


Effective and efficient detection and interpretation of road direction signs

Marco Buzzelli , Stefano Talamona
Department of Informatics, Systems and Communication
University of Milano – Bicocca
Milan, Italy

Abstract—Road direction signs depict textual and directional information to convey instructions about how to reach a given destination. Detecting and interpreting road direction signs constitutes a key component in self-driving vehicles, therefore we propose a hybrid pipeline that combines deep learning with traditional handcrafted image processing, aiming for a combination of effectiveness and efficiency. Our solution includes a procedure for the identification of the orientation of arrows, generalizing on a wide variety of pictorial styles for direction signs across the globe. Our pipeline is evaluated in terms of accuracy and inference time of its individual steps, demonstrating excellent performance. Experiments are performed over two variants of the pipeline, assuming the availability of different levels of computational resources. We also test the system’s dependence on annotated supervision by performing evaluation with a varying number of training instances.

Index Terms—road direction signs, street scenes, object detection, arrow orientation

I. INTRODUCTION

A self-driving vehicle is a vehicle capable of analyzing and understanding its surrounding environment, in order to develop a strategy to bring passengers safely and in the shortest possible time to their destination. The perception system of such a vehicle must be able to analyze and understand a potentially infinite amount of different situations, having to adapt to each of them in order to ensure maximum effectiveness, maximum efficiency, and above all maximum safety. The behavior of a self-driving vehicle can be, therefore, guided by a number of inferences to be performed on street scenes, ranging from depth estimation [1] to vehicle recognition [2]. In this paper we specifically focus on addressing the presence of direction signs, i.e. road signs that include two complementary pieces of information:

- textual information, composed of written text indicating which place or places will be encountered continuing in a certain direction;
- directional information, consisting of one or more arrows indicating which direction to follow to reach the destination indicated by the text.

The processing pipeline scheme is graphically represented in Figure 1. To the best of our knowledge, this is the first published instance of a method for the detection and interpretation of road direction signs with a focus on arbitrarily

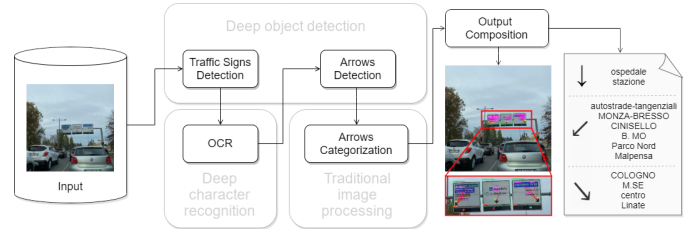


Fig. 1: Overall pipeline for detection and recognition of road direction signs.

shaped and oriented arrows. Wenzel et al. [3] employed a scale-invariant approach based on ACF features (Aggregated Channel Features) [4] to locate the four corners of a sign and to train four models of Support Vector Machines, thus obtaining four different detectors. These were used through a sliding-window approach to generate hypotheses about the presence of potential corners belonging to the signs. Finally these hypotheses, consisting of pairs formed by coordinate points expressed in pixels and a score value, were filtered on the basis of the latter and geometric constraints. This work was however limited to the detection of direction signs, without further analysis or interpretation of their content, similarly to the works by Choi et al. [5] and Tabernik et al. [6]. A pivotal step in the interpretation of road direction sign is in fact the classification of arrows based on the direction they indicate. On this domain, limited research has been documented within the scientific literature. Existing works [7]–[9] focus on an extremely reduced set of arrows shapes and colors, exploiting their geometric characteristics to define handcrafted features with which to classify the type of arrow (and consequently the direction represented). This limitation is in part attributed to the different application domain, focusing on road markings as opposed to road traffic signs.

II. PROPOSED METHOD

In the following, we describe our pipeline for the detection and interpretation of road direction sign. Its design is motivated towards generalizing for the wide variety of appearances that may be encountered on different roads all around the world, as depicted in Figure 2.



Fig. 2: Sample of road signs from around the world, depicting the intrinsic variability of this class.

A. Sign detection

The first step of our pipeline involves identifying all the road signs present in the image. The YOLOv7 neural architecture was used for this task [10], [11], due to the extensively proven efficacy on a wide variety of detection applications. In order to tune the solution to the problem at hand, domain-specific data augmentation operations have been applied to the data. Perspective deformations [12] simulate the effect of the sign being acquired from different road lanes, at different distances, and account for a misplacement of the imaging device, whose positioning on the vehicle might change through time due to the impact of road bumps. A combination of CutMix [13], copy paste [14] and mosaic [15] operations is used to recreate additional scenarios of multiple road signs present in one image. Pseudo-random modifications of the Hue-Saturation-Value representation of the input recreate the impact of atmospheric phenomena, and generalize to the acquisition with different cameras. Additionally, a random application of the image negative of signs portions has been implemented, to balance the amount of traffic signs that are written with light text on a dark background, and vice-versa. Preliminary experiments revealed that this solution reduced the final model accuracy, possibly due to the larger amount of one category in the test set, and as such it has not been included in our final solution. Detection instances that are entirely contained within others are removed (this occurs when particular rectangular graphic elements with different colors and writings appear inside a sign).

B. Text detection and recognition

The extraction and categorization of textual information relies on EasyOCR [16]. The main steps of the pipeline proposed by the authors consist of a feature extraction process with a CRAFT feature approach [17], used for character identification, which are then categorized using ResNet [18] as a feature extractor, and LSTM (Long Short Term Memory) [19] and CTC (Connectionist Temporal Classification) [20] to recognize character sequences. In case the procedure produces a null output, the current candidate road sign is discarded. Otherwise, the road sign portion is given as input to the subsequent arrow detection step. The decision to perform the Optical Character Recognition (OCR) step first with respect to the

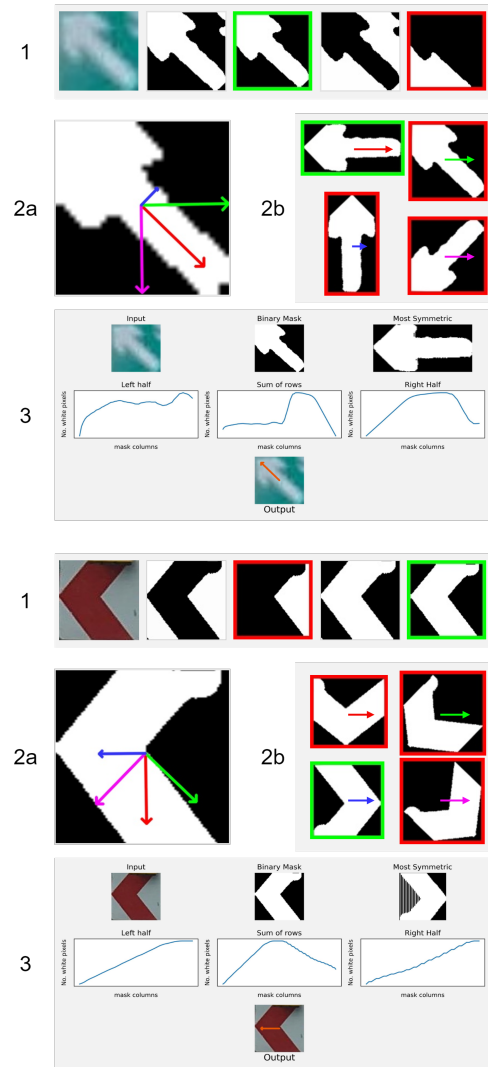


Fig. 3: Two example executions of the proposed arrow orientation categorization procedure.

arrow detection step was taken on the basis of preliminary studies, which proved that this procedure requires less time than arrow detection, thus leading to a potential reduction in calculation time. The output of this step includes both the detected text and its position within the input: the spatial information will be used later in the last step of the pipeline to compose the final result.

C. Arrow detection and orientation categorization

Arrows inside road signs from all around the world show an extremely variable range of shapes and colors, as shown in Figure 2, making the use of handcrafted features for detection impractical. For this reason, we relied on a second YOLOv7 model for the arrow detection (trained on an appropriate custom dataset), and we defined a procedure based only on traditional image processing techniques to categorize the arrow's orientation, synthesized in Figure 3. The set of possible directions indicated by the arrows is here discretized into eight

TABLE I: Results obtained with YOLOv7 and YOLOv7-tiny architectures for the two object detection models: road traffic signs, and arrows.

Model	Training Instances	Class	Validation Precision	Validation Recall	Validation mAP@.5	Validation mAP@.5:.95	Test Precision	Test Recall	Test mAP@.5	Test mAP@.5:.95	
YOLOv7	10'033	all	0.707	0.601	0.65	0.447	0.697	0.596	0.633	0.431	
		d-or-i	0.68	0.672	0.704	0.513	0.662	0.659	0.671	0.484	
		other	0.733	0.53	0.597	0.382	0.732	0.533	0.594	0.378	
	5'017	all	0.718	0.574	0.619	0.42	0.7	0.554	0.595	0.399	
		d-or-i	0.691	0.646	0.667	0.479	0.662	0.609	0.621	0.442	
		other	0.745	0.503	0.572	0.362	0.738	0.498	0.568	0.356	
	2'509	all	0.676	0.526	0.565	0.372	0.66	0.518	0.541	0.355	
		d-or-i	0.644	0.574	0.6	0.418	0.627	0.559	0.558	0.388	
		other	0.707	0.479	0.529	0.326	0.693	0.476	0.525	0.322	
	824	arrow	0.966	0.986	0.991	0.735	0.974	0.95	0.983	0.749	
	YOLOv7-tiny	10'033	all	0.658	0.482	0.516	0.325	0.64	0.474	0.502	0.314
			d-or-i	0.636	0.561	0.577	0.381	0.627	0.533	0.55	0.359
other			0.679	0.403	0.455	0.269	0.654	0.414	0.455	0.27	
824		arrow	0.912	0.938	0.949	0.668	0.922	0.899	0.95	0.66	

categories: up, down, left, right, north-west, north-east, south-west and south-east.

The first step in orientation categorization is a binarization of the arrow region. Since a threshold value could not be established a priori, the Otsu binarization technique was used [21]. To account both for light-on-dark and dark-on-light arrows, it is assumed that the region of the arrow is the “most central” connected component, i.e. that it intersects the center of the image.

The second step is identifying the arrow’s rotation angle, regardless of its pointing direction. This is achieved by first determining a set of four rotation candidates through the application of Principal Component Analysis (PCA) [22], and then performing a symmetry analysis on these candidates. PCA is first applied to the binary mask of the arrow, obtaining two resulting orthogonal eigenvectors, which provide information about the direction of white pixels distribution within the mask. This idea is based on the fact that every arrow, regardless of its shape, has a head that takes up a great part of the total area of the arrow; this head constitutes a morphological component that extends towards a well-defined direction. Four directions emerging from PCA are therefore taken into consideration: the major and minor orthogonal eigenvectors, plus the two intermediate rotations to account for mis-identifications. To determine which of these rotations is the most likely, a symmetry analysis is performed. The arrow binary mask is rotated according to each candidate, and an inverse symmetry metric is computed as the accumulated difference between the left half and the reversed right half. The most symmetrical rotation will be the one with the minimum accumulation.

The third step is a categorization of the arrows pointing direction, given its rotation. We start by rotating the binarized arrow to be horizontal, following the information estimated at the previous step. The goal is to determine the position of the vertex (left half, or right half) by studying the trend of the sums per column of the arrow mask. These sums are used to create two plots, one for each half. The ascending (or descending) curve of maximum extension identifies the arrow half containing the vertex, thus uniquely classifying the

pointing direction. In this case, the extension is computed as the sum of columns with a monotonic trend. In some cases, arrows without a body contain a convexity that leads to a failure in this algorithm. For this reason, vertical convexities are filled before computing the column sums.

D. Output composition

The output of the various steps of the pipeline just described are eventually combined to obtain the final output of the system. It was decided to associate textual and directional information based on their spatial position within the sign, by resorting to Euclidean distance. This is calculated between all the text-arrow pairs in order to determine which of these are closest to each other. Eventually, the pipeline output is grouped by direction, so as to have textual data composed of pairs of elements where the first represents the discretized direction, and the second all the destinations that can be reached by proceeding in that direction.

III. EXPERIMENTS AND RESULTS

In this section we describe the experimental setup and results, as documented in Table I.

A. Experimental setup

The dataset “Mapillary Traffic Sign Dataset” (MTSD) [23] has been used for method training and assessment. The circa 42,000 examples provided with relative annotation file, in fact, guarantee the possibility of experimenting with different configurations of the training data for the sign recognition models. All analyzed images have an excellent general quality, ensuring sufficient legibility of the contents of the signs. Furthermore, the data was collected from many different countries, guaranteeing a very high variety in terms of immortalized scenes. Key aspect of this dataset is the presence of a class of sign instances called “direction_or_information”, which includes, but is not limited to, direction signs.

Training, validation and test sets were defined for sign detection. These were created by selecting the instances of the MTSD dataset for which the related annotation file reports

TABLE II: Quantitative results related to the OCR procedure performed using the EasyOCR library.

Metric considered	Resulting value
Minimum edit distance	0
Maximum edit distance	15
Average edit distance	2.16
Minimum text length (ground truth)	4
Minimum text length (OCR results)	4
Maximum text length (ground truth)	106
Maximum text length (OCR results)	106
Average text length (ground truth)	24.24
Average text length (OCR results)	24.0

the presence of at least one sign instance belonging to the “direction_or_information” class. In total these turned out to be 12,539, and were divided as follows: training set 10,033 images (80%), validation set 1,253 images (10%), test set 1,253 images (10%). Two classes were used to train the detector: “direction_or_information”, and “other” in which are aggregated all the other classes of signs present in the MTSD. During the OCR step there is no discrimination on the input regarding the classes to which the bounding boxes belong: this aims to handle the case in which a direction sign has been recognized as a “sign” but not as a “direction sign”. The metrics considered for detection evaluation are: accuracy, recall, and mean average precision (mAP). mAP is computed respectively at 0.5 Intersection over Union (mAP@.5) and averaged between 0.5 and 0.95 Intersection over Union (mAP@.5:.95).

Evaluation of the OCR procedure required the creation of a set of images with relative annotations, since MTSD does not provide this piece of information. Fifty bounding boxes of signs from the dataset were selected, automatically extracted by the object detection procedure. For each bounding box, a text file was manually produced containing a single line of text which represents the transcription of all the textual information present in the sign. The Levenshtein distance [24] (or “edit” distance), was used for evaluation, converting both annotation and prediction to lowercase, and removing blank spaces between words.

For the arrow direction categorization, fifty instances of arrow images were automatically extracted by the object detection procedure from images of the MTSD test set, and their orientation was manually annotated. Assessment is measured as absolute matched classes over the eight possible directions, ignoring the intensity of the orientation error.

To the best of our knowledge, no other methods for the detection and interpretation of road direction signs have been published. As such, our results constitute an initial benchmark for future comparison by other methods.

B. Experimental results

Table I reports the results on object detection, both for road signs and arrows. Results are presented divided according to the set of data on which they were produced

TABLE III: Quantitative results related to the process of arrows direction categorization.

Direction	Total	Correct	Wrong
All	50	47	3
up	7	6	1
down	6	6	0
left	7	7	0
right	7	5	2
north-west	6	6	0
north-east	7	7	0
south-west	5	5	0
south-east	5	5	0

(validation/test), according to two variants of the YOLO detector (YOLOv7 and YOLOv7-tiny), the number of training instances, and the class for evaluation (“d-or-i” is short for “direction_or_information”). The YOLOv7-tiny variant is considered to account for limited computational resources and battery consumption, in an hypothetical scenario of on-board processing. The different configurations of training instances are evaluated to test the system’s dependence on annotated supervision.

The experimental results show that the values of the metrics for YOLOv7-tiny are lower than those of the models with the use of YOLOv7. This is coherent with the fact that the YOLOv7 model has approximately 36.9 million parameters, while YOLOv7-tiny has 6.2 million parameters.

OCR results are reported in Table II, reporting an average edit distance of 2.16. In table are also reported general statistics on the set of annotations and outputs.

The results for arrow categorization are reported in Table III in terms of absolute matches, along with the number of the respective instances for each direction, showing excellent performance. Despite the direction-detailed analysis, our proposed algorithm is invariant by design, and as such direction-specific biases are to be mostly attributed to the underlying features of the tested images.

A key part of the quantitative analysis is the profiling of execution times. These were computed both for the whole system and for the individual steps of the pipeline, by feeding all 1,253 images of the MTSD test set. A preliminary warmup execution was launched so as to guarantee a consistent profiling of the execution times, avoiding the “cold start” problem. Three independent executions were performed and the average among them was reported. The recorded results are reported in Table IV both for YOLOv7 and YOLOv7-tiny, where the part referring to YOLOv7-tiny also includes the optimization of the arrow categorization procedure. All time values refer to computation on Intel Core i5-8300H 2.30GHz CPU and GeForce GTX 1050 Ti 4G GPU.

Qualitative results are provided in Figure 4, illustrating the system behavior in different setups.

Total / Per image	YOLOv7	YOLOv7-tiny
Time for the whole pipeline		
Total	857.761s	325.923s
Per image	0.684s \approx 1.5 FPS	0.260s \approx 4 FPS
Time for road signs detection		
Total	81.201s	23.282s
Per image	0.064s	0.019s
Time for arrows detection		
Total	370.787s	75.159s
Per image	0.296s	0.060s
Time for arrows direction classification		
Total	5.921s	4.054s
Per arrow	0.005s	0.004s
Time for the OCR procedure		
Total	140.503s	87.117s
Per image	0.112s	0.070s

TABLE IV: Execution time for the proposed pipeline.

IV. CONCLUSIONS

We have addressed the problem of detecting road direction signs and analyzing their content, associating textual information to directional information. The proposed pipeline is a hybrid solution that combines deep learning techniques for object detection and text recognition, with traditional image processing techniques for the identification of the directions indicated by the arrows. Extensive experiments have been conducted to evaluate all aspects of our solution: including detection accuracy, text recognition accuracy, and arrow orientation accuracy, displaying excellent results. Furthermore, we considered two solutions, defined as a function of the available hardware resources, and we evaluated their effectiveness as well as their efficiency. The eventual integration of the presented solutions into the software of a self-driving vehicle is promising, however depending on rigorous quality management testing, as well as on a significant reduction of the computational cost tailored to System-on-Chip-specific hardware optimization. As future developments, we consider integrating higher level logic for the association of textual elements to arrows, taking into account Gestalt laws for the interpretation of road signs that are ultimately designed towards fruition by human beings.

REFERENCES

- [1] Simone Bianco, Marco Buzzelli, and Raimondo Schettini, "A unifying representation for pixel-precise distance estimation," *Multimedia Tools and Applications*, vol. 78, pp. 13767–13786, 2019.
- [2] Marco Buzzelli and Luca Segantini, "Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments, and results," *Sensors*, vol. 21, no. 2, pp. 596, 2021.
- [3] Thomas Wenzel, Ta-Wei Chou, Steffen Brueggert, and Joachim Denzler, "From corners to rectangles—directional road sign detection using learned corner representations," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1039–1044.
- [4] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona, "Fast feature pyramids for object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.
- [5] Kyoungtaek Choi, Jae Kyu Suhr, and Ho Gi Jung, "Fast pre-filtering-based real time road sign detection for low-cost vehicle localization," *Sensors*, vol. 18, no. 10, pp. 3590, 2018.



Fig. 4: Qualitative results of the proposed pipeline applied to instances of the MTSD test set.

- [6] Domen Tabernik and Danijel Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 4, pp. 1427–1440, 2019.
- [7] Georg Maier, Sebastian Pangerl, and Andreas Schindler, "Real-time detection and classification of arrow markings using curve-based prototype fitting," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 442–447.
- [8] Gustavo Henrique de Oliveira, Francisco Assis da Silva, Danilo Roberto Pereira, Leandro Luiz de Almeida, Almir Olivette Artero, Alex Fernando Bonora, and Victor Hugo C de Albuquerque, "Automatic detection and recognition of text-based traffic signs from images," *IEEE Latin America Transactions*, vol. 16, no. 12, pp. 2947–2953, 2018.
- [9] VE Perlin, DB Johnson, MM Rohde, RM Lupa, G Fiorani, and S Mohammad, "Esarr: enhanced situational awareness via road sign recognition," in *Unmanned Systems Technology XII*. SPIE, 2010, vol. 7692, pp. 156–168.
- [10] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.
- [11] Wong Kin-Yiu, "yolov7," 2022.
- [12] Ke Wang, Bin Fang, Jiye Qian, Su Yang, Xin Zhou, and Jie Zhou, "Perspective transformation data augmentation for object detection," *IEEE Access*, vol. PP, pp. 1–1, 12 2019.
- [13] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 2019.
- [14] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," 2020.

- [15] Zhiwei Wei, Chenzhen Duan, Xinghao Song, Ye Tian, and Hongpeng Wang, "Amrnet: Chips augmentation in aerial images object detection," 2020.
- [16] JaidedAI, "Easyocr," <https://github.com/JaidedAI/EasyOCR>, 2020.
- [17] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoon Yun, and Hwalsuk Lee, "Character region awareness for text detection," 2019.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.
- [19] Sepp Hochreiter and Jurgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.
- [20] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," 2006.
- [21] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [22] Karl Pearson F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
- [23] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang, "The mapillary traffic sign dataset for detection and classification on a global scale," 2020.
- [24] Vladimir I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Cybernetics and Control Theory*, vol. 10, no. 8, pp. 707–710, 1966.