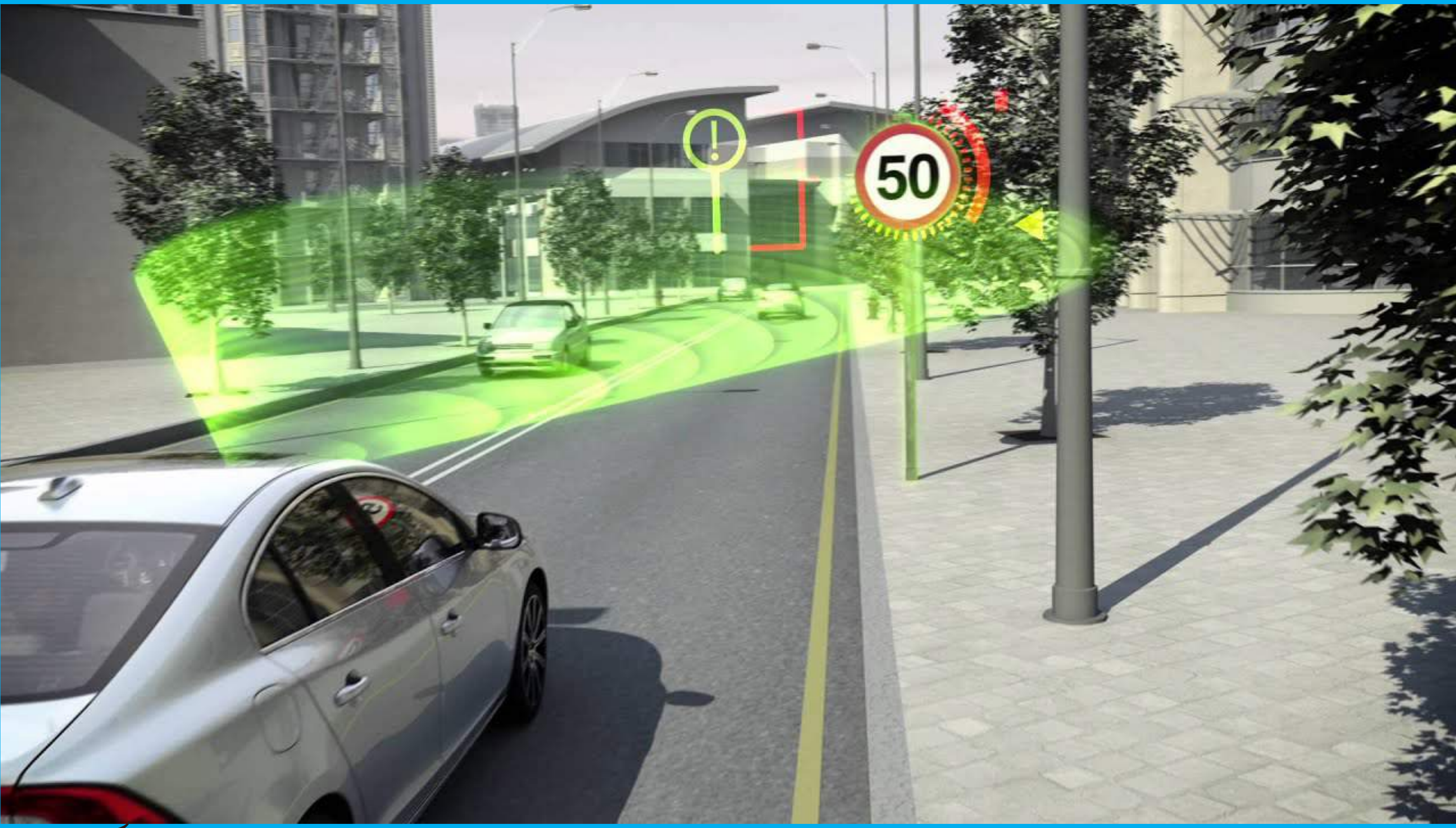
The Potential of Tiling on Traffic Sign Detection

Thesis

N. Skenderi

Delft university of technology
Department of mechanical engineering

4177754



TU Delft

The Potential of Tiling on Traffic Sign Detection

Thesis

by

N. Skenderi

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 21, 2022 at 11:00 AM.

Student number:	4177754	
Project duration:	Febuary 4, 2021 – March 3, 2022	
Thesis committee:	Dr. J. F. P. Kooij,	TU Delft, supervisor
	Prof. dr. ir. - ,	TU Delft

TU Delft

Preface

"The present is theirs; the future, for which I really worked, is mine." — Nikola Tesla

N. Skenderi
Delft, June 2021

Contents

1 Introduction 1
1.1 Background information . . . 1
1.1.1 Advanced Driver Assistance System, Intelligent Speed Assistance . . . 1
1.1.2 Autonomous vehicles . . . 2
1.1.3 Traffic sign maintenance . . . 3
1.2 Problem statement . . . 4
1.3 Research Questions . . . 5
1.4 Thesis layout . . . 5

2 Related Work 6
2.1 The evolution of traffic sign detection systems . . . 6
2.2 Traffic sign detection datasets . . . 8
2.3 Contributions . . . 10

3 Methodology 11
3.1 Method foundation . . . 11
3.2 Tiling method guide line . . . 15
3.3 German Traffic Sign Detection Benchmark Dataset . . . 16
3.4 Reproducing the results obtained by Arcos et al. . . . 19
3.5 Evaluation . . . 19

4 Experiments 24
4.1 Resize to batch . . . 24
4.2 Tile adaptation . . . 26
4.3 Hyperparameter optimization . . . 29
4.3.1 Non Maximum Suppression . . . 29
4.3.2 Confidence score . . . 32
4.4 Batch of Bordered Tiles . . . 33
4.5 Performance . . . 36

5 Conclusions 38

A Literature list 40

Bibliography 43

1 Introduction

1.1. Background information

Road safety is crucial to ensure the lives of all the road users. With year in, year out increase in traffic, it is becoming more and more important due to the increase in complexity and the increase of the scale of accidents. According to the World Health Organization [1] approximately 1.35 million people die each year as a result of road traffic accidents.

Obeying the traffic regulations enforced through traffic signs is obviously a necessity to avoid dangerous traffic situations, especially speed limits. The increase in average speed is directly related both to the probability of an incident occurring and to the severity of the consequences of the incident. Just a one percent increase in average speed results in a four percent increase in a fatal accident and a three percent increase in a serious injuries. This just shows how important abiding by the traffic rules can be in terms of lives but there is more. The expenses that have to be made to resolve and clean up all these incidents can go up to three percent of the gross domestic product for most countries. On top off the safety and the cost benefits, upholding the speed limits on high speed roads reduces the emission volume of greenhouse gasses. With the increase in world temperature and climate change, this is of coarse a very desirable side effect.

To prevent drivers from speeding a lot of behavioral research in combination with adjusted traffic regulations and innovative technologies have been conducted. A future proof and heavily studied system that detects and recognizes encountered traffic signs is the Traffic Sign Detection system (TSD). In doing so the system can assist the driver to uphold the traffic regulations enforced by encountered traffic signs. The TSD is a system that relies on on-board live camera footage and computer vision techniques in order to extract the desired information from the captured images, in this case traffic signs. Due to the increase in computational power throughout the recent years and the transition to automation, TSD has stirred interest.

The interest has propelled the development of such systems up to a point where some research papers [31] dare to claim that the recognition performance of such systems even exceeds human capabilities under certain assumptions. The achieved performance has made it possible for Traffic Sign Detection to be used for many different use cases where the emphasise lies at detecting traffic signs in order to perform a certain task. The most prominent and beneficial use cases are:

- Advanced Driver Assistance System, Intelligent Speed Assistance
- Autonomous vehicles
- Traffic sign maintenance

1.1.1. Advanced Driver Assistance System, Intelligent Speed Assistance

Advanced Driver Assistance Systems also known as ADAS's are almost indispensable but not always as noticeable in modern vehicles. Without drivers realizing it, all kinds of ADAS's are managing the vehicle in order to lighten the driving task and securing the safety of the driver and the surrounding. The Intelligent Speed Assistance (ISA), also known as Intelligent Speed Adaptation is an example of an ADAS that is literally expected to be a live saver.

ISA can either rely on an up to date digital map data containing all traffic signs in combination with a localisation system or on a Traffic Sign Detection system. For redundancy a combination of the two systems is desirable however the Traffic Sign Detection system is of course the future oriented solution.

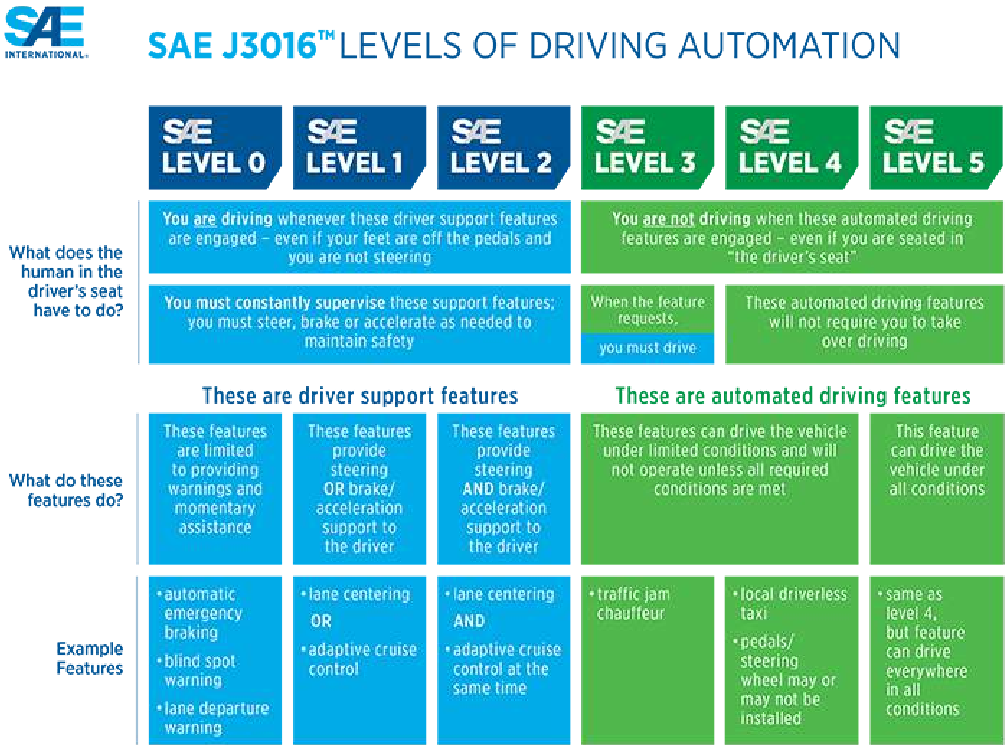
The Intelligent speed Assistance is intended to prevent drivers from unnecessary speeding. The system can be:

- Advisory, the system displays the speed limit and signals the driver whenever the speed limit changes as well as when the vehicle speed exceeds the speed limit.
- Voluntary, the system can prevent the driver to exceed the speed limit through an active acceleration paddle for example however the driver has the option to override the system by exerting more force on the acceleration paddle.
- Mandatory, the system overrides the drivers input whenever the speed limit is being surpassed by limiting the engines power up to the speed limit.

The European Union has introduced a new law that will set for launch this year. The new law will restricts all car manufacturers to design vehicles equipped with ISA. By 2024 all existing vehicles must have an Intelligent Speed Assistance system on-board in the European Union.

1.1.2. Autonomous vehicles

Self driving vehicles or vehicles with Autopilots are already among us but this is only the beginning of the road to fully autonomous vehicles. The current self driving commercial vehicles can be driven hands free on the highway for a short amount of time, but the drivers are required to constantly monitor the actions of the vehicle and keep their attention on the road. These vehicles rely on Lane Keeping System (LKS), Adaptive Cruise Control (ACC) and Traffic Sign Detection to manage the vehicle on the highway. As mentioned before this is only the beginning as the driving automation can be divided in six levels of automation as shown in figure 1.1.



SAE INTERNATIONAL

SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/ acceleration support to the driver	These features provide steering AND brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	•automatic emergency braking •blind spot warning •lane departure warning	•lane centering OR •adaptive cruise control	•lane centering AND •adaptive cruise control at the same time	•traffic jam chauffeur	•local driverless taxi •pedals/ steering wheel may or may not be installed	•same as level 4, but feature can drive everywhere in all conditions

Figure 1.1: The Society of Automotive Engineers (SAE): Six levels of automation

Traffic Sign Detection is an essential part of Autonomous driving. The higher the automation level the more disastrous the consequences of a miss call from the Traffic Sign Detection system can be. Therefore each level demands a better performance from the TSD system with ideally a perfect performance at level 5, full automation.

The benefits of autonomous vehicles heavily depend on degree of participation of autonomous vehicles in traffic and the level of automation. There can be personal benefits like greater mobility for people who cannot drive such as the elderly, children or people with disabilities. Greater convenience given that the driving task is done by the vehicle itself the driver or better said passenger is able to watch a movie or do some work while on the road. With more autonomous vehicles on the road, the roads are expected to become safer considering that almost all accidents are caused by human error. Ghost jams are a thing of the past when a certain degree autonomous vehicles on the road is achieved. Vehicle to vehicle communication and vehicle platooning can reduce energy consumption and thus environment pollution especially for large vehicles such as trucks.

1.1.3. Traffic sign maintenance

As aforementioned traffic signs are of utmost importance for road safety. Therefore it is important for traffic signs to be in recognisable conditions. There are certain regulations by law that every traffic sign must satisfy. For example the traffic signs has to have the right colour, shape and retroreflectivity. The importance of retroreflectivity must not be undermined. In figure 1.2 A number of stop signs are captured during the day. In figure 1.3 the exact same picture is taken during the night. Some of the stop signs appear to have disappeared in figure 1.3 due to the lack of retroreflectivity.

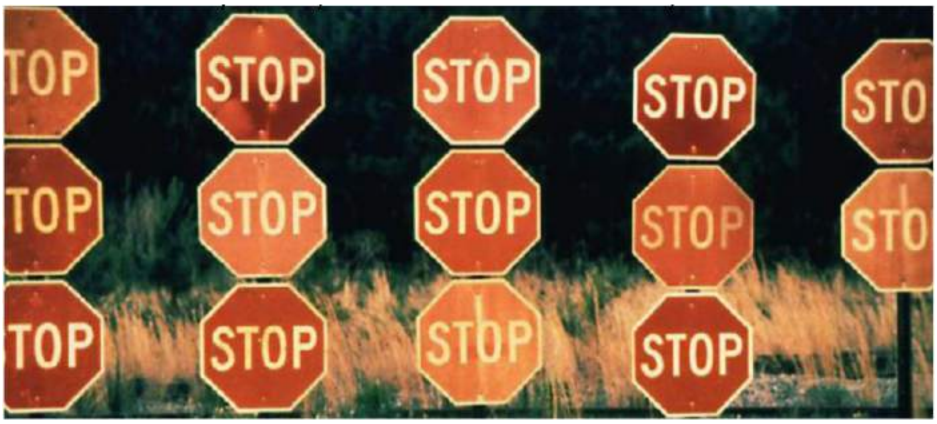


Figure 1.2: Retroreflectivity test[1], stop signs during day time

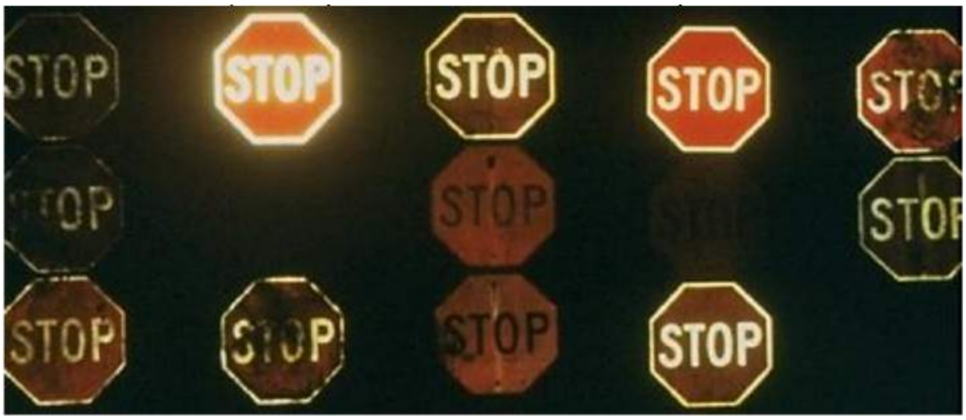


Figure 1.3: Retroreflectivity test, stop signs during night time

[1] https://saferroadsconference.com/wp-content/uploads/2017/06/1330_2-Urban-Camenzind-v1.pdf

In the Manual on Uniform Traffic Control Devices (MUTCD) and the Vienna Convention on Road Signs and Signals the minimal requirements to ensure and the methods to uphold road safety are stated. When a traffic sign does not meet these conditions, it essential to clean, fix or even replace the damaged signs as soon as possible in order to avoid dangerous traffic situations. Certain companies are employed for systematic management and maintenance of traffic signs by the local government. The methods to analyse the conditions of traffic signs manually are either labour intensive and thus costly or not precise and specific for each traffic sign. Traffic Sign Detection is already being used to tackle this problem to reduce cost and produce a good evaluation of each sign resulting in safer roads.

1.2. Problem statement

In general high accuracy object detection models are too slow to be implemented for real time applications because of the processing power that is needed to achieve the necessary speed. Models that do achieve real time object detection have in most cases a very low accuracy especially when dealing with small objects.

The TSD system obtains an input image 1.4 from the on board camera. By making use of a Traffic Sign Detection model the image is analysed and processed. The result is visualized in figure 1.5. One traffic sign is detected correctly however the smaller sign in the distance is overlooked. A correct detection of all signs in the image is shown in figure 1.6.

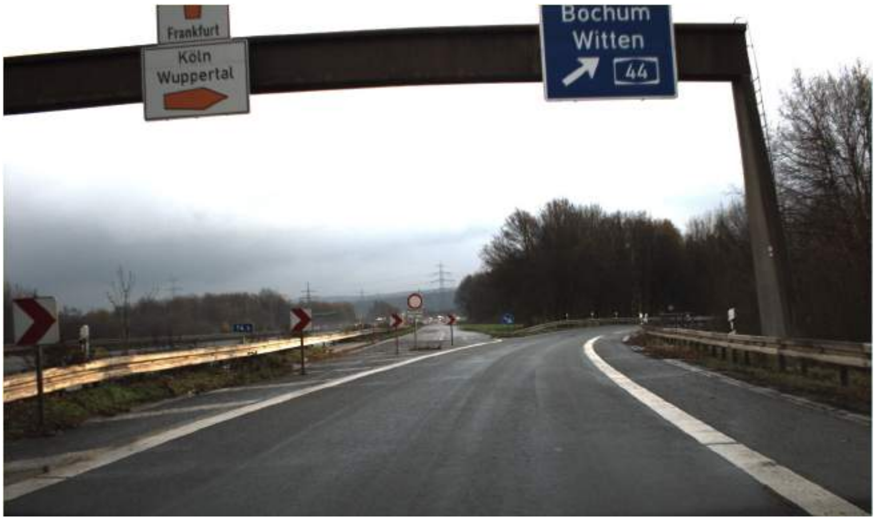


Figure 1.4: Input image

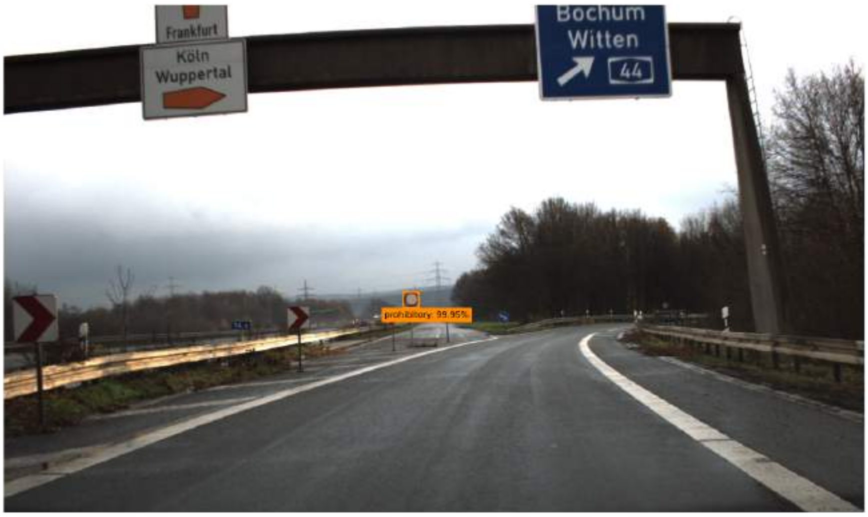


Figure 1.5: The challenge of small traffic signs detection, unable to detect small traffic signs.

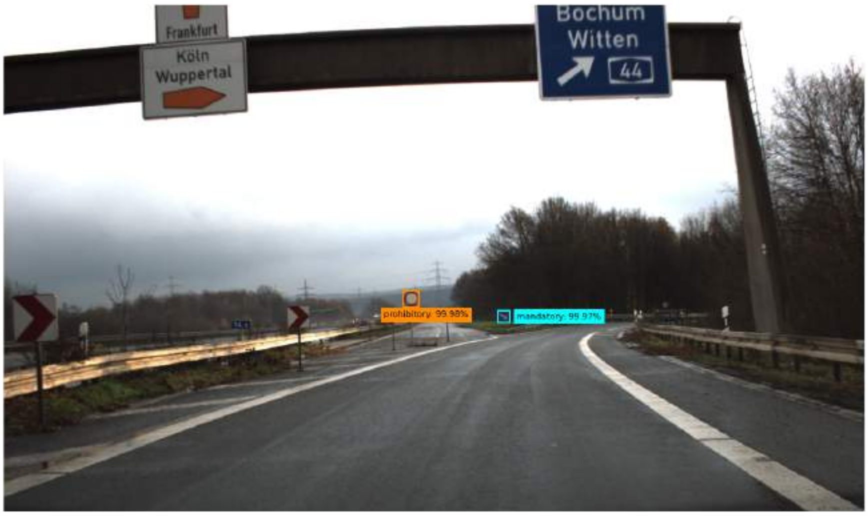


Figure 1.6: Small traffic signs detection

The goal of this thesis is to enhance a traffic sign detection model that requires relatively low processing power in such a manner that it can compete with traffic sign models that show great results and achieve high accuracy but require high processing power maintaining the ability to operate in real time.

1.3. Research Questions

In order to achieve the goals set for this research the existing computer vision techniques are analysed. The Tiling method is an interesting technique that is often used in the object detection domain but is yet to be used in the TSD domain. To obtain better understanding of the benefits and the possible drawbacks of the Tiling method the following research questions are stated:

- How to apply the Tiling Method to Traffic Sign Detection?
- How to keep inference time low while incorporating the Tiling Method to the Traffic Sign Detection Model?
- What are the optimal hyper-parameters of the entire model?
- How effective is adding borders to the tiles?
- How does applying the Tiling Method to a real time Traffic Sign Detection model compete with the state of the art models?

1.4. Thesis layout

A lot of research is already available on Traffic Sign Detection. Before building upon that a review of some excellent work that stand out is given in chapter 2. In chapter 3 the decisions and options to tackle the small traffic sign detection problem are stated. These options are then examined and researched one by one in chapter 4. Form the obtained results a conclusion is drown and with it the main questions are answered in the last chapter.

2
Related Work

In this chapter an overview of published articles on related problems is provided to acquire a deeper understanding of Traffic Sign Detection, associated challenges and past solutions. The importance of previous work must not be neglected. Most new research findings as in this thesis are inspired by existing solutions. Therefore in the next section the chronological development of Traffic Sign Detection methods is described. Followed by the state of the art Traffic Sign Detection models and the available Traffic Sign Detection datasets. The chapter ends with the contribution of this research to the field of Traffic Sign Detection.

2.1. The evolution of traffic sign detection systems

The first traffic sign detection systems used hand engineered features to detect certain properties of traffic signs, like color and shape. This is of course a great intuition because traffic signs are designed in predetermined distinct colors and shapes. During the traditional object detection period the focus was more on classification rather than localisation because classification was still a challenge to overcome.

The most common classical computer vision techniques are:

- Color threshold
- Haar wavelets [13] (Haar et al. 1910)
- Hough transform [8] Duda et al. 1972)
- Support Vector Machine [5] (Cortes et al. 1995)
- Adaptive Boosting [11] (Freund et al. 1997)
- Histograms of oriented gradients [7] (Dala et al. 2005)

Stand alone, shallow hand engineered systems were not enough to overcome the classification challenge. To solve this challenge, multiple algorithms were combined and cascades of classifiers were used. Many added their own implantation to the algorithms. All this was of course at the expense of inference time. The processing power is the limiting factor that makes solving the traffic sign detection challenges a challenge in itself. As we speak there is still a trade off to be made between accuracy and inference time.

The most influential paper over the decade declared by the International Association for Pattern Recognition is awarded to "In-vehicle camera traffic sign detection and recognition" by ruta et al. (2009) [29]. What made this work stand out was the fact that it could achieve good accuracy in real time. The algorithm uses for detection so called Quad-tree Region of Interest (RoI) extraction with color thresholds and Hough transform in combination with mean shift clustering which reduces detection duplicates. An integral feature map is used to speed up the process. Haar [13] and HOG [7] features were used in combination with the color information, acquired in the detection phase, and SimBoost, a purposed version of AdaBoost [11], for classification. This work was made possible due to the laid foundation by Viola et al. (2001) "Robust real-time face detection" [37]. A face detection algorithm which was by far the fastest algorithm at the time under comparable detection accuracy.

The best accuracy achieved on the German Traffic Sign Detection Benchmark is "A robust, coarse-to-fine traffic sign detection method" [38] by Wang et al. in 2013. The algorithm uses the sliding window method. In the so called coarse filtering stage regions of interest are selected using HOG as a feature extractor with Linear Discriminant Analysis (LDA) [10] as a classifier. In the fine filtering stage the candidate ROIs are up-scaled and reexamined using again HOG and Support Vector machine as a classifier. The result are outstanding on the GTSDB dataset but the inference time of a single image is several seconds which is of coarse not favorable. This is to be expected with the sliding window method which dense samples the entire input image thus demands high computational power.

Eventually the advancement of traditional methods stagnated. That was the case for the traffic sign domain as well as the entire object detection field. In 2012 the focus shifted form classical computer vision techniques to deep learning based detection method, the so called Alexnet moment. This was due to the jump in classification achieved by Krizhevsky et al. with "ImageNet Classification with Deep Convolutional Neural Networks" [18]. It was not the first DCNN deployed for detection and classification, other architectures already spurred interest.

A Convolutional Neural Network is in general a set of convolutional layers combined with an activation function and pooling layers that performs the feature extraction followed by fully connected, dense layers for classification and bounding box regression, figure 2.1.

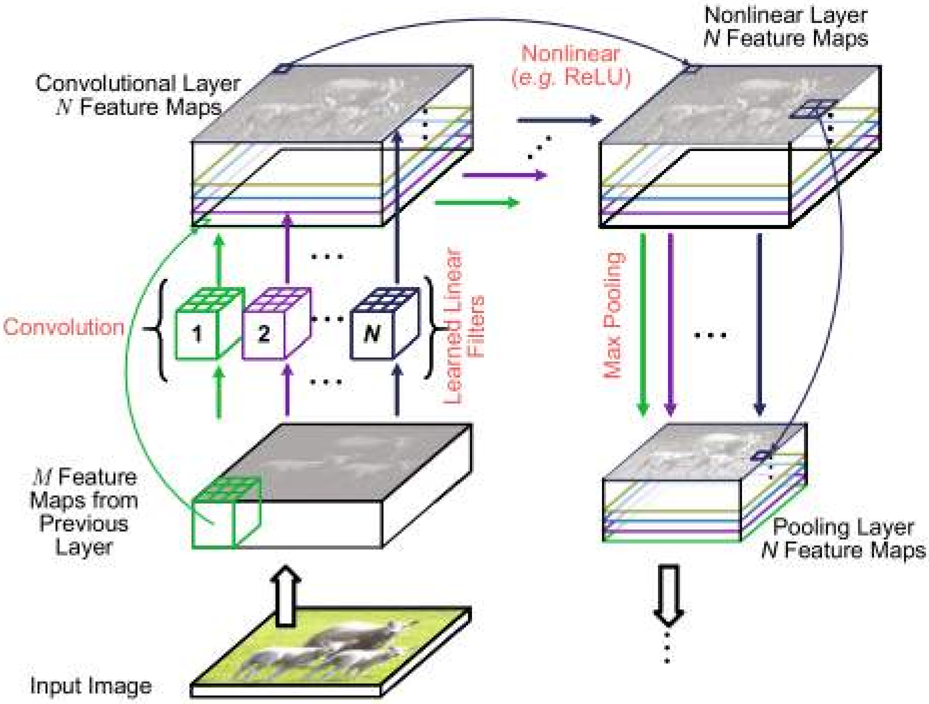


Figure 2.1: Repeating a general set of convolutional layers [23]

With each layer, the dimensions of the input image are reduced into an abstracted semantic output feature map. Through back propagation during the training phase the CNN adjusts the weights in the internal model in order to match the ground truth data. Batch by batch the result of the loss function is reduced up to the point of saturation. This is essentially how the CNN is able to learn detect certain features and eventually objects of interest. The features that the model learns to detect cannot be directly influenced. Only by training the model with an abundant and carefully chosen data samples, can the model learn the desired features. Unlike the traditional classification methods, the deep learning based classification methods are more robust to external conditions and are able to learn high level feature representations of images.

Once the usefulness of CNNs became clear, it did not take long before hybrid detection systems which used hand engineered algorithms for region proposals and used CNN for classification and bounding box regression. Region Based Convolutional Neural Network (R-CNN) [12] proposed by Girshick et al. in 2014 is an example of a hybrid detection system using selective search [30] for extracting Region of interests and a CNN for classification.

For performance speed up and end to end training Faster Region Based Convolutional Neural Network (Faster R-CNN) [28] used a Region Proposal Network (RPN) to extract regions of interest making the system a full convolutional neural network. These type of convolutional neural networks with region proposal modules are referred as two stage detectors. You Only Look Once (YOLO) [27] and Single Shot multibox Detector (SSD) [24] are one stage detectors in which the whole detection process is completed in one feed forward step. With the one stage detectors it is finally possible to easily achieve real time detection. By utilising similar methods like lightweight backbone networks and fewer ROIs proposals two stage detectors as Region-based Fully Convolutional Networks (R-FCN) [6] can achieve real time performance as well.

2.2. Traffic sign detection datasets

A thorough dataset is an essential requirement for computer vision based algorithms, especially for Convolutional Neural Networks. A dataset can literally make or break a CNN. Not only is it crucial for training, validation and testing but also for comparing different traffic sign detection techniques. Creating such datasets is very time consuming because apart from having to take a huge amount of images that include every possible traffic scenario, each object and preferably every aspect of the scene has to be most of the time manually annotated as well. During the classical computer vision era every research made use of small self made datasets making it unable to truly compare traffic sign detection system. Luckily that era came to an end.

The German Traffic Sign Detection Benchmark dataset (Houben et al. 2013) [15], figure 2.2 is the most widely used dataset in the traffic sign detection domain. It is originally used for computer vision and machine learning competition, introduced in IEEE International Joint Conference on Neural Network (IJCNN) 2013. The aim of the competition is to rank all traffic sign detection approaches and reveal the state-of-the-art traffic sign detection architectures. As the name implies the images were taken near Bochum in Germany throughout the year 2010. The GTSDB dataset contains 900 images (FullIJCNN2013), divided in 600 training images (TrainIJCNN2013) and 300 test images (TestIJCNN2013). Each image can contain up to 6 traffic signs in different view angles, optical distortions and lighting condition. The images were taken with a Prosilica GC 1380CH camera with a resolution of 1360 × 1024 pixels. Due to the setup the images were clipped to 1360 × 800 for the dataset. The sizes of the traffic signs in an image vary from 16 to 128 pixels with respect to the longer side. There are 43 different annotated traffic signs divided in four categories, prohibitory, danger, mandatory and others.

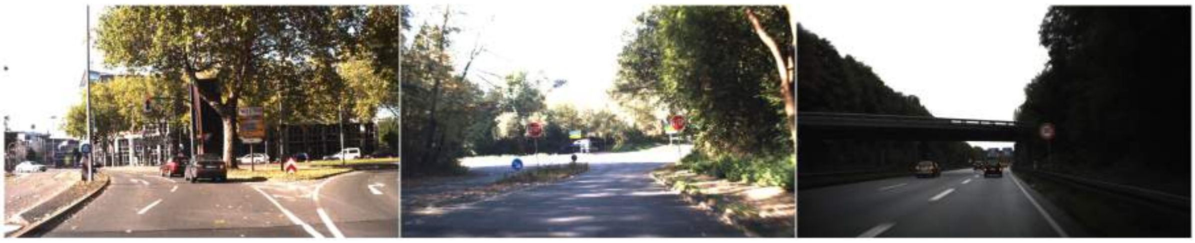


Figure 2.2: The German Traffic Sign Detection Benchmark dataset

Just like Germany, quite a few countries have there own traffic sign dataset. The Swedish Traffic Sign dataset , figure 2.3 was created by Larsson et al. (2011) primarily for their own research [21] [20]. The dataset contains 3488 traffic signs in more than 20000 images from over 350 km of Swedish city roads and highways. The 1280 by 960 images are from recorded sequences taken with a Point-Grey Chameleon, a 1.3 mega-pixel color camera. 20% of all images are labeled. The annotations contains 15 sign type (pedestrian crossing, designated lane right, no standing or parking, priority road, give way, 30 to 120 kph speed signs and others), four categories (information, prohibitory, danger, mandatory and others), image sign conditions (visible, occluded or blurred) and whether the signs is on the driven road or on a side road. The size of the traffic signs on the images range from 3 by 5 up to 263 by 248 pixels. Aside from Ireland, Malta and the United Kingdom, all European countries abide the Vienna Convention on Road Signs and Signals regulations.

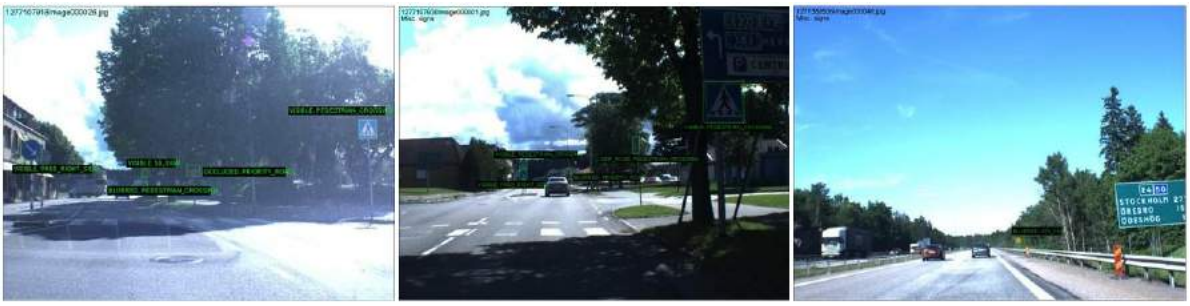


Figure 2.3: The Swedish Traffic Sign dataset

Tsinghua-Tencent 100K Benchmark dataset (zhu et al 2016)[41], figure 2.4 uses images from the Chinese Tencent Street View. Because of this, all the images are preprocessed by image enhancing techniques such as exposure adjustment. The dataset contains 30000 traffic signs from 100000 images, covering a large variation in lighting and weather conditions. The original 360 degree images were recorded by six SLR cameras and then pasted together. The original images were divided into four 2048 by 2028 images for the Tsinghua-Tencent 100K Benchmark dataset. There are 128 traffic signs classes of which only 45 are used and divided into four categories prohibitory, danger, mandatory and others. Asides from the class label and bounding box, the dataset provides the pixel mask as well. The size of a traffic sign can go from 2 by 7 to 397 by 394 pixels. The Chinese traffic sign regulations are very similar to the ones used in Europe.

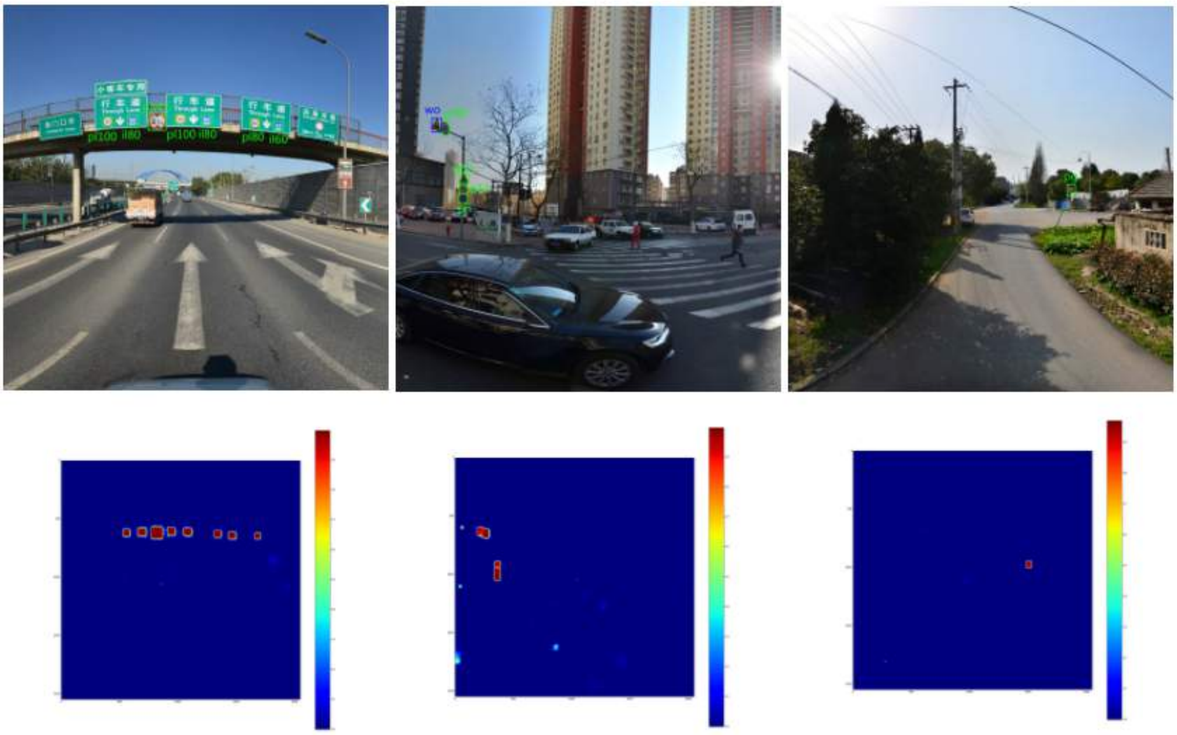


Figure 2.4: The Swedish Traffic Sign dataset

Challenging Unreal and Real Environments for Traffic Sign Detection (CURE-TSD), figure 2.5 is an interesting dataset created by temel et al. 2019 [34] in Belgium. The dataset exists of real world and synthesized video sequences generated in a virtual environment. This is a trend in different artificial intelligent fields where data is either scarce or restricted. Data augmentation is used to increase the amount of data. The real world video sequences are processed with 12 different types of effects with each five different levels of challenging conditions. These applied image processing types along with 14 distinct sign classes and associated bounding boxes are annotated. There are 896700 images containing 648186 annotated traffic signs. The sign size can range from 10 by 11 to 206 by 277 pixels.

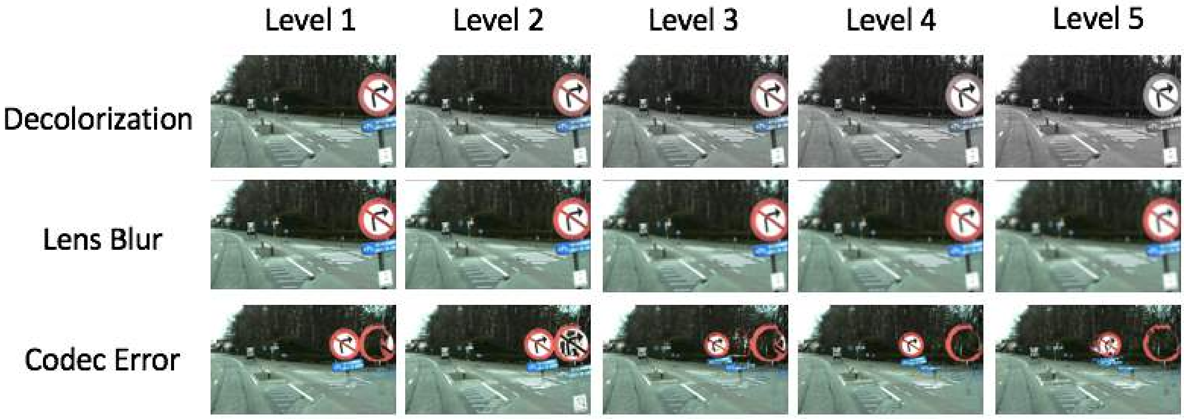


Figure 2.5: Challenging Unreal and Real Environments for Traffic Sign Detection

The LISA Traffic Sign Dataset [25], figure 2.6 consists of 47 different traffic sign types from the United States. All traffic signs in the US must uphold the Manual on Uniform Traffic Control Devices (MUTCD) regulations. The images are taken with different cameras resulting in image resolutions from 640 by 480 to 1024 by 522 pixels.The traffic sign size in the images can vary from 6 by 6 to 167 by 168 pixels. There are in total 7855 annotations on 6610 images. The annotations include sign type, bounding box, sign size, occlusion status and whether the signs is on the driven road.

Figure 2.6: The LISA Traffic Sign Dataset

Each of the five aforementioned traffic sign datasets have their own approach which makes them stand out among the other datasets, The Belgium Traffic Sign Dataset (BTSD) [35], The Dataset of Italian Traffic Signs (DITS) [40], Mapping and Assessing the State of Traffic InFrastructure (MASTIF) dataset [42]. There are also more recent datasets like the Waymo dataset [32] where the focus is more towards autonomous driving rather than traffic sign detection specifically.

2.3. Contributions

While there are object detection paper that depend on the Tiling Method there are not any Traffic Sign Detection papers that explore this method.

There are different approaches to the Tiling Method with each a specific use case. In this thesis the options of applying the Tiling Method to a pretrained model are explored while keeping the reference time as low as possible. The positive effects of the Tiling method have to be maximized for Traffic Sign Detection by finding the most beneficial tiling approach. In doing so the undesired side effects of the Tiling method have to be resolved. The optimal hyper-parameters of the entire model will be analysed for an increase in accuracy. Eventually to evaluate the potential of applying the Tiling Method, the model will be compared with the base model and the state of the art Traffic Sign Detection algorithms.

3

Methodology

This chapter elaborates on the methodology used to increase the accuracy of traffic sign detection models by improving the ability to detect small traffic signs in an image. The baseline method is explained in detail in section 1. This will represent the staring point of the experiments to keep track of the possible improvements that are made. The developments made to detect small objects in the object detection domain are reviewed in section 2. This knowledge will then be applied in section 3. In which the course of actions taken in this research to resolve the small traffic sign detection problem are presented. Eventually, in section 4, the evaluation methods that are used to understand the performance of the traffic sign detection model are described.

3.1. Method foundation

TSD is a specific branch of object detection. Object detection models are trained on TSD datasets to be used for Traffic Sign Detection. In most cases the models are not trained from square one, instead the filter weights of the initial layers of the model are fixed and the top layers are adjusted/retrained for the particular use case, Traffic Sign Detection. This is know as Transfer Learning bozinovski et al. [4]. In doing so a lot of training time can be spared without a noticeable loss in accuracy. The initial layers detect simple lines, edges and shapes, while the top layers detect complex features.

This Research is build up on the work of Arcos et al. (2018) "Evaluation of Deep Neural Networks for traffic sign detection systems" [3]. This paper provides a great overview of the performance of open source state of the art object detection systems particularly modified and adapted to the traffic sign detection domain. These models are pre-trained on the Microsoft COCO dataset and were fine-tuned through transfer learning on the German Traffic Sign Detection Benchmark dataset. To Analyse and evaluate the state of the art object detection model for the traffic sign detection task the following matrices were used: mean Average Precision, running time, the effect of traffic sign image sizes, memory usage and number of floating point operations (FLOPs). A visual representation of the result are presented in figure 3.1 One of the conclusions from this research is that object detection systems fine tuned through transfer learning produce close results to those obtained from specifically engineered state of the art systems for traffic sign detection.

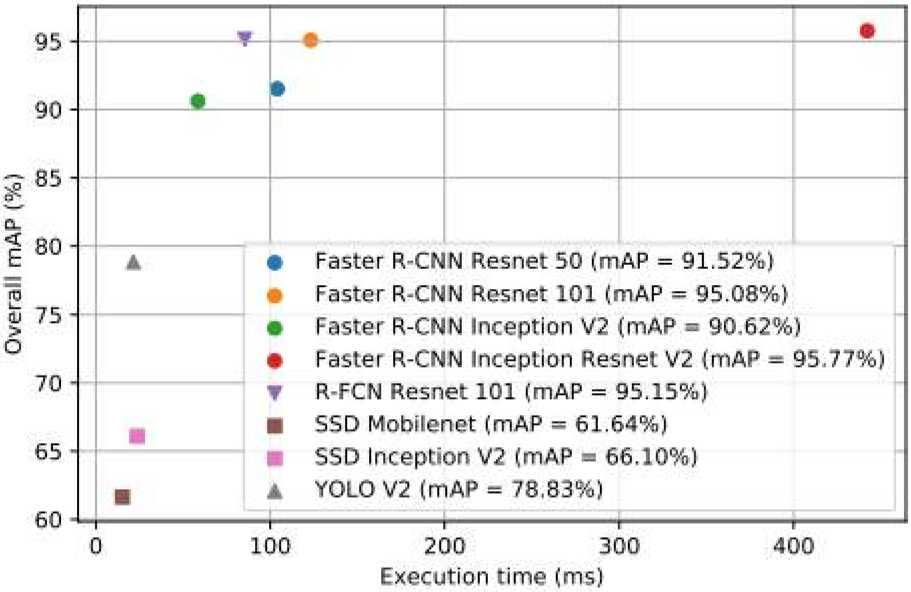
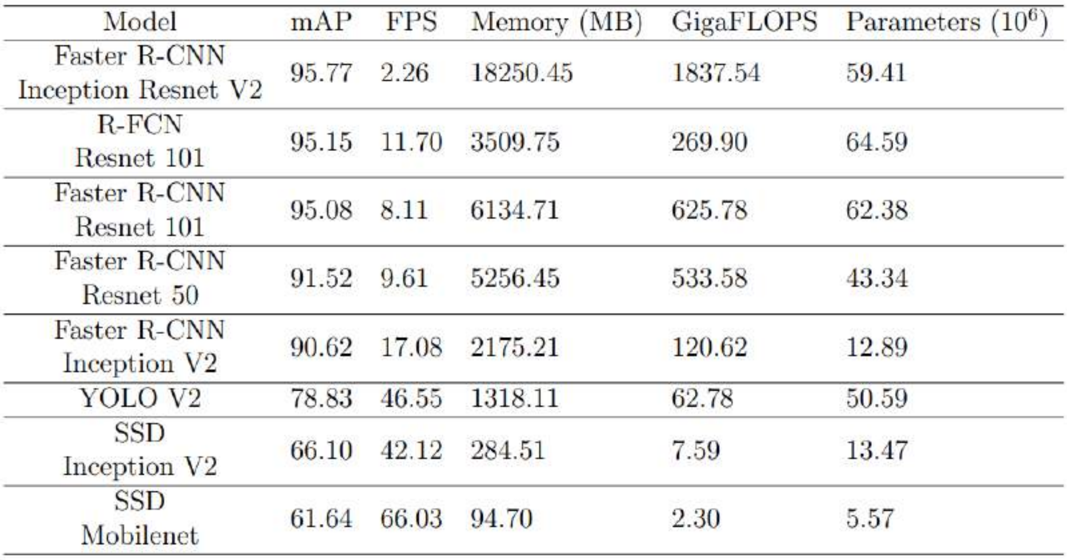


Figure 3.1: Visualisation results from Arcos et al.

The models used in this paper are all fully based on convolutional neural networks. Out of the eight (Faster R-CNN Resnet V1 50, Faster R-CNN Resnet V1 101, Faster R-CNN Inception V2, SSD Inception V2, Faster R-CNN Inception Resnet V2, R-FCN Resnet 101, SSD Mobilenet V1, and YOLO V2 Darknet-19), table 3.1, object detection systems the following systems produced noteworthy results.



Model	mAP	FPS	Memory (MB)	GigaFLOPS	Parameters (10^6)
Faster R-CNN Inception Resnet V2	95.77	2.26	18250.45	1837.54	59.41
R-FCN Resnet 101	95.15	11.70	3509.75	269.90	64.59
Faster R-CNN Resnet 101	95.08	8.11	6134.71	625.78	62.38
Faster R-CNN Resnet 50	91.52	9.61	5256.45	533.58	43.34
Faster R-CNN Inception V2	90.62	17.08	2175.21	120.62	12.89
YOLO V2	78.83	46.55	1318.11	62.78	50.59
SSD Inception V2	66.10	42.12	284.51	7.59	13.47
SSD Mobilenet	61.64	66.03	94.70	2.30	5.57

Table 3.1: Results from Arcos et al.

Faster R-CNN Inception Resnet V2 exhibits the highest accuracy with a mAP of 95.77 percent. A major drawback of this feature extractor is the inference or as stated by arcos et al. in figure 3.1 the execution time, 2.26 Frames Per Second (FPS) resulting in the slowest system by far. Being the deepest of the models with 164 layers this is of course to be expected.

The feature extractor, Inception Resnet V2 [33], uses Residual Inception Blocks that combines the computation efficiency of Inception modules and the optimization benefits of the residual learning framework of Residual neural network (ResNet) by He et al. (2016) [14].

Faster R-CNN [28], figure 3.2 is the first detector that is entirely based on deep learning due to the introduction of Region Proposals through a CNN called Region Proposal Network (RPN). Like every detector that uses region proposals Faster R-CNN has a two stage detection architecture which in the first stage Region of Interest (RoI) also called agnostic detection are proposed and in the second stage the proposed detections are refined and classified.

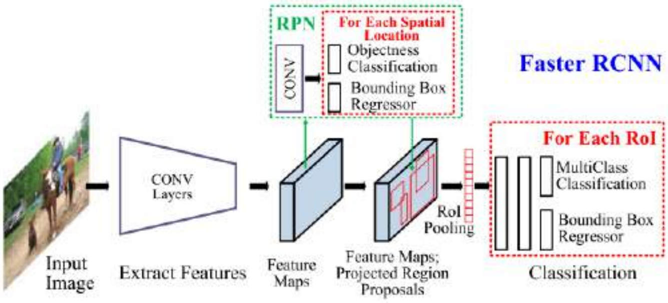


Figure 3.2: High level diagram of Faster R-CNN [23]

Note that in some papers, the feature extractor and meta architecture of an algorithm are called the backbone and head of a system.

R-FCN ResNet 101 produces the best trade off between latency and accuracy with 11.70 FPS and 95.15 percent respectively. A slight decrease in mAP of 0.62 percent but a substantial increase in latency namely 9.44 FPS with respect to Faster R-CNN Inception Resnet V2.

ResNet 101 (Residual neural network) [14] as the number suggests is 101 layers deep. Such deep networks are made possible because of the deep residual learning framework that is introduced. Normally after a certain amount of layers, depth, the accuracy would saturate and slightly after that it would even degrade rapidly. Instead ResNet gains accuracy with increasing layers.

Region-based Fully Convolutional Networks (R-FCN) [6] 3.3 is an improved Faster RCNN with comparable accuracy and lower latency. The increase in process speed is because of the shared computations throughout the system. The fully convolutional detector has to be trained for two contradicting concepts though, location invariance, to be able to classify an object wherever on an image and location variance, to be able to precisely locate the object on an image. This contradiction is compromised by using position sensitive score maps. These position sensitive score maps are convolutional feature map that are trained to recognize certain parts of the objects, figure 3.4.

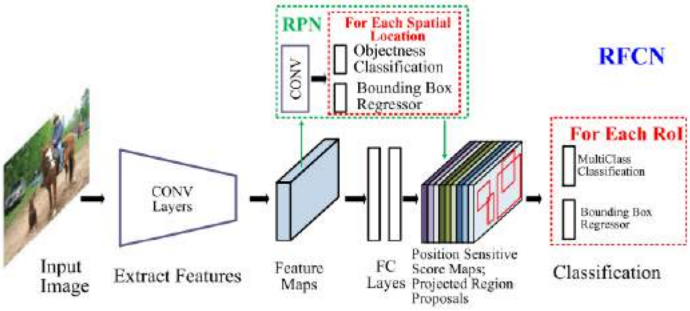


Figure 3.3: High level diagram of R-FCN [23]

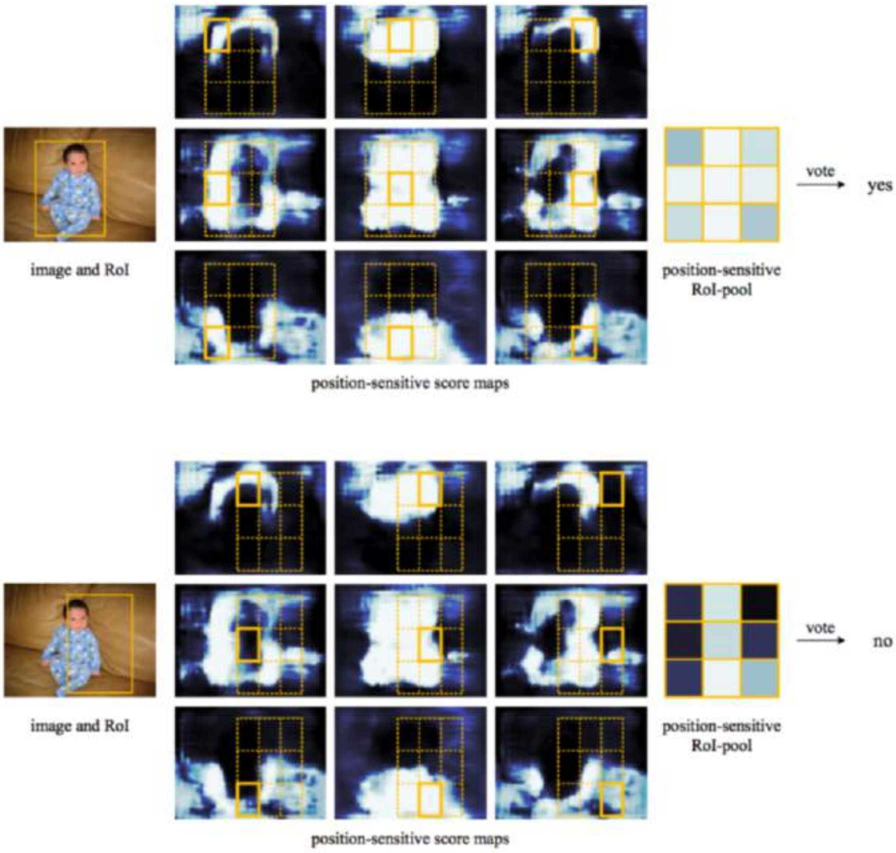
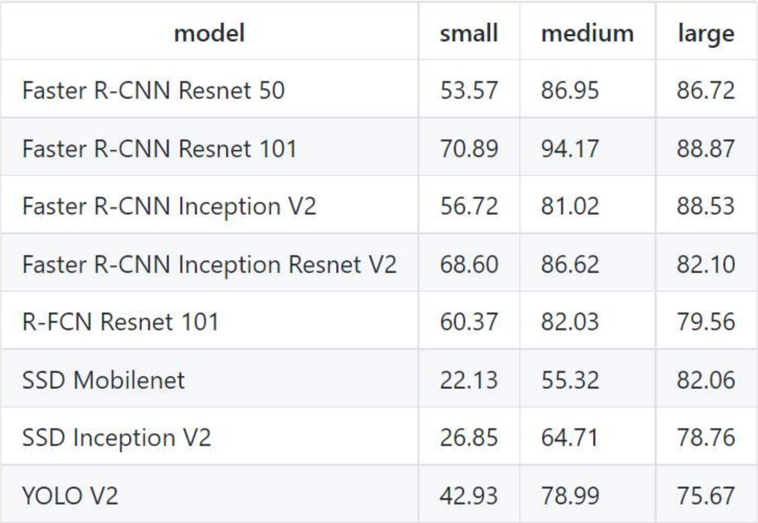


image and RoI
position-sensitive score maps
position-sensitive RoI-pool
vote
yes
image and RoI
position-sensitive score maps
position-sensitive RoI-pool
vote
no

Figure 3.4: Position sensitive score maps

SSD MobileNet v1 is the fastest model with 66.03 FPS but with the worst mAP of 61.64 percent. There is a huge difference in accuracy as well as latency compared to the previously mentioned traffic sign detection systems. The main reason of the low accuracy becomes clear by dividing the German Traffic Sign Detection Benchmark dataset into small, medium and large traffic signs table 3.2. With decreasing size, the difference in mAP become larger.



model	small	medium	large
Faster R-CNN Resnet 50	53.57	86.95	86.72
Faster R-CNN Resnet 101	70.89	94.17	88.87
Faster R-CNN Inception V2	56.72	81.02	88.53
Faster R-CNN Inception Resnet V2	68.60	86.62	82.10
R-FCN Resnet 101	60.37	82.03	79.56
SSD Mobilenet	22.13	55.32	82.06
SSD Inception V2	26.85	64.71	78.76
YOLO V2	42.93	78.99	75.67

Table 3.2: Mean Average Precision according to detection size

MobileNet v1 engineered by Howard et al. (2017) [16] is a 28 layer lightweight deep neural network that uses depthwise separable convolution. It separates standard convolution into depth-wise convolution, the filter stage and Point-wise Convolution, the combination stage. In doing so the amount of computations and model size are drastically reduced at the expense of a slight accuracy loss.

Single Shot MultiBox Detector [24] figure 3.5 has one stage meta architecture. It uses a single feed-forward convolutional network to directly predict anchor offsets and classes. This was the second meta architecture to do so but the novelty in SSD is the introduction of the multi reference and multi resolution detection technique. This technique significantly improves the detection accuracy of one stage detectors, especially for small objects which does not particularly show in this results. The theory behind it is that the first layers are better at detection small object due to the higher resolution and the last layers are better at detecting large objects due to the ability to extract complex features.

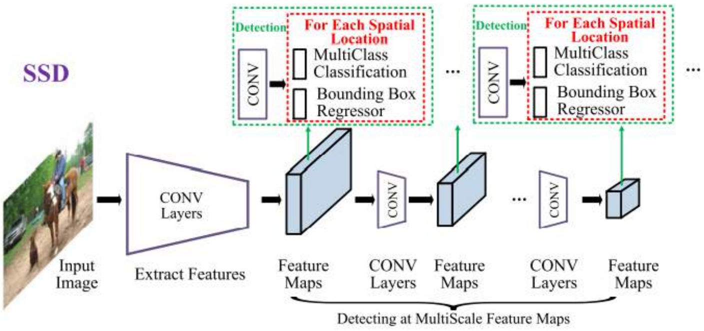


Figure 3.5: High level diagram of SSD [23]

The three aforementioned models were built in Tensorflow [2], which is an artificial intelligence library, in combination with the TensorFlow Object Detection API [17]. This API, Application Programming Interface simplifies the code to construct, train and deploy object detection models. The pre-trained models used by arcos et al. are provided by TensorFlow Object Detection API inside the Tensoflow Model Zoo.

3.2. Tiling method guide line

In "The Power of Tiling for Small Object Detection" Unel et al.(2019) [36] addresses the tiling method for the detection of pedestrians and vehicles onboard a Micro Aerial Vehicle (MAV). A step wise overview of the approach is shown in figure 3.6. The input image is tiled in six equally sized tiles. Subsequently the input image and the six tiles are processed one by one by the CNN. In the last step all the detections are merged and visualized in the final image, the output.

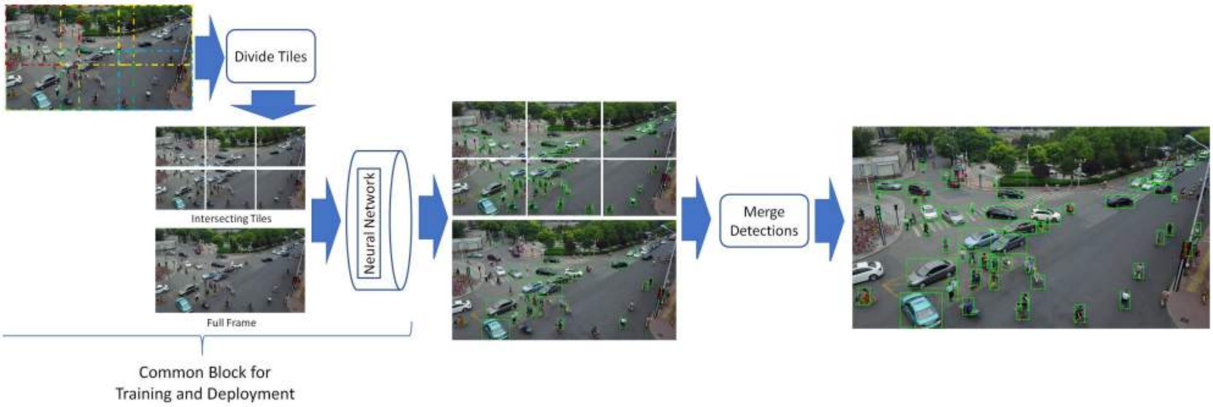


Figure 3.6: Unel et al. tiling approach.

The Pelee convolutional neural network [39] recommended by Unal et al. is a lightweight feature extraction framework has been chosen as backbone of the algorithm with SSD as the head. Different tile dimension were combined and researched during the training and inference phase. Training with slightly smaller tiles than used during the inference phase produced the best performance in terms of mean Average Position. Through the proposed method an increase from 11.48 percent to 35.88 percent mAP is achieved. The frames per second (FPS) went down from 101 to 6.3 on a TX2 GPU. According to the Embedded Real time Inference Challenge 5 FPS is the bare minimum to run real time.

The Power of Tiling for Small Object Detection will be more the guide line of this research adapted to the traffic sign detection domain. Unfortunately due to the absence of a powerful GPU like the TX2 the training steps will not take place. Therefore the pre-trained traffic sign detection models provided by Arcos et al. are used. By using these pre-trained models the time that is necessary for training these models, which could be weeks, is spared. This provides the opportunity to completely focus on applying and adjusting the tiling method to the traffic sign detection domain.

3.3. German Traffic Sign Detection Benchmark Dataset

To optimize, evaluate and compare the traffic sign detection model, the GTSDB dataset has been chosen. The 43 traffic sign classes divided in four categories are shown in figure 3.7. The frequency of the sign classes and the distribution of the categories are shown in 3.3 and 3.4 respectively. The annotation file contains the file names of the image were the respective traffic signs can be found. The ground truth bounding box coordinates and the class id of the traffic signs from 0 to 42 are represented as (file.ppm;ymin;xmin;ymax;xmax;id) in the CSV file.

It is worth mentioning that the GTSDB dataset is not flawless. In figure 3.8 encountered examples of missing annotations are displayed. Small traffic signs are more likely to be overlooked than larger traffic signs, especially when the images are not in sequence as is the case with the GTSDB dataset. This can lead to missing annotations in the train dataset making it more difficult for detection models to distinguish small traffic signs from the background. Annotation errors in the test dataset result in true positives to be calculated as false positives resulting in incorrectly lower accuracy values.

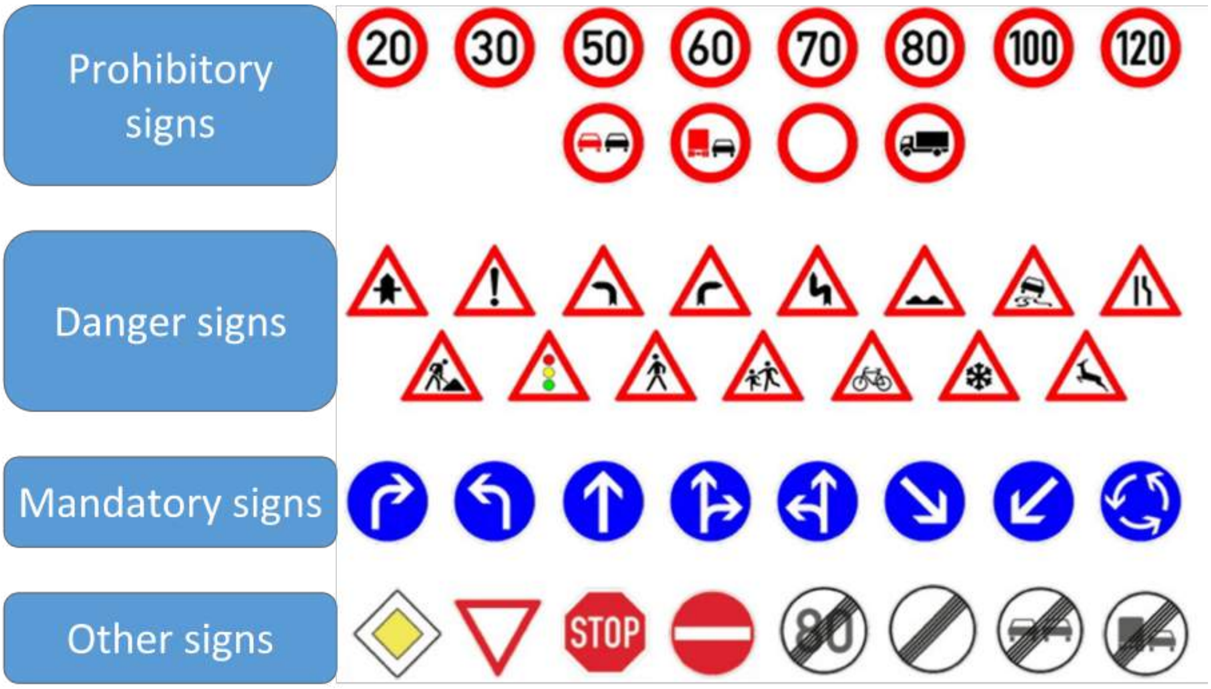


Figure 3.7: GTSDB Dataset traffic sign classes and categories

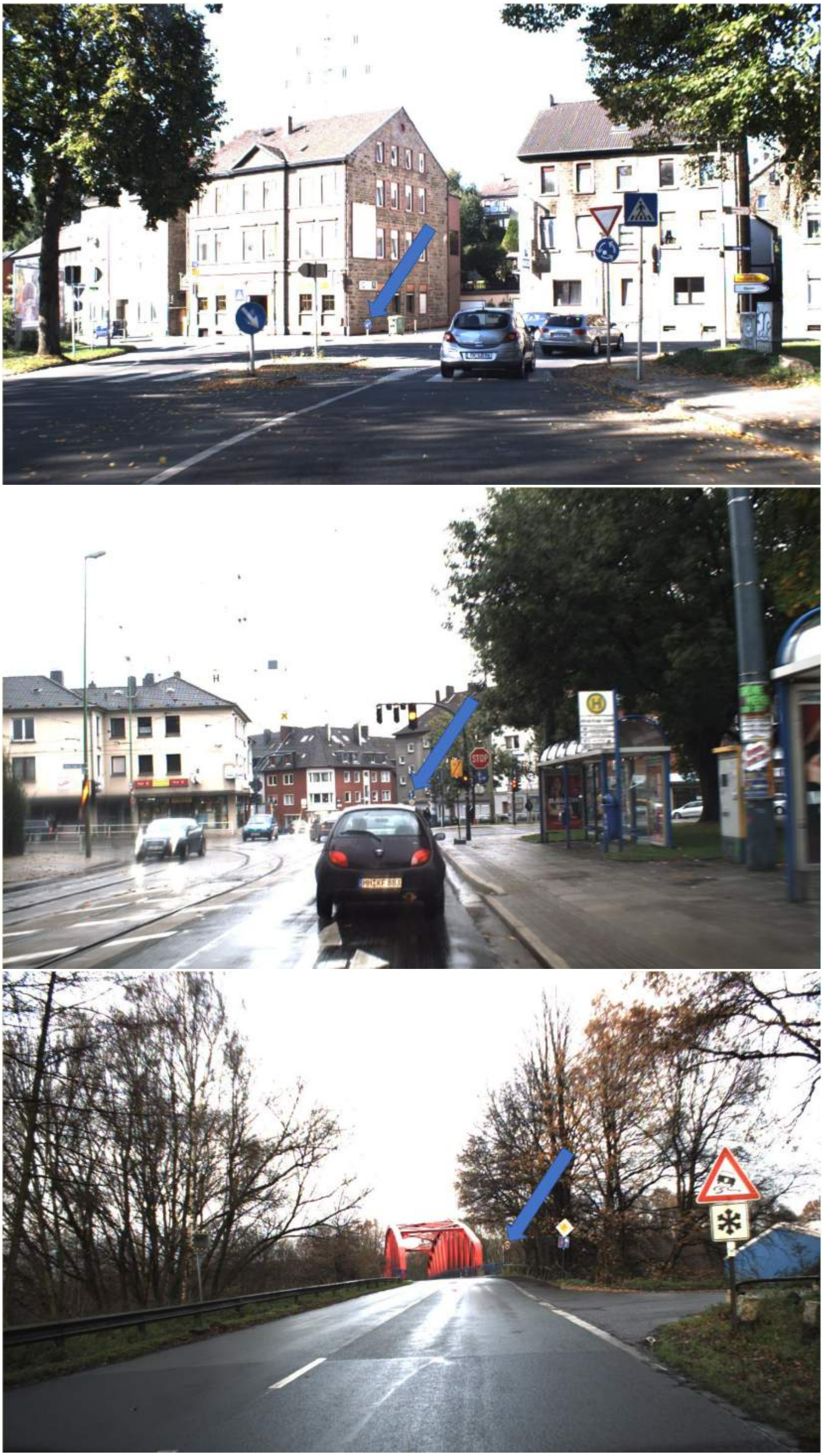


Figure 3.8: GTSDB images with overlooked small traffic signs, 00001 ,00548 and 00855

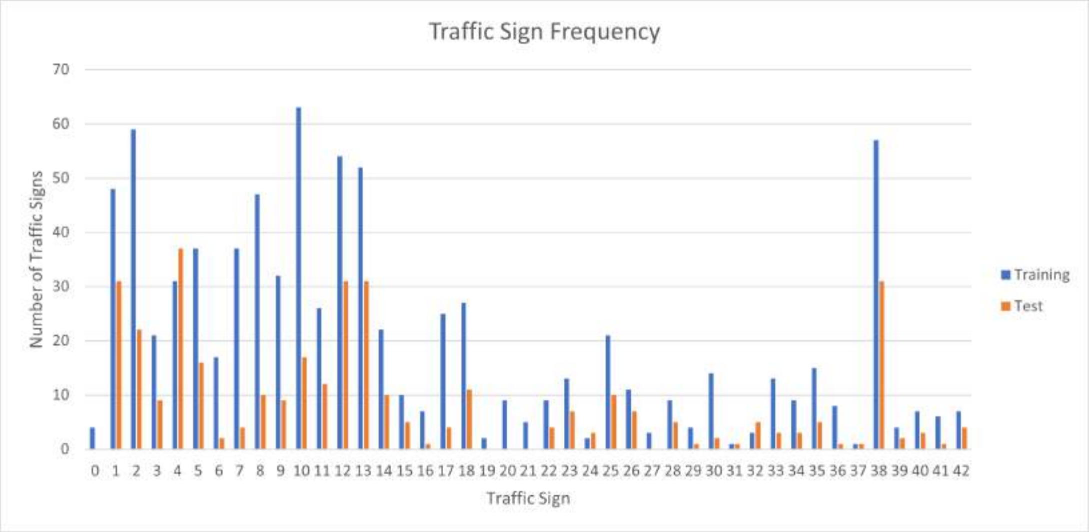


Table 3.3: Frequency traffic sign classes

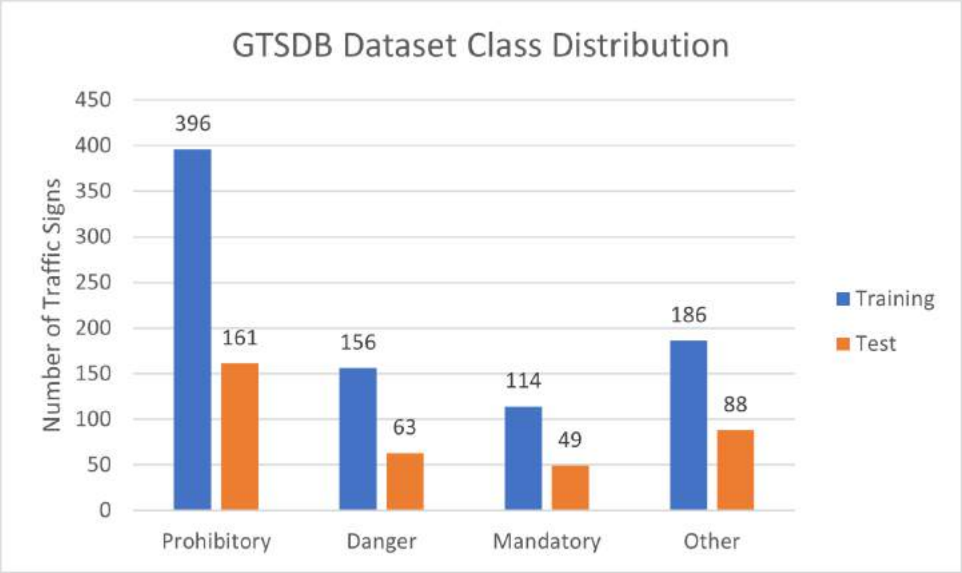


Table 3.4: Distribution of the categories

As mentioned before The German Traffic Sign Detection Benchmark dataset (Houben et al. 2013)[15] is the most widely used dataset in the traffic sign detection domain and above all Arcos et al. have trained the used models on the GTSDB Training dataset. The 600 training set images were all used for training and the 300 test set images were used for validation as well as testing. This is of course not the best way to approach validating and testing a model. In fact the best way to do that is to have a separate set of images for validation and another set for testing the model. Ideally, the images used for testing the model should be completely new encounters to examine whether the model can give the desired results in new situations. This approach gives better insight on how the model performs because the results obtained from validation which is actually part of training the model, are not due to overfitting.

Overfitting is a concept in machine learning that occurs whenever the model is trained up to the point for which it fits the training dataset exactly. The model is in that case that well adjusted to the training dataset that it has learned irrelevant information such as noise within the training dataset. That is why the main indication of overfitting is when training results are considerably better than the test results. This can be seen when comparing the results using the GTSDB training dataset and the GTSDB test dataset, table 3.5.

	Results Arcos et al. Microsoft COCO mean Average Precision	Reproduced Microsoft COCO mean Average Precision
GTSDB test dataset	37.68	37.36
GTSDB train dataset	59.07	58.08

Table 3.5: Reproduction of the results. The difference in train and test data.

3.4. Reproducing the results obtained by Arcos et al.

Reproducing the exact result is not an easy task. Most researchers do not explicitly write down every step they have taken. As it shows in table 3.5 even making use of the same pre-trained model will not grant the same result. Comparing CNN models is not as straightforward as one might think. This is apart from the different use cases in which one might prefer accuracy above speed or the other way around. This is the main reason why there is no clear cut best CNN model.

The exact same reproduced results were obtained when using a Zbook G3 and Google Colab. Therefore the difference in hardware and slight difference in software versions should not be the cause of this discrepancy between the Arcos et al. results and the reproduced results. Something that does stands out is that, while not stated, arcos et al. uses a Joint Photographic Expert Group (jpg) image file extension for the traffic sign detection during the training and test phase. This can be found in the provided code. The GTSDB dataset is provided in Portable Pixmap (ppm) format. The reason why arcos et al. chose to do this is unclear but this might be the cause of the discrepancy between Arcos et al. results and the reproduced results.

An attempt to still reproduce the same results as Arcos et al. is carried out by converting the image extension from ppm to jpg format using Python Image Library. As can be expected the conversion method from ppm to jpg has an impact on the detection. The conversion did produce different results but it still did not match the results obtained from Arcos et al. Unfortunately the conversion method is not stated in their research. With this can be concluded that it is not possible to match the exact results provided by Arcos et al. due to the lack of information. Therefore for this research the original image file extension, ppm, provided by GTSDB dataset will be used keeping in mind the discrepancy in the results.

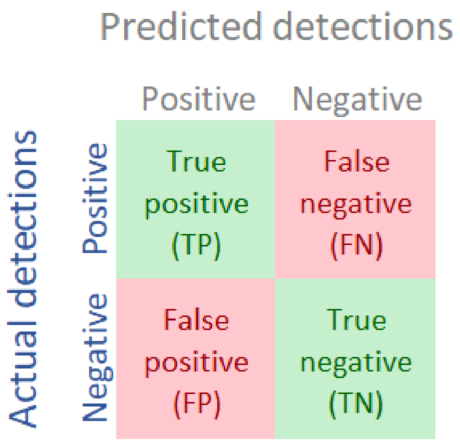
3.5. Evaluation

Once a model outputs the desired results, it is time for the evaluation. This is done in order to gain insight into the behaviour of the model. By doing so, the parameters of the model can be tweaked and tuned to a certain extend to fit its use case. The model is subjected to widely used standard metrices so that the results can be compared with the results of other models to determine the performance.

Traffic sign detection is just a specific branch of object detection. Which is why the metrices used for object detection are just as relevant for traffic sign detection. The most important metrices that are used in the German traffic sign detection benchmark/challenge [15] and most other object detection challenge like the PASCAL VOC Challenge [9], the COCO Object Detection Challenge [22] and the Open Images Challenge [19] are the precision recall curve, area under the curve, mean average precision and the average overlap. Each have a slightly different approach to these metrices. Before diving into these metrices it is important to mention the confusion matrix for binary classification figure 3.9, the foundation of the aforementioned evaluation metrices.

The ideal model would predict every actual detection exactly. Meaning detecting every true positive detection and nothing else. Unlike in medical research for example, the true negative does not have any meaning in traffic sign detection. Because any other pixel that is not part an object of interest is in fact a true negative. I order for a predicted detection to be classified as a true positive, false positive, false negative or true positive it is subjected to certain thresholds.

- The classification, the predicted detection must have the same class as the actual detection.
- The confidence score is the probability of a predicted detection being an actual detection.
- Intersection over Union (IoU) figure 3.10, also known as the Jaccard index is the area of the intersection divided by the area of the union of the predicted detection bounding box and actual detection bounding



		Predicted detections	
		Positive	Negative
Actual detections	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Figure 3.9: Confusion Matrix for Binary Classification

box also labelled as the ground truth bounding box.

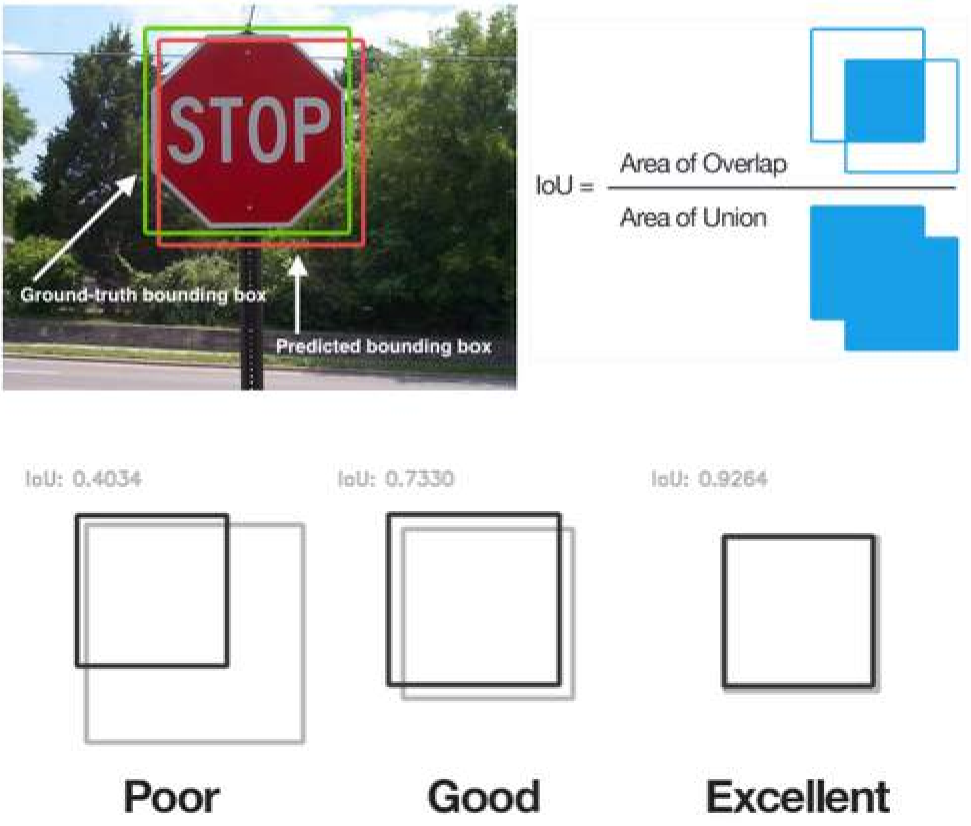


Figure 3.10: Intersection over Union[1]

[1] https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

For a predicted detection to be classified as a:

- True positive, the predicted detection must have the same class as the actual detection and satisfy the confidence and intersection over union threshold.
- False positive, the predicted detection does not satisfy one of the criteria or is a multiple predicted detection with a lower score. The latter is the case for the PASCAL VOC Challenge but in the German Traffic Sign Detection challenge and the other mentioned challenges a multiple predicted detection is neither classified as a true positive nor as a false positive but is ignored.
- False negative is an actual detection that is not predicted or the predicted detection does not meet one of the three criteria and the predicted detection results in a false positive.
- True negative can be seen as any other place/pixel on an image that is not covered by an actual detection or a predicted detection.

To go back to the ideal scenario, an exactly predicted detection would have the same class, a confidence score and an intersection over union of 1.

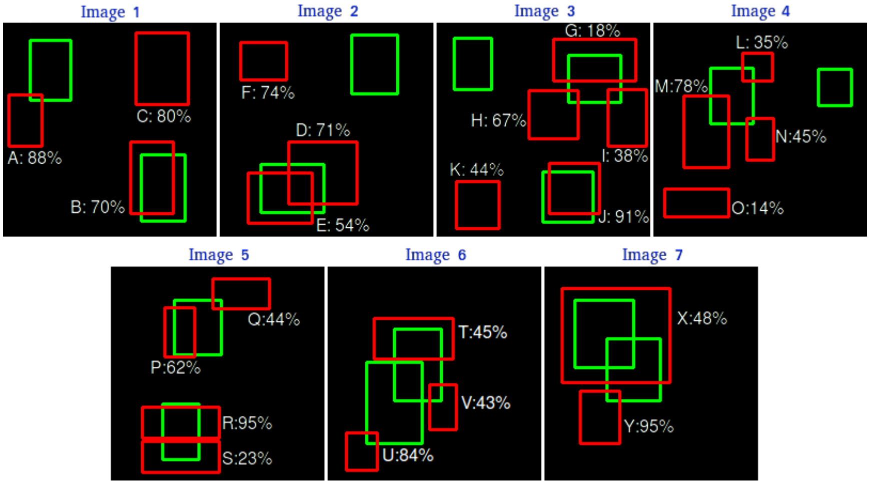


Figure 3.11: Images with groud truth bounding boxes (green) and predicted detections (red) [26]

Once every predicted detection of the dataset is classified as True Positive, False Positive and False negative, the precision and the recall can be calculated.

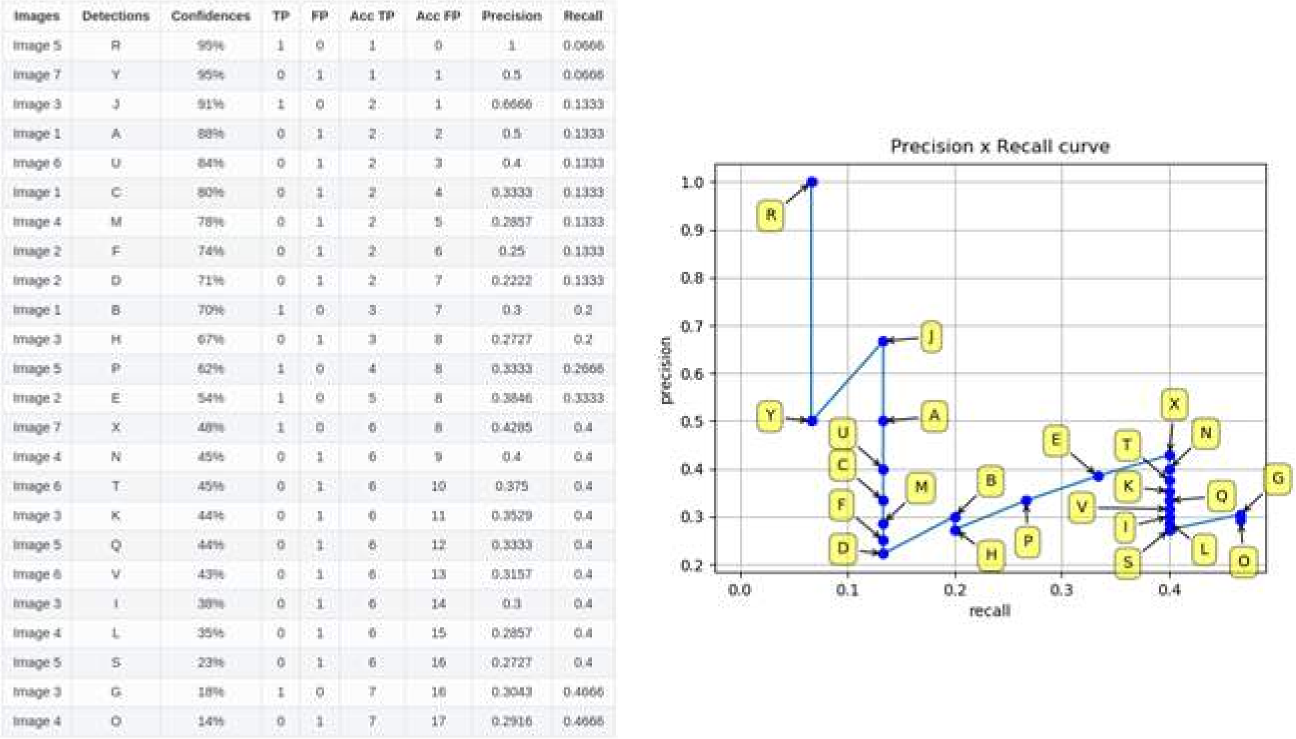
$$precision = \frac{TP}{TP+FP} = \frac{TP}{All\ predicted\ detections} \tag{3.1}$$

Precision, also known as positive predictive value, indicates the fraction of all predicted detections that are correctly detected. In other words, how “many” actual detections are detected per predicted detection. Notice that the true positives and false positives combined results in all predicted detections.

$$recall = \frac{TP}{TP+FN} = \frac{TP}{All\ ground\ truth\ bounding\ boxes} \tag{3.2}$$

The Recall, also known as the sensitivity, indicates the fraction of all actual detections that are correctly detected. In other words, the ability to detect actual detections. Notice that the true positives and the false negatives combined results in all ground truth bounding boxes.

By varying the confidence threshold from high to low values, the confusion matrix will change and with it the precision and the recall. When plotting the precision and recall for each confidence threshold with the precision on the horizontal axis and the recall on the vertical axis the precision recall curve is obtained, figure 3.12. The precision recall curve gives insight into the balance/trade-off between precision and recall. If undetected traffic signs are adverse for the use case and false detection are not an issue, as is the case for traffic sign maintenance for example, then a higher recall has the priority.



Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Figure 3.12: Precision recall curve construction[26]

By calculating the area under the interpolated precision recall curve, the average precision also known as Area under the Curve (AUC) is obtained. The r_{i+1} - r_i stands for the end and the start of the recall interval for which the the interpolated precision stays unchanged and $p_{interp}(r_{i\,+\,1})$ for the interpolated precision at that recall interval. Ideally the average precision would be of course one.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \tag{3.3}$$

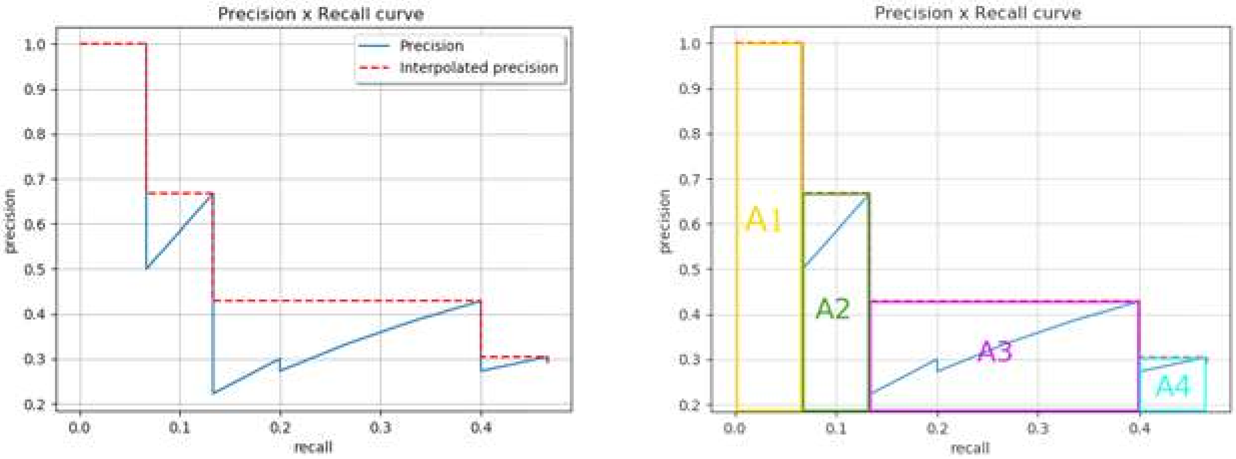


Figure 3.13: The average precision is the sum of the areas A1, A2, A3 and A4 [26]

The average precision is calculated for each class independently. The mean average precision is calculated by taking the mean of all average precision over all classes (K).

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{3.4}$$

The mean average precision can also be taken for different sized detection bounding boxes, small, medium and large for example. In this case K stand for the different size category.

The mean average precision used in the Microsoft COCO Object Detection Challenge evaluation takes it a step further and takes the mean average precision over different intersection over union thresholds. Varying from 0.5 to 0.95 with steps of 0.05. The Average of all steps results in the Microsoft COCO mean Average Precision evaluation metric.

$$Microsoft\ COCO\ mAP^{IoU = .50:.05:.95} \tag{3.5}$$

When there is an imbalance in classes in the dataset the micro mean average precision gives a better insight into the performance of the model. The micro mean average precision is measured by weighting each average precision according to the predicted detections of the corresponding class fraction (di) of all the predicted predictions (d)

$$Micro\ mAP = \frac{\sum_{i}^{K} (AP_i \times \frac{d_i}{d})}{K} \tag{3.6}$$

The mean average precision is a numerical performance indicator of the model. The higher the score the better the performance of the model. To get an even better understanding of the performance the average overlap is taken into account as well. This indicates how well each correctly predicted detection (true positive) matches the actual detection.

4 Experiments

This Chapter elaborates on the experiments and the obtained results. Firstly, the recorded image and the tiles must be of the same size in order to feed the recorded image and the tiles as a batch to the model. The resize options are compared in the search for the optimal performance. Secondly, the optimal position of the tiles is explored. Once the design of the system is set, the threshold of non maximum suppression and the confidence score of the system are analysed and adjusted to their optimal values. In the next section a solution to add borders to the tiles to reduce the increased false positives is examined. Finally, the improvements achieved by applying the new tile method to the SSD MobileNet v1 are compered with the best performing models mentioned by arcos et al. Al the experiments are carried out on the CPU of a HP Ultrabook Studio G3 with the following specifications:
Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59GHZ
8GB 2133 MHz (1 module van 8GB)
256GB SATA 3 - SSD
Windows 10 Home 64 bit multi
On power supply

4.1. Resize to batch

The adaptation of the tile method starts with the tiles. There are of course different approaches to apply the tiles, think about the size of the tiles, the different aspect ratios, the position and the number of tiles. The idea is to exclude unwanted parts of the recorded images and to zoom into the region of interest. The most obvious and straightforward method is to use a two times digital zoom in other words use dimensions for the tiles that are twice as small as the original image. That would lead to a tile of 680 by 400 with the same aspect ration as the original image. This is of course necessary to feed the original image and the tiles as one batch to the model in one inference cycle. By making use of a batch it is possible to utilize the computational resources more efficiently and speed up the inference time. The inference time per image in a batch should be theoretically less than using each image and tile individually as input to the internal model. Three option are tested to resize the image and tiles to the same dimensions.

1. Resize the tiles to the size of the recorded image. The advantage of this option is that the recorded image is untouched and the up scaled tiles could produce better results.

2. Resize the recorded image to the size of the tiles. This would mean that only one image would be down scaled and thus less computations are necessary resulting in a faster performance.

3. Resize the recorded image and the tiles directly to 300 by 300. These are the dimensions the SSD MobileNet v1 resizes every input to before processing.

The micro Microsoft COCO mean Average Precision evaluation metric is used to compare the three resize options. The results are shown in table 4.1.

	Option 1	Option 2	Option 3
Microsoft COCO mean Average Precision	68.83%	**70.02%**	66.50%

Table 4.1: Resize options

Surprisingly, the second option produces the best results. Apparently resizing images has a negative effect on the performance of the model. It is understandable to observe a decrease in performance for up scaling images because of the extra pixels that have to be created making the image not true to nature. With down scaling there is information loss which leads to a decrease in performance as well. However down scaling the image and the tiles to 300 by 300 should not make any difference because the resize to 300 by 300 is incorporated in the SSD MobileNet v1 model. This indicates that cv2.INTER_AREA, which is the function used for downs scaling the images, produces different results from the inbuilt SSD MobileNet v1 resize function. Different function with the same means to an end can produce different result as is the case for PIL.Image.BICUBIC and cv2.INTER_CUBIC for example. To compare these two functions the same image is processed by each function. The difference when subtracting an image obtained from supposedly the same function but different modules is shown in figure 4.1. To obtain a numerical understanding the three color channels of each image are added in order to have one channel to represent on the colorbar. The two images are then subtracted by one and other resulting in figure 4.2. The difference can go from -32 up to 38 combined color difference. This is not just a shift but actually pixel wise an entirely new image. This difference certainly has an effect on the performance of the model. By training the model with tiles using the same resize function the accuracy of the model can be improved. The impact of image processing on the performance was also considerable with different image file extensions as mentioned before.

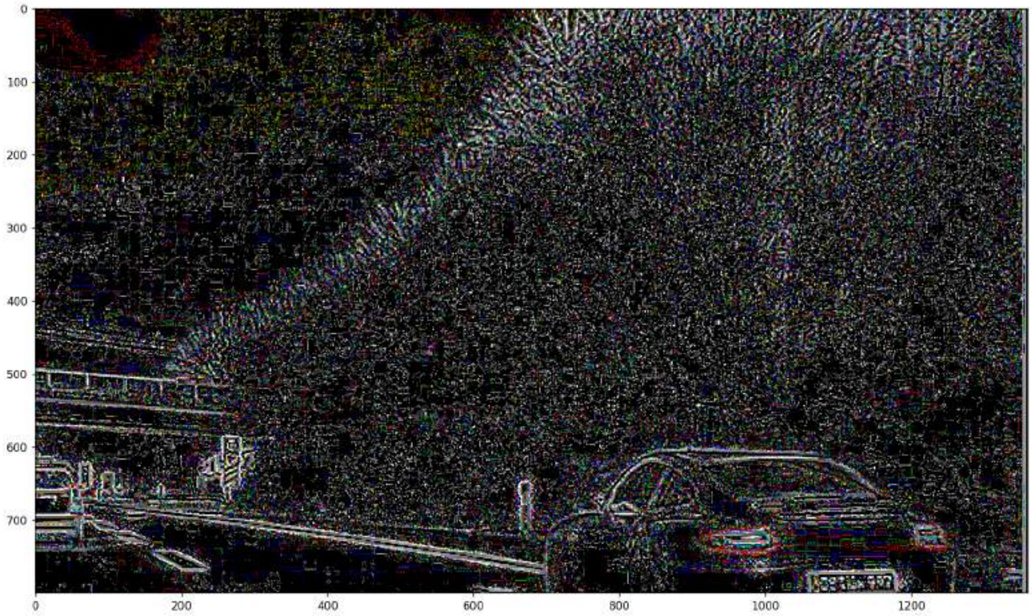


Figure 4.1: The difference when subtracting an image obtained from supposedly the same function but different modules.

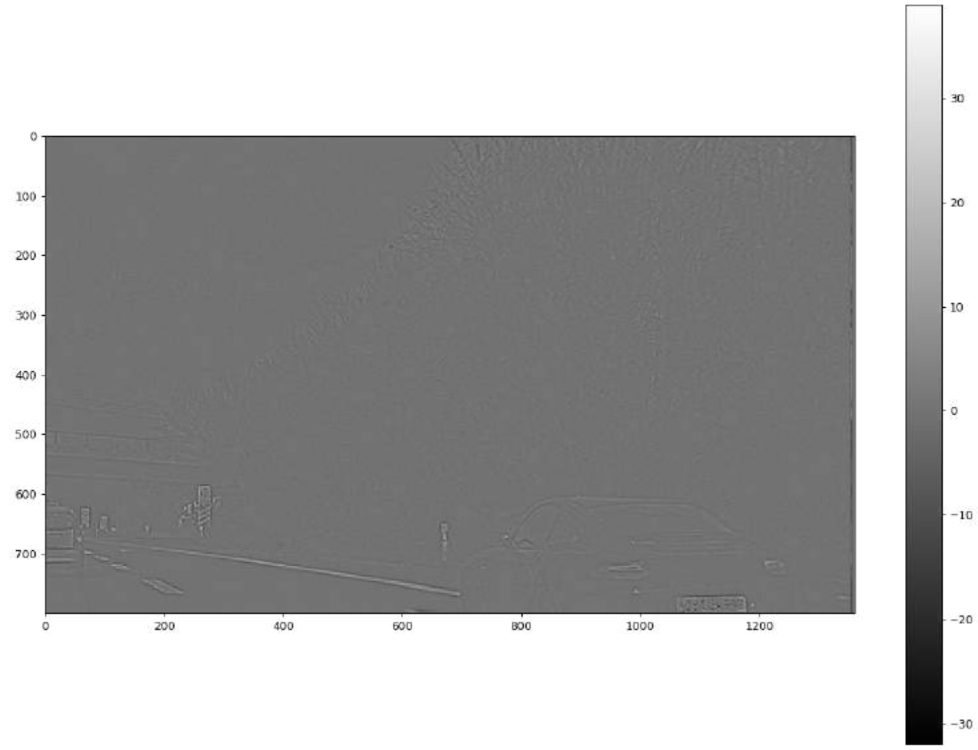


Figure 4.2: The difference when subtracting an one color channel image obtained from supposedly the same function but different modules in gray scale.

The difference in performance speed between the three options is not taken into account because the fluctuation in process time for the same test setup are relatively larger making the difference in performance speed negligible.

4.2. Tile adaptation

Traffic signs can always be expected to be found at more or less the same position with respect to the road due to the traffic sign guidelines of the Vienna Convention on Road Signs and Signals. That means that traffic signs will mostly show up on certain part of the recorded images. This should be especially the case with small, further away traffic sign. Covering the entire recorded image with tiles like Unel et al. proposes is unnecessary for traffic sign detection. To make use of this fact an analysis of the position of the ground truth bounding boxes in the training set images is performed. The results will give a good estimation of where and how the tiles should be placed in order to optimize the system for traffic sign detection.

A heatmap with the position of all ground truth bounding boxes of the train samples is visualized in figure 4.3 . The ground truth bounding boxes are spread throughout the length but are mostly within a certain range in the width of the images. Covering the whole image would mostly result in an increase in computation and thus inference time given that the bulk of the traffic signs are within a small range, especially for small traffic signs, boxplot 4.4. Traffic sign bounding boxes with a width less than or equal to 32 pixels are considered small signs. From 33 up to 46 pixels are considered medium size and bounding boxes with a width larger than 46 pixels are considered large.

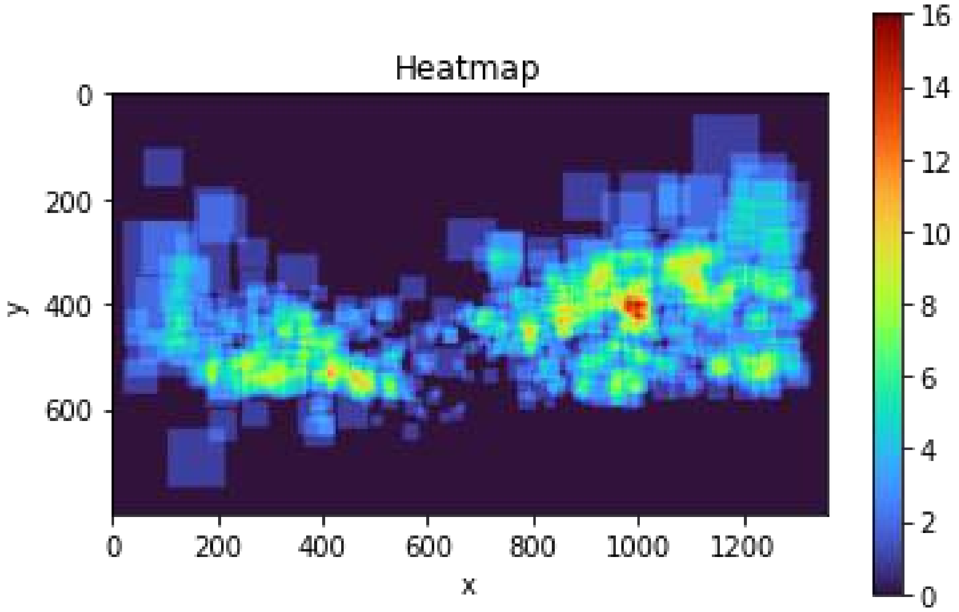


Figure 4.3: Heatmap of all GTSDB ground truth bounding boxes

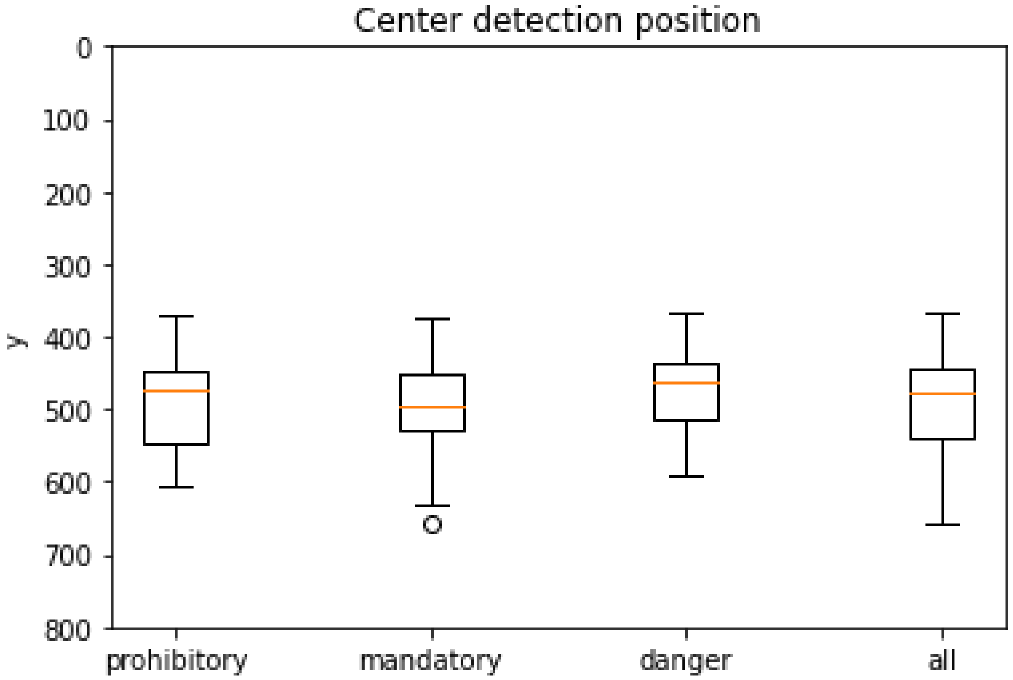


Figure 4.4: Boxplot of the vertical dispersion of all small ground truth bounding boxes

In order to incorporate truncated traffic signs at the border of the tiles, three tiles were used that will cover the horizon-line of the image, resulting into a left and right tile with both a 50 percent overlap with the center tile see figure 4.5. Since the length of the image is covered entirely by the three tiles, the vertical position of the tiles is the only and most important parameter left to find in order to optimize Microsoft COCO micro mAP performance. To find this optimal position where the tiles will capture the most ground truth bounding boxes, the position in the width of the image is used as a variable. The center-line of the tiles varies form 0 to 800 in the width of the image. This results in figure 4.6.

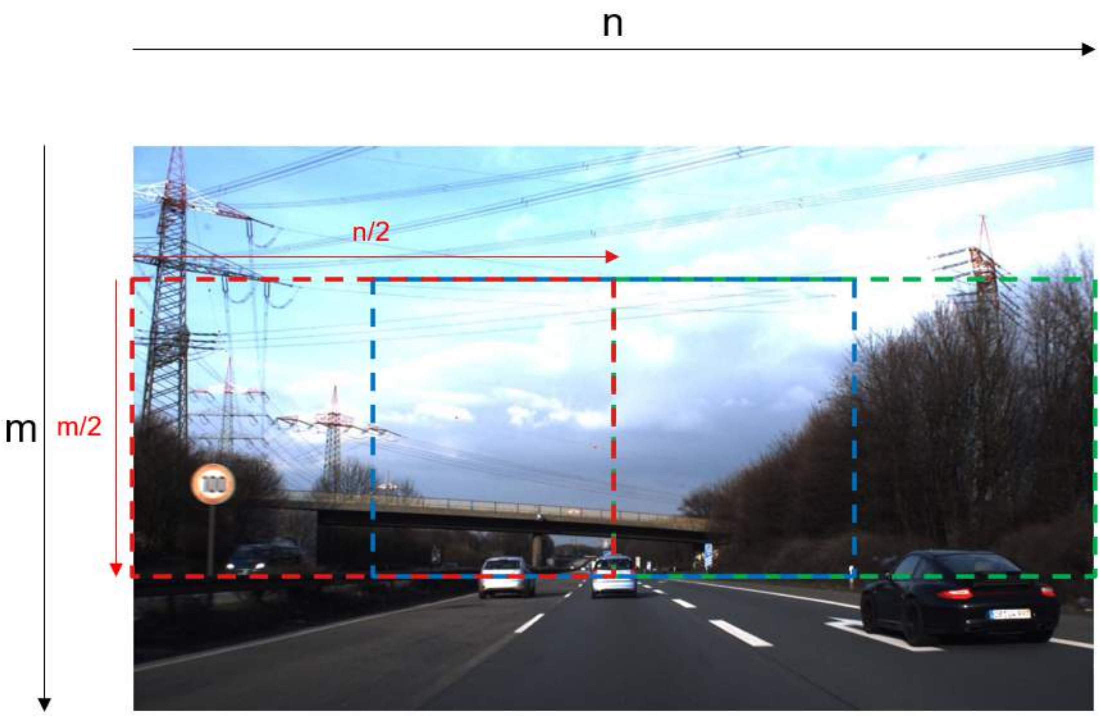


Figure 4.5: Tile dimensions

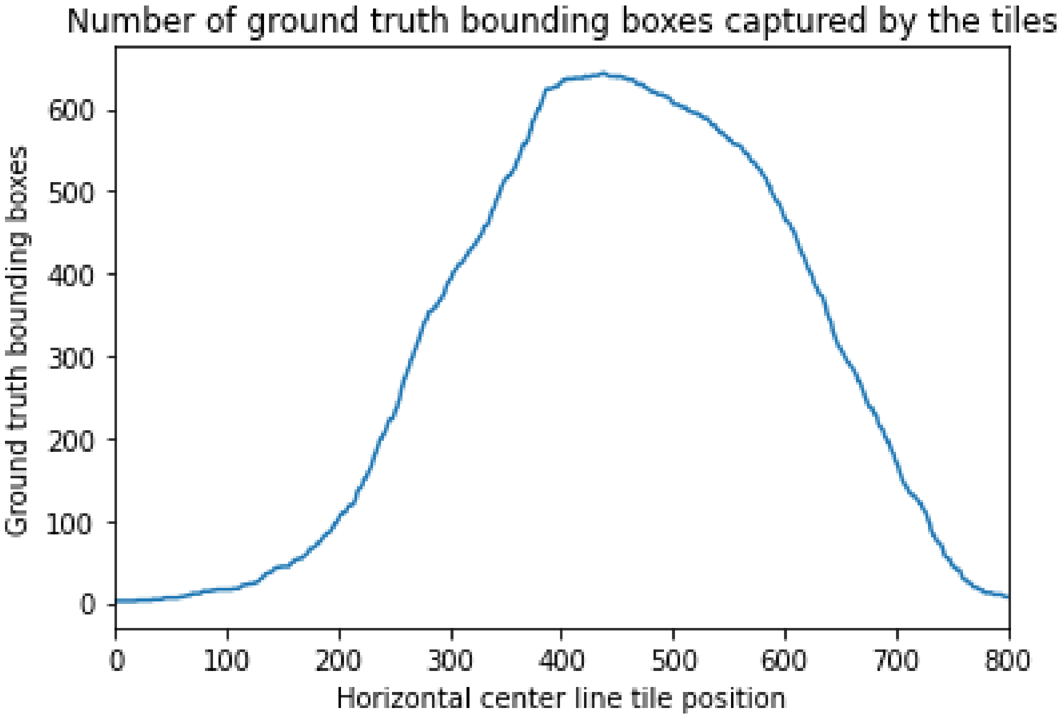


Figure 4.6: Number of bouding boxes captured when moving the slices in the y axis

The bounding boxes are divided into three size categories, small, medium and large. With the three tiles all the small ground truth bounding boxes are captured with tile center-line position from 460 to 567. The same is true for all medium sized ground truth bounding boxes for the range from 452 to 484. While for the large sized not all ground truth bounding boxes can be included and the maximum is reached at 438.

With this information available instead of examining all 800 positions of the tile centre line a smaller ranger can be strategically chosen to save a lot of time. The range from 435 to 485 seemed most promising and was further examined. The Microsoft COCO micro mean Average Precision evaluation metric is used to evaluate the performance at different tile center-line positions. This produced fluctuating results within the aforementioned range however it did not show a downward trend thus the examination was extended up to 570. The best results were found at 492 tile center-line position which seems to be at the top of the trend of figure 4.7 With the average of all small ground truth bounding boxes being at 490, 492 is quite a logical result because the tile method is supposed to improve the detection of small objects. Up till this point the tile approach showed very promising results. An increase from 36.36% to 70.73% on the Microsoft COCO mean Average Precision metric.

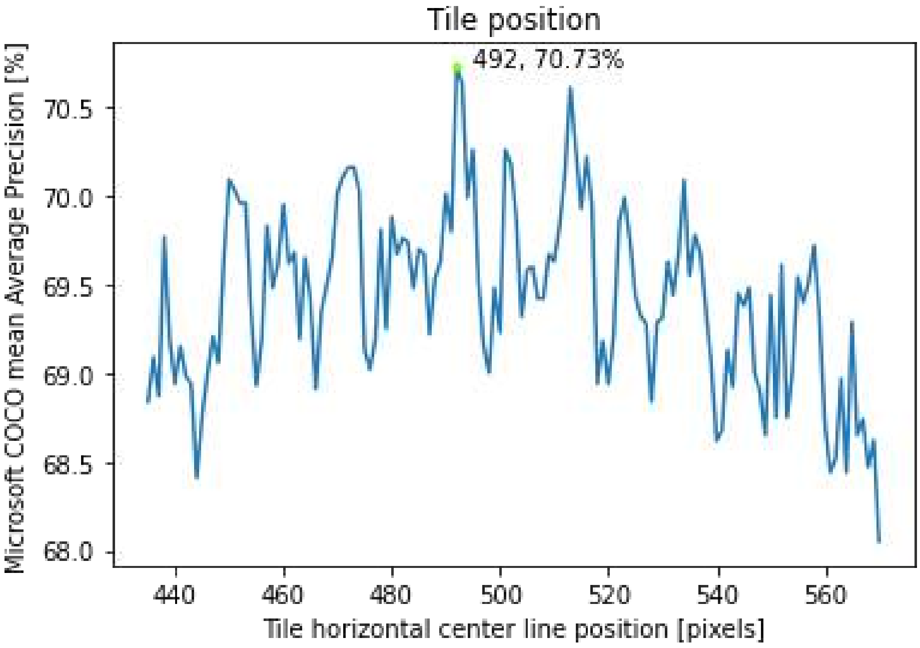


Figure 4.7

4.3. Hyperparameter optimization

Unlike model parameters that can be learned from data in this case the weights of the SSD MobileNet v1 model, hyperparameter are a choice made to fit a certain use case. The systems design choices like the batch size and the tiles are in fact hyperparameter as well however in this section the focus lies on optimizing the Non Maximum Suppression and the confidence score threshold for small traffic sign detection.

4.3.1. Non Maximum Suppression

The SSD MobileNet v1 model produces 100 prediction per image. With the three added tiles a total of 400 predictions per recorded image have to be reduced to only a few detections. With the Non Maximum Suppression a lot of multiple predictions can be discarded. Non-Maximum Suppression is a post-processing step to filter the proposed detections based on the following:

1. First the algorithm sorts all proposed bounding boxes based on their confidence scores from high to low.

2. The highest scoring bounding box is saved and compared with every other bounding box.

3. If the Intersection over Union is greater than the NMS threshold, the selected bounding box is removed/suppressed.

4. This loop is repeated until every proposed bounding box is either saved or removed.

An NMS threshold close to one results in almost all predictions to get through. This is because the predictions have to be almost a perfect match, in other words the predictions must have an Intersection over Union close to one for the predictions with a lower confidence score to be discarded. An NMS close to zero is thus a hard threshold. A small overlap between predictions results in the lower confidence scoring prediction to be discarded.

In figure 4.8 the Microsoft COCO micro mAP increases with increasing NMS threshold (IoU) value. This is not an expected result because of the increase in false positives which should have a negative impact on the Microsoft COCO micro mAP. There are two explanations for this behaviour though. The first reason is due to the low confidence score of the false positives. The lower the confidence score of a false positive the less impact it will have on the micro Microsoft COCO mAP. The Mean Average Precision is very sensitive to true positives and in this case substantially less to false positives. Secondly, because the predicted bounding boxes with the highest confidence score have not necessarily the highest IoU/overlap with the ground truth bounding boxes figure 4.104.11. That means that at higher NMS threshold values these predictions satisfy the NMS threshold as well as the COCO IoU threshold whenever the higher confidence scoring predictions do not satisfy the last mentioned threshold. This can be observed at Microsoft COCO IoU 0,70, 0,75, 0,80 and 0,85 values when setting the mAP against the Microsoft COCO IoU for different NMS thresholds, figure 4.9. Here the NMS threshold 0.85 has a higher average precision than the other two which seem almost identical. The mean of a line results in the Microsoft COCO mean Average Precision which of course corresponds to a point on the line in figure 4.8. The increase in Microsoft COCO mAP with increasing NMS threshold is a flaw in the system because it wrongfully suppresses better predictions. This is especially amplified by the tile method due to the increase in predictions.

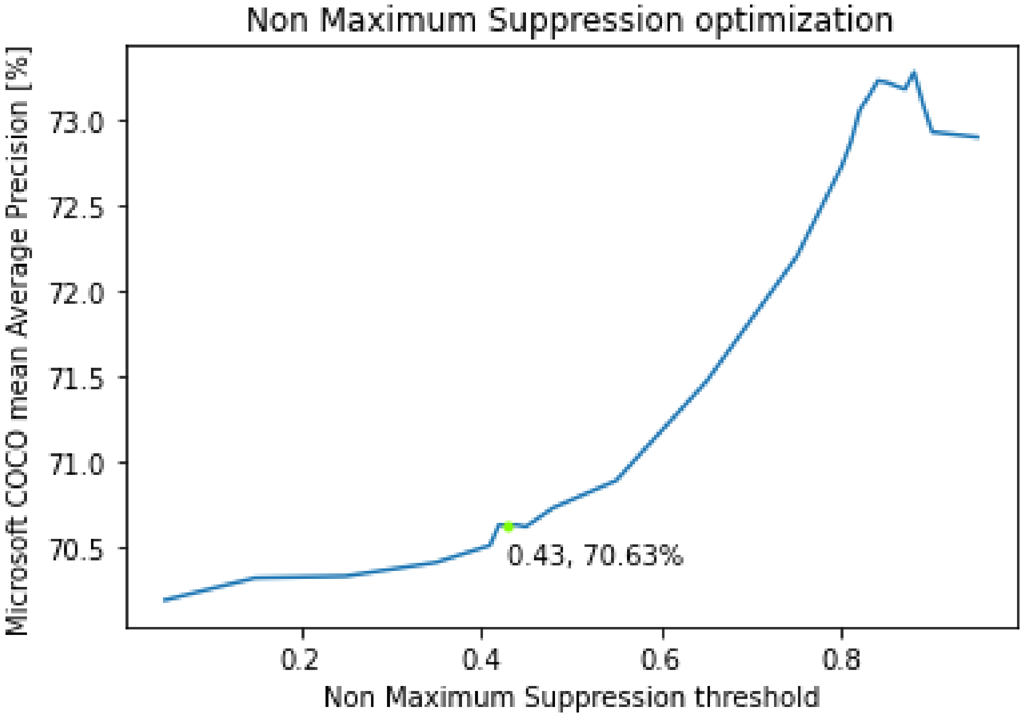


Figure 4.8

Unfortunately in order to optimize the NMS threshold of this system, it is not possible to solely rely on the Microsoft COCO micro mAP. In this case the amount of true positives and the amount of false negatives will be taken into consideration in optimizing the system.

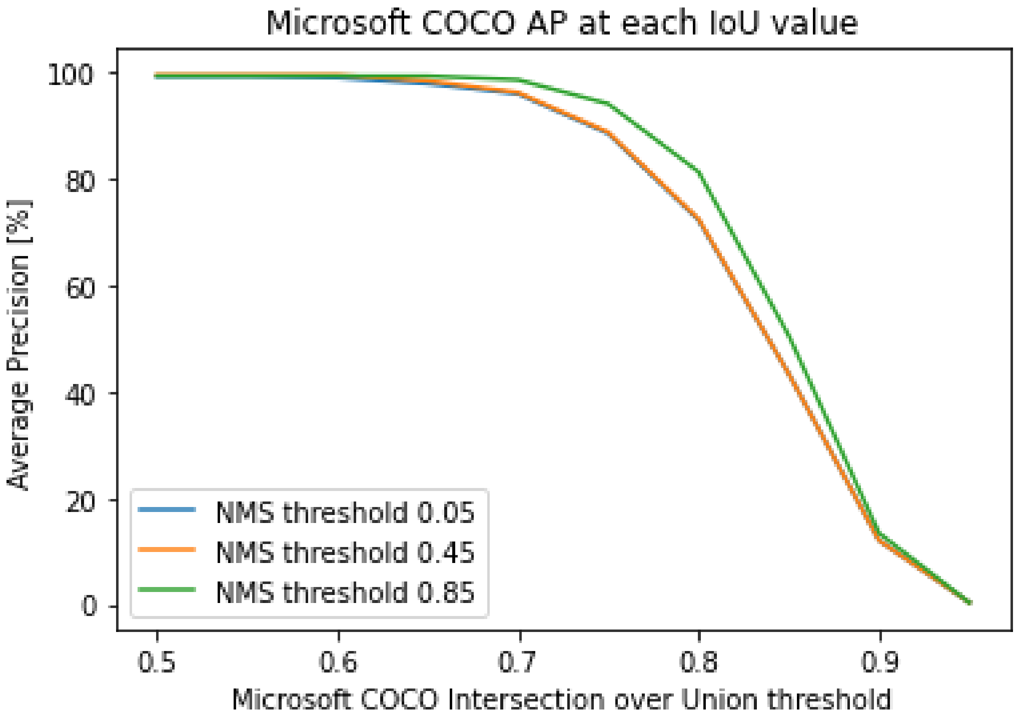


Figure 4.9

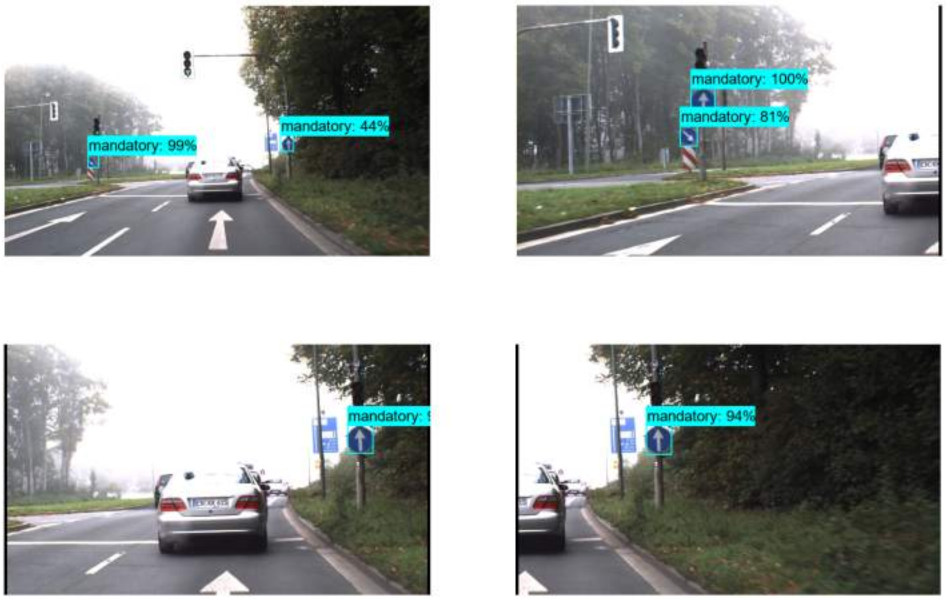


Figure 4.10: Proposed detections from the batch

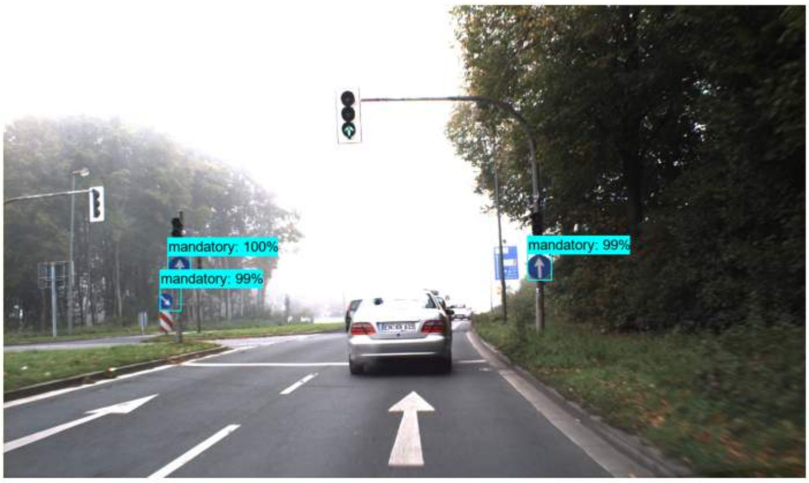


Figure 4.11: End result of a high confidence scoring bad bounding box, bottom left bounding box.

4.3.2. Confidence score

With the confidence score up to a certain degree the same behavior can be observed as the NMS threshold. At low confidence threshold values more prediction are accepted where some of them have incorrectly a lower confidence score. Luckily the difference is negligible and the hardest threshold at just slightly less than the best micro Microsoft COCO mAP performance is chosen. At this point the highest accuracy with the lowest amount of false positives is achieved, figure 4.12.

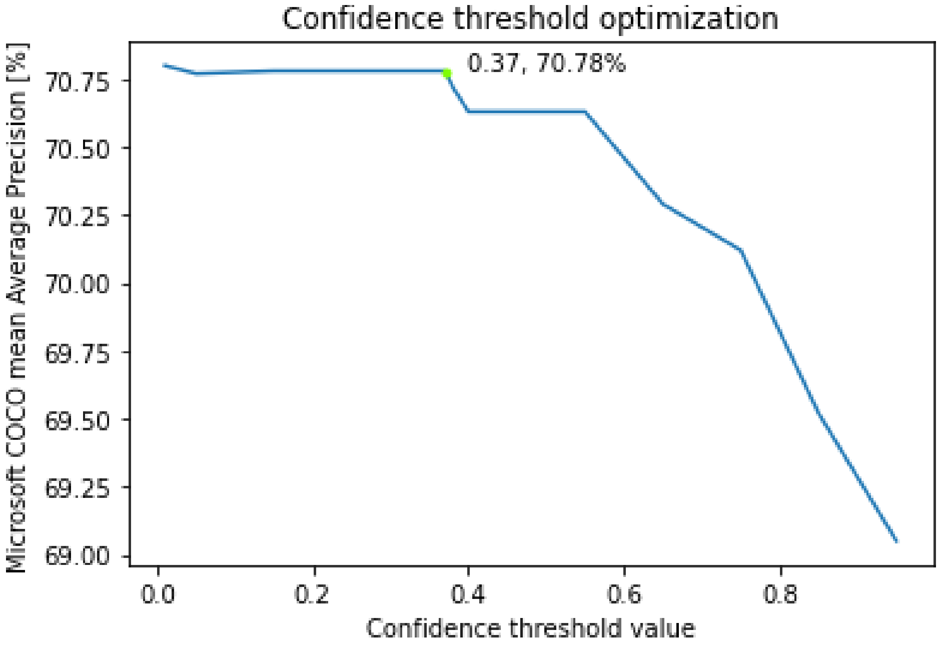


Figure 4.12

4.4. Batch of Bordered Tiles

An unwanted side effect of the tile method is the increase in false positive. To get a better understanding of why these extra false positives occur the bounding boxes are plotted alongside the borders of the tiles in figure 4.13. A considerable amount of bounding boxes are at the borders of the tiles. From there the idea came forth to add borders to the tiles at regions where the tiles overlap. In figure 4.14 the batch with bordered tiles is presented. On the top left is the resized input image, top right is the right tile with on the left side a barely visible black border of five pixels, bottom right is the centre tile with a border on the left and the right side and the bottom left the left tile with a border on the right side can be found.

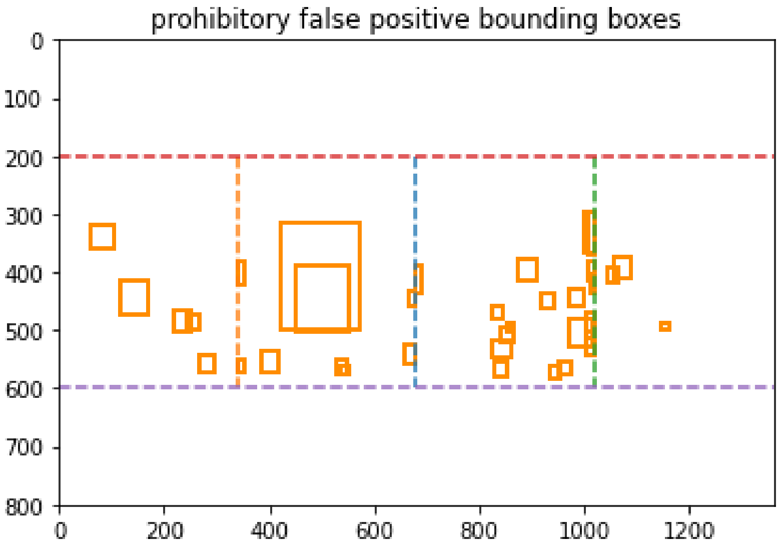


Figure 4.13

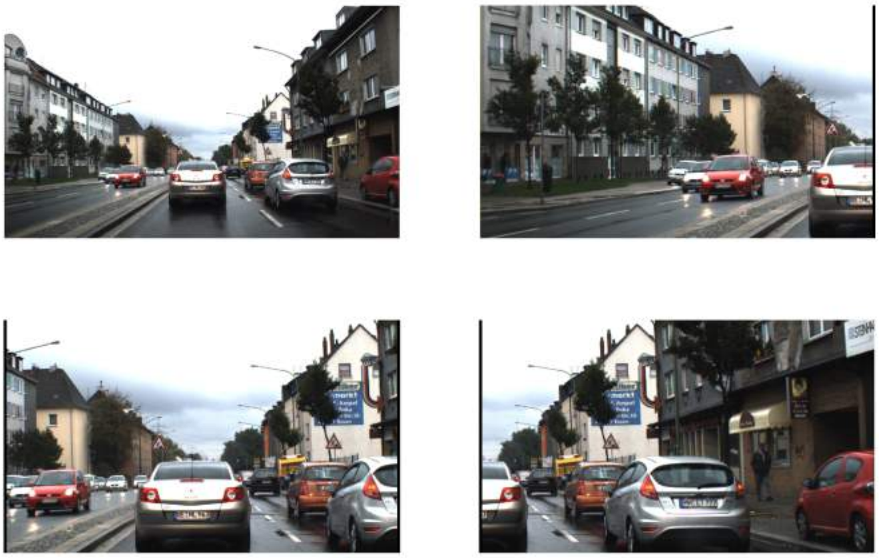


Figure 4.14: The batch with bordered tiles

By adding the borders to the tiles the Microsoft COCO micro mAP value slightly increases and the amount of false positives decreases without a noticeable influence on the inference speed. Border thickness of 0, 1, 2, 3, and 5 pixels are tested and the results are presented in table 4.15. With this final adjustment the Batch of Bordered Tiles method is completed. An overview of the steps that are taken is shown in figure 4.16.

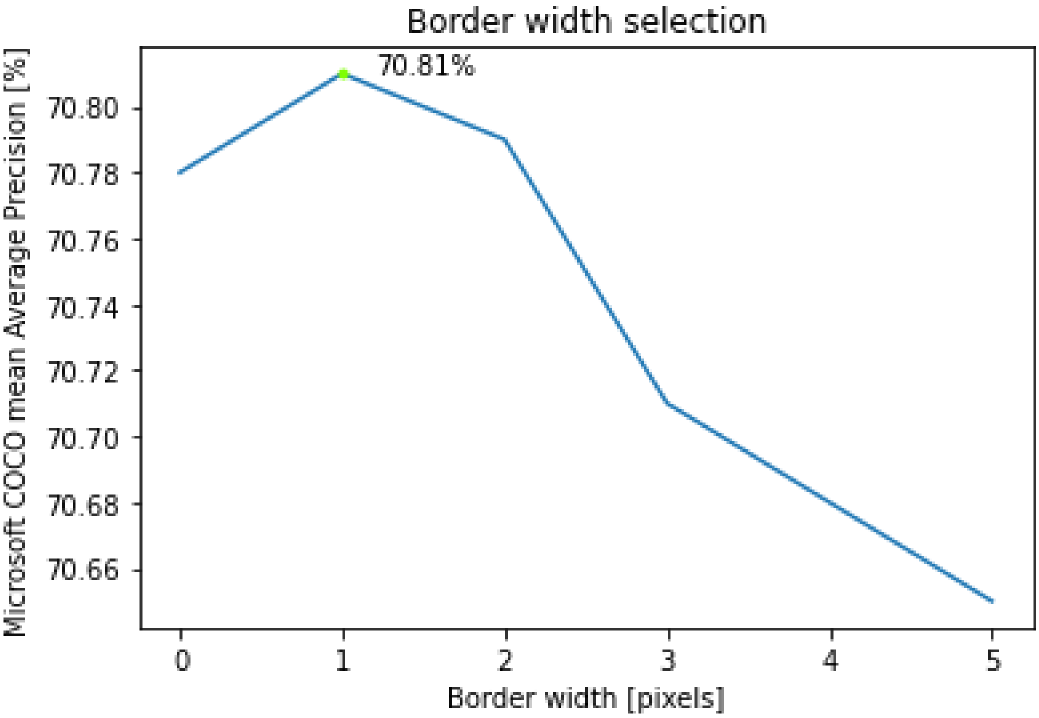


Figure 4.15

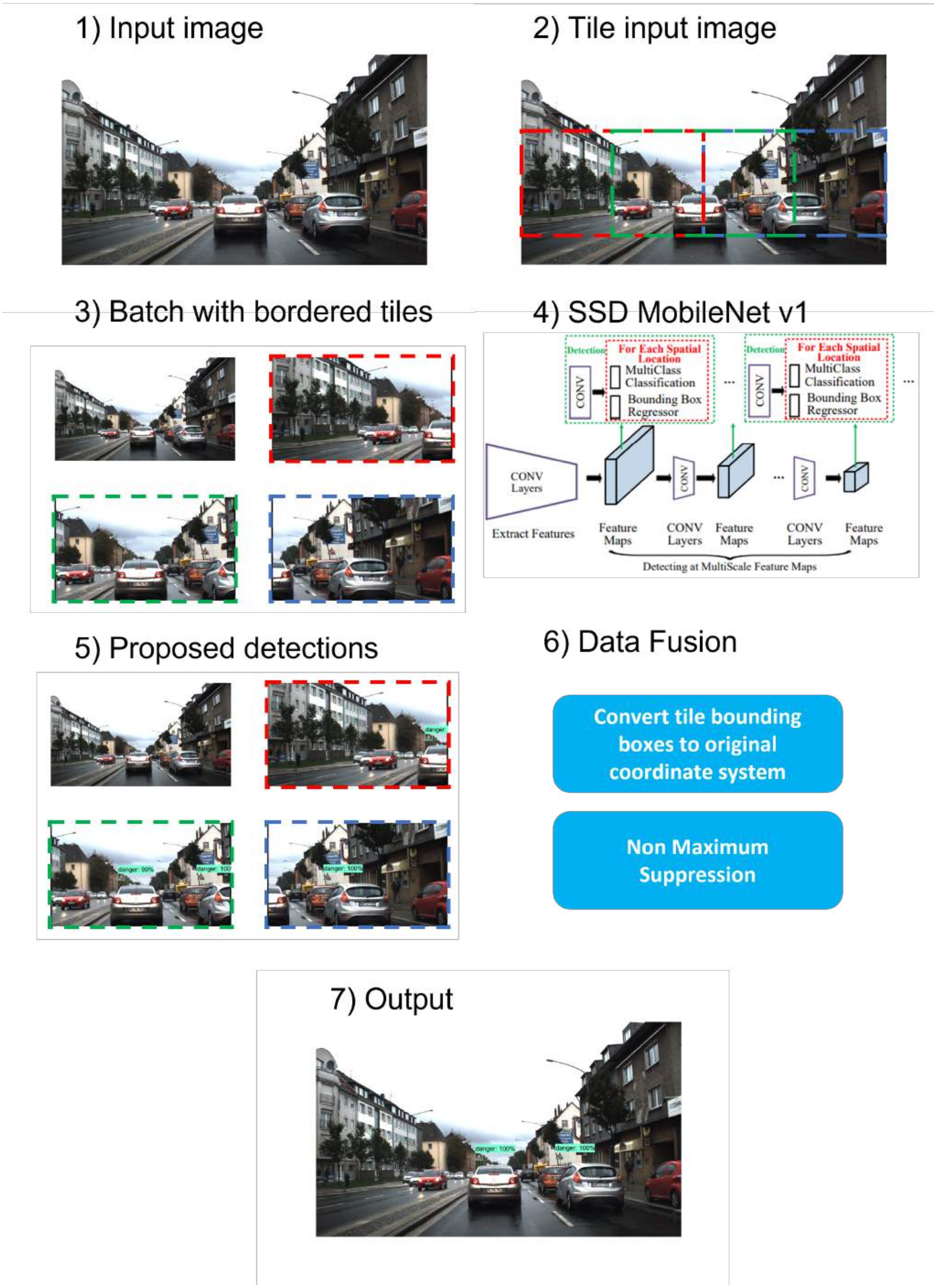


Figure 4.16: Batch of Bordered Tiles approach

4.5. Performance

The Batch of Bordered Tiles delivers a substantial increase in performance on the SSD MobileNet v1 model according to the Microsoft COCO evaluation metric. To visualize this performance increase, a comparison is made with the models that stood out for their performance according to arcos et al., in figure 4.17. The BoBT SSD MobileNet v1 clearly had to give in performance speed in order to achieve an increase in accuracy which was to be expected. However, the system still surpasses the R-FCN Resnet 101 in terms of inference time and has made a huge leap in accuracy being only 9.09 percent off the R-FCN Resnet 101 micro Microsoft COCO mean Average Precision. It is worth mentioning again that the R-FCN Resnet 101 model is praised by Arcos et al. as the best model in terms of trade-off between accuracy and inference time. In terms of true positives and false positives it is right there with the best model, Faster R-CNN Inception Resnet V2.This can be confirmed when applying the VOC Pascal mean Average Precision metric which is less sensitive to the quality of the detections, figure 4.18.

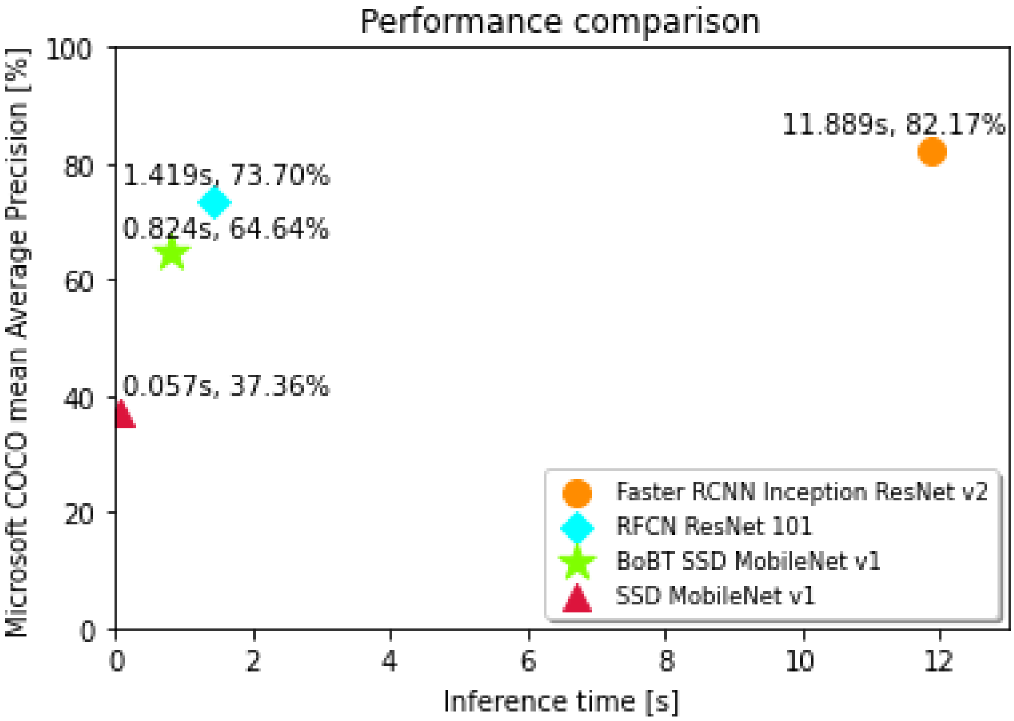


Figure 4.17

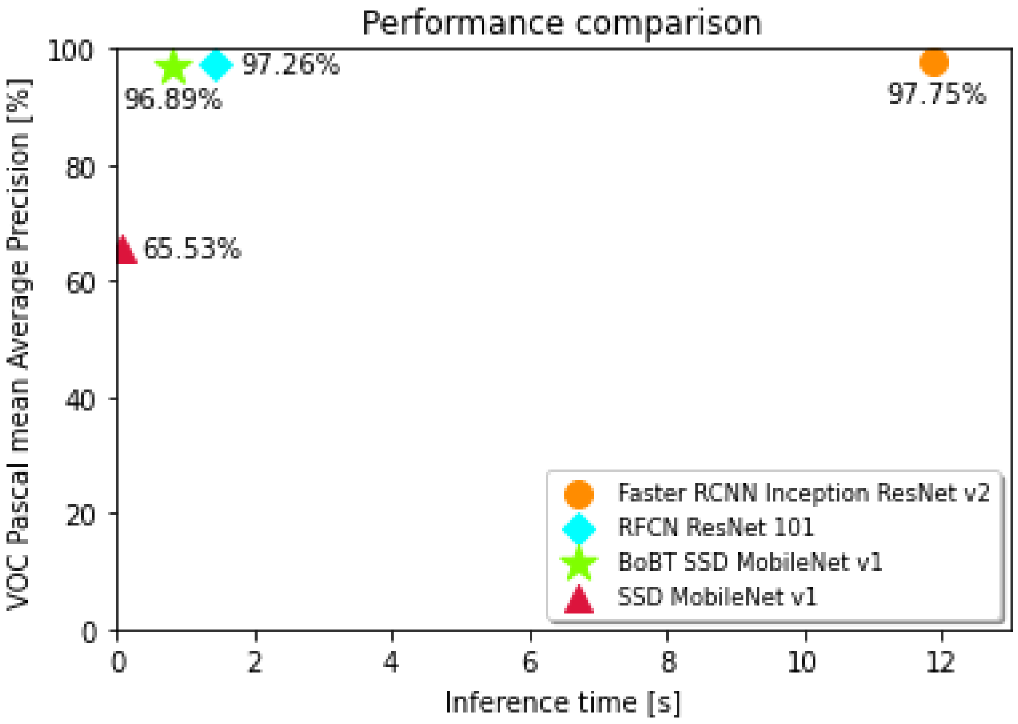

Performance comparison
VOC Pascal mean Average Precision [%]
0
20
40
60
80
100
Inference time [s]
0
2
4
6
8
10
12
96.89%
97.26%
97.75%
65.53%
Faster RCNN Inception ResNet v2
RFCN ResNet 101
BoBT SSD MobileNet v1
SSD MobileNet v1

Figure 4.18

5 Conclusions

Traffic sign detection is gaining more and more importance with the introduction of level two autonomous vehicles like the Tesla Model S. It started of with trying to classify traffic signs with hand crafted systems and has evolved into detecting traffic signs through neural networks in the past decade. Eventually the traffic sign detection will only be a part of the Artificial Intelligence of the autonomous vehicles that are bound to come sooner or later. In order to get to that point the small traffic sign detection problem is examined throughout this research. This problem is especially present in fast lightweight models like the SSD MobileNet v1.

In the object detection domain the tiling method is used as a solution for small object detection. This is however not yet applied to traffic sign detection. To tackle the small traffic sign detection problem a new tile method was introduced, the Batch of Bordered Tiles. By using tiles of the recorded image with half the dimensions, a two time digital zoom is essentially created. Unlike in other use cases it is not necessary to cover the whole recorded image with tiles as long as the small traffic signs that appear near the horizon line of the image are covered. Therefore three overlapping tiles will cover the entire horizon line. For the German Traffic Sign Detection Benchmark dataset a position at 492 of the horizontal center line of the tiles yields the optimal performance. Using less tiles results in a higher inference time.

To keep the increase in inference time to a minimum by utilizing the computational resources more efficiently a batch is used as an input. The batch is made of the three tiles and the recorded image resized to the dimensions of the tiles.

During the optimization of the Non Maximum Suppression a flaw in the tiling method in combination with the Microsoft COCO evaluation metric arises. Aside from the Microsoft COCO evaluation metric to be almost insensitive to false negatives with a relatively low confidence score, it also incorrectly indicates a performance improvement caused by duplicates of detections which during inference are false negatives. On the other side this indicates that the model incorrectly assigns higher confidence score to less precise bounding boxes. Due to this the true positives and false positives were taken into consideration to find the optimal NMS threshold at 0.43. The optimal confidence score threshold is found at 0.37.

Much like the optimization of the hyperparameters, the added borders on the overlapping sides of the tiles did not have a substantial impact on the performance. A reduction of the false positives at the borders and a slight increase in the micro Microsoft COCO mAP without any noticeable drawbacks is achieved. All this adjustments lead to the Batch of bordered tile method.

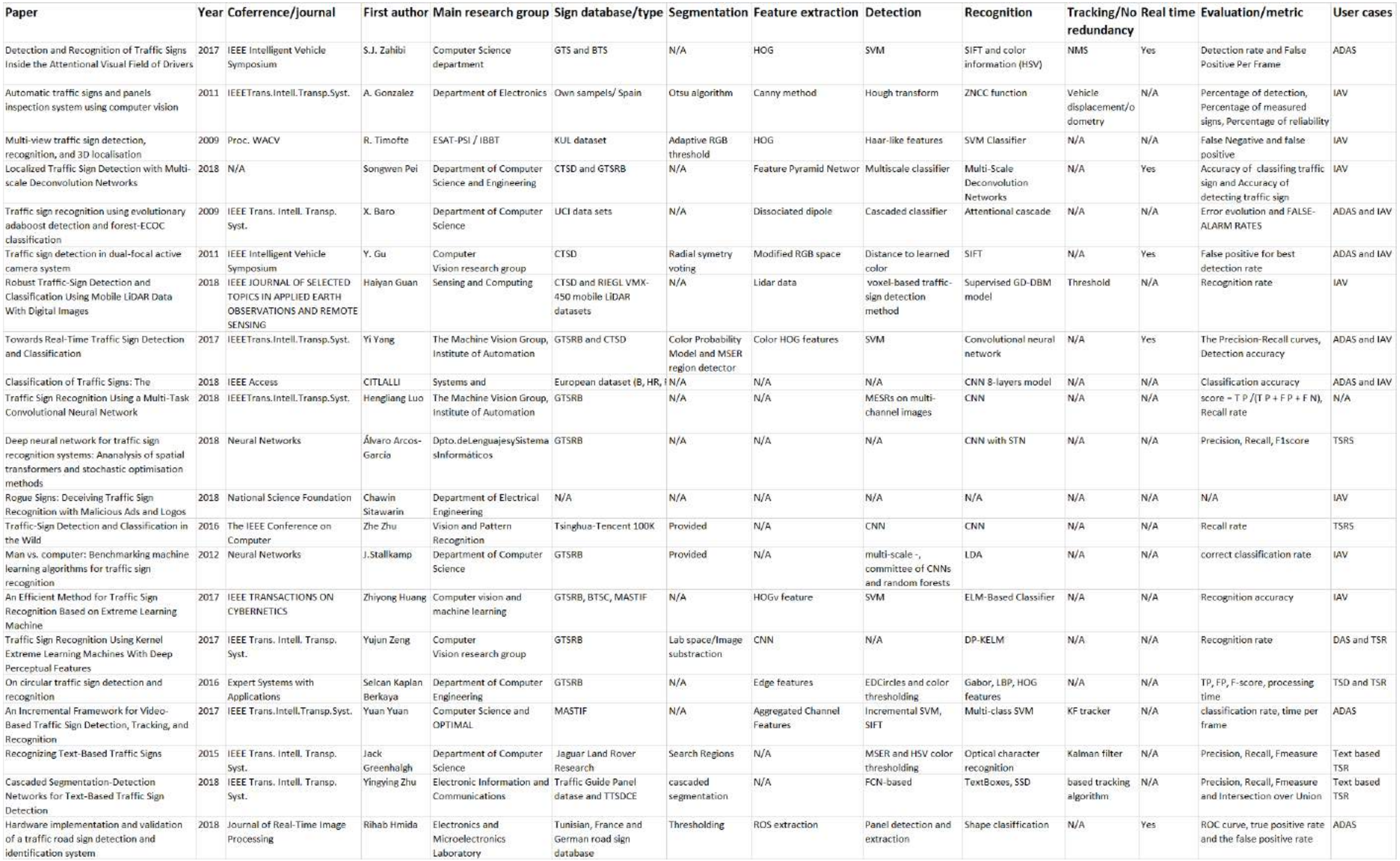
The BoBT method increases the accuracy of a model at the cost of inference speed. There is a small difference in precisely localizing the traffic signs compared to state of the art models but in terms of detecting traffic signs the difference is less than a percent with the highest accuracy model. Not to mention that it still outperforms these models in terms of speed. A Graphics processing unit can speeds up the inference time about five times. With it the real time objective is easily achieve realizing the goal of this research: To enhance a traffic sign detection model that requires relatively low processing power to compete with models that achieve high accuracy but require high processing power, maintaining the ability to operate in real time.

Currently new object detection models and even new versions of the aforementioned models are proposed. According to their respective papers these models deliver better performance, may that be in latency or accuracy. The increase in performance is not ground breaking though and does not undermine or affect the findings obtained from this research. The BoBT method can be applied to these new models as well.

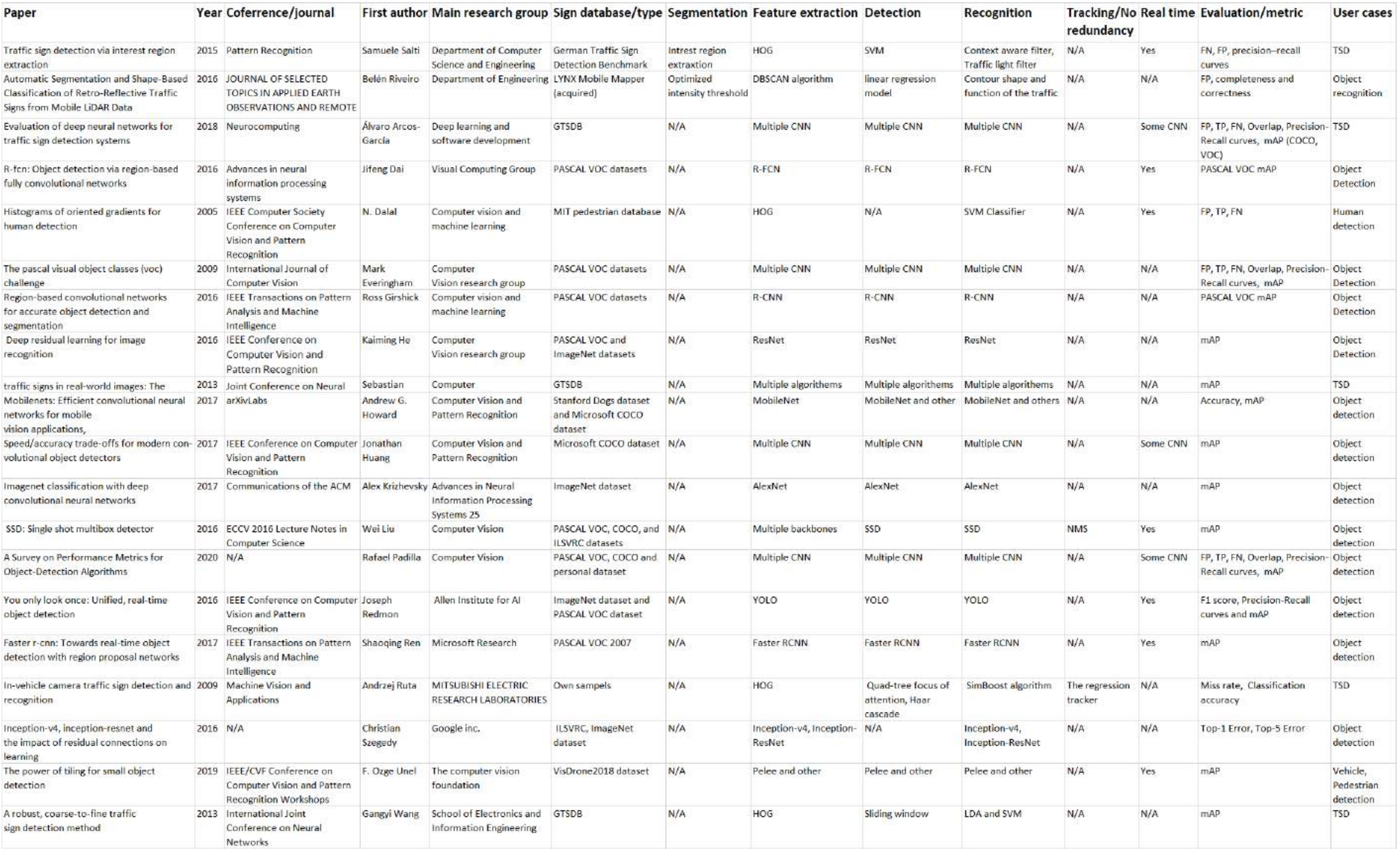
Despite the substantial increase in accuracy, it still appears that a lot of potential is being literally suppressed. This hidden potential is brought to light in chapter 4. The Non Maximum Suppression is not only a bottleneck in terms of speed but apparently also in terms of accuracy. For future work it would be interesting to uncover the true potential of the Batch of Bordered Tiles.

A

Literature list



Paper	Year	Coferrence/journal	First author	Main research group	Sign database/type	Segmentation	Feature extraction	Detection	Recognition	Tracking/No redundancy	Real time	Evaluation/metric	User cases
Detection and Recognition of Traffic Signs Inside the Attentional Visual Field of Drivers	2017	IEEE Intelligent Vehicle Symposium	S.J. Zahibi	Computer Science department	GTS and BTS	N/A	HOG	SVM	SIFT and color information (HSV)	NMS	Yes	Detection rate and False Positive Per Frame	ADAS
Automatic traffic signs and panels inspection system using computer vision	2011	IEEETrans.Intell.Transp.Syst.	A. Gonzalez	Department of Electronics	Own sampels/ Spain	Otsu algorithm	Canny method	Hough transform	ZNCC function	Vehicle displacement/odometry	N/A	Percentage of detection, Percentage of measured signs, Percentage of reliability	IAV
Multi-view traffic sign detection, recognition, and 3D localisation	2009	Proc. WACV	R. Timofte	ESAT-PSI / IBBT	KUL dataset	Adaptive RGB threshold	HOG	Haar-like features	SVM Classifier	N/A	N/A	False Negative and false positive	IAV
Localized Traffic Sign Detection with Multi-scale Deconvolution Networks	2018	N/A	Songwen Pei	Department of Computer Science and Engineering	CTSD and GTSRB	N/A	Feature Pyramid Networ	Multiscale classifier	Multi-Scale Deconvolution Networks	N/A	Yes	Accuracy of classifing traffic sign and Accuracy of detecting traffic sign	IAV
Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification	2009	IEEE Trans. Intell. Transp. Syst.	X. Baro	Department of Computer Science	UCI data sets	N/A	Dissociated dipole	Cascaded classifier	Attentional cascade	N/A	N/A	Error evolution and FALSE-ALARM RATES	ADAS and IAV
Traffic sign detection in dual-focal active camera system	2011	IEEE Intelligent Vehicle Symposium	Y. Gu	Computer Vision research group	CTSD	Radial symetry voting	Modified RGB space	Distance to learned color	SIFT	N/A	Yes	False positive for best detection rate	ADAS and IAV
Robust Traffic-Sign Detection and Classification Using Mobile LiDAR Data With Digital Images	2018	IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING	Haiyan Guan	Sensing and Computing	CTSD and RIEGL VMX-450 mobile LiDAR datasets	N/A	Lidar data	voxel-based traffic-sign detection method	Supervised GD-DBM model	Threshold	N/A	Recognition rate	IAV
Towards Real-Time Traffic Sign Detection and Classification	2017	IEEETrans.Intell.Transp.Syst.	Yi Yang	The Machine Vision Group, Institute of Automation	GTSRB and CTSD	Color Probability Model and MSER region detector	Color HOG features	SVM	Convolutional neural network	N/A	Yes	The Precision-Recall curves, Detection accuracy	ADAS and IAV
Classification of Traffic Signs: The	2018	IEEE Access	CITLALLI	Systems and	European dataset (B, HR, I	N/A	N/A	N/A	CNN 8-layers model	N/A	N/A	Classification accuracy	ADAS and IAV
Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network	2018	IEEETrans.Intell.Transp.Syst.	Hengliang Luo	The Machine Vision Group, Institute of Automation	GTSRB	N/A	N/A	MESRs on multi-channel images	CNN	N/A	N/A	score = T P /(T P + F P + F N), Recall rate	N/A
Deep neural network for traffic sign recognition systems: Ananlysis of spatial transformers and stochastic optimisation methods	2018	Neural Networks	Álvaro Arcos-Garcia	Dpto.deLenguajesySistemasInformáticos	GTSRB	N/A	N/A	N/A	CNN with STN	N/A	N/A	Precision, Recall, F1score	TSRS
Rogue Signs: Deceiving Traffic Sign Recognition with Malicious Ads and Logos	2018	National Science Foundation	Chawin Sitawarin	Department of Electrical Engineering	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	IAV
Traffic-Sign Detection and Classification in the Wild	2016	The IEEE Conference on Computer	Zhe Zhu	Vision and Pattern Recognition	Tsinghua-Tencent 100K	Provided	N/A	CNN	CNN	N/A	N/A	Recall rate	TSRS
Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition	2012	Neural Networks	J.Stallkamp	Department of Computer Science	GTSRB	Provided	N/A	multi-scale -, committee of CNNs and random forests	LDA	N/A	N/A	correct classification rate	IAV
An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine	2017	IEEE TRANSACTIONS ON CYBERNETICS	Zhiyong Huang	Computer vision and machine learning	GTSRB, BTSC, MASTIF	N/A	HOGv feature	SVM	ELM-Based Classifier	N/A	N/A	Recognition accuracy	IAV
Traffic Sign Recognition Using Kernel Extreme Learning Machines With Deep Perceptual Features	2017	IEEE Trans. Intell. Transp. Syst.	Yujun Zeng	Computer Vision research group	GTSRB	Lab space/Image substraction	CNN	N/A	DP-KELM	N/A	N/A	Recognition rate	DAS and TSR
On circular traffic sign detection and recognition	2016	Expert Systems with Applications	Selcan Kaplan Berkaya	Department of Computer Engineering	GTSRB	N/A	Edge features	EDCircles and color thresholding	Gabor, LBP, HOG features	N/A	N/A	TP, FP, F-score, processing time	TSD and TSR
An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition	2017	IEEE Trans.Intell.Transp.Syst.	Yuan Yuan	Computer Science and OPTIMAL	MASTIF	N/A	Aggregated Channel Features	Incremental SVM, SIFT	Multi-class SVM	KF tracker	N/A	classification rate, time per frame	ADAS
Recognizing Text-Based Traffic Signs	2015	IEEE Trans. Intell. Transp. Syst.	Jack Greenhalgh	Department of Computer Science	Jaguar Land Rover Research	Search Regions	N/A	MSER and HSV color thresholding	Optical character recognition	Kalman filter	N/A	Precision, Recall, Fmeasure	Text based TSR
Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection	2018	IEEE Trans. Intell. Transp. Syst.	Yingying Zhu	Electronic Information and Communications	Traffic Guide Panel datase and TTSDCE	cascaded segmentation	N/A	FCN-based	TextBoxes, SSD	based tracking algorithm	N/A	Precision, Recall, Fmeasure and Intersection over Union	Text based TSR
Hardware implementation and validation of a traffic road sign detection and identification system	2018	Journal of Real-Time Image Processing	Rihab Hmida	Electronics and Microelectronics Laboratory	Tunisian, France and German road sign database	Thresholding	ROS extraction	Panel detection and extraction	Shape clasiffication	N/A	Yes	ROC curve, true positive rate and the false positive rate	ADAS



Paper	Year	Coferrence/journal	First author	Main research group	Sign database/type	Segmentation	Feature extraction	Detection	Recognition	Tracking/No redundancy	Real time	Evaluation/metric	User cases
Traffic sign detection via interest region extraction	2015	Pattern Recognition	Samuele Salti	Department of Computer Science and Engineering	German Traffic Sign Detection Benchmark	Intrest region extraxtion	HOG	SVM	Context aware filter, Traffic light filter	N/A	Yes	FN, FP, precision–recall curves	TSD
Automatic Segmentation and Shape-Based Classification of Retro-Reflective Traffic Signs from Mobile LiDAR Data	2016	JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE	Belén Riveiro	Department of Engineering	LYNX Mobile Mapper (acquired)	Optimized intensity threshold	DBSCAN algorithm	linear regression model	Contour shape and function of the traffic	N/A	N/A	FP, completeness and correctness	Object recognition
Evaluation of deep neural networks for traffic sign detection systems	2018	Neurocomputing	Álvaro Arcos-García	Deep learning and software development	GTSDB	N/A	Multiple CNN	Multiple CNN	Multiple CNN	N/A	Some CNN	FP, TP, FN, Overlap, Precision-Recall curves, mAP (COCO, VOC)	TSD
R-fcn: Object detection via region-based fully convolutional networks	2016	Advances in neural information processing systems	Jifeng Dai	Visual Computing Group	PASCAL VOC datasets	N/A	R-FCN	R-FCN	R-FCN	N/A	Yes	PASCAL VOC mAP	Object Detection
Histograms of oriented gradients for human detection	2005	IEEE Computer Society Conference on Computer Vision and Pattern Recognition	N. Dalal	Computer vision and machine learning	MIT pedestrian database	N/A	HOG	N/A	SVM Classifier	N/A	Yes	FP, TP, FN	Human detection
The pascal visual object classes (voc) challenge	2009	International Journal of Computer Vision	Mark Everingham	Computer Vision research group	PASCAL VOC datasets	N/A	Multiple CNN	Multiple CNN	Multiple CNN	N/A	N/A	FP, TP, FN, Overlap, Precision-Recall curves, mAP	Object Detection
Region-based convolutional networks for accurate object detection and segmentation	2016	IEEE Transactions on Pattern Analysis and Machine Intelligence	Ross Girshick	Computer vision and machine learning	PASCAL VOC datasets	N/A	R-CNN	R-CNN	R-CNN	N/A	N/A	PASCAL VOC mAP	Object Detection
Deep residual learning for image recognition	2016	IEEE Conference on Computer Vision and Pattern Recognition	Kaiming He	Computer Vision research group	PASCAL VOC and ImageNet datasets	N/A	ResNet	ResNet	ResNet	N/A	N/A	mAP	Object Detection
traffic signs in real-world images: The	2013	Joint Conference on Neural	Sebastian	Computer	GTSDB	N/A	Multiple algorithems	Multiple algorithems	Multiple algorithems	N/A	N/A	mAP	TSD
Mobilenets: Efficient convolutional neural networks for mobile vision applications,	2017	arXivLabs	Andrew G. Howard	Computer Vision and Pattern Recognition	Stanford Dogs dataset and Microsoft COCO dataset	N/A	MobileNet	MobileNet and other	MobileNet and others	N/A	N/A	Accuracy, mAP	Object detection
Speed/accuracy trade-offs for modern convolutional object detectors	2017	IEEE Conference on Computer Vision and Pattern Recognition	Jonathan Huang	Computer Vision and Pattern Recognition	Microsoft COCO dataset	N/A	Multiple CNN	Multiple CNN	Multiple CNN	N/A	Some CNN	mAP	Object detection
Imagenet classification with deep convolutional neural networks	2017	Communications of the ACM	Alex Krizhevsky	Advances in Neural Information Processing Systems 25	ImageNet dataset	N/A	AlexNet	AlexNet	AlexNet	N/A	N/A	mAP	Object detection
SSD: Single shot multibox detector	2016	ECCV 2016 Lecture Notes in Computer Science	Wei Liu	Computer Vision	PASCAL VOC, COCO, and ILSVRC datasets	N/A	Multiple backbones	SSD	SSD	NMS	Yes	mAP	Object detection
A Survey on Performance Metrics for Object-Detection Algorithms	2020	N/A	Rafael Padilla	Computer Vision	PASCAL VOC, COCO and personal dataset	N/A	Multiple CNN	Multiple CNN	Multiple CNN	N/A	Some CNN	FP, TP, FN, Overlap, Precision-Recall curves, mAP	Object detection
You only look once: Unified, real-time object detection	2016	IEEE Conference on Computer Vision and Pattern Recognition	Joseph Redmon	Allen Institute for AI	ImageNet dataset and PASCAL VOC dataset	N/A	YOLO	YOLO	YOLO	N/A	Yes	F1 score, Precision-Recall curves and mAP	Object detection
Faster r-cnn: Towards real-time object detection with region proposal networks	2017	IEEE Transactions on Pattern Analysis and Machine Intelligence	Shaoqing Ren	Microsoft Research	PASCAL VOC 2007	N/A	Faster RCNN	Faster RCNN	Faster RCNN	N/A	Yes	mAP	Object detection
In-vehicle camera traffic sign detection and recognition	2009	Machine Vision and Applications	Andrzej Ruta	MITSUBISHI ELECTRIC RESEARCH LABORATORIES	Own sampels	N/A	HOG	Quad-tree focus of attention, Haar cascade	SimBoost algorithm	The regression tracker	N/A	Miss rate, Classification accuracy	TSD
Inception-v4, inception-resnet and the impact of residual connections on learning	2016	N/A	Christian Szegedy	Google inc.	ILSVRC, ImageNet dataset	N/A	Inception-v4, Inception-ResNet	N/A	Inception-v4, Inception-ResNet	N/A	N/A	Top-1 Error, Top-5 Error	Object detection
The power of tiling for small object detection	2019	IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops	F. Ozge Unel	The computer vision foundation	VisDrone2018 dataset	N/A	Pelee and other	Pelee and other	Pelee and other	N/A	Yes	mAP	Vehicle, Pedestrian detection
A robust, coarse-to-fine traffic sign detection method	2013	International Joint Conference on Neural Networks	Gangyi Wang	School of Electronics and Information Engineering	GTSDB	N/A	HOG	Sliding window	LDA and SVM	N/A	N/A	mAP	TSD

Bibliography

[1] *Global status report on road safety 2018*. World Health Organization, 2018.

[2] M Abadi. Large-scale machine learning for species distributions. *Large-Scale Machine Learning in the Earth Sciences*, page 73–94, 2017. doi: 10.1201/9781315371740-6.

[3] Álvaro Arcos-García, Juan A. Álvarez García, and Luis M. Soria-Morillo. Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing*, 316:332–344, 2018. doi: 10.1016/j.neucom.2018.08.009.

[4] S Bozinovski and A Fulgosi. The influence of pattern similarity and transfer learning upon training of a base perceptron b2. In *Proceedings of Symposium Informatica*, pages 3–121, 1976.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/bf00994018.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*, 2005. doi: 10.1109/cvpr.2005.177.

[8] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. doi: 10.1145/361237.361242.

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2009. doi: 10.1007/s11263-009-0275-4.

[10] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2): 179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

[11] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016. doi: 10.1109/tpami.2015.2437384.

[13] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910. doi: 10.1007/bf01456326.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/cvpr.2016.90.

[15] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013. doi: 10.1109/ijcnn.2013.6706807.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and et al. Speed/accuracy trade-offs for modern convolutional object detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/cvpr.2017.351.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386.

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. doi: 10.1007/s11263-020-01316-z.

[20] F. Larsson, M. Felsberg, and Forssen P.-E. Correlating fourier descriptors of local patches for road sign recognition. *IET Computer Vision*, 5(4):244, 2011. doi: 10.1049/iet-cvi.2010.0040.

[21] Fredrik Larsson and Michael Felsberg. Using fourier descriptors and spatial models for traffic sign recognition. *Image Analysis Lecture Notes in Computer Science*, page 238–249, 2011. doi: 10.1007/978-3-642-21227-7_23.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Computer Vision – ECCV 2014 Lecture Notes in Computer Science*, page 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

[23] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128 (2):261–318, 2019. doi: 10.1007/s11263-019-01247-4.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, page 21–37, 2016. doi: 10.1007/978-3-319-46448-0_2.

[25] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012. doi: 10.1109/tits.2012.2209421.

[26] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. Da Silva. A survey on performance metrics for object-detection algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020. doi: 10.1109/iwssip48289.2020.9145130.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/cvpr.2016.91.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1137–1149, 2017. doi: 10.1109/tpami.2016.2577031.

[29] Andrzej Ruta, Fatih Porikli, Shintaro Watanabe, and Yongmin Li. In-vehicle camera traffic sign detection and recognition. *Machine Vision and Applications*, 22(2):359–375, 2009. doi: 10.1007/s00138-009-0231-x.

[30] Koen E. A. Van De Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. *2011 International Conference on Computer Vision*, 2011. doi: 10.1109/iccv.2011.6126456.

[31] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. doi: 10.1016/j.neunet.2012.02.016.

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, and et al. Scalability in perception for autonomous driving: Waymo open dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. doi: 10.1109/cvpr42600.2020.00252.

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[34] Dogancan Temel, Min-Hung Chen, and Ghassan Alregib. Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3663–3673, 2020. doi: 10.1109/tits.2019.2931429.

[35] Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. *2009 Workshop on Applications of Computer Vision (WACV)*, 2009. doi: 10.1109/wacv.2009.5403121.

[36] F. Ozge Unel, Burak O. Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. doi: 10.1109/cvprw.2019.00084.

[37] P. Viola and M. Jones. Robust real-time face detection. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV*, 2001. doi: 10.1109/iccv.2001.937709.

[38] Gangyi Wang, Guanghui Ren, Zhilu Wu, Yaqin Zhao, and Lihui Jiang. A robust, coarse-to-fine traffic sign detection method. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013. doi: 10.1109/ijcnn.2013.6706812.

[39] Robert J. Wang, Xiang Li, and Charles X. Ling. Pelee: A real-time object detection system on mobile devices, 2019.

[40] Ali Youssef, Dario Albani, Daniele Nardi, and Domenico Daniele Bloisi. Fast traffic sign recognition using color segmentation and deep convolutional networks. *Advanced Concepts for Intelligent Vision Systems Lecture Notes in Computer Science*, page 205–216, 2016. doi: 10.1007/978-3-319-48680-2_19.

[41] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/cvpr.2016.232.

[42] Siniša Šegvić, Karla Brkić, Zoran Kalafatić, and Axel Pinz. Exploiting temporal and spatial constraints in traffic sign detection from a moving vehicle. *Machine Vision and Applications*, 25(3):649–665, 2011. doi: 10.1007/s00138-011-0396-y.