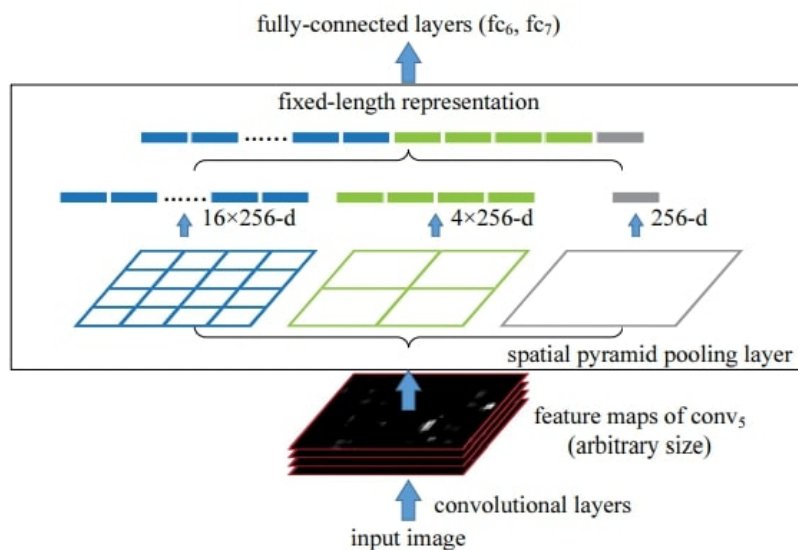


Figure 10 – SPP-Net architecture. Extracted from (HE et al., 2014).

- Fast Region-based Convolutional Neural Network (Fast R-CNN): To tackle the above problems, (GIRSHICK, 2015) introduced a multi-task loss on classification and bounding box regression by proposing a novel CNN architecture named Fast R-CNN. In Fast R-CNN, firstly the whole image is processed with standard convolution architecture like VGG16 to produce a feature map, this step is similar to SPP-Net and after that, a fixed-length feature vector is extracted from each region proposal with a region of interest (RoI) pooling layer.

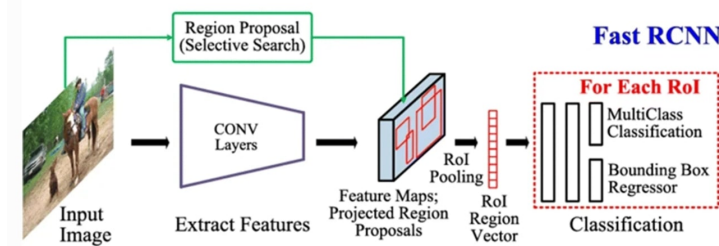


Figure 11 – Fast R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEMBHURNE, 2023).

- Faster Region-based Convolutional Neural Network (Faster R-CNN): To solve this problem, (REN et al., 2015) introduced an additional Region Proposal Network (RPN), which acts in a nearly cost-free way by sharing full-image conv features with detection networks i.e instead of using a selective search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals. RPN is achieved with a fully convolutional network, which has the ability to predict object bounds and scores at each position simultaneously. RPN takes an image of arbitrary size to generate a set of rectangular object proposals. The important point here is to note that RPN operates on a specific conv layer with the preceding layers shared with the object detection network. In other words, to generate “proposals” for the region where the object lies, a small network is slid over a convolutional feature map that is the output by the last convolutional layer.

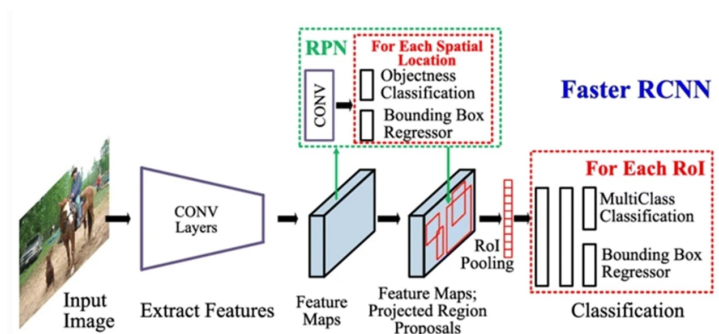


Figure 12 – Faster R-CNN architecture. Extracted from (DIWAN; ANIRUDH; TEMBHURNE, 2023).

- **You Only Look Once (YOLO):** All of the previous object detection algorithms use regions to localize the object within the image. The network does not look at the complete image, instead, it looks at parts of the image which have high probabilities of containing the object. YOLO or You Only Look Once, proposed by Redmon et al. is a novel object detection algorithm much different from the region-based algorithms seen above. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes. YOLO makes use of the whole topmost feature map to predict both confidences for multiple categories and bounding boxes. YOLO divides the input image into an $S \times S$ grid and each grid cell is responsible for predicting the object centered in that grid cell. Each grid cell predicts bounding boxes and their corresponding confidence scores.

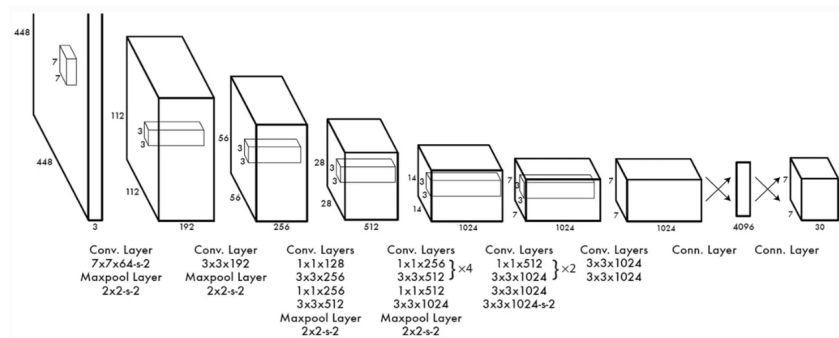


Figure 13 – YOLO architecture. Extracted from (DIWAN; ANIRUDH; TEMBHURNE, 2023).

- **Single Shot MultiBox Detector (SSD):** To avoid some of the limitations of YOLO, (LIU et al., 2016) proposed a Single Shot MultiBox Detector (SSD), which was inspired by the anchors adopted in MultiBox, RPN, and multi-scale representation. Given a specific feature map, instead of fixed grids adopted in YOLO, the SSD takes advantage of a set of default anchor boxes with different aspect ratios and scales to discretize the output space of bounding boxes. To handle objects of various sizes, the network fuses predictions from multiple feature maps with different resolutions. In the context of our project that object detection type will be used as it is a lightweight type combined with the mobilenet architecture. There is also a variant called SSD-Lite, which is the same as SSD but implemented with depthwise-separable convolutions rather than regular convolution layers, it's much faster than regular SSD and perfectly suited for use on mobile devices. SSD is the second one-stage method, and its main contribution is to propose multi-scale features for object detection. It significantly improves the accuracy of the one-stage method, especially for small objects. RetinaNet proposes a new loss function named focal loss to solve

the extreme foreground background class imbalance encountered during the training of dense detectors.

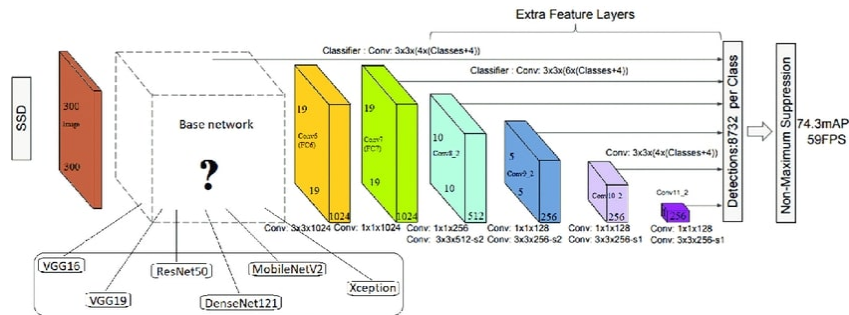


Figure 14 – SSD architecture. Extracted from (JEE et al., 2021).

2.3.4 Machine learning

According to (JO, 2020), machine learning is a field of inquiry devoted to understanding and building methods from data that humans can understand. According to (DULHARE et al., 2020), Machine Learning is a field in which computers learn by repeating several tasks which are considered as experiences.

The definition of (DULHARE et al., 2020) is clearer in the context of this study but both are related to the learning of computers and to do such learning we need some artificial neural network as we will see in the section below.

2.3.4.1 Artificial Neural Network

According to (ZHANG, W., 2010), an artificial neural network is a digital reproduction of biological neurons, composed of artificial neurons or nodes and able to learn thanks to what we call weights by adjusting them.

2.3.5 Deep learning

According to (PEDRAM ATAEE, 2021), deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. The learning process can be supervised, semi-supervised, unsupervised, or by reinforcement. In the case of this study we will be using supervised learning in which the desired output is already known and is represented as labels for the data.

2.3.5.1 Convolutional neural network

According to (GOODFELLOW; BENGIO; COURVILLE, 2016), a convolutional neural network is a type of artificial neural network that is designed to process data with a grid-like topology, such as images or audio spectrograms. It consists of multiple layers, including convolutional layers, which extract increasingly complex features from the input data, and fully connected layers, which use these features to classify or regress the data.

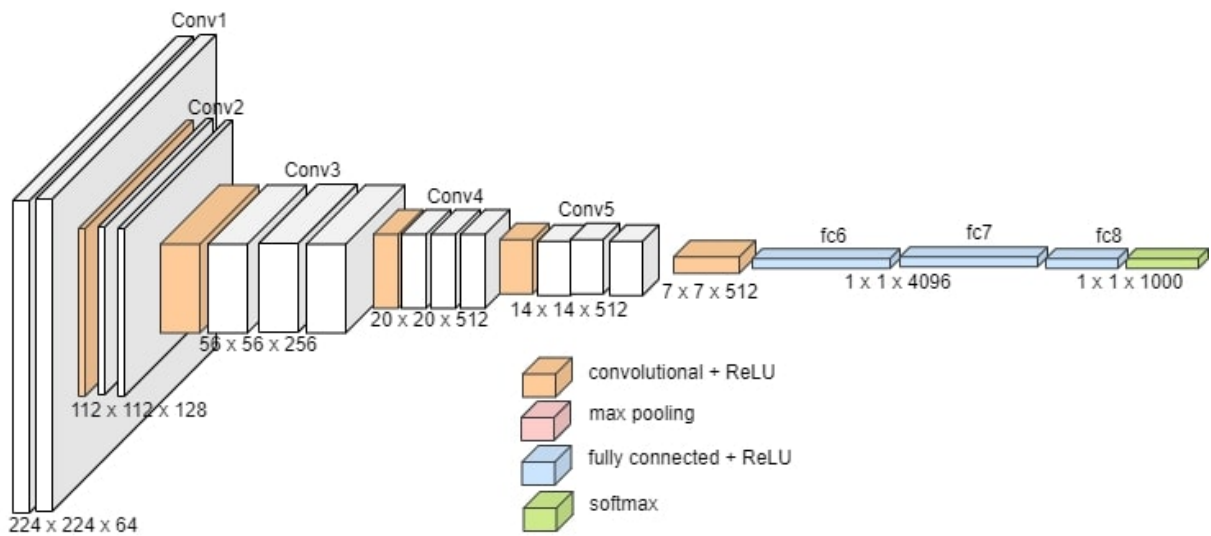


Figure 15 – CNN architecture. Source: (The authors, 2023).

2.3.5.2 Transfer learning

Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks (PAN; YANG, 2010).

2.3.5.3 Dropout

Within the realm of machine learning, dropout serves as a valuable method to combat overfitting. By selectively and randomly disregarding particular nodes within a layer during the training process, it fosters a more independent and robust behavior among the units, thereby enhancing the model's generalization capabilities. (MARIMUTHU, 2022).

2.3.5.4 Learning rate

In machine learning, the learning rate is like a speed dial that controls how quickly a model learns from data. Choosing the right learning rate is important. If it's too low, learning will be slow and might get stuck. If it's too high, learning will be fast but may become unstable. Finding the sweet spot is essential for effective and stable model training. (BROWNLEE, 2019).

2.3.5.5 Layers

In a convolutional neural network, a layer consists of a collection of filters, which are also known as kernels. These filters slide over the input data in a systematic way, computing the dot product between themselves and the local region of the input that they

are currently covering. The output of this computation is a new feature map that highlights the presence or absence of certain patterns or features in the input data. These filters are learnable, meaning that they are adjusted during training to optimize the network's performance on a specific task (GOODFELLOW; BENGIO; COURVILLE, 2016).

In a convolutional layer, a set of filters are applied to small, overlapping regions of the input data, which are known as receptive fields. The filters are designed to look for specific patterns or features in the input data. By applying these filters to the receptive fields, the layer produces a set of feature maps that encode the presence or absence of these patterns or features. This allows the network to learn to recognize more complex patterns and objects as the depth of the layers increases (LECUN et al., 1998).

2.3.5.6 ReLU

ReLU is a non-linear activation function that is used in multi-layer neural networks or deep neural networks. The output of ReLU is the maximum value between zero and the input value. An output is equal to zero when the input value is negative and the input value when the input is positive (NAIR; HINTON, 2010).

2.3.5.7 Pooling

Convolutional layers in a convolutional neural network systematically apply learned filters to input images in order to create feature maps that summarize the presence of those features in the input. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.5.8 Residual layer

A residual layer, also called a residual block, is a building block used in deep neural networks that helps the network learn more efficiently by reusing output from previous layers. This allows the network to better capture complex relationships between input features, and improve accuracy. The residual layer consists of one or more convolutional layers followed by an element-wise addition of the input to the block, which enables the network to learn residual functions. These residual functions capture the difference between the input and output of the block, rather than having the network learn the entire function from scratch. This approach can help address the problem of vanishing gradients in deep networks, and make the network easier to train (SHAFIQ; GU, 2022).

2.3.6 Mobilenet

Several CNN architectures are known to be very big. Nevertheless, there are some architectures that are quite small. According to (HOWARD et al., 2017), MobileNet is a deep learning architecture that can be used in mobile and embedded vision applications that uses depth wise separable convolutions to build light depth weight neural networks.

2.3.6.1 Mobilenet Architecture

MobileNets are built on depth wise separable convolution layers. Each depth wise separable convolution layer consists of a depthwise convolution and a pointwise convolution. Counting depthwise and pointwise convolutions as separate layers, a MobileNet has 28 layers. A standard MobileNet has 4.2 million parameters which can be further reduced by tuning the width multiplier hyperparameter appropriately. The size of the input image is $224 \times 224 \times 3$.

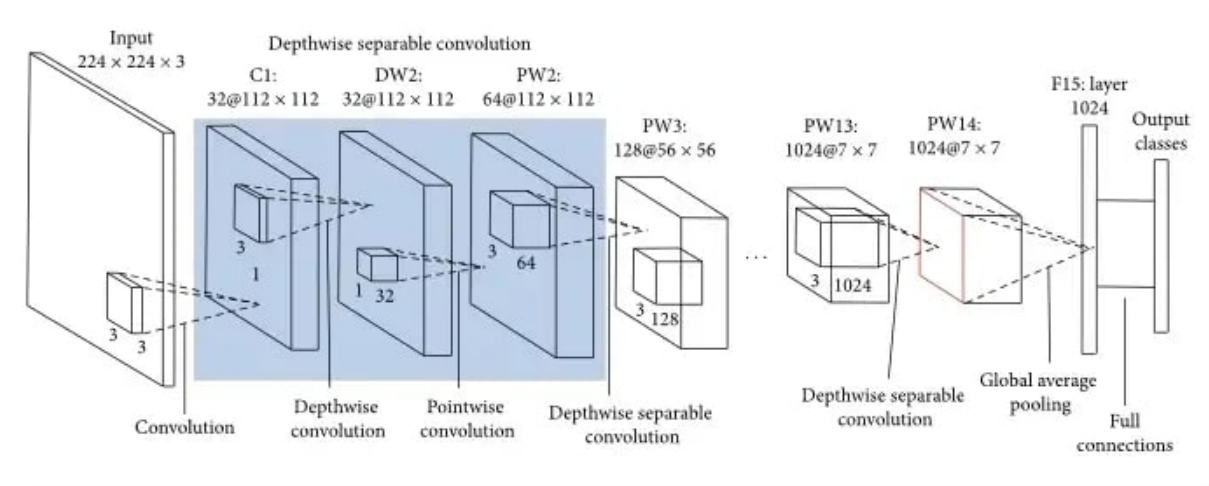


Figure 16 – Mobilenet architecture. Extracted from (PALMIERI, 2021).

2.4 DATASET

A dataset is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the dataset in question. The data set lists values for each of the variables, such as for example height and weight of an object, for each member of the dataset. Datasets can also consist of a collection of documents or files (MARSLAND, 2014).

2.4.1 Data augmentation technique

Data augmentation technique is a process of artificially increasing the amount of data by generating new data points from existing data. This includes adding minor

alterations to data or using machine learning models to generate new data points in the latent space of original data to amplify the dataset (CUNHA, 2022).

2.5 TRAINING

Training is the process of providing a neural network with large amounts of data in order to enable it to learn from patterns and make accurate predictions or decisions. During training, the network adjusts its internal parameters to minimize the difference between its predictions and the actual outputs using an optimization algorithm such as stochastic gradient descent. The goal of training is to produce a neural network that can accurately predict or classify new inputs it has not seen before (GOODFELLOW; BENGIO; COURVILLE, 2016).

- **Pytorch:** It is an open-source machine learning (ML) framework based on the Python programming language and the Torch library. Torch is an open-source ML library used for creating deep neural networks and is written in the Lua scripting language (PASZKE et al., 2019).
- **Over-fitting:** It is an undesirable machine learning behavior that occurs when the machine learning model gives accurate predictions for training data but not for new data (GOODFELLOW; BENGIO; COURVILLE, 2016).
- **Loss function:** In mathematical optimization and decision theory, a loss function or cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization problem seeks to minimize a loss function. An objective function is either a loss function or its opposite, in which case it is to be maximized. The loss function could include terms from several levels of the hierarchy (CUNHA, 2022).
- **Optimizer:** According to (PERE, 2020), an optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate. Thus, it helps in reducing the overall loss and improving the accuracy.
- **Type of optimizer:** Below are the definitions of the four most commonly used types of optimizers in deep learning.
 1. The term stochastic in Stochastic Gradient Descent means randomness on which the algorithm is based upon. In stochastic gradient descent, instead of taking the whole dataset for each iteration, we randomly select the batches of data. That means we only take few samples from the dataset (GOODFELLOW; BENGIO; COURVILLE, 2016).

2. The adaptive gradient descent algorithm is slightly different from other gradient descent algorithms. This is because it uses different learning rates for each iteration. The change in learning rate depends upon the difference in the parameters during training. The more the parameters change, the more minor the learning rate changes. This modification is highly beneficial because real-world datasets contain sparse as well as dense features. So it is unfair to have the same value of learning rate for all the features. The Adagrad algorithm uses the below formula to update the weights. Here the $\alpha(t)$ denotes the different learning rates at each iteration, n is a constant, and E is a small positive to avoid division by 0 (DUCHI; HAZAN; SINGER, 2011).
 3. RMS prop is one of the popular optimizers among deep learning enthusiasts. This is maybe because it hasn't been published but still very well known in the community. RMS prop is ideally an extension of the work RPPROP. RPPROP resolves the problem of varying gradients. The problem with the gradients is that some of them were small while others may be huge. So, defining a single learning rate might not be the best idea. RPPROP uses the sign of the gradient adapting the step size individually for each weight. In this algorithm, the two gradients are first compared for signs. If they have the same sign, we're going in the right direction and hence increase the step size by a small fraction. Whereas, if they have opposite signs, we have to decrease the step size. Then we limit the step size, and now we can go for the weight update (DURYEA; GANGER; HU, 2016).
 4. The name Adam is derived from adaptive moment estimation. This optimization algorithm is a further extension of stochastic gradient descent to update network weights during training. Unlike maintaining a single learning rate through training in SGD, Adam optimizer updates the learning rate for each network weight individually. The creators of the Adam optimization algorithm know the benefits of Adagrad and RMSProp algorithms, which are extensions of the stochastic gradient descent algorithms. Hence the Adam optimizers inherit the features of both Adagrad and RMS prop algorithms. In adam, instead of adapting learning rates based upon the first moment(mean) as in RMS Prop, it also uses the second moment of the gradients. We mean the uncentered variance by the second moment of the gradients (KINGMA; BA, 2014).
- LabelImg: As described by (ALTIS, 2022), it is a user-friendly tool for image

annotation. It provides a graphical interface that allows you to manually draw bounding boxes around objects in images and automatically saves the annotations that contains information about the image in XML or text format for each labeled image.

2.6 PERFORMANCE INDICATOR

In machine learning, a performance indicator is a metric that quantifies the quality or effectiveness of a model in solving a specific task. It allows for a quantitative assessment of how well the model is performing in relation to a particular objective or criterion. Performance indicators are frequently employed to evaluate model performance on classification, regression, and clustering tasks, among others (ALPAYDIN, 2010). In the context of our proposal the mean Average Precision (mAP), described in chapter 4, is used as metric to know how well the model learned from the data.

2.6.1 SSD components

The following components work together in the SSD algorithm to efficiently detect and localize objects in an input image. By leveraging multiple feature maps at different scales and aspect ratios, SSD achieves accurate and real-time object detection performance.

- **Grid cell:** Just like the YOLO algorithm, the SSD algorithm divides the bounding box into a 5x5 grid. Each grid cell is responsible to output the shape, location, color, and label of the object it contains.
- **Anchor box:** As the CNN divides the image into a grid, each cell in the grid is assigned more than one anchor box. SSD model uses a template matching technique during the training period to match the bounding box with each ground truth object of the image.
- **Aspect ratio:** Every object has a different shape and configuration. Some are rounder and larger, while others are shrunk and short. The SSD architecture helps declare aspect ratios beforehand through a ratio parameter.
- **Zoom level:** The zoom parameter can magnify smaller objects in each grid cell to identify their presence, category and location. For example, if we need to identify a building and a park from a helicopter, we need to scale the SSD algorithm in a way that it detects both the larger and the smaller objects.
- **Receptive field:** Receptive field is defined as that moving set of pixels of the image that the algorithm is currently working on. Different layers of a CNN model compute different regions of an input image. As it goes deeper, the size of the object increases. Just like a microscope, a CNN model magnifies every pixel of the object to compute which category it belongs to.

2.7 PARTIAL CONCLUSION

In this chapter, we introduced and explained various terms associated with traffic sign and object detection. We discussed different types of models that are commonly used in this field and provided an understanding of their functioning. Additionally, we presented technical terminology related to deep learning, artificial intelligence, and other concepts relevant to the training process. By familiarizing readers with these terms, we aim to enhance their understanding of the subsequent discussions and analyses presented in the following chapters.

3 RELATED WORKS

This section provides a concise overview of the existing literature on traffic signal detection, focusing on the types of models employed and the datasets utilized to accomplish their objectives. It encompasses a range of techniques utilized in these studies.

The work of (ALGHMGMHAM et al., 2019) presented a study on vertical Arabia Saudi traffic sign classification using Deep CNN and the different angles and including other parameters and conditions. The images for the dataset, which is a total of 2,718, were collected from three different cities in Arabia Saudi. These images were then transformed into Gray-scale with a dimension of 30x30 pixels as they were from different dimensions and as they were in RGB format. Figure 17 is an image of the obtained images after pre-processing. After the training process the authors obtained an accuracy of 100% within 150 epochs in 16 different experiments on different number of epochs and batch size numbers.



Figure 17 – Sample of the traffic signs after the pre-processing. Extracted From (ALGHMGMHAM et al., 2019).

The work of (PON et al., 2018) proposed a hierarchical model built upon the ResNet-50 version of R-CNN which is part of the two-stage model detection algorithm and the images of the dataset are exclusively from the United States. After the experiments 54% as accuracy was obtained but this work is more related to our work as it use two data sets that is part of our work including traffic sign and lights. Figure 18 shows the results of that model on different traffic scenes.



Figure 18 – Results of the hierarchical model on images from Los Angeles, United States. Extracted From (PON et al., 2018).

The work of (HOELSCHER, 2017) presented a study on vertical traffic sign detection and classification techniques in images of complex traffic scenarios. The author used two approaches for image segmentation and selection of regions of interest were tested. The first one is a color thresholding with Fourier descriptors but was not satisfactory and the second one is a color filtering using Fuzzy Logic together with an algorithm that selects stable regions in different shades of gray. The model was used with a created Brazilian dataset by the author and along with the German dataset to obtain 93% as extraction accuracy and 95% as classification accuracy. Figure 19 illustrates an overview of the proposed approach of the author:

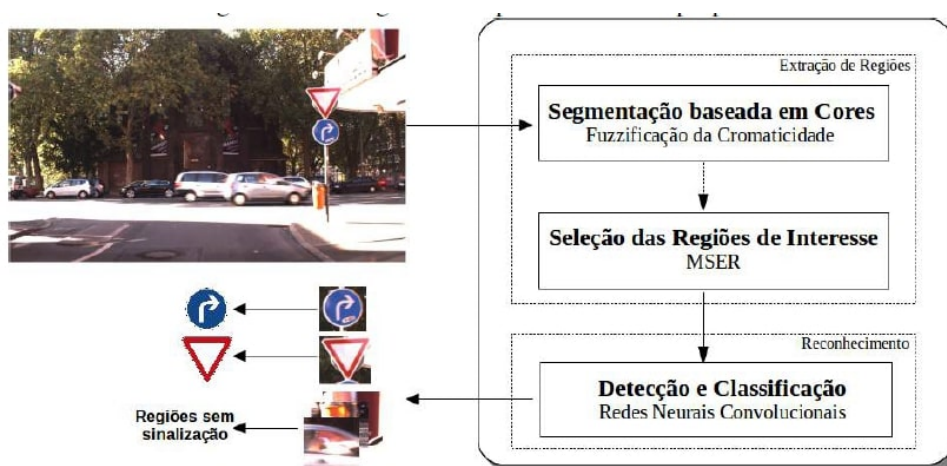


Figure 19 – The system overview of the author. Extracted From (HOELSCHER, 2017).

The work of (SILVA, F. A. d. et al., 2020) proposed a real-time traffic sign detection and recognition algorithm using neural network. Three different classes of traffic signs are used as shown in Figure 20. For the network architecture they used a Faster R-CNN

model with VGG-16 and Inception V4 which had the best result as for feature extraction network and 128x128 pixel as the input of the images. Those images were collected from videos that they took using a camera in front of a car and the frames of the video were extracted to later be augmented using different degrees of rotation on those images where 90% were used for training and 10% for validation. 82% is the accuracy obtained by the best architecture they used.



Figure 20 – The classes used by the authors. Extracted From (SILVA, F. A. d. et al., 2020).

The work of (BHATT; LALDAS; LOBO, 2022) proposed a model for traffic sign detection and recognition using deep learning with convolutional neural networks and a hybrid dataset that includes a reference dataset for German traffic sign recognition from Kaggle and a self-generated Indian traffic sign dataset with an hybrid dataset from the previous ones. For the experiments, 50 epochs were used to train on the German dataset, 15 for the Indian created dataset as we can see the different signs that it contains in Figure 21 and 25 epochs on the hybrid dataset which results an accuracy of 95.45% for the hybrid datasets, 91.08% for the Indian dataset, and 99.85% for the German dataset.



Figure 21 – Indian traffic sign dataset used by the Authors. Extracted From (BHATT; LALDAS; LOBO, 2022).

The work of (YONEDA et al., 2020) proposed an algorithm which is exclusively about traffic lights and arrow lights where the method achieved 91.8% and 56.7% as accuracy for the traffic lights and the arrow lights respectively. Different processes on the region of interest to be able to detect the arrow lights and the architecture used is YOLOv3 which is one of the one-stage model detection that has an F-value of 91.8% for the traffic lights which is calculated in the equation (1).

$$F - value = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (1)$$

The work of (WALI et al., 2019), presented a comprehensive survey that delves into the domains of traffic sign detection, tracking, and classification. The authors conduct a thorough investigation of algorithms, methods, and specifications pertaining to each aspect, presenting the findings in well-structured tables along with relevant references. The survey incorporates a comparative analysis that evaluates TSDR data, performance metrics, and the accessibility of different techniques. Furthermore, the review sheds light on the prevailing challenges and issues faced by existing technologies, accompanied by succinct suggestions for potential enhancements. Figure 22 illustrates the block diagram of the Block diagram of the traffic sign recognition system from the authors.

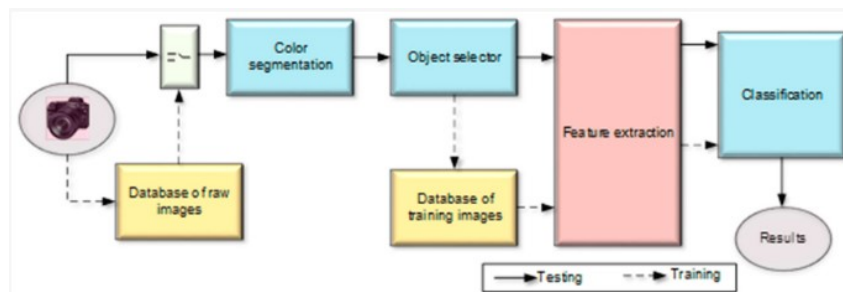


Figure 22 – Block diagram of the traffic sign recognition system. Extracted From (WALI et al., 2019).

The work of (WILLIAM et al., 2019), proposed an effective solution for real-time traffic sign detection and recognition, specifically addressing challenges related to weather conditions, illumination, and visibility. To achieve this, the authors explored advanced multi-object detection systems, such as Faster R-CNN and SSD, along with various feature extractors including MobileNet v1, Inception v2, and Tiny-YOLOv2 but the focus was on evaluating the performance of F-RCNN Inception v2 and Tiny YOLO v2, as they demonstrated the most promising results. The architecture of the network they used is in figure 23.

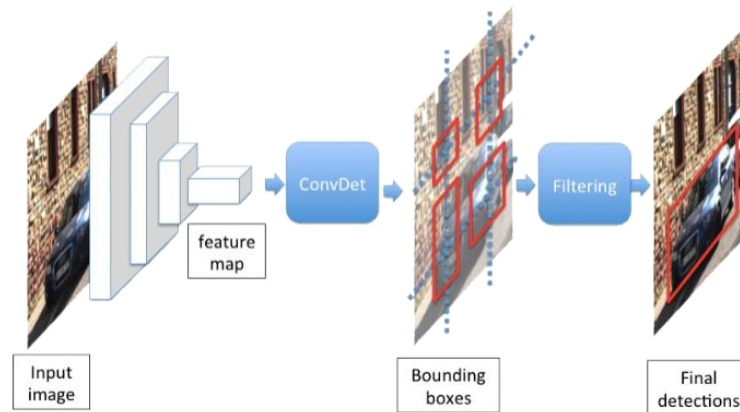


Figure 23 – Testing architecture. Extracted From (WILLIAM et al., 2019).

The work of (DALBORGO et al., 2023) focused on Traffic Sign Recognition systems enabled by embedded systems with internet connections. The implementation of TSR systems using Convolutional Neural Networks and datasets for AI training was discussed. The datasets included a new class for TSR called vegetation occlusion. The results demonstrated that this approach facilitates faster traffic sign maintenance by utilizing vehicles as moving sensors. The proposed technique enables the identification of irregularities in traffic signs, allowing for timely reporting and fixing of issues, ultimately enhancing traffic safety. The paper also evaluated the performance of various YOLO models based on case studies. Figure 24 illustrates the annotation process used by the authors.

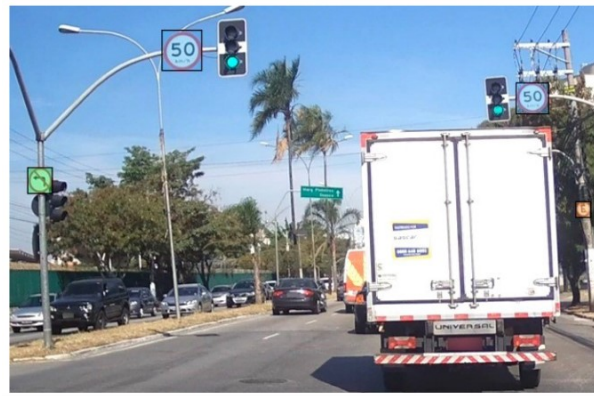


Figure 24 – Image annotation. Extracted From (DALBORGO et al., 2023).

The work of (CHEN et al., 2021) focused on the recognition of small-size traffic signs by proposing an efficient method called "traffic-signs recognition small-aware," inspired by state-of-the-art object detection frameworks like YOLOv4 and YOLOv5. The authors also presented four key contributions of their work. Firstly, they enhance the model's backbone by incorporating high-level features to improve the detector head. Secondly, they utilized the receptive field block-cross in the model's neck to capture contextual information from the feature map. Thirdly, they refined the detector head grid to achieve more accurate detection specifically for small traffic signs. Lastly, they introduced a data augmentation

method called Random Erasing-Attention for the input, which increases the difficulty of samples and enhances the model’s robustness. The authors conducted real experiments on the challenging dataset TT100K and demonstrate that their proposed method significantly improves performance compared to the state-of-the-art approaches. Additionally, their method operates in real-time, making it highly promising for applications in advanced driver assistance systems and autonomous driving systems. Figure 25 illustrates some detection examples on the TT100K testing set by the authors.



Figure 25 – Detection examples on the TT100K testing set by the authors. Extracted From (CHEN et al., 2021).

The work of (ZHU; YAN, 2022) presented an experiment evaluating the performance of the latest version of YOLOv5, a deep learning model, for Traffic Sign Recognition (TSR) using a dataset created by the authors. The objective was to demonstrate the suitability of deep learning models for TSR by comparing YOLOv5 with Single Shot Multibox Detector, another popular object detection algorithm. The experiments utilized the authors’ custom dataset. The experimental results showed that YOLOv5 achieved a mean Average Precision (mAP) of 97.70% for all classes at a threshold of 0.5, whereas SSD achieves a mAP of 90.14% under the same conditions. Furthermore, YOLOv5 demonstrated superior recognition speed compared to SSD. Figure 26 illustrates some examples of the classes they used.









| Class | Sample | Num. | Class | Sample | Num. |
|-----------------|-----------------------------------------------------------------------------------|------|-----------------------------|-------------------------------------------------------------------------------------|------|
| No U-turn |  | 271 | Road bump |  | 329 |
| Road works |  | 294 | Watch for children crossing |  | 176 |
| Crosswalk ahead |  | 313 | Give way |  | 317 |
| Stop |  | 286 | No entry |  | 196 |

Figure 26 – Classes used by the authors. Extracted From (ZHU; YAN, 2022).

The work of (ZHANG, J. et al., 2019) focused on the development of lightweight neural networks for traffic sign recognition, specifically designed for resource-constrained environments. The authors proposed two novel lightweight networks that achieve higher recognition precision while minimizing the number of trainable parameters. They utilized knowledge distillation to transfer knowledge from a larger trained model called teacher network to a smaller model called student network. Additionally, the authors pruned redundant channels from the student network by identifying insignificant channels based on the values of batch normalization (BN) scaling factors. This resulted in a compact model with comparable accuracy to more complex models. The teacher network achieved an accuracy rate of 93.16% on the CIFAR-10 general dataset. Using the knowledge from the teacher network, the student network was trained on the GTSRB and BTSC traffic sign datasets, achieving high accuracy rates of 99.61% and 99.13% respectively, with only 0.8 million parameters. Figure 27 illustrate the confusion matrix (CM) they used as evaluation metrics for their student network on the GTSRB dataset.

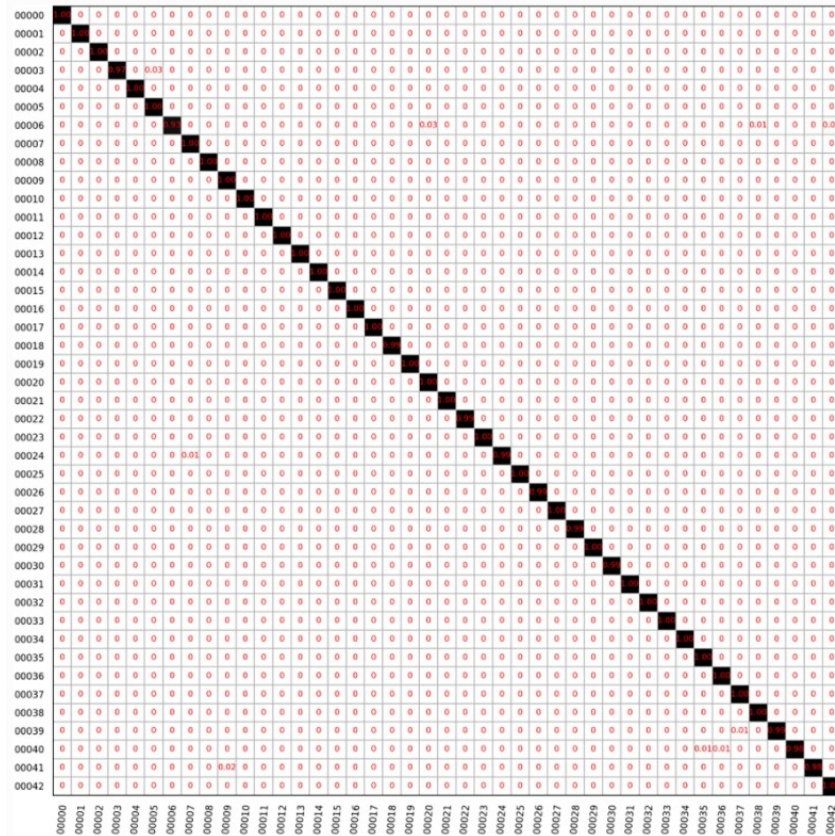


Figure 27 – Confusion matrix on the GTSRB dataset. Extracted From (ZHU; YAN, 2022).

The work of (FREDJ et al., 2023) addressed the importance of road traffic signs for driver safety and the potential benefits of multi-step traffic forecasting on road networks. The authors focused on traffic sign recognition using Deep Convolutional Neural Networks, which have shown excellent results in this domain. The authors proposed a Traffic and Road Sign recognition system based on CNNs and evaluate its performance using a novel dataset called the Tunisian traffic signs dataset. To enhance efficiency, the authors reduced the number of layers in the LeNet network, thereby decreasing the network’s parameters and accelerating computation. They experimented with different parameters to optimize recognition rates in challenging real-world scenarios, including varied weather conditions, complex backgrounds, variable illumination, and sign color fading. The experimental results demonstrated that the proposed CNN architecture achieved significant accuracy, surpassing the performance of similar previous studies. This highlights the effectiveness of the CNN-based approach for traffic sign recognition, particularly in challenging and uncontrolled environments. Figure 28 illustrates the proposed method of the authors.

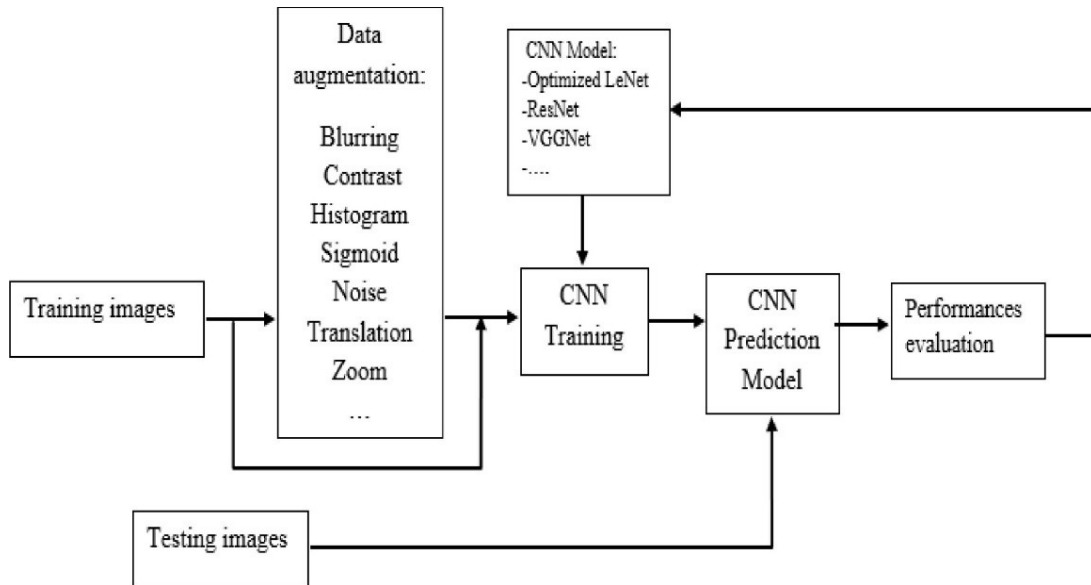


Figure 28 – Proposed method of the authors. Extracted From (FREDJ et al., 2023).

3.1 FINAL REMARKS

In this chapter, we conducted a comprehensive review of various works pertaining to our research problem. These studies primarily focused on the detection of traffic signs, traffic lights, and arrow lights. However, it is worth noting that most of these works were limited to classification tasks and did not consider the simultaneous detection of traffic signs and lights, unlike the works of (BHATT; LALDAS; LOBO, 2022) and (ALGHMGMHAM et al., 2019). Among the related works, (PON et al., 2018) and (HOELSCHER, 2017) employed object detection techniques similar to ours. However, they utilized traffic signs from different countries or did not encompass the comprehensive detection of all states of traffic lights, as demonstrated in (PON et al., 2018). On the other hand, (YONEDA et al., 2020) utilized a one-stage model detection approach, which aligns with our work. Nevertheless, this is the only aspect in common with our research purpose.

The work of (WALI et al., 2019) is about a comprehensive survey on traffic sign detection, tracking, and classification. Conducts a thorough investigation of algorithms, methods, and specifications. Provides comparative analysis, performance metrics, and accessibility of different techniques. Highlights challenges and suggests potential enhancements while the work of (WILLIAM et al., 2019) proposes a real-time traffic sign detection and recognition solution. Addresses challenges related to weather conditions, illumination, and visibility. (DALBORGO et al., 2023) discuss Traffic Sign Recognition systems enabled by embedded systems with internet connections. Utilizes Convolutional Neural Networks and datasets for AI training. Incorporates a new class for TSR called vegetation occlusion. Focuses on utilizing vehicles as moving sensors for traffic sign maintenance. The work of (CHEN et al., 2021) focuses on the recognition of small-sized traffic signs method inspired by YOLOv4 and YOLOv5 while the work of (ZHU; YAN, 2022) evaluates the perfor-

mance of YOLOv5 for Traffic Sign Recognition (TSR) using a custom dataset. Compares YOLOv5 with SSD (Single Shot Multibox Detector). Highlights the superior recognition speed of YOLOv5.

The work of (ZHANG, J. et al., 2019) develops lightweight neural networks for traffic sign recognition in resource-constrained environments like in our work and the work of (FREDJ et al., 2023) emphasizes the importance of road traffic signs and multi-step traffic forecasting. Proposes a Traffic and Road Sign recognition system based on CNNs. Evaluates performance using the Tunisian traffic signs dataset. Optimizes recognition rates in challenging real-world scenarios.

Given the aforementioned gaps in the existing literature, we aim to contribute by developing a novel system that combines the detection of Brazilian vertical traffic signs and lights. Additionally, we strive to create a meticulously annotated dataset of these objects. By leveraging object detection techniques, we aim to address the unique challenges posed by the simultaneous detection of traffic signs and lights, providing a valuable resource for researchers and practitioners in the field.

4 METHODOLOGY

This chapter outlines the methodology employed in the document.

With the primary objective of reducing computation resources, we conducted a thorough literature review on the Single Shot Multibox Detector model and identified a suitable variant called SSD-Lite that offered a lighter computational footprint.

This model has been chosen due to its ability to strike a balance between accuracy and efficiency. SSD-Lite, a lightweight variant of the original SSD model, excels in real-time object detection tasks, making it well-suited for the rapid processing required in traffic management systems. Its impressive speed and respectable accuracy ensure swift and reliable identification of traffic signs and lights, thereby enhancing road safety and traffic flow. Additionally, its efficiency is particularly valuable for resource-constrained applications, making it an ideal choice for deployment in various hardware configurations, from edge devices to cloud-based solutions, ultimately contributing to more effective and scalable traffic management solutions.

We also focused on the creation of our dataset, specifically comprising images of Brazilian vertical traffic signs and lights. Additionally, we discuss the evaluation metrics commonly used in related works and our work to assess the performance of object detection models.

Finally, we present our proposed approach, offering a more detailed view of the methodology employed in our research. Figure 29 provides a visual representation of our methodology.

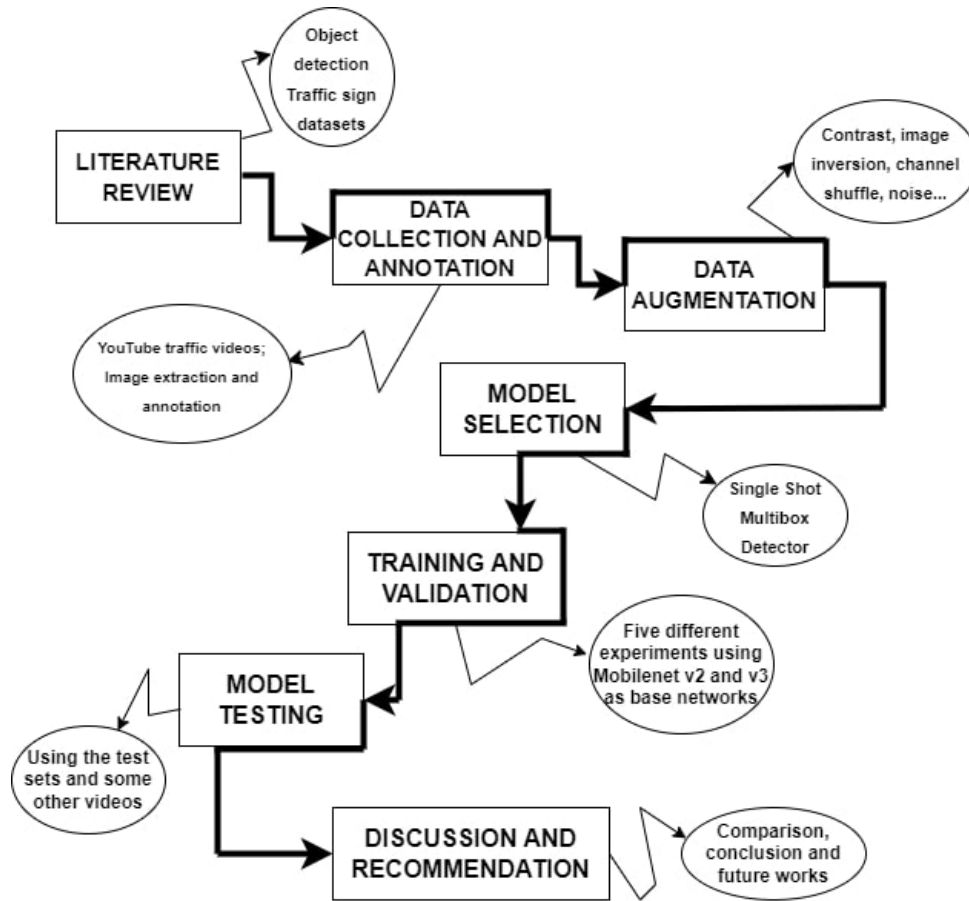


Figure 29 – Methodology used in our work. Source: (The authors, 2023).

4.1 DATA COLLECTION AND ANNOTATION

This section refers to the process of gathering and preparing data for use in our work and data analysis tasks. It involves two main steps: data collection and data annotation. But the first step of the methodology, the literature review, which guides us during our work, is presented in section 3 where we learn from other works.

4.1.1 Data collection

Based on the various types of traffic signs depicted in Figure 1, our goal was to gather a collection of images containing these signs for the purpose of training our model. To achieve this, we acquired videos from a relevant [YouTube channel](#) and employed OpenCV to extract frames from these videos. Subsequently, we meticulously reviewed the videos, removing redundant frames and those that did not contain objects relevant to our study. Through this meticulous process, we obtained a dataset consisting of 1,363 images. It is worth noting that due to the absence of certain traffic signs in the images, we only utilized 16 specific types, resulting in a total of 16 classes for our model. Figure 30 presents an overview of these classes used in our project, accompanied by their corresponding descriptions in both English and Portuguese. Additionally, the figure displays the count

of objects present for each class after the data augmentation process that is detailed in section 4.1.3.

| # | ENGLISH NAME | PORTUGUESE NAME | LABEL | SIGN IMAGE | # OF OBJECTS | # | ENGLISH NAME | PORTUGUESE NAME | LABEL | SIGN IMAGE | # OF OBJECTS |
|--------------------------------------|------------------|------------------------------|--------|-----------------------------------------------------------------------------------|--------------|----|------------------------------|---------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------|--------------|
| 1 | Stop sign | Parada obrigatória | 000000 |  | 2.461 | 9 | Road hump | Alfândega | 000025 |  | 4.018 |
| 2 | Give away | Dê a preferência | 000001 |  | 1.518 | 10 | Sense of the way circulation | Sentido de circulação da via/pista | 000028 |  | 9.499 |
| 3 | No left turn | Proibido virar à esquerda | 000003 |  | 2.415 | 11 | Truck keeps right | Ônibus, caminhões e veículos de grande porte mantenham-se à direita | 000035 |  | 20.705 |
| 4 | No right turn | Proibido virar à direita | 000004 |  | 1.155 | 12 | Bus route | Circulação exclusiva de ônibus | 000040 |  | 1.590 |
| 5 | No parking | Estacionamento proibido | 000007 |  | 6.809 | 13 | Cycling | Circulação exclusiva de bicicleta | 000042 |  | 1.505 |
| 6 | Regular Parking | Estacionamento regulamentado | 000008 |  | 2.581 | 14 | Yellow light | Atenção veículos | 000051 |  | 1.550 |
| 7 | No park and stop | Proibido parar e estacionar | 000009 |  | 5.401 | 15 | Red light | Parada para veículos | 000052 |  | 4.967 |
| 8 | Speed limit | Velocidade máxima permitida | 000023 |  | 20.703 | 16 | Green light | Veículos podem seguir | 000053 |  | 16.376 |
| TOTAL: 85.253 objects for 16 classes | | | | | | | | | | | |

Figure 30 – The classes of the created dataset. Adapted from (ALGHMGMHAM et al., 2019).

4.1.2 Data annotation

In this experiment, the object recognition method employed is a form of supervised learning and given the nature of our object detection problem, precise annotation of object locations and corresponding descriptions was crucial, requiring labeled information for the traffic signs and lights to be detected in the images. This information includes the category of the traffic signs as well as their precise location within the image. To facilitate this process, we utilized a Python-based tool called Labeling for the annotation of the images, as described in section 2.5, which offers a user-friendly interface with shortcut keys and follows the labeling format consistent with PASCAL VOC. Figure 31 illustrates the image annotation process using Labeling. After labeling the images, the corresponding information for each image, including the class labels and object locations, was saved in XML files with matching names. These XML files contain all the necessary information for training the network. Once the image annotation task was completed, across the 1,363 images, we annotated a total number of 2.107 objects, categorized into 16 classes representing common objects found in real-life traffic scenes.

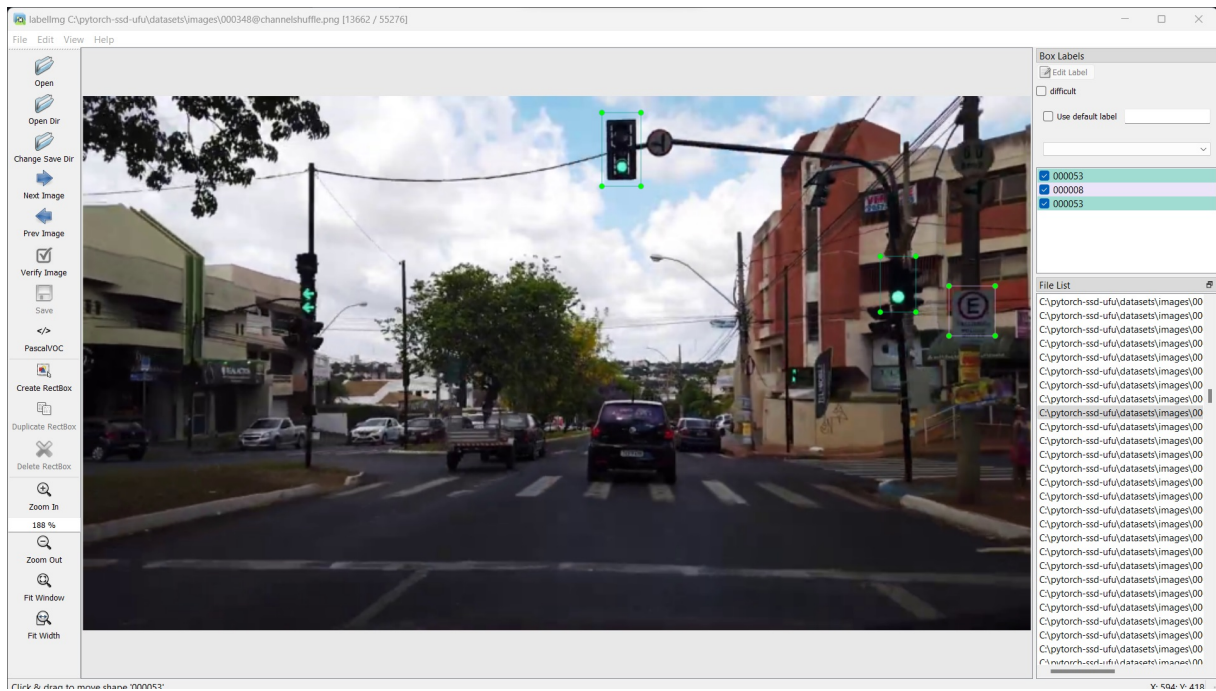


Figure 31 – Labeling tool for image labeling. Source: (The authors, 2023).

4.1.3 Data augmentation

To enhance the model’s learning capabilities, data augmentation was performed to increase the diversity of examples. We employed seven augmentation techniques, including contrast adjustments, noise addition, linear and sigmoid contrast modifications, channel shuffling, image solarization, and image inversion demonstrating their impact on the image appearance.

The careful selection of those augmentation methods is paramount to artificially expand our dataset and prevent overfitting, enabling our model to generalize better to real-world scenarios despite the data scarcity. By searching the internet and introducing controlled variations through these augmentation techniques, we were able to imbue our model with the adaptability needed to handle diverse conditions, ultimately enhancing its performance, even when working with a small and finite dataset.

Through these augmentation methods, we generated a total of 55,276 augmented images, with a total of 85,253 objects for all the classes although the distribution was imbalanced because of the difficulty to find the same number of objects for all the classes in one image. Thus, that unbalanced dataset has been considered after employing Labeling for efficient image annotation and employing various augmentation techniques where we obtained a comprehensive dataset with labeled traffic sign and light information. This dataset served as the foundation for training our network, enabling robust object recognition and detection. Figure 33 serves as an illustration of the seven augmentation methods applied to a single image of our dataset and figure 32 shows the class distribution after the augmentation. For anyone interested, our final dataset is available at (PIERRE,

2023).

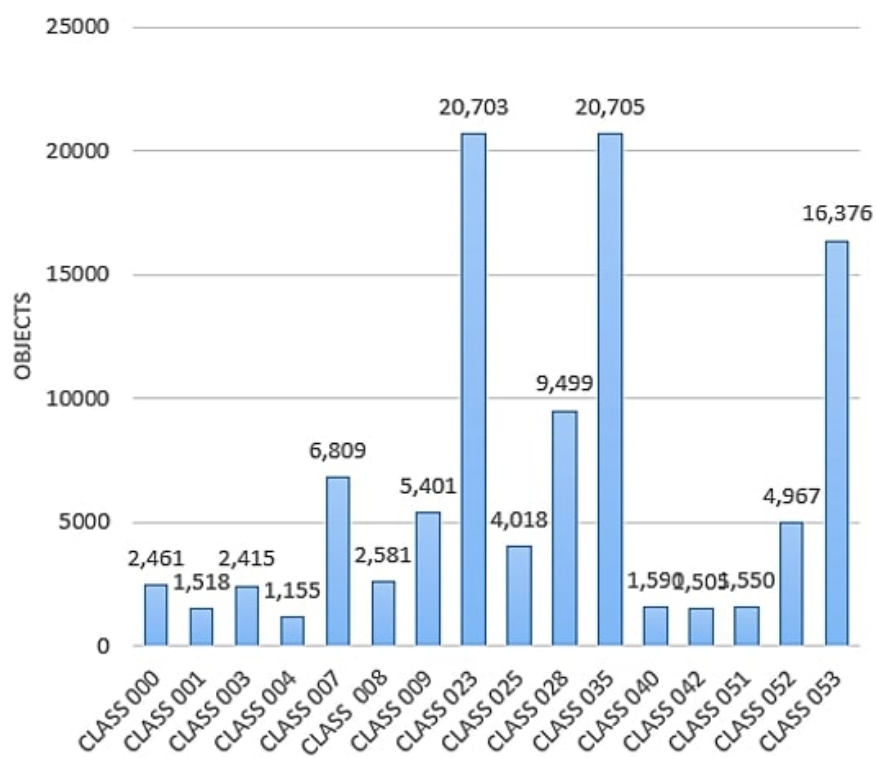


Figure 32 – Dataset class distribution. Source: (The authors, 2023).



Figure 33 – Example of the augmentation methods on one image. Source: (The authors, 2023).

4.2 MODEL SELECTION

The core architecture employed in our work is the Single Shot Multibox Detector, whose choice is explained in Section 4, consists of three main components, as depicted in Figure 15. The first component is responsible for extracting features from the input images. That first part can adopt VGG-16 network like in the original paper without dropout layer, FC8 and soft-max classification layers. It replaces the fully connected layers

FC6 and FC7 in the ordinary VGG network with convolutional layers Conv6 and Conv7.

The second component is the detection heads, which are responsible for generating bounding boxes and class confidence scores. To create a lighter version of SSD, known as SSD-Lite, certain layers were removed from this component. This optimization allows for a more efficient and streamlined detection process. In that second part, four convolutional layers of Conv8, Conv9, Conv10, and Conv11 have been newly added. Each convolutional layer utilizes a 1×1 convolution kernel for dimensionality reduction and then makes use of a 3×3 convolution kernel for feature extraction. The loss function of the SSD model consists of two parts: The localization loss (Lloc) and the confidence loss (Lconf). The entire loss function is weighted sum of localization loss and the confidence loss.

The final component is crucial for eliminating redundant detections and ensuring the best predictions for each object. It employs a mechanism to remove duplicate detections and retain only the most accurate and relevant results. This is achieved by applying a predefined threshold value, typically set at 0.5 or 0.7, depending on the specific dataset and requirements.

In wholesome, our work utilizes the SSD architecture, featuring feature extraction using multiple MobileNet models, detection heads for generating bounding boxes and class confidence scores, and a mechanism for removing duplicate detections. The three SSD-Lite final models has a total of 97 layers, 99 layers and 114 layers using Mobilenet version 2, MobileNet version 3 small and MobileNet version 3 large respectively. This architecture forms the foundation of our system, where its extra layers and its detection heads are illustrated in Table 2 and Table 1 respectively.

It's to be noted that the first Conv2d layers from every module are in a group of ConvBNReLU layers where they are followed by a BatchNom2d and ReLu6 layers, except for the last Conv2d layer which is apart from any other group.

4.2.1 SSD-Lite Base network

The base network of the SSD-Lite architecture is MobileNet. In our implementation, we utilized three different models from the MobileNet family: MobileNet v2, MobileNet v3 (small), and MobileNet v3 (large) which are lightweight convolutional neural network architectures designed for mobile and embedded devices with limited computational resources. In table 3 is shown the layers taken out from the final best model which is Mobilenet v2 for feature extractions and it is composed of 19 modules. It's to be noted that, like the extra layers of the SSD-Lite, the first Conv2d layers of the mobilenet v2 described in table 2 are in a group but that time it's a Conv2dNormActivation layers where every Conv2d layer is followed by a BatchNom2d and ReLu6 layers, and except for the first and last module where we can find the conv2d layers apart from any other group.

| Layer | In | Out | Kernel | Stride |
|-------------|------|------|--------|--------|
| Conv2d | 576 | 576 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 576 | 68 | (1, 1) | (1, 1) |
| Conv2d | 1280 | 1280 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 1280 | 102 | (1, 1) | (1, 1) |
| Conv2d | 512 | 512 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 512 | 102 | (1, 1) | (1, 1) |
| Conv2d | 256 | 256 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 256 | 68 | (1, 1) | (1, 1) |
| Conv2d | 256 | 256 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 256 | 68 | (1, 1) | (1, 1) |
| Conv2d | 128 | 128 | (3, 3) | (1, 1) |
| BatchNorm2d | 576 | — | — | — |
| RELU6 | — | — | — | — |
| Conv2d | 128 | 68 | (1, 1) | (1, 1) |

Table 1 – SSD-Lite regression/classification heads.

| Layer | In | Out | Kernel | Stride |
|--------|------|-----|--------|--------|
| Conv2d | 1280 | 256 | (1,1) | (1,1) |
| Conv2d | 256 | 256 | (3,3) | (2,2) |
| Conv2d | 256 | 512 | (1,1) | (1,1) |
| Conv2d | 512 | 128 | (1,1) | (1,1) |
| Conv2d | 128 | 128 | (3,3) | (1,1) |
| Conv2d | 128 | 256 | (1,1) | (1,1) |
| Conv2d | 256 | 128 | (1,1) | (1,1) |
| Conv2d | 128 | 128 | (3,3) | (2,2) |
| Conv2d | 128 | 256 | (1,1) | (1,1) |
| Conv2d | 256 | 64 | (1,1) | (1,1) |
| Conv2d | 64 | 64 | (3,3) | (2,2) |
| Conv2d | 64 | 128 | (1,1) | (1,1) |

Table 2 – SSD-Lite extra layers.

4.3 TRAINING AND VALIDATION

During the training phase, the prepared dataset was utilized to train the SSD-Lite model. We employed an optimization algorithm, set appropriate hyper-parameters, and iteratively trained the model on the training data. Monitoring the training progress, we

recorded relevant metrics and adjusted parameters as necessary to achieve optimal results.

4.3.1 Hardware and software Setup

To deploy the system, a computer or laptop is necessary. Regarding the hardware requirements, the system necessitates a processing unit capable of handling the computational demands of the object detection algorithm in real-time. A reliable power supply is essential to operate the processing unit continuously. The power supply should provide sufficient power throughout the duration of data capture. Furthermore, the system requires a storage device to store the captured images and processed data.

To develop and train the object detection model, the system relies on a deep learning framework like PyTorch. Such frameworks offer a user-friendly interface for constructing and training deep neural networks. Additionally, the system necessitates image processing libraries such as OpenCV for preprocessing the input images prior to their utilization in the object detection model. These libraries offer a range of functions for tasks such as image resizing, normalization, and filtering. As for the operating system requirements, the system is compatible with any OS that supports the selected deep learning framework and image processing libraries. Linux and Windows are commonly used operating systems for machine learning and computer vision applications, offering extensive support for the required tools and libraries.

4.3.2 Evaluation metrics

Evaluation metrics play a crucial role in assessing the performance of object detection models. Among these metrics, Average Precision (AP) and mean Average Precision (mAP) are widely used to evaluate the effectiveness of various object detection models, including Faster R-CNN, Mask R-CNN, SSD, YOLO, and others. To understand these metrics, below are the definitions of some terms:

- True Positive (TP) — Correct detection made by the model.
- False Positive (FP) — Incorrect detection made by the detector.
- False Negative (FN) — A Ground-truth missed (not detected) by the object detector.
- True Negative (TN) — This is the background region correctly not detected by the model. This metric is not used in object detection because such regions are not explicitly annotated when preparing the annotations.

After defining the aforementioned terms, there are several other metrics used to assess the performance of a model on data. These metrics provide additional insights into the model's effectiveness in object detection:

- Precision is a metric that quantifies the accuracy or exactness of a model in correctly identifying relevant objects. It is calculated as the ratio of true positives

(TP) to the total number of detections made by the model. Precision focuses on minimizing false positives, meaning it measures how many of the model's predicted positives are actually true positives.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall measures the model's ability to detect all relevant objects or ground truths. It is calculated as the ratio of true positives to the total number of ground truths. Recall aims to minimize false negatives, indicating how well the model captures all the positives in the dataset.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

In summary, precision evaluates the model's precision in making correct positive predictions, while recall assesses the model's ability to capture all positive instances in the dataset. Both metrics are important for assessing the performance of object detection models. Equations (2) and (3) illustrate the formula for each one of them. The f-value is also another metric shown in equation (1).

4.3.3 Intersection over Union

Commonly called (IoU), it is a used metric in object detection to measure the degree of overlap between a predicted bounding box and the ground-truth bounding box, which is manually annotated. It helps evaluate the accuracy of object detection by quantifying the similarity between the predicted and ground-truth bounding boxes. The IoU value ranges from 0 to 1, where a value of 0 indicates no overlap between the boxes, and a value of 1 represents a perfect overlap. A higher IoU value indicates a better alignment between the predicted and ground-truth bounding boxes.

When considering an IoU threshold of α , a True Positive (TP) refers to a detection where the $\text{IoU}(\text{ground-truth, predicted}) > \alpha$. A False Positive (FP) occurs when the $\text{IoU}(\text{ground-truth, predicted}) \leq \alpha$. A False Negative (FN) is a ground truth that was missed when the $\text{IoU}(\text{ground-truth, predicted}) \leq \alpha$. The formula for calculating IoU is shown in Figure 34, which represents the ratio of the intersection area of the predicted and ground-truth bounding boxes to the union area of the two boxes.

In summary, IoU provides a quantitative measure of the overlap between predicted and ground-truth bounding boxes, assisting in evaluating the accuracy and correctness of object detection models.

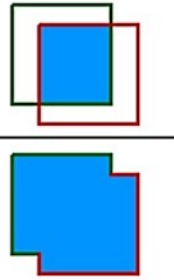
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Diagram 1}}{\text{Diagram 2}}$$


Figure 34 – Intersection over union formula. Extracted from (PADILLA; NETTO; SILVA, E. A. B. da, 2020).

4.3.4 Mean Average Precision

AP@ α refers to the Area Under the Precision-Recall Curve (AUC-PR) evaluated at the alpha Intersection over Union (IoU) threshold. It quantifies the performance of object detection models by measuring the precision and recall trade-off at a specific IoU threshold. A higher value of Area Under the PR Curve indicates higher precision and recall rates. The PR curve typically exhibits a zig-zag pattern, as it is not necessarily monotonically decreasing. AP is calculated individually for each class, resulting in as many AP values as there are classes. These AP values are then averaged to obtain the mean Average Precision (mAP) metric. The mAP provides an overall assessment of the model's performance by taking into account the AP values across all classes. Equations (4) and (5) give the formulas related to the AP and the mAp respectively.

$$AP = \sum_{i=0}^{N-1} [Recalls_{(i)} - Recalls_{(i+1)}] * Precisions_{(i)} \quad (4)$$

$$mAp = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

4.3.5 Proposed Approach

The system proposed in this study is designed as an object detection system specifically tailored for detecting traffic signs and lights in images captured by a moving car. To accomplish this, we employ a lightweight version of the Single Shot Detector (SSD-Lite) algorithm, utilizing the MobileNet V2 as the base network for efficient feature extraction.

The SSD algorithm is renowned for its real-time object detection capabilities, performing detection in a single pass through a deep neural network. By employing the Lite version of the SSD algorithm, which is optimized for low-power and low-latency devices, we ensure suitability for real-time applications.

The proposed system comprises three key components: image acquisition, image preprocessing, and object detection. Image acquisition involves capturing images from a moving car, providing the input for subsequent analysis. These images then undergo preprocessing steps to enhance their quality and optimize them for efficient detection. Finally, the object detection component utilizes the trained model to identify and localize traffic signs and lights within the images. Our proposed system aims to accurately detect and classify traffic signs and lights in real-world scenarios.

Moving on to our proposal, Figure 35 depicts the functioning of our system, showcasing how it operates in practice and highlighting its key components and processes. This architecture forms the foundation of our system.

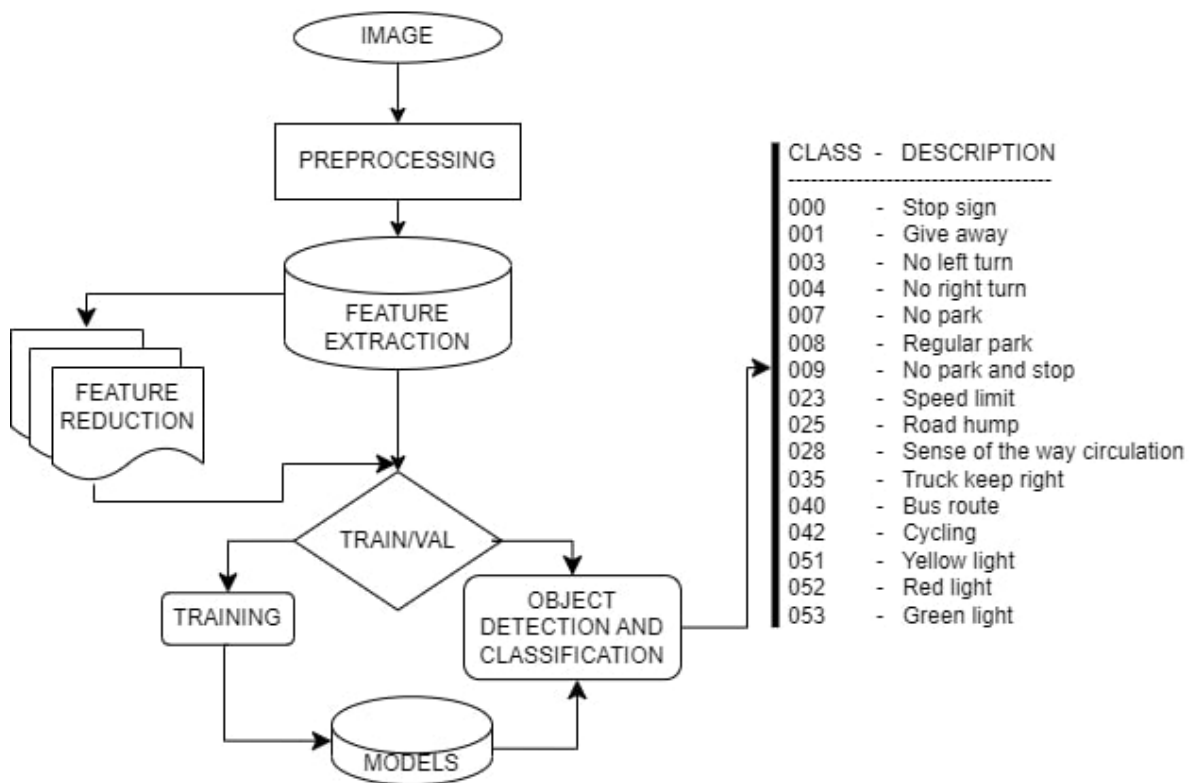


Figure 35 – System overview. Adapted from (ALGHMGMHAM et al., 2019).

4.3.6 Data Preparation

After augmenting the dataset and organizing the images and corresponding annotations into appropriate formats suitable for training the SSD-Lite model. This involved splitting the data into training and validation sets, ensuring proper data organization for efficient model training. To ensure uniformity and facilitate data processing, we normalized the images by applying a consistent color mode and resizing them to a predetermined size. Furthermore, we processed each image in binary format, simplifying the manipulation of the resulting binary matrix. By examining each pixel in the image, we assigned a value of

0 to represent darkness and a value of 1 to represent brightness. By following these steps, we established a well-structured and augmented dataset that includes relevant classes of traffic signs and lights. The normalization and conversion to binary format facilitate subsequent image processing and analysis in our system.

4.4 PARTIAL CONCLUSION

In this chapter, we provided an in-depth overview of the methodology employed in our research, as illustrated in Figure 29. We began by explaining the process of creating the database and annotating the images, highlighting the tools and techniques utilized to accomplish our objectives. Additionally, we delved into the methods employed to augment the dataset, enhancing its diversity and enabling robust training of our model.

Furthermore, we introduced the model we have selected and the performance indicator used to evaluate its effectiveness, along with the relevant terminologies associated with it. This indicator serves as a benchmark for assessing the accuracy and reliability of our traffic sign and light recognition system.

Finally, we presented the proposed approach of our work and data preparation, encompassing the hardware and software requirements necessary for implementing and deploying our system. These requirements ensure the smooth operation and optimal performance of our model throughout its development and practical application.

| Layer | In | Out | Kernel | Stride |
|--------------|-----------|------------|---------------|---------------|
| Conv2d | 3 | 32 | (3,3) | (2,2) |
| Conv2d | 32 | 32 | (3,3) | (1,1) |
| Conv2d | 32 | 16 | (1,1) | (1,1) |
| Conv2d | 16 | 96 | (1,1) | (1,1) |
| Conv2d | 96 | 96 | (3,3) | (2,2) |
| Conv2d | 96 | 24 | (1,1) | (1,1) |
| Conv2d | 24 | 144 | (1,1) | (1,1) |
| Conv2d | 144 | 144 | (3,3) | (1,1) |
| Conv2d | 144 | 24 | (1,1) | (1,1) |
| Conv2d | 24 | 144 | (1,1) | (1,1) |
| Conv2d | 144 | 144 | (3,3) | (2,2) |
| Conv2d | 144 | 32 | (1,1) | (1,1) |
| Conv2d | 32 | 192 | (1,1) | (1,1) |
| Conv2d | 192 | 192 | (3,3) | (1,1) |
| Conv2d | 192 | 32 | (1,1) | (1,1) |
| Conv2d | 32 | 192 | (1,1) | (1,1) |
| Conv2d | 192 | 192 | (3,3) | (1,1) |
| Conv2d | 192 | 64 | (1,1) | (1,1) |
| Conv2d | 32 | 192 | (1,1) | (1,1) |
| Conv2d | 192 | 192 | (3,3) | (2,2) |
| Conv2d | 192 | 64 | (1,1) | (1,1) |
| Conv2d | 64 | 384 | (1,1) | (1,1) |
| Conv2d | 384 | 384 | (3,3) | (1,1) |
| Conv2d | 384 | 64 | (1,1) | (1,1) |
| Conv2d | 64 | 384 | (1,1) | (1,1) |
| Conv2d | 384 | 384 | (3,3) | (1,1) |
| Conv2d | 384 | 64 | (1,1) | (1,1) |
| Conv2d | 64 | 384 | (1,1) | (1,1) |
| Conv2d | 384 | 384 | (3,3) | (1,1) |
| Conv2d | 384 | 64 | (1,1) | (1,1) |
| Conv2d | 64 | 384 | (1,1) | (1,1) |
| Conv2d | 384 | 384 | (3,3) | (1,1) |
| Conv2d | 384 | 96 | (1,1) | (1,1) |
| Conv2d | 96 | 576 | (1,1) | (1,1) |
| Conv2d | 576 | 576 | (3,3) | (1,1) |
| Conv2d | 576 | 96 | (1,1) | (1,1) |
| Conv2d | 96 | 576 | (1,1) | (1,1) |
| Conv2d | 576 | 576 | (3,3) | (1,1) |
| Conv2d | 576 | 96 | (1,1) | (1,1) |
| Conv2d | 96 | 576 | (1,1) | (1,1) |
| Conv2d | 576 | 576 | (3,3) | (2,2) |
| Conv2d | 576 | 160 | (1,1) | (1,1) |
| Conv2d | 160 | 960 | (1,1) | (1,1) |
| Conv2d | 960 | 960 | (3,3) | (1,1) |
| Conv2d | 960 | 160 | (1,1) | (1,1) |
| Conv2d | 160 | 960 | (1,1) | (1,1) |
| Conv2d | 960 | 960 | (3,3) | (1,1) |
| Conv2d | 960 | 160 | (1,1) | (1,1) |
| Conv2d | 160 | 960 | (1,1) | (1,1) |
| Conv2d | 960 | 960 | (3,3) | (1,1) |
| Conv2d | 960 | 320 | (1,1) | (1,1) |
| Conv2d | 320 | 1280 | (1,1) | (1,1) |

Table 3 – SSD-Lite extraction feature layers from mobilenet v2.

5 EXPERIMENTS

In this chapter, we provide a detailed overview of the hyper-parameters employed in our selected model as follows : firstly, we outline the specific hyper-parameters utilized in our chosen model, which play a crucial role in its performance. We emphasize the importance of fine-tuning these parameters to optimize the model’s accuracy and efficiency.

Next, we describe the machines utilized during the training and testing phases. We provide detailed specifications for each machine, including the processor, RAM, and graphics card. These machines were carefully selected to provide the necessary computational resources for executing our experiments effectively.

Subsequently, we present the results obtained by applying the best-performing model to real-life traffic scenario videos and images from the dedicated test set. The model’s performance is evaluated based on its ability to accurately detect and classify traffic signs and lights in various real-world contexts.

Finally, we engage in a comprehensive discussion of the obtained results, considering factors such as model accuracy, computational efficiency, and potential areas for improvement. We analyze the strengths and limitations of our approach, addressing any discrepancies or unexpected outcomes.

5.1 TRAINING PROCESS

In this section, the architecture of the neural network, the training dataset and also some configurations are presented. Aiming at improving the precision of the best results, it was executed with three different image input sizes, where the displayed precision correspond to the arithmetic mean of the achieved values for every class. For the assembly of the neural network, we used the following architectures:

- SSD-Lite with mobilenet v2 as base network;
- SSD-Lite with mobilenet v3 small as base network;
- SSD-Lite with mobilenet v3 large as base network.

To achieve our goals, we utilized two different machines running the Windows 11 operating system. The first machine was equipped with an Intel® Core™ i7-6700K CPU clocked at 4.00GHz, 8 cores in total, 24.0 GiB of RAM, and an NVIDIA GeForce GTX 960 with 2.0 GB of RAM. The second machine featured an 11th Gen Intel® Core™ i3-1115G4 CPU operating at 3.00GHz with 4 cores, 12 GB of RAM, and a Mesa Intel® UHD Graphics (TGL GT2) graphics card. The second machine was primarily used for conducting tests and implementing data augmentation techniques, while the first machine was utilized for the training process.

To identify the most suitable model architecture that would facilitate the training process and offer the best accuracy, an extensive battery of tests was conducted. These

tests involved experimenting with various configurations and fine-tuning adjustments for each architecture. Table 4 provides a detailed list of the hyper-parameters used in these tests and table 5 the number of parameters for each model. The dataset was divided into three distinct phases: training, validation, and testing. Approximately 70% of the images, totaling 38,693 images, were utilized for the training phase. The validation phase consisted of 20% of the images, approximately 11,055 images, while the remaining 10% (5,527 images) were reserved for the testing phase.

By carefully allocating the dataset and conducting rigorous tests, we aimed to find the optimal configuration for each architecture, ultimately leading to a model with simplified architecture for efficient training and another model with the highest achievable accuracy. These steps ensured a thorough evaluation and selection of the most suitable models for our objectives.

| Hyperparameter | Value |
|-----------------------|--------------|
| Learning rate | 0.001 |
| Batch size | 32 |
| Optimizer | SGD |
| Number of epochs | 25 |
| Weight decay | 0.00004 |
| Gamma | 0.1 |

Table 4 – Hyperparameter list.

| Model | Parameters |
|-----------------------------|-------------------|
| Mobilenet v2 SSD-Lite | 3.286.326 |
| Mobilenet v3 small SSD-Lite | 1.304.522 |
| Mobilenet v3 large SSD-Lite | 3.881.522 |

Table 5 – Number of parameters for each model.

5.2 RESULTS EVALUATION

Following the experiments involving different input sizes, we proceeded to evaluate the best-performing method using the dedicated test set. Table 6 provides an overview of the results obtained during the training phase. Additionally, Table 7 presents the accuracy achieved for each individual class when utilizing the best model.

It is worth noting that the training process took a longer time when using the higher input size of 512x512. Surprisingly, the accuracy obtained with this larger input size did not surpass that of the 320x320 input size. Furthermore, for the Mobilenet v3 small model, the input size of 128x128 actually yielded better accuracy results compared to the higher input size, but inferior to the best-performing model.

Interestingly, adjusting the batch size did not yield any significant changes in the training process. Regardless of the batch size, all models reached a point of convergence

within less than 20 epochs. Consequently, we decided to conclude the training within 25 epochs, as subsequent changes in model performance were minimal and marginal. It's to be noted that some other experiments have also been made using different values for the batch size like 8, 16 and 64 but only with the value 32 we had a best result presented here.

During the experiments, we used an IoU threshold of 0.5 to help on a better object detection, as we know that, that value is used to compare the probability of the detected bounding box with the ground-truth box and keep the one with the highest probability.

Overall, these findings highlight the impact of input size on training time and accuracy. While larger input sizes may require more computational resources and time, they do not necessarily guarantee improved accuracy. These insights guide our decision-making process and contribute to the overall understanding of model training dynamics.

| # | Input | Base network | VAL DATA | |
|---|---------|--------------------|----------|--------------|
| | | | Day | mAP@0.5 |
| 1 | 128x128 | Mobilenet v2 | 1.49 | 0.64 |
| | | Mobilenet v3 small | 1.11 | 0.11 |
| | | Mobilenet v3 large | 1.38 | 0.54 |
| 2 | 320x320 | Mobilenet v2 | 3.89 | *0.87 |
| | | Mobilenet v3 small | 7.15 | 0.46 |
| | | Mobilenet v3 large | 5.35 | 0.84 |
| 3 | 320x320 | Mobilenet v2 | 3.77 | 0.79 |
| 4 | 320x320 | Mobilenet v2 | 3.30 | 0.78 |
| 5 | 512x512 | Mobilenet v2 | 8.9 | 0.77 |
| | | Mobilenet v3 small | 5.7 | 0.54 |

Table 6 – Training results for every input size where the threshold value for the IoU is 0.5.

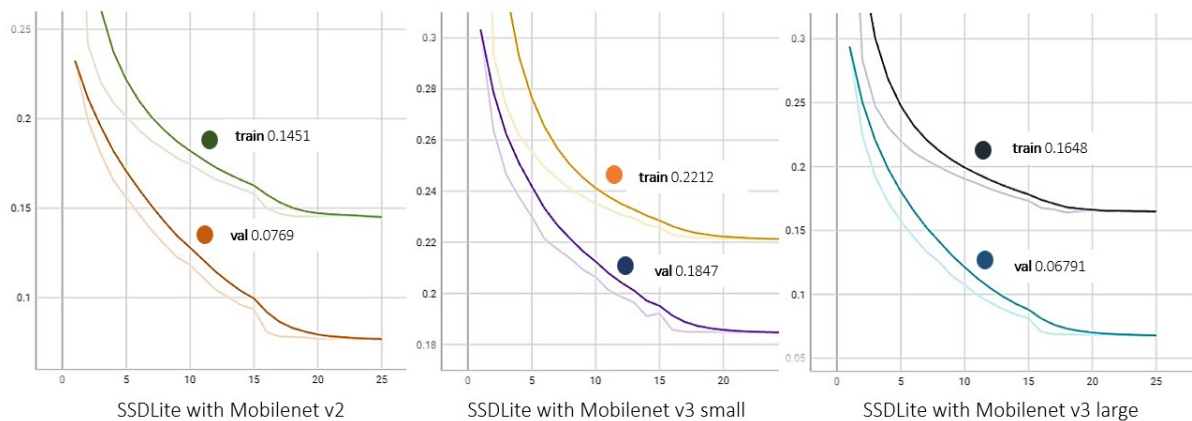


Figure 36 – Loss for input size 128. Source: (The Authors, 2023).

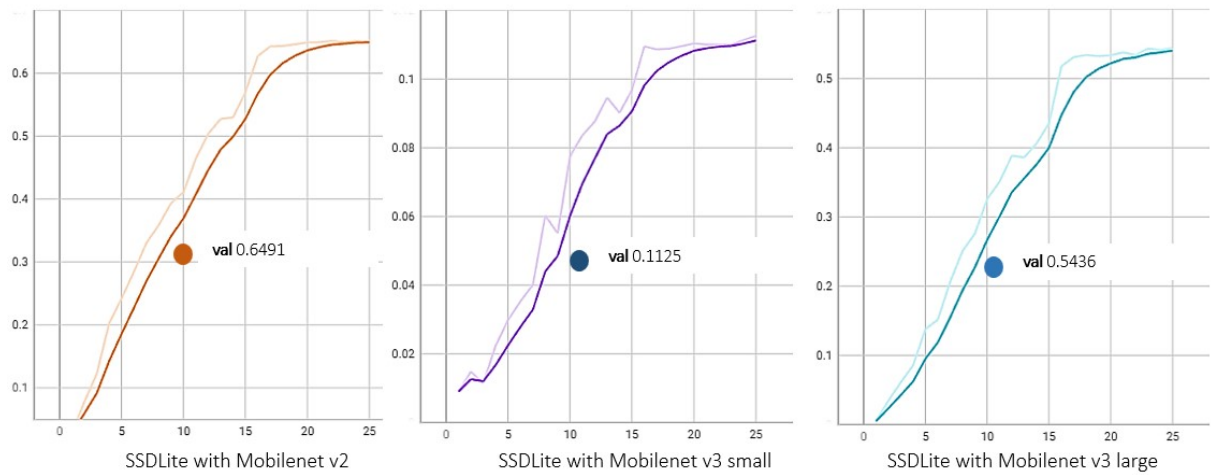


Figure 37 – Accuracy for input size 128. Source: (The Authors, 2023).

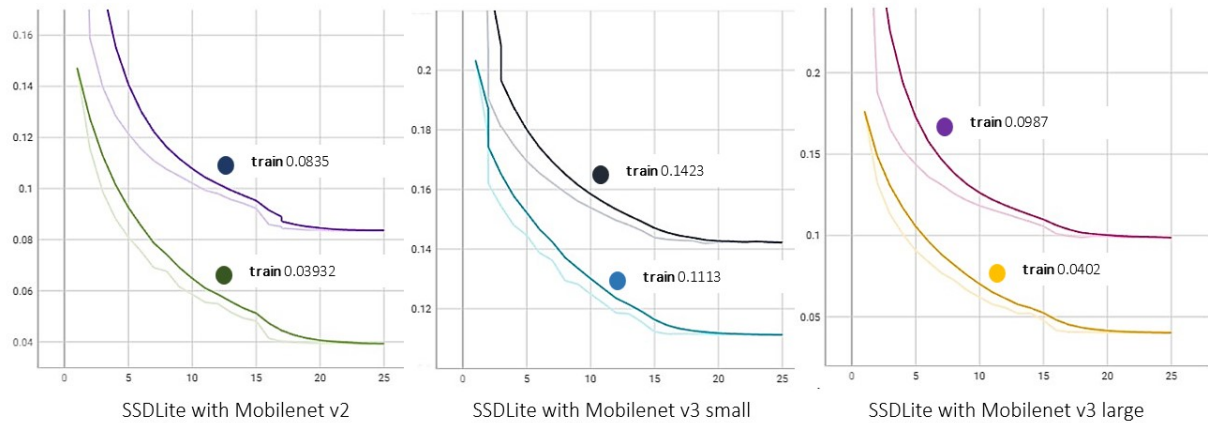


Figure 38 – Loss for input size 320. Source: (The authors, 2023).

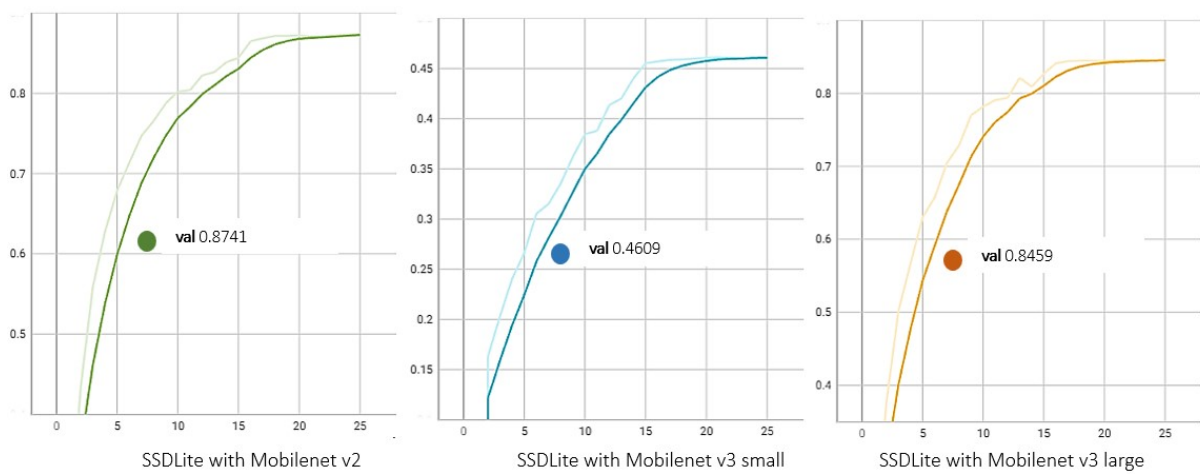


Figure 39 – Accuracy for input size 320. Source: (The authors, 2023).

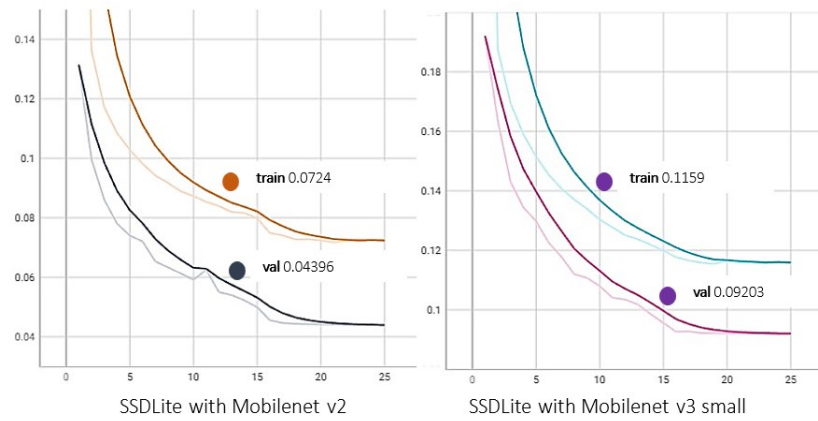


Figure 40 – Loss for input size 512. Source: (The authors, 2023).

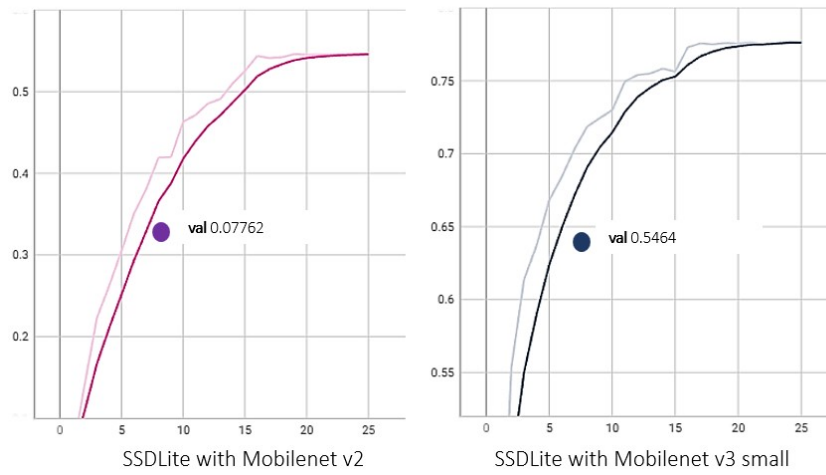


Figure 41 – Loss for input size 512. Source: (The authors, 2023).

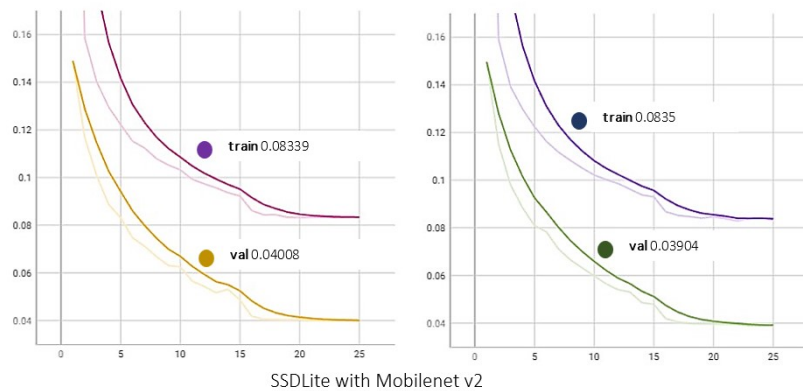


Figure 42 – Second and third experiment loss for input size 320. Source: (The authors, 2023).

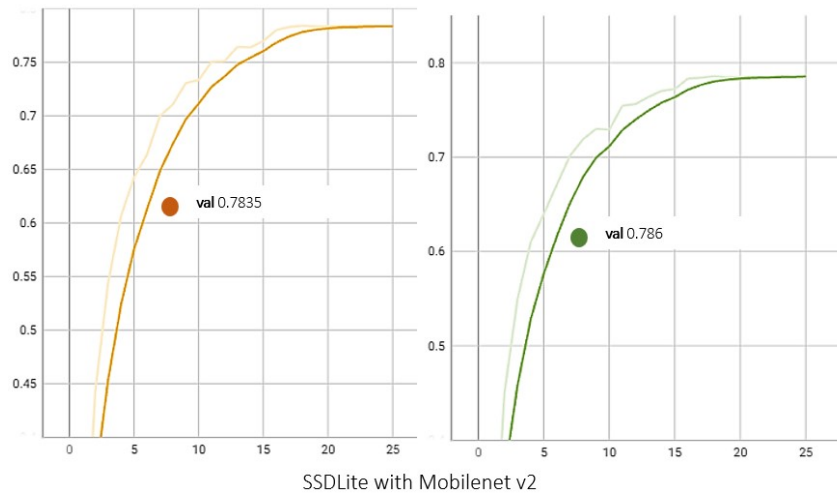


Figure 43 – Second and third experiment accuracy for input size 320. Source: (The authors, 2023).

| TEST DATA | | |
|------------------------------|--------------|---------------|
| Class name | Label | AP@0.5 |
| Stop sign | 000 | 0.80 |
| Give away | 001 | 0.76 |
| No left turn | 003 | 0.78 |
| No right turn | 004 | 0.60 |
| No park | 007 | 0.79 |
| Regular park | 008 | 0.79 |
| No park and stop | 009 | 0.79 |
| Speed limit | 023 | 0.79 |
| Road hump | 025 | 0.80 |
| Sense of the way circulation | 028 | 0.79 |
| Trucks keep right | 035 | 0.79 |
| Bus route | 040 | 0.79 |
| Cycling | 042 | 0.79 |
| Yellow light | 051 | 0.80 |
| Red light | 052 | 0.76 |
| Green light | 053 | 0.79 |
| mAP@0.5 | | 0.78 |

Table 7 – Accuracy of every class using the best model with the test data.

It's to be noted that we do not have an accuracy for the training process here but, in fact, the model was evaluated after each epoch and only the best model after the evaluation with its information was kept. For using different the input sizes, it was faster with 128x128 Mobilenet v3 small and slower with 320x320 Mobilenet v3 large but we kept the 320x320 Mobilenet v2 as it has a better accuracy and an acceptable speed which 30 FPS (Frame Per Seconds) as we can see in Figure 44.

5.3 DISCUSSION

During the experiments, we utilized an IoU threshold of 0.5. With this threshold, our mean Average Precision was 0.87%. The best model, which employed the second version of Mobilenet as the base network and had an input size of 320x320, outperformed the other models in terms of accuracy. However, the models with larger input sizes, such as 512x512 pixels, suffered from increased computational time and slower object detection in videos or images. Due to these limitations, we only conducted two experiments using the 512x512 input size.

Alternatively, the 128x128 input size showed faster object detection but resulted in a decrease in mAP due to the loss of certain features during training. Despite this, our object detection model achieved a good result compared to existing literature. It differs from previous works by considering the classification and localization of Brazilian traffic signs and lights in images, which is not typically addressed. In Figures 44 we can find the results obtained by applying the best model to the test set.

Moreover, after the testing step we noticed a lower accuracy specially for the class "004" and after investigation it came out that happened because of the presence of more images with noise in the test data than the training and validation data. In spite of that low accuracy during the test step, the model was able to detect objects in videos with better accuracy as we can see in Figure 45 to Figure 48, which are the results obtained by applying the best model to some videos.

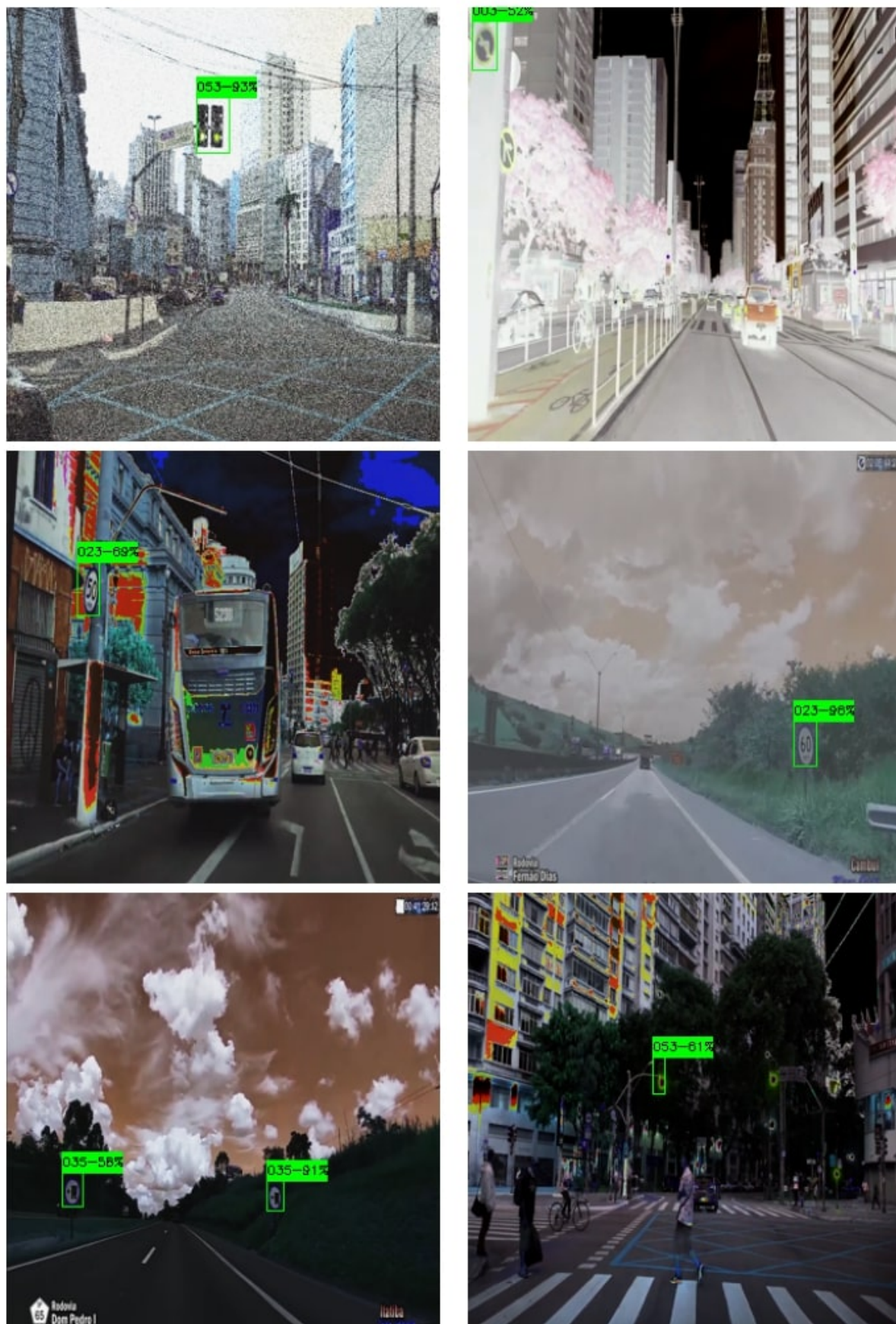


Figure 44 – Example of detection from video. Source: (The authors, 2023).



Figure 45 – Example of detection from video. Source: (The authors, 2023).



Figure 46 – Example of detection from video. Source: (The authors, 2023).



Figure 47 – Example of detection from video. Source: (The authors, 2023).



Figure 48 – Example of detection from video. Source: (The authors, 2023).

5.4 PARTIAL CONCLUSION

In this chapter, we provided a comprehensive overview of our experimental process, highlighting key hyper-parameters, dataset splitting, hardware configuration, and the evaluation of obtained results. As previously mentioned, the architecture utilizing the Mobilenet v2 base network demonstrated superior performance, exhibiting higher precision while requiring less computation time compared to its counterpart with an input size of 512x512.

Furthermore, we presented detailed results, including the accuracy of each individual class, shedding light on the model's performance across different traffic sign and light categories. Additionally, we showcased the effectiveness of our best-performing model by applying it to various images and videos, resulting in a commendable 30 frames per second (FPS) and a mean Average Precision (mAP) of 0.87%.

In summary, this chapter provided a comprehensive understanding of the experimental process, highlighting the superiority of the architecture utilizing the Mobilenet v2 base network in terms of precision and computational efficiency. The results presented underscored the model's accuracy in detecting and classifying traffic signs and lights, reaffirming its effectiveness and potential for real-world applications.

6 CONCLUSIONS

In this research paper, our primary focus was on developing a robust system for the recognition of vertical traffic signs and traffic lights. To achieve our goal, we conducted a series of experiments using three different base networks in combination with the lightweight version of Single Shot MultiBox Detector.

After rigorous experimentation and analysis, we obtained remarkable results. Our system achieved an impressive accuracy rate of 87.4% in recognizing vertical traffic signs and traffic lights. This achievement can be attributed to the meticulous selection of the second version of Mobilenet as the best-performing base network, combined with an input image size of 320x320 pixels.

Furthermore, we compared our results with existing literature and observed that our approach outperformed most previous works in the field. What sets our research apart is the comprehensive consideration of both classification and localization aspects, specifically tailored to Brazilian traffic signs and lights. This distinction allowed us to address the unique characteristics and challenges associated with this specific context.

In conclusion, our research presents a highly accurate and efficient system for the recognition of vertical traffic signs and traffic lights. We believe that our findings contribute significantly to the field of object detection and can pave the way for improved traffic management systems, ensuring safer and more efficient road transportation.

6.1 CONTRIBUTIONS

By extensively evaluating our model using various videos and images, we successfully validated our hypothesis and identified several significant contributions in our research. First and foremost, we introduced a comprehensive dataset exclusively focused on traffic signs and lights in Brazil. This dataset fills a crucial gap in the availability of resources for object detection in this specific context.

Moreover, our work introduced a lightweight model based on the Single Shot Multi-box Detector. This model was specifically tailored to the task of traffic sign and light detection, offering a balance between accuracy and computational efficiency. The utilization of this one-stage detection model significantly reduced the computation memory required while maintaining reliable detection performance.

Additionally, through our experiments, we gained a deeper understanding of the traffic signs and lights, as well as the detection process employed by our chosen model. This enhanced understanding can contribute to further advancements in the field of object detection, particularly in the domain of traffic sign and light recognition.

In summary, our research has made valuable contributions by providing a dedicated dataset for traffic sign and light detection in Brazil, introducing a lightweight SSD-based model, and improving our understanding of the detection process. These contributions

collectively advance the field of object detection, particularly in the context of traffic management and safety.

6.2 FUTURE WORK

In terms of future work, there are several potential implementations that align with the theme of our research. Firstly, it would be valuable to conduct a comparative analysis with RetinaNet, another SSD-based model, to assess its performance as a classification technique for traffic sign and light recognition. Such a comparison could provide insights into the strengths and weaknesses of different SSD base models and potentially lead to improvements in accuracy and efficiency.

Expanding the scope of the dataset to include a broader range of traffic signs and the new type of traffic lights with four colors like in figure 49 and augmenting the number of objects for certain classes would also be a worthwhile endeavor by taking using less images with noise. By encompassing a more diverse set of objects, the model's robustness and generalization capabilities can be further enhanced. We can also consider the use of another type of optimizer like ADAM and the use of the F-Value as another metric to better understand how the model learns.

For practical applications, it would be beneficial to convert the trained model into the ONNX format, enabling its integration into mobile applications for real-life testing. This would facilitate the deployment of the system in a practical setting, allowing for validation and performance evaluation under real-world conditions.

Lastly, incorporating text-to-speech functionality in the system could enhance driver safety and attention on the road. By providing auditory descriptions of detected signs, drivers can focus on the road instead of constantly looking at the screen. This feature can contribute to a more user-friendly and distraction-free experience.

In wholesome, these potential future implementations aim to further enhance the accuracy, efficiency, and practical applicability of our traffic sign and light recognition system. By exploring these avenues, we can continue advancing the field and contribute to improved road safety and traffic management.



Figure 49 – Traffic light with a new color in Brazil. Extracted from (RIBAS, 2023).

REFERENCES

- ALGHMGMHAM, Danyah A.; LATIF, Ghazanfar; ALGHAZO, Jaafar; ALZUBAIDI, Loay. Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN. v. 163. [S.l.]: Elsevier BV, 2019. P. 266–274. DOI <https://doi.org/10.1016/j.procs.2019.12.108>.
- ALPAYDIN, Ethem. Introduction to machine learning. 2nd ed. Cambridge, Mass: MIT Press, 2010. (Adaptive computation and machine learning). DOI <https://doi.org/doi:10.1017/S1351324912000290>.
- ALTIS. Labelling Images for Object Detection with LabelImg — altisconsulting.com. [S.l.: s.n.], 2022. Disponível em: <<https://www.altisconsulting.com/insights/labelling-images-for-object-detection-with-labelimg>>.
- AMISHA; MALIK, Paras; PATHANIA, Monika; RATHAUR, VyasKumar. Overview of artificial intelligence in medicine. en. v. 8. [S.l.: s.n.], 2019. P. 2328. DOI https://doi.org/10.4103/jfmpe.jfmpe_440_19.
- AUTOMOTIVO, Meu Guia. O Que Significa as Cores do Semáforo? Guia Simplificado — meuguiaautomotivo.com. [S.l.: s.n.], 2022. Disponível em: <<https://meuguiaautomotivo.com/o-que-significa-as-cores-do-semaforo>>.
- BAZILIO, Gabriela Silvério; GUIMARÃES, Rafael Alves; NAZIF-MUNOZ, José Ignacio; OUIMET, Marie Claude; MAMRI, Asma; NETO, Otaliba Libânio Morais. Estimate of the magnitude of risky and protective behaviors associated with road traffic injuries in capitals participating in the Life in Traffic Project of Brazil. Ed. by Yanyong Guo. v. 17. [S.l.]: Public Library of Science (PLoS), Oct. 2022. DOI <https://doi.org/10.1371/journal.pone.0275537>.
- BHATT, Neel; LALDAS, Pratiksha; LOBO, Vivian Brian. A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN. Coimbatore, India: IEEE, June 2022. P. 1354–1358. DOI: <https://doi.org/10.1109/ICCES54183.2022.9835954>.
- BROWNLEE, Jason. A Gentle Introduction to Object Recognition With Deep Learning, 2019. v. 8. [S.l.: s.n.], 2019. P. 30. Disponível em:

<https://machinelearningmastery.com/object-recognition-with-deep-learning>.

CHEN, Junzhou; JIA, Kunkun; CHEN, Wenquan; LV, Zhihan; ZHANG, Ronghui. A real-time and high-precision method for small traffic-signs recognition. v. 34. [S.l.]: Springer Science and Business Media LLC, Sept. 2021. P. 2233–2245. DOI <https://doi.org/10.1007/s00521-021-06526-1>.

CHOUDHURY, Ambika. Top 8 Algorithms For Object Detection. en-US. [S.l.: s.n.], June 2020. Disponível em: <<https://analyticsindiamag.com/top-8-algorithms-for-object-detection>>.

CUNHA, Luis. Deep learning with Python (2^a ed) - François Chollet - Manning, outubro 2021, 504 pp. [S.l.]: Instituto Superior Miguel Torga, June 2022. P. 113–115. DOI <https://doi.org/10.31211/interacoes.n42.2022.r1>.

DAI, Jifeng; LI, Yi; HE, Kaiming; SUN, Jian. R-FCN: Object Detection via Region-based Fully Convolutional Networks. [S.l.]: arXiv, 2016. DOI <https://doi.org/10.48550/arXiv.1605.06409>.

DALBORGO, Vanessa; MURARI, Thiago B.; MADUREIRA, Vinicius S.; MORAES, João Gabriel L.; BEZERRA, Vitor Magno O. S.; SANTOS, Filipe Q.; SILVA, Alexandre; MONTEIRO, Roberto L. S. Traffic Sign Recognition with Deep Learning: Vegetation Occlusion Detection in Brazilian Environments. v. 23. [S.l.]: MDPI AG, June 2023. P. 5919. DOI <https://doi.org/10.3390/s23135919>.

DIWAN, Tausif; ANIRUDH, G.; TEMBHURNE, Jitendra V. Object detection using YOLO: challenges, architectural successors, datasets and applications. en. v. 82. [S.l.: s.n.], Mar. 2023. P. 9243–9275. DOI <https://doi.org/10.1007/s11042-022-13644-y>.

DUCHI, John; HAZAN, Elad; SINGER, Yoram. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. v. 12. [S.l.: s.n.], 2011. P. 2121–2159. Disponível em: <<http://jmlr.org/papers/v12/duchi11a.html>>.

DULHARE, Uma N.; AHMAD, Khaleel; KHAIROL AMALI BIN AHMAD; JOHN WILEY & SONS, INC (Eds.). Machine learning and big data: concepts, algorithms, tools and applications. Hoboken, NJ: Wiley-Scrivener, 2020. DOI <https://doi.org/10.1002/9781119654834.ch5>.

- DURYEA, Ethan; GANGER, Michael; HU, Wei. Exploring Deep Reinforcement Learning with Multi Q-Learning. v. 07. [S.l.]: Scientific Research Publishing, Inc., 2016. P. 129–144. DOI <https://doi.org/10.4236/ica.2016.74012>.
- ELAINA, Jeniffer. Quais são as maiores causas de acidentes de trânsito? | Smartia. pt-BR. [S.l.: s.n.], Aug. 2021. Disponível em: <<https://www.smartia.com.br/blog/maiiores-causas-acidentes-transito>>.
- ENGLISH GRAMMAR HERE. Traffic Symbol Signs and Road Symbols. en-US. [S.l.: s.n.], Feb. 2020. Disponível em: <<https://englishgrammarhere.com/symbols-and-signs/traffic-symbol-signs-and-road-symbols>>.
- EWAN. Image Recognition : A Complete Guide. en-US. [S.l.: s.n.], Jan. 2019. Disponível em: <<https://deepomatic.com/what-is-image-recognition>>.
- FREDJ, Hana Ben; CHABBAH, Amani; BAILI, Jamel; FAIEDH, Hassen; SOUANI, Chokri. An efficient implementation of traffic signs recognition system using CNN. v. 98. [S.l.]: Elsevier BV, Apr. 2023. P. 104791. DOI <https://doi.org/10.1016/j.micpro.2023.104791>.
- GIOVANNA. Brasil é o terceiro país com mais mortes de trânsito. pt-br. [S.l.: s.n.], May 2022. Disponível em: <<https://www.apm.org.br/ultimas-noticias/brasil-e-o-terceiro-pais-com-mais-mortes-de-transito>>.
- GIRSHICK, Ross. Fast R-CNN. [S.l.]: arXiv, 2015. DOI <https://doi.org/10.48550/arXiv.1504.08083>.
- GIRSHICK, Ross; DONAHUE, Jeff; DARRELL, Trevor; MALIK, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation.[S.l.]: arXiv, 2013. DOI <https://doi.org/10.48550/arXiv.1311.2524>.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep learning. Cambridge, Massachusetts: The MIT Press, 2016. (Adaptive computation and machine learning). DOI <https://doi.org/10.1007/s10710-017-9314-z>.

GOVERNMENT, Northern Territory. Road signs in the Northern Territory. en. [S.l.: s.n.], Dec. 2022. Disponível em: < <https://nt.gov.au/driving/safety/road-signs>>.

HAENLEIN, Michael; KAPLAN, Andreas; TAN, Chee-Wee; ZHANG, Pengzhu. Artificial intelligence (AI) and management analytics. en. v. 6. [S.l.: s.n.], Oct. 2019. P. 341–343. DOI <https://doi.org/10.1080/23270012.2019.1699876>.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. [S.l.: s.n.], 2014. DOI <https://doi.org/10.48550/arXiv.1406.4729>.

HOELSCHER, Igor Gustavo. Detecção e classificação de sinalização vertical de trânsito em cenários complexos. por. [S.l.: s.n.], 2017. Disponível em: <<https://lume.ufrgs.br/handle/10183/163777>>.

HOWARD, Andrew G.; ZHU, Menglong; CHEN, Bo; KALENICHENKO, Dmitry; WANG, Weijun; WEYAND, Tobias; ANDREETTO, Marco; ADAM, Hartwig. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. [S.l.: s.n.], 2017. DOI <https://doi.org/10.48550/arXiv.1704.04861>.

JEE, Gaurav; GM, Harshvardhan; GOURISARIA, Mahendra Kumar; SINGH, Vijander; RAUTARAY, Siddharth Swarup; PANDEY, Manjusha. Efficacy Determination of Various Base Networks in Single Shot Detector for Automatic Mask Localisation in a Post COVID Setup. v. 35. [S.l.]: Informa UK Limited, Oct. 2021. P. 345–364. DOI <https://doi.org/10.1080/0952813x.2021.1960638>.

JO, Taeho. Simple Machine Learning Algorithms. [S.l.]: Springer International Publishing, Dec. 2020. P. 69–90. DOI https://doi.org/10.1007/978-3-030-65900-4_4.

KARN, Aryan. ARTIFICIAL INTELLIGENCE IN COMPUTER VISION. v. 6. [S.l.: s.n.], May 2021. DOI <https://doi.org/10.33564/IJEAST.2021.v06i01.037>.

KINGMA, Diederik P.; BA, Jimmy. Adam: A Method for Stochastic Optimization. [S.l.]: arXiv, 2014. DOI <https://doi.org/10.48550/arXiv.1412.6980>.

LAFORE, Bruno. Erros humanos contribuem para 8 em cada 10 acidentes em rodovias federais — cnnbrasil.com.br. [S.l.: s.n.], 2022. Disponível em: <<https://www.cnnbrasil.com.br/nacional/erros-humanos-contribuem-para-8-em-cada-10-acidentes-em-rodovias-federais>>.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. v. 86. [S.l.: s.n.], Nov. 1998. P. 2278–2324. DOI <https://doi.org/10.1109/5.726791>.

LIU, Wei; ANGUELOV, Dragomir; ERHAN, Dumitru; SZEGEDY, Christian; REED, Scott; FU, Cheng-Yang; BERG, Alexander C. SSD: Single Shot MultiBox Detector. [S.l.: s.n.], 2016. P. 21–37. DOI https://doi.org/10.1007/978-3-319-46448-0_2.

MARIMUTHU, Parthiban. Dropout Regularization in Deep Learning — analyticsvidhya.com. [S.l.: s.n.], 2022. Disponível em: <<https://www.analyticsvidhya.com/blog/2022/08/dropout-regularization-in-deep-learning>>.

MARSLAND, Stephen. Probabilistic Learning. [S.l.]: Chapman and Hall/CRC, Oct. 2014. P. 153–16. DOI <https://doi.org/10.1201/b17476-7>.

MIN SU KIM; JI-HYE SEO; NAM KYU KWON; JU-MAN SONG; POOGYEON PARK. Real-time moving object detection using a vehicle-mounted monocular rear-view fisheye camera. Chiang Mai, Thailand: IEEE, June 2016. P. 1–6. DOI <http://doi.org/10.1109/ECTIcon.2016.7561385>.

NAIR, Vinod; HINTON, Geoffrey E. Rectified Linear Units Improve Restricted Boltzmann Machines. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2010. Disponível em: <<https://api.semanticscholar.org/CorpusID:15539264>>.

PA, Srudeep. An Overview on MobileNet: An Efficient Mobile Vision CNN. en. [S.l.: s.n.], June 2020. Disponível em: <<https://medium.com/@godeep48/an-overview-on-mobilenet-an-efficient-mobile-vision-cnn-f301141db94d>>.

- PADILLA, Rafael; NETTO, Sergio L.; SILVA, Eduardo A. B. da. A Survey on Performance Metrics for Object-Detection Algorithms. [S.l.]: IEEE, July 2020. DOI <https://doi.org/10.1109/iwSSIP48289.2020.9145130>.
- PALMIERI, Nayara. Sinais de Trânsito que todo motorista precisa conhecer. pt-BR. [S.l.: s.n.], Jan. 2021. Disponível em: <<https://www.despachantedok.com.br/blog/multas-de-transito/sinais-de-transito-que-todo-motorista-precisa-conhecer>>.
- PAN, Sinno Jialin; YANG, Qiang. A Survey on Transfer Learning. v. 22. [S.l.: s.n.], Oct. 2010. P. 1345–1359. DOI <https://doi.org/10.1109/TKDE.2009.191>.
- PASZKE, Adam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett. v. 32. [S.l.]: Curran Associates, Inc., 2019. DOI <https://doi.org/10.48550/arXiv.1912.01703>.
- PEDRAM ATAEE, PhD. Deep Learning in Simple Words — towardsdatascience.com. [S.l.: s.n.], 2021. Disponível em: <<https://towardsdatascience.com/deep-learning-in-simple-words-448e2c7f6ebe>>.
- PERE, Christophe. What are Loss Functions? en. [S.l.: s.n.], June 2020. Disponível em: <<https://towardsdatascience.com/what-is-loss-function-1e2605aeb904>>.
- PIERRE, Monhel Maudoony. thesis-ufu-mmp. [S.l.]: Mendeley, 2023. DOI <https://doi.org/10.17632/jbpsr4fvg9.1>.
- PON, Alex; ADRIENKO, Oles; HARAKEH, Ali; WASLANDER, Steven L. A Hierarchical Deep Architecture and Mini-batch Selection Method for Joint Traffic Sign and Light Detection. [S.l.]: IEEE, May 2018. DOI <https://doi.org/10.1109/crv.2018.00024>.
- REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [S.l.: s.n.], 2015. DOI <https://doi.org/10.48550/arXiv.1504.08083>.

- RIBAS, Hayana. Novidades nos semáforos: conheça a nova cor e a 4ª luz que prometem agilizar o tráfego urbano — multiversonoticias.com.br. [S.l.: s.n.], 2023. Disponível em: <<https://multiversonoticias.com.br/novidades-nos-semaforos-conheca-a-nova-cor-e-a-4a-luz-que-prometem-agilizar-o-trafego-urbano>>.
- SHAFIQ, Muhammad; GU, Zhaoquan. Deep Residual Learning for Image Recognition: A Survey. v. 12. [S.l.]: MDPI AG, Sept. 2022. P. 8972. DOI <https://doi.org/10.3390/app12188972>.
- SILVA, Francisco Assis da; SANTOS, Daniel Castriani; PEREIRA, Danillo Roberto; ALMEIDA, Leandro Luiz de; ARTERO, Almir Olivette; PITERI, Marco Antônio; ALBUQUERQUE, Victor Hugo C. de. Real-Time Traffic Sign Detection and Recognition using CNN. IEEE Latin America Transactions, v. 18, n. 3, p. 522–529, Apr. 2020. Disponível em: <<https://latam.ieceer9.org/index.php/transactions/article/view/680>>.
- SINGH, Aditya. Top 6 Object Detection Algorithms. en. [S.l.: s.n.], Sept. 2021. Disponível em: <<https://medium.com/augmented-startups/top-6-object-detection-algorithms-b8e5c41b952f>>.
- TAN, Mingxing; LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. INTERNATIONAL conference on machine learning. [S.l.: s.n.], 2019. P. 6105–6114. DOI <https://doi.org/10.48550/arXiv.1905.11946>.
- WALI, Safat B.; ABDULLAH, Majid A.; HANNAN, Mahammad A.; HUSSAIN, Aini; SAMAD, Salina A.; KER, Pin J.; MANSOR, Muhamad Bin. Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges. v. 19. [S.l.]: MDPI AG, May 2019. P. 2093. DOI <https://doi.org/10.3390/s19092093>.
- WHO. Road traffic injuries. [S.l.]: World Health Organization: WHO, June 2022. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>>.
- WILLIAM, Marco Magdy; ZAKI, Pavly Salah; SOLIMAN, Bolis Karam; ALEXSAN, Kerolos Gamal; MANSOUR, Maher; EL-MOURSAY, Magdy; KHALIL, Kerolos. Traffic Signs Detection and Recognition System using Deep Learning. [S.l.]: IEEE, Dec. 2019. DOI <https://doi.org/10.1109/iciis46948.2019.9014763>.

WONTORCZYK, Antoni; GACA, Stanislaw. Study on the Relationship between Drivers' Personal Characters and Non-Standard Traffic Signs Comprehensibility. en. *International Journal of Environmental Research and Public Health*, v. 18, n. 5, p. 2678, Mar. 2021. DOI <https://doi.org/10.3390/ijerph18052678>.

XU, Shuyuan; WANG, Jun; SHOU, Wenchi; NGO, Tuan; SADICK, Abdul-Manan; WANG, Xiangyu. Computer Vision Techniques in Construction: A Critical Review. en. *Archives of Computational Methods in Engineering*, v. 28, n. 5, p. 3383–3397, Aug. 2021. DOI <https://doi.org/10.1007/s11831-020-09504-3>.

YONEDA, Keisuke; KURAMOTO, Akisuke; SUGANUMA, Naoki; ASAKA, Toru; ALDIBAJA, Mohammad; YANASE, Ryo. Robust Traffic Light and Arrow Detection Using Digital Map with Spatial Prior Information for Automated Driving. v. 20. [S.l.]: MDPI AG, Feb. 2020. P. 1181. DOI <https://doi.org/10.3390/s20041181>.

ZHANG, Jianming; WANG, Wei; LU, Chaoquan; WANG, Jin; SANGAIAH, Arun Kumar. Lightweight deep network for traffic sign classification. v. 75. [S.l.]: Springer Science and Business Media LLC, July 2019. P. 369–379. DOI <https://doi.org/10.1007/s12243-019-00731-9>.

ZHANG, Wenjun. *Computational ecology: artificial neural networks and their applications*. Hackensack, NJ: World Scientific, 2010. DOI <https://doi.org/10.1142/7436>.

ZHU, Yanzhao; YAN, Wei Qi. Traffic sign recognition based on deep learning. v. 81. [S.l.]: Springer Science and Business Media LLC, Mar. 2022. P. 17779–17791. DOI <https://doi.org/10.1007/s11042-022-12163-0>.