UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

DATASET ANONYMIZATION AND ROAD SIGNS DETECTION

BACHELOR FINAL THESIS

Document:
THESIS

Author:
ILIAS KHAYAT EL GHARBI FELLAH

Director /Co-director:
RAMON MORROS RUBIO

Degree:
Bachelor in Audiovisual Systems ENGINEERING

Examination session:
Spring

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Abstract

Micro mobility vehicles and renting services have seen an unprecedented spike due to the growth of population in urban areas. Simultaneously, automotive technology for autonomous driving has drastically improved and entered the global market. In this thesis we propose the testbed for a future assisted driving application. This prototype is based on an object detector using a Region Based Convolutional Neural Network trained to detect traffic road signs specific to micro mobility vehicles. In order to train this model, it's necessary to use a dataset that contains confidential data of many citizens, we also introduce a solution to manage this sensitive data under the General Data Protection Regulation using pre-trained models for face and number plate detection.

Table of contents

List of tables

List of figures

1. Introduction

1.1 Object

The main objective of this research work is to develop a proof of concept for an assisted driving software for micro mobility vehicles. This study takes part of a broader project seeking to improve the safety of the users as well as making them aware about the regulations that may be emplaced by the city, ultimately the goal is not to aim for autonomous driving but to serv as an addition to the driver experience.

In order to provide fundamental information such as maximum speed, wrong way driving or threatening areas involving pedestrians or cars to name a few, it's necessary to use a traffic sign detector powered by deep learning models. The dataset used to train this model poses the other main objective of the thesis. For the purpose of making this dataset public it must comply with the GDPR [1] regulation which implies that no sensitive information like faces or vehicles plates can be recognizable. This task will also be accomplished with deep learning but for this case, state of the art pre-trained models will be used.

1.2 Scope

- Detection and blurring of faces and license plates.
- Model training for traffic sign and road marking detection.

More aspects like signal tracking or optimization to run on mobile phones in real-time are necessary for the final implementation, but for the sake of this research, workload will be entirely focused in object detection.

1.3 Requirements

The main requirements to elaborate this project is to run the training and inference on Detectron 2 [2] and ultimately compile the script to anonymize the dataset on UPC CALCULA servers.

Detectron 2 is a Facebook AI team library that provides CUDA and PyTorch implementation of state-of-the-art models for detection tasks such as bounding-box detection, instance and semantic segmentation, and person key point detection.object de

The licensing system of the Detectron2 is one of its most significant features: the library is distributed under the Apache 2.0 license, while pre-trained models are released under the CC BY-SA 3.0 license. Enabling any change in the code and use for personal, scientific, or even commercial purposes by just giving proper credit to the team. It's unusual in the scientific community, which frequently employs licenses that require publishing the source code and non-commercial use. These conditions fit perfectly in the frame of this research

1.4 Rationale

The way we get around town is evolving. We are progressively choosing more environmentally friendly and practical personal vehicles that give us the freedom to travel whenever and wherever we want. Micro-mobility has emerged as an essential aspect to resolve transportation issues in large cities, as well as a key element for future urban management models while enabling for the creation of profitable initiatives from an economic standpoint

Since July 2017, personal mobility vehicles (PMVs) have had their own set of regulations [37] to ensure the safety and security of its passengers, as well as good coexistence with other pedestrians and cars. PMVs and cycles for personal use, as well as for commercial usage, whether motor or mechanical, are subject to these restrictions.

On November 2021, a royal decree was approved amending the General Traffic Regulations and the General Vehicle Regulations [3], in regard to urban traffic measures. The legislation governs, among other aspects, the technical requirements and conditions of PMVs, which have now been formally designated as vehicles and are thus prohibited from driving on sidewalks or pedestrian areas.

The Royal Decree's action establishes a state framework that classifies PMVs as vehicles and specifies the technical requirements for them to be able to drive, based on a technical characteristic manual, while also establishing the demand for a traffic certificate depending on the VMPs specifications such as maximum speed or weight. As this last stipulation is certainly more difficult to regulate and PMVs must be governed by the municipal traffic regulations of the city where they are located, which may change for foreigners, there’s a clear need for a software to facilitate all this crucial information to the drivers in order to prevent accidents and confusion.

2 Background and review of the state of the art

In this section, we will start examining with a broader perspective the state of the art of object detection in order to better understand more complex approaches for traffic sign and road sign detection.

2.1 Computer vision and Object Detection

Object detection is a computer vision technique for identifying and locating objects in images and videos. This technique is usually confused with image recognition, so before we proceed it's crucial to understand the differences between the two.

Image recognition assigns a label to images. In *figure 2.1* outputs the label ''pedestrian''. Contrarily, object detection, shows a box where each member and labels the box "pedestrian". The model provides more information than just recognition because it predicts the location and label for every object. This distinction is necessary as there are multiple papers entirely focused on signal recognition and not detection.

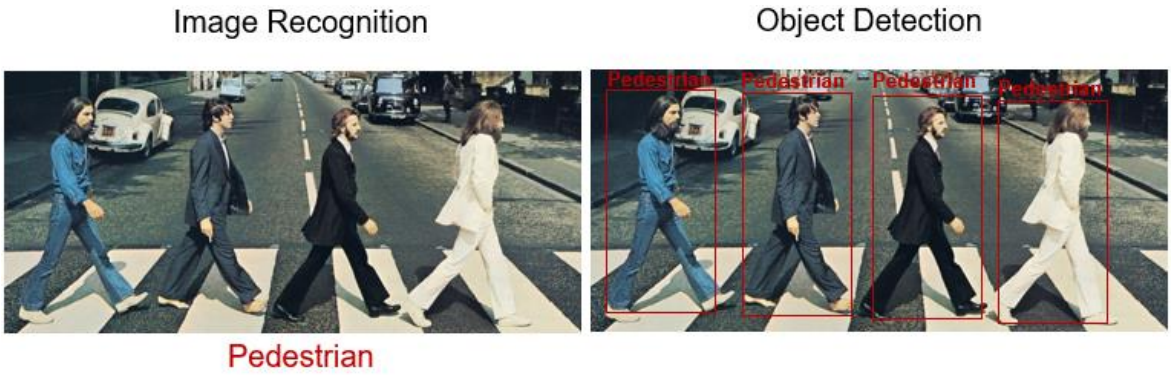


Figure 2.1 Difference between image recognition and object Detection

Object detection can be divided into two categories: machine learning-based approaches and deep learning-based approaches. Traditional machine learning systems use computer vision algorithms to recognize several aspects of an image, such as edges or the color histogram, in order to recognize groups of pixels that could be an object. These characteristics are then loaded into a regression model that predicts the object's position as well as its label.
In contrast deep learning-based techniques use convolutional neural networks (CNNs) [4] to do end-to-end, unsupervised object detection, which avoids the need to define and extract characteristics separately.
Deep learning methods have become the state-of-the-art approaches for object detection due to the outperforming results and reliability in comparison to their machine learning counterpart. For these stated reasons a Deep learning approach is elected for this project. In the next section the basic structure and various deep learning-based approaches will be assessed.

2.2 Basic structure and current models

Deep learning-based object detection models usually can be split in two parts. When an image is entered into an encoder or backbone, it is passed through a number of layers and blocks that are trained to extract statistical features. The decoder receives the encoder's outputs and guesses the bounding boxes and labels for each item. There are two main approaches to this task: two-stage detectors and one-stage detectors.

The most basic two-stage detector is a region proposal network (RPN) [5]**.** This network is primarily based on two layers, a regressor and a classifier as seen in figure 2.2. Each bounding box's position and dimension are predicted by the regressor, which is connected to the encoder's output. These outputs are the X, Y coordinates for the object and the width and height. The pixels located in these areas are passed to a classification subnetwork to assign a label or reject the proposal. The advantage of this method is that it produces a more accurate and flexible model that can propose an arbitrary number of regions that might contain a bounding box. However, the increased accuracy comes at the expense of computing efficiency. Detectron2 models are based on this approach, including RPN & Fast R-CNN [6] and Faster R-CNN [5].

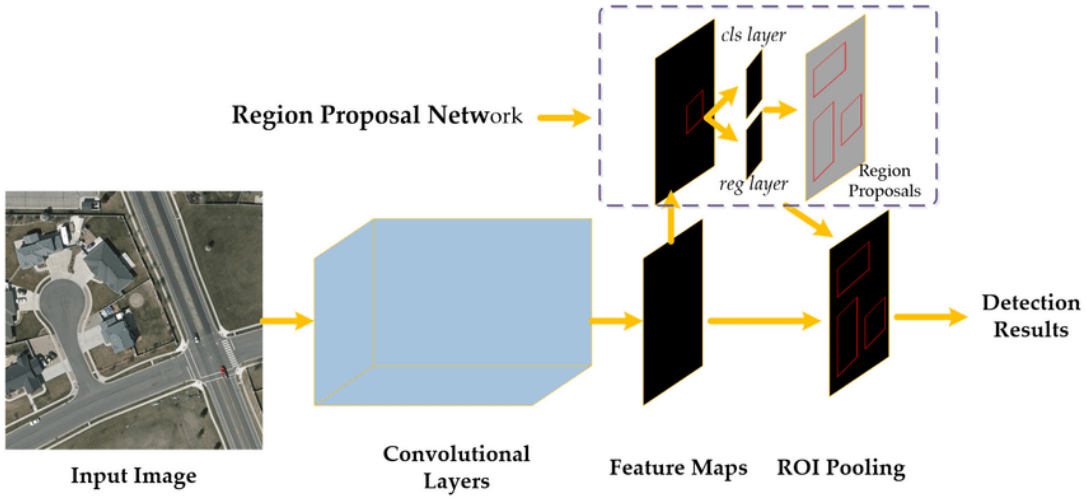


Figure 2.2 Region Proposal Network. "Cls layer" denotes classification layer, "Reg layer" denotes regression layer and "ROI pooling" denotes Region of interest pooling. (source [32])

In the other hand, single shot detectors (SSDs) [7] are one-stage and they are designed for object detection in real-time. SSDs rely on a set of specified regions instead of a subnetwork to propose regions. The input image is covered with an array of anchor points where each anchor point contains boxes of multiple shapes and sizes that serve as regions. For each box, the model outputs a prediction to denote how well an object matches in this particular grid and updates the box's location and size to increase the accuracy. As a result of the many boxes contained at every anchor point, SSDs produce multiple potential detections that overlap. Post-processing is necessary in order to refine these predictions and pick the best one. A widely used post-processing method is known as non-maximum suppression. SSDs are the predecessors to modern models like YOLO [8] (you only look once).

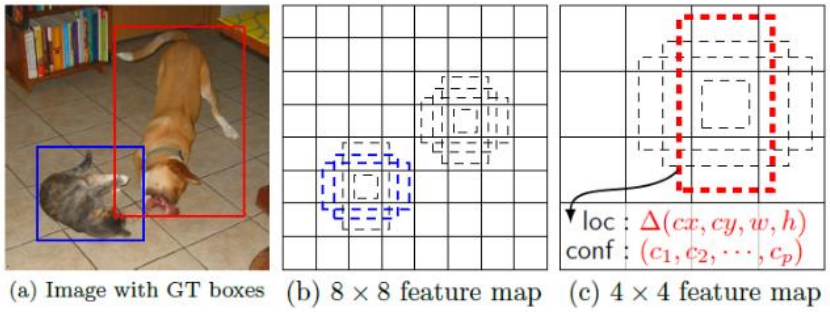


Figure 2.3 SSD multi-scale example (source [7])

As seen in *Figure 2.3* smaller grids are used to detect the cat, in contrast, the dog size grid is increased which makes the SSD more efficient.

The position and label for each object are provided by object detectors, but is necessary to know the reliability of this prediction. The most popular metric is called intersection-over-union (IOU). Given two bounding boxes, the area of the intersection is divided by the area of the union. This value ranges from 0 to 1.

2.3 Traffic Sign and Road Marking Detection

To ensure the recognition of traffic signs and road signs, researchers adopt multiple methods and techniques that can be divided in two main methods, which are classical and machine learning approaches. In our case we must approach traffic signs and road markings differently as the pose different sets of challenges.

Large fluctuations in illumination, meteorological conditions and also human factors are a challenge in both circumstances. The road scene also poses a challenge for recognition systems, as it comprises many objects that resemble traffic signs, as well as many cluttered objects that the make recognition more difficult. Other conditions such as motion blur due to the vehicles' motion or damaged signs have to be taken in account.

Some of the initial efforts to traffic sign recognition and classification rely on more standard computer vision algorithms that exploited the geometry (triangles, squares, circle, and octagon) and color of the signs in parallel combined with tracking. Integrating a tracking technique into these traditional approaches can help solve some of the problems by predicting the position of road signs in subsequent frames, thereby limiting the research zone to a smaller area, reducing the number of false positives. Despite the significant improvements proposed by traditional methodologies, these types of technologies cannot guarantee the effectiveness of traffic sign identification on their own and for these reasons researchers explore machine learning approaches.

Deep learning approaches have been researched for traffic sign detection as techniques in this field have improved. They are particularly efficient since they are usually based on a large collection of annotated data like German Traffic Sign Recognition Benchmark (GTSRB)[9] that helps train the algorithms to recognize road signs in a variety of settings, by introducing data augmentation a more robust model is accomplished against lightning changes, bad conditions, occlusions, damaged signs, and so on.

2.4 Face detection

In the fields of computer vision and pattern recognition, face detection is a crucial study area. Faces in real-world photographs exhibit a high degree of variability in scale, occlusion, expression and illumination, making it a difficult assignment. The development of precise and effective algorithms and techniques for face detection systems has recently attracted the attention of many scientists and engineers. Most approaches are based on models described in section 2.2. The standard datasets for this task are WIDER face [18], LFW [25] FDDB [26]. Best preforming networks released in recent years are Retina Face [27], DSFD [28] and Pyramid Box [29]. Usually, this technique is mostly related to face recognition as it enables many services and applications that range from Instagram or snapchat face filter for entertainment to more crucial implementations like biometrics or security surveillance systems.

2.5 License plate detection

This technique is also mostly related to a recognition task. Automated license plate recognition (ALPR) intends to use image processing and pattern recognition methods to extract and identify license plate (LP) characters from photos or videos of moving vehicles. Intelligent transport system (ITS) applications [30] such as traffic monitoring and control or parking management all rely heavily on ALPR techniques. Just as traffic sign and road marking detection in section 2.3 the foundation of conventional methods for LP detection is the extraction of characteristics, such as edge, color or character features [31]. These features are, however, sensitive to the background's complexity, such as items with a similar shape, color, texture, or character set. In this case as well CNNs have outperformed traditional approaches in terms of feature representation and performance, becoming the de-facto standard detectors.

3 Methodology

The pipeline of this thesis will be split in two main blocks. First of all, is necessary to further inspect the dataset and review its features, as seen in 3.1, in order to proceed with the removal of sensitive data. To accomplish this last task, it will be necessary to test the metrics of various state-of-the-art pre-trained models for face and license plate detection as described in section 3.2. Finally, in section 3.3, model will be trained with faster R-CNN network [5]. This dataset was designed to train in YOLOV5[10], all annotations must be converted to COCO [11] format before Detectron2 can process them.

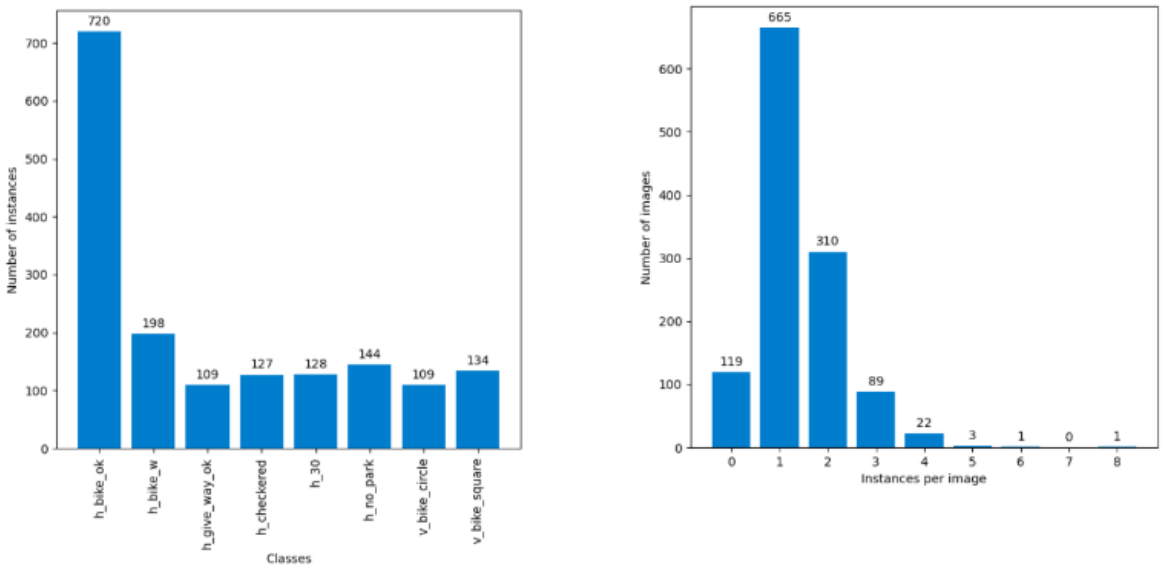
3.1 Dataset

This database was proposed by Elisabet Bayo in [12], mainly focusing on signs that might alert the riders from obstacles such as pedestrians and cars, or markings referring bike lanes and priority. The majority of the photographs are generated from snapshots of clips taken by the author while riding his bike. Video was captured at various times of the day and in various weather situations, with some bike lanes being repeated. Unfocused photos are also used as shown in figure 3.1.

Figure 3.1 Random samples that the display some features like variety of: aspect ratios a), focus f), light conditions, angle and size of bounding boxes to name a few.

3.1.1 Classes

This dataset is composed of 1210 annotated images and has been restructured in 5 folders, 3 folders for training (60%), one for validation (20%) and another one for testing (20%). These instances are composed by 8 classes as described in table 1



a) Number of instances of each class b) Number of instances per image

Figure 3.2 Dataset statistics (source [12])

Analyzing figure 3.2 we can relate some features of this histogram to Barcelona's bike lane model [13]. We can notice the large disparity in the number of instances of h bike ok, this is due to the fact that this marker is used 2 or 3 times each block, whereas h bike w is only used in bidirectional lanes. These signs are used to differentiate between unidirectional and bidirectional or whether the driver is riding in the correct direction. Frequency of other classes varies depending on the street. Another feature is that images have mostly one or two instances of objects since most road marking appear alone or in pairs. The images with more instances correspond to bidirectional lanes, where markings pairs are in both directions, and to bike lanes crossings.

It's also crucial to include background images that don't have any signs or marking to improve training. These are the images with 0 instances and account for 10% of the dataset.

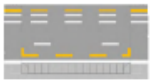
Name	Type	Use	Description	Image
h_bike_ok	RM	Mark a bike lane and right direction	Bike	
h_bike_w	RM	Mark a bike lane and wrong direction	Bike upside down	
h_give_way_ok	RM	Pedestrian priority section ahead	Triangle pointing down	
h_checkered	RM	Pedestrian priority for service access	Checkered area on the ground	
h_30	RM	30km/h speed limit, bikes have priority overs cars	Number 30 inside a circle	
h_no_park	RM	Areas where cars may come out from garages or similar	Yellow line on the side of the road delimited by straight angles	
v_bike_circle	TS	Bike lane use compulsory for bikes (in use when going against car direction)	Bike in a blue circle with a white outline	
v_bike_square	TS	Bike lane use recommended but not compulsory	Bike in a blue square with a white outline	

Table 1 Description of road markings (RM) and traffic signs (TS)

(source [12])

3.1.2 Format conversion

This dataset was made for Yolov5 [10], and it is structured in 5 folders with their respective Data config file, each one containing a folder with images and another folder with their respective labels.

YOLO v5 requires each image to have annotations in the form of a.txt file, with each line describing a bounding box. The bounding boxes are written as the following image shows.

Each row is related to one instance in the image, and it is composed by class, x_center, y_center, width and height, respectively. Box coordinates are normalized by the dimensions of the image and classes starting from 0. These annotations have the same name as their image correspondence. The images are then loaded with a data config yaml file that specifies the number of classes, as well as their names and directories for the train and validation folders.

COCO saves its annotations in JSON format describing object classes, bounding boxes, and bitmasks. COCO has five different annotation types but for this training we will use the object detection one, it has a list of categories and annotations. Categories are made by classes, their ids and weather they belong to a super category. The annotations section contains a list of every individual object annotation from every image in the dataset, containing: Area, image id, bounding box (top left x position, top left y position, width, height), category id corresponds to a class in the categories section and finally a unique id for every annotation object.

Starting with yolo is a little tricky because the yolo format saves the normalized image's size and width. As a result, you must read the image file to determine the original image height and width. The PyLabel [15] package is used to make the conversion. The current dataset is rearranged in 3 folders for train, validation. After the conversion 3 COCO datasets are obtained each one with their respective JSON file.

3.2 Anonymization

Obfuscation of the face [16], such as facial blurring, has been demonstrated to be successful in ensuring privacy. However, entire, unobfuscated images are often used in object detection, object recognition, and image segmentation. It has been proven for trainings in ImageNet that datasets with blurred faces account for negligible accuracy drops (≤ 1.0%) [16]. The same applies for license plate blurring.

In this section we'll examine pre-trained models used to detect faces in section 3.2.1 and license plates in section 3.2.2.

3.2.1 Face detection

For this task, YOLOV5-face [17] had the best performance both in accuracy and speed. This model was trained on WiderFace dataset [18] which is the largest face detection dataset, containing 32,203 images and 393,703 faces. It is realistic and complex due to the wide range of scale, position, occlusion, expression and illumination. The medium sized yolov5m pre-trained model performed well enough for this task. After the inference we run Non maximum suppression with IuO_threshold=0.5. Finally archiving bounding boxes in x, y, w, h format and applying a gaussian blur to the area as seen in figure 3.3.

Figure 3.3 Anonymized snapshot from dataset

3.2.2 License plate detection

In this case it was significantly harder to find a reliable model since LP usually present a smaller area and are usually distorted due to oblique views. Another consideration, is that LP have significant changes in different countries, so many pre-trained models for US or other countries did not fit four our case.

The method is pretty similar to section 3.2.1, in this case we proceed using ALPR [19]. This approach is based on a combination of car detection network with YOLOV2 [20] and LP detection with WPOD-NET[21], which allows the correction of the LP area to a rectangle resembling a frontal view by regressing one affine transformation per detection. In this case bounding boxes come in pairs of (x1, x2, x3, x4) and (y1, y2, y3, y4), indicating the pixel position of every corner. These boxes are not constrained to a rectangle but act as mask as seen in *Figure 3.4*.

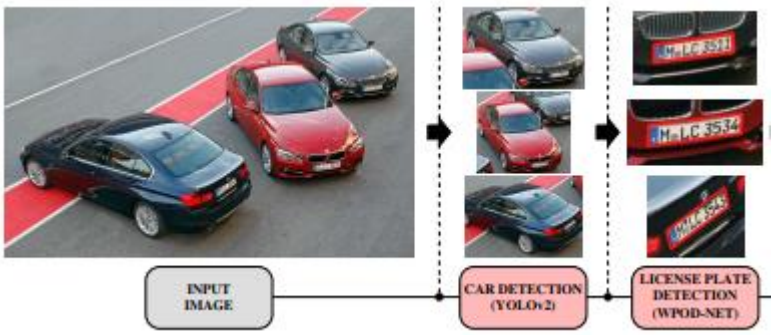


Figure 3.4 ALPR pipeline (source [19])

3.3 Road sign and road marking detection

As previously mentioned, the network of choice for this task is a Faster RCNN with FPN backbone. This model is provided by Detectron2 in their model zoo API. First step will be further examination of the network architecture in section 3.3.1 and then we will proceed to the training algorithm in 3.3.2.

3.3.1 Architecture

This is a multi-scale detector that accomplishes high accuracy while detecting tiny to large objects, making itself the go-to standard detector.

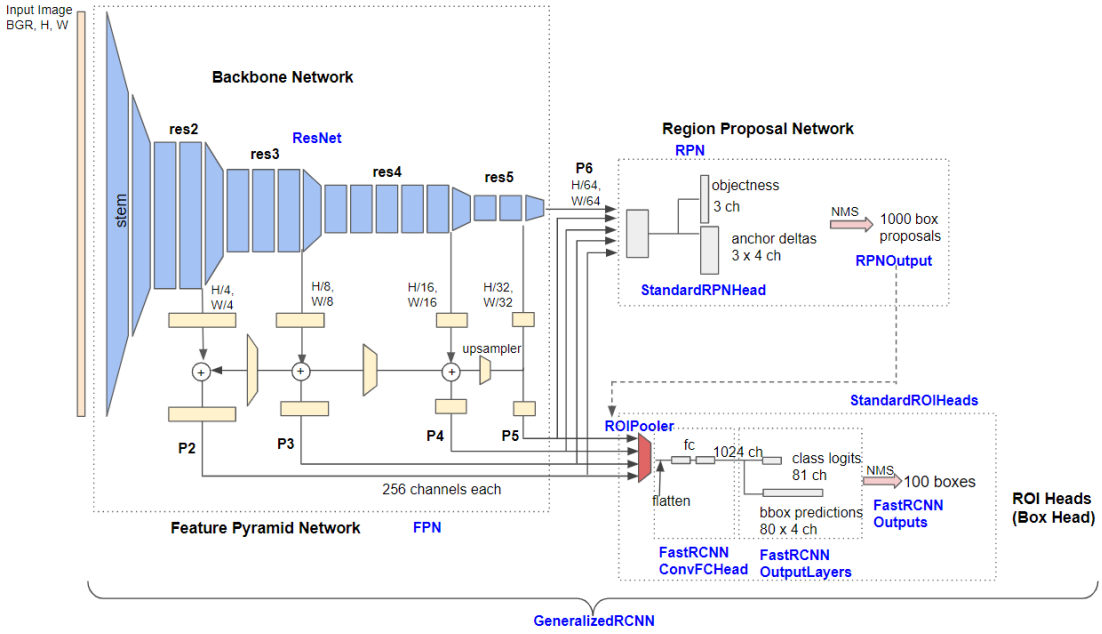


Figure 3.5 Faster RCNN architecture (source [33])

The network's architecture is depicted in *Figure 3.5*. It consists of three main blocks, more specifically:

Backbone Network: from the input image this network extracts feature maps at different scales such ad P2 (1/4 scale), P3 (1/8), P4 (1/16), P5 (1/32) and P6 (1/64) are the output features of Base-RCNN-FPN. This network takes the image input and returns a dictionary of feature maps:

- **Input** is an image stored in a tensor composed by batch size, color (BGR), height and width.
- **ResNet** is right after input, made by various convolutional blocks such as stem blocks and bottle necks that gradually down-sample the input, for this project Resnet 101 was chosen having (3, 4, 23, 3) blocks. These blocks are connected to lateral and output convolution layers with up-samplers in between and a last-level maxpool layer for P6. This is a brief overview of the **feature pyramid network**.
- **Output** is a dictionary containing the Feature maps at different scales and channels, 256 by default, described in *figure 3.6*.

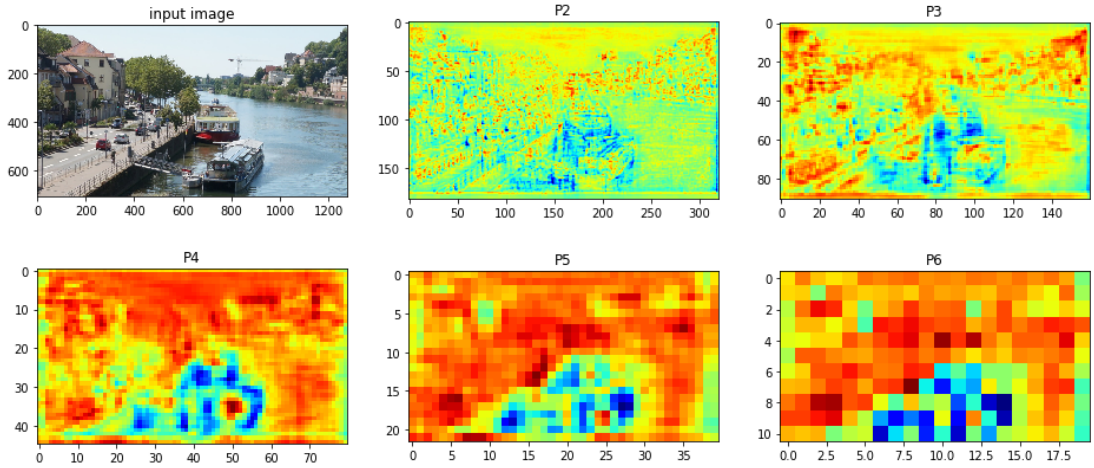


Figure 3.6 feature maps output (source [34])

Region Proposal Network: predicts the objectness and the object boundary box at each point by sliding a window over the feature maps. By default, 1000 box proposals with confidence scores are generated. For every scale level, a 3×3 convolution filter is applied over the feature maps followed by separate 1×1 logits convolution for objectness predictions and 1x1 anchor deltas convolution for boundary box regression. The objectness evaluates if an object is present in the box. These convolutional layers are called RPN head. All the feature maps at their various scale levels of are treated with the same head.

The same head is applied to all different scale levels of feature maps. Usually, small objects are detected at P2 and P3 and the larger ones at P4 to P6 as seen in *figure 3.7*.

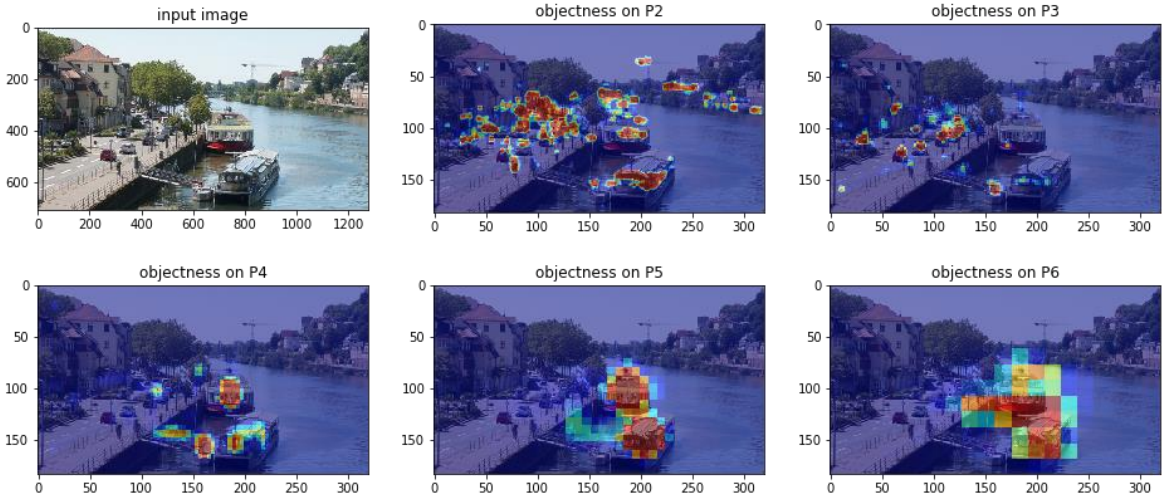


Figure 3.7 objectness maps overlayed on input image (source [35])

Box Head: ROI pooling uses fully-connected layers to generate fine-tuned box positions and classification results cropping and warping feature maps into many fixed-size features. We select the feature map layer at the most appropriate scale to extract the feature patches based on the size of the ROI. The formula to choose the feature map is related to the width w and height h.

$$k = \lfloor 4 + \log_2(\sqrt{wh}/224) \rfloor.$$

Where 224 is the canonical box size. If the box area is 448 it is assigned 5th level (P5). Finally, using non-maximum suppression (NMS), repeated detections are filtered. This structure is based on *figure 2.2* ROI block.

3.3.2 Training

Training was done on 654 images with table 3 distribution and table 4 parameters.

Category	Instances	Category	Instances	Category	Instances
h_bike_ok	441	h_bike_w	126	h_give_way_ok	63
h_checkered	73	h_30	72	h_no_park	85
v_bike_circle	61	v_bike_square	81		
total	1002				

Table 2 Training instance distribution

Model	*Faster RCNN X 101 FPN 3X*
Number of epochs	*1.5 k*
Data augmentation (performed in roboflow)	*Resize: Stretch to 416x416 (for faster training)* *Shear: ±8° Horizontal, ±8° Vertical* *Crop: 0% Minimum Zoom, 20% Maximum Zoom* *Brightness: Between -30% and +30%*
Number of classes	*8*
Training samples	*654*
Parameters	*Initial learning rate 0.001* *Learning rate decay: 5 1e^{-2} every 1500 steps* *IoU threshold: 0.7* *Batch size: 64*

Table 3 Training parameters

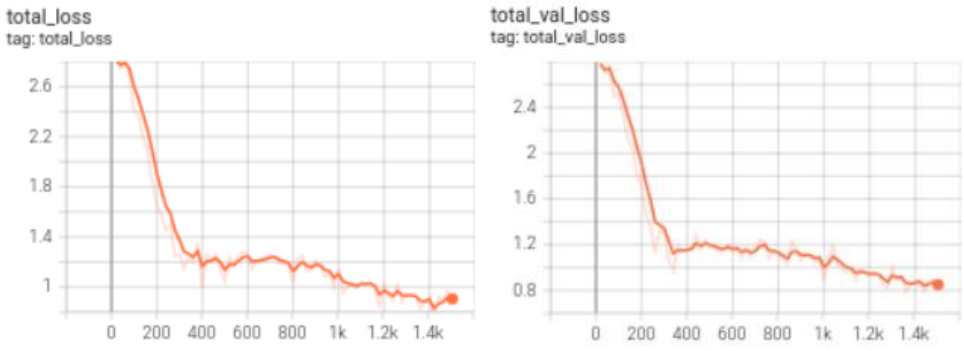


Figure 3.8 Loss metrics during training

If we analyze the loss curves in *figure 3.8*, values of training loss (left) and validation loss (right) seem to depict a good bias, as both curves start with a slight offset but quickly align at epoch 200 and keep decreasing at a similar rate. Validation doesn't go up so we can assure that the model is not overfitting. Both finish approximately at 0.8, which is not a bad result but there's room for improvement. This improvement mostly comes from finding better parameters for training and balancing the data.

4 Discussion of the solution

4.1 Anonymization

As we can see in the in figure 4.1 and figure 4.2, the script for the dataset anonymization works as expected for faces and LP's. Some artifacts caused by false detections are visible in some frames, that could be resolved by discarding inferences that doesn't match a certain confidence score but this entails a higher chance for false negatives to occur.

Figure 4.1 Anonymized face frame

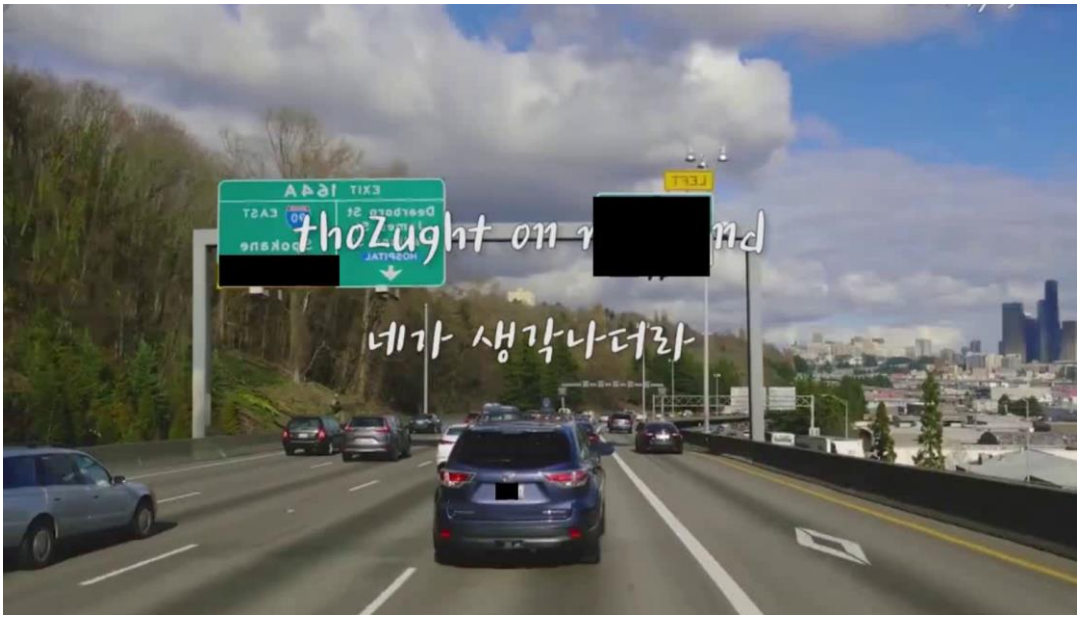
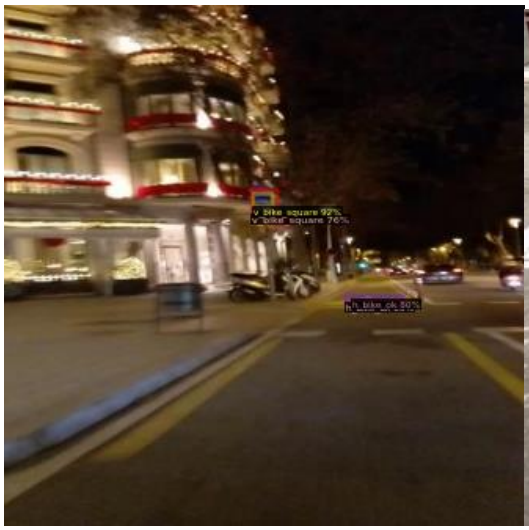


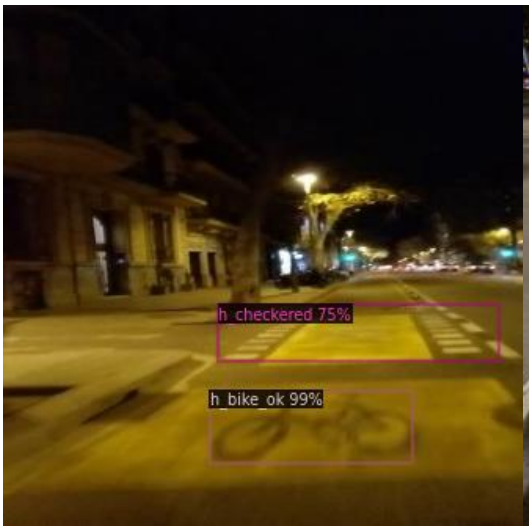
Figure 4.2 Anonymized LP frame (source[36])

4.2 Road sign and road marking detection

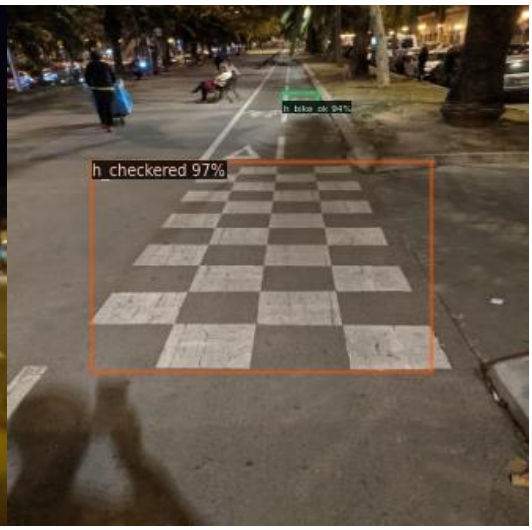


a)

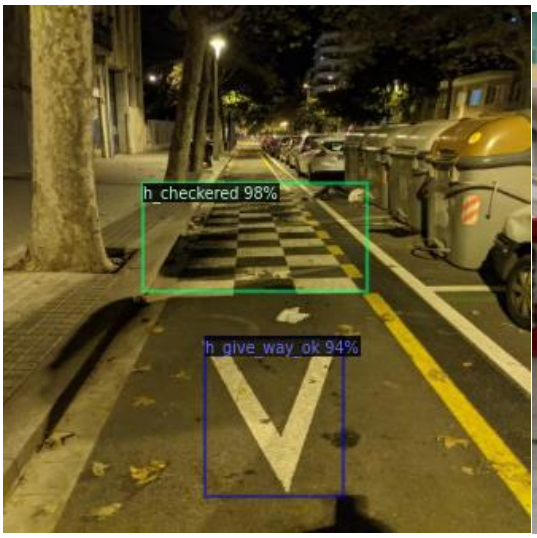
b)



c)

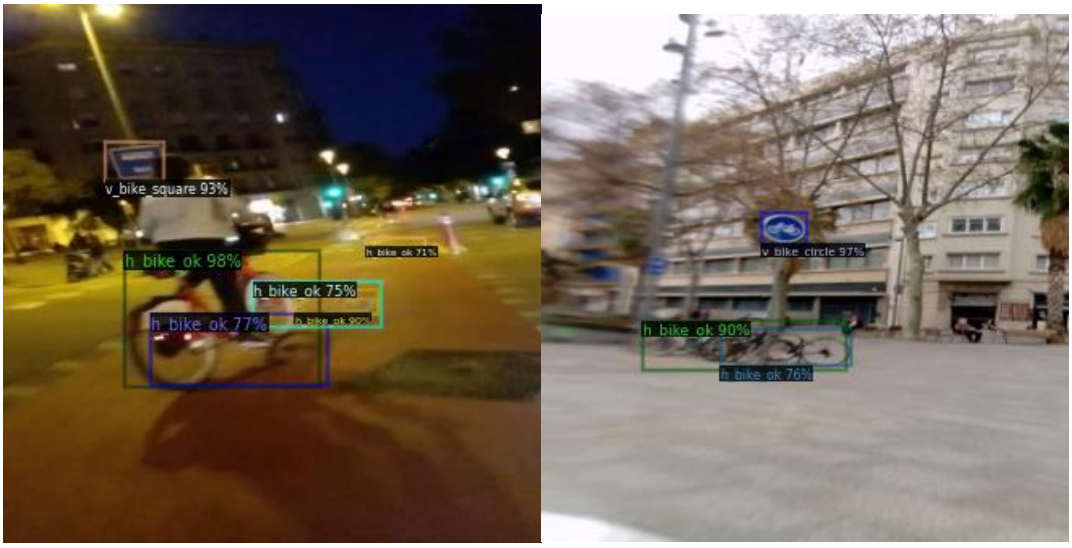


d)



e)

f)



g) h)

Figure 4.3 Random dataset samples inferenced with trained model.

As we can see in *figure 4.3* this model has a good general performance, it's capable of detecting majority of classes in easy and medium difficulty cases even in some harder ones like a) that has really poor lighting and bad angle. As seen in d) and e) most of the times checkered boxes are usually situated after h_bike_ok or h_give_way_ok which causes the model to make fake positives when these signs are present and floor is not plain c). Another recurrent issue is that detects sewers as h_30 and most of the time detects real bicycles as traffic sign h_bike_ok.

category	ap	category	ap	category	ap
h_no_park	25.357	h_30	68.374	h_give_way_ok	38.479
h_bike_w	33.065	h_checkered	43.714	v_bike_square	39.522
h_bike_ok	56.317	v_bike_circle	56.404		
AP	AP50	AP75	APs	APm	APl
50.349	75.426	56.359	28.078	60.137	60.137

Table 4 Average precisión tables.

Model accomplishes an average precision of 50 which is not excellent but does a decent work detecting the classes in the majority of possible scenarios. Possible upgrades will be stated in section 6.

5 Analysis and assessment of environmental and social implications

As mentioned in section 1.4 micro mobility plays a big factor in both environmental and social aspects, especially in the years to come. The insight provided by this project serves as a testbed for the development of an assisted driving application for PMVs. This application would have a direct social impact that would imply an indirect environmental impact as well. In section 6.1 it is described how it is going to affect the society and subsequently in sector 6.2 It is explained how this translates to the environmental aspect.

5.1 Social impact

The final objective of thesis and the whole project is to improve the safety of pedestrians, drivers and riders, which ultimately leads to a better welfare for all citizens. Making the riders aware of their surroundings as well as the traffic rules will reduce the rate of intra-urban accidents which account for multiple retentions in the already outdated and congested model for mobility in big cities. Furthermore, PMVs help less mobile user groups, like elderly or physically impaired people who can drive themselves, become more mobile and, as a result, more socially active, which is a really important factor as far as inclusivity goes. [22][23].

5.2 Environmental impact

As these enhancements in safety and welfare are proved, public opinion and usage will improve making governs and city councils invest more in PMVs and the necessary infrastructure thus making a faster transition to sustainable mobility. According to Ujet and SustainAbility, if 8% of current road cars are replaced by electric vehicles, by 2050, emissions would be minimized by 80% [24]. That has a huge long-term positive influence on the environment.

6 Conclusions and future work

In this research, we proposed a solution to treat with sensitive information in datasets as well as a model for traffic sign and road marking detection, this testbed relies on a model trained on FASTER RCNN serving as first step to developing an application for assisted driving. Workload has been entirely focused in signal processing, computer vision and deep learning.

After running multiple trainings in order to yield a better accuracy, various aspects about the whole approach became clear that need to be tackled different. The main subject to work on Is the unbalanced distribution of the dataset, this can be addressed with multiple techniques such as using a focal loss [14], down-weighting inliers in order to decrease the contribution of a largely unbalanced class to the total loss. Another simple technique is to resample the dataset such that all classes have similar distributions.

In the long run, it is interesting to implement tracking and optimize this system to run on mobile devices on real time. It is also interesting to explore other approaches for object detection such as YOLO or TensorFlow, since multiple methods based on these systems proved to be equally capable and less resource intensive. Besides, Detectron2 is convoluted and required some effort to understand, a large portion of the project was spent to get a grasp on how the library works in general and in many cases is not possible to extrapolate the examples they provide to custom scenarios.

7 References

[1] European Parliament, Council of the European Union GDPR Regulations
https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32016R0679 Online, Apr 2016

[2] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, Ross Girshick. Facebook research/detectron2. https://github.com/facebookresearch/detectron2 2019

[3]«BOE» núm. 297, Ministerio de la Presidencia, Relaciones con las Cortes y Memoria Democrática https://www.boe.es/diario_boe/txt.php?id=BOE-A-2020-13969 Nov 2020 pages (98638-98643).

[4] Kaidong Li, Wenchi Ma, Usman Sajid, Yuanwei Wu, Guanghui Wang. Object Detection with Convolutional Neural Networks arXiv:1511.08458v2 [cs] Dec 2015

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks arXiv:1506.01497v3 [cs.CV] Jan 2016

[6] Ross Girshick. Fast R-CNN arXiv:1504.08083v2 [cs.CV] *Sep 2015*

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector arXiv:1512.02325v5 [cs.CV] Dec 2016

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection arXiv:1506.02640v5 [cs.CV] *May 2016*

[9] Johannes Stallkamp; Marc Schlipsing; Jan Salmen; Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition
The 2011 International Joint Conference on Neural Networks, 2011, pp. 1453-1460, doi: 10.1109/IJCNN.2011.6033395.

[10] Glenn Jocher; Ayush Chaurasia; Alex Stoken; Jirka Borovec; NanoCode012; Yonghye Kwon; TaoXie; Jiacong Fang; imyhxy; Kalen Michael; Lorna; Abhiram V; Diego Montes; Jebastin Nadar; Laughing; tkianai; yxNONG; Piotr Skalski; Zhiqiang Wang; Adam Hogan; Cristi Fati; Lorenzo Mammana; AlexWang1900; Deep Patel; Ding Yiwei; Felix You; Jan Hajek; Laurentiu Diaconu; Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. Doi: 10.5281/zenodo.6222936

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. Microsoft COCO: Common Objects in Context arXiv:1405.0312v3 [cs.CV] Feb 2015

[12] Bayó Puxan, Elisabet. Traffic Sign Detection for Micromobility http://hdl.handle.net/2117/350044 May 2021

[13] Manuel Crespo Yáñez and Joan Valls Fantova. Manual de disseny de carrils bici de Barcelona. https://bacc.cat/wp-content/uploads/2016/03/ESBORRANY_Manual-de-disseny-de-carrils-bici-barcelona-2016.pdf Mar 2016.

[15] Jeremy Fraenkel, Alex Heaton, and Derek Topper. PyLabel https://github.com/pylabel-project/pylabel Nov 2021

[16] Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, Olga Russakovsky. A Study of Face Obfuscation in ImageNet arXiv:2103.06191v3 [cs.CV] Jun 2022

[17] Delong Qi, Weijun Tan, Qi Yao, Jingfeng Liu. YOLO5Face: Why Reinventing a Face Detector. arXiv:2105.12931v3 [cs.CV] Jan 2022

[18] Shuo Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. WIDER FACE: A Face Detection Benchmark. arXiv:1511.06523v1 [cs.CV] Nov 2015

[19] Sérgio Montazzolli, Claudio Rosito Jung. License Plate Detection and Recognition in Unconstrained Scenarios doi: 10.1007/978-3-030-01258-8_36 Sep 2018

[20] Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger arXiv:1612.08242v1 [cs.CV] Dec 2016

[21] Phat Nguyen Huu, Cuong Vu Quoc. Proposing WPOD-NET combining SVM system for detecting car number plate. DOI: http://doi.org/10.11591/ijai.v10.i3.pp657-665 Sep 2021

[22] Rebecca Johnso. Mobility scooters in the UK: Public perception of their role. DOI:10.1680/jtran.16.00140. June 2017

[23] David Metz. Future Transport Technologies for an Ageing Society: Practice and Policy. DOI:10.1108/S2044-994120170000010009. December 2017

[24] Tommi Inkinen, Tan Yigitcanlar, Mark Wilson. Sustainable Mobility and Transport. (ISSN 2071-1050). August 2021

[25] Gary B. Huang,Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. 2007

[26] Vidit Jain and Erik Learned-Miller. FDDB: A Benchmark for Face Detection in Unconstrained Settings. UM-CS-2010-009. 2010

[27] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, Stefanos Zafeiriou. RetinaFace: Single-stage Dense Face Localisation in the Wild. arXiv:1905.00641v2 [cs.CV]. May 2019

[28] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, Feiyue Huang. DSFD: Dual Shot Face Detector. arXiv:1810.10220v3 [cs.CV]. Apr 2019

[29] Xu Tang, Daniel K. Du, Zeqiang He, Jingtuo Liu. PyramidBox: A Context-assisted Single Shot Face Detector. arXiv:1803.07737v2 [cs.CV]. Aug 2018

[30] Ying Wen; Yue Lu; Jingqi Yan; Zhenyu Zhou; Karen M. von Deneen; Pengfei Shi. An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. DOI: 10.1109/TITS.2011.2114346. March 2011

[31] Yuanxing Zhao, Jing Gu, Chui Liu, Shumin Han. License Plate Location Based on Haar-Like Cascade Classifiers and Edges. DOI:10.1109/GCIS.2010.55. Dec 2010

[32] Wei Chen, Funing Zhong, Tan. Multiple-Oriented and Small Object Detection with Convolutional Neural Networks for Aerial Image DOI:10.3390/rs11182176. Sep 2019

[33] Hiroto Honda. Digging into Detectron 2 — part 1. https://medium.com/p/47b2e794fabd Online, Jan 2020

[34] Hiroto Honda. Digging into Detectron 2 — part 2. https://medium.com/@hirotoschwert/digging-into-detectron-2-part-2-dd6e8b0526e Online, Jan 2020

[35] Hiroto Honda. Digging into Detectron 2 — part 4. https://medium.com/@hirotoschwert/digging-into-detectron-2-part-4-3d1436f91266 Online, Mar 2020

[36] Oregon Tropics. Spokane, Washington | 4k Driving Tour | Dashcam. https://www.youtube.com/watch?v=Zi40on9J2Zg&ab_channel=OregonTropicst

[37] Regulations https://www.barcelona.cat/mobilitat/en/noticia/new-regulations-on-the-circulation-of-personal-mobility-vehicles_1027137