Distance Estimation of Fixed Objects in Driving Environments

Giorgio Leporoni[1], Valerio Ponzi[1,2], Francesco Pro[1] and Christian Napoli[1,2]

[1]*Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma, 00185, Italy*
[2]*Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19, Roma, 00185, Italy*

Abstract
Autonomous driving is a highly relevant topic today, particularly among major car manufacturers attempting to lead in technological innovation and enhance driving safety. An autonomous vehicle must possess the capability to sense its environment and navigate without human intervention. Thus, it serves as both a driver support system and, in some cases, a substitute. A crucial aspect involves identifying the positions of pedestrians, traffic signs, traffic lights, and other vehicles while computing distances from them. This enables the vehicle to emit alerts to the driver in potentially dangerous situations, such as impending obstacles due to external factors or driver distraction. In this paper, we introduce an approach for identifying traffic signs and determining the distance from them. Our method utilizes the YOLOv4 network for identification and a customized network for distance computation. This integration of AI technologies facilitates the timely detection of hazards and enables proactive alert mechanisms, thereby advancing the capabilities of autonomous vehicles and enhancing driving safety.

Keywords
Machine Learning, Deep Learning, Yolo, Autonomous Driving

1. Introduction

Road safety is a major global concern, impacting the well-being of individuals and communities worldwide. The development and adoption of advanced technologies, such as driver assistance systems and autonomous vehicles, offer significant potential to further enhance road safety in the long term. This is possible by creating systems based on cameras or sensors mounted on the vehicles that process the acquired images and can identify the typical objects of a road environment by doing some computation on them, such as looking at their distances. In this way, the vehicle could be able to make quick decisions autonomously in case of necessity. A classical example is when there is a stop signal and the system detects that the driver is not reducing the velocity, at this point it can brake autonomously the vehicle or easily alarms the driver with acoustic signals.

In the last years, attempts have begun to approach this field of research by exploiting artificial intelligence. Previous methods involved the use of geometry with the assumption of fixed dimensions for objects such as vehicles. Other methods were based on IPM (Inverse Perspective Mapping) using the lines present on the carriageway, these methods are all dependent on the parameters of the used camera.

One of the main problems in this field of research is the dataset. We are talking about a very delicate area, so to be sure of the system's accuracy the dataset should be composed of a huge number of samples representing different objects in very different contexts [1, 2, 3]. So, what we did was to record video on short routes from a dash cam mounted on our vehicle, extracting frames on which we then calculated ground truth in an automated way to finally make an ad hoc dataset for us.

In this paper, we focus on computing the distances between the vehicle and the detected traffic signs using single images captured by a monocular camera. We decided to use this type of camera because it is the most common and affordable. The method foresees two phases, one for the detection of the traffic signs on the captured images and a second phase for inferring distances from them. For this second phase we built a network based on a modern paper [4] that tries to solve the problem with a pure base artificial intelligence approach.

Our main contributions arise from our endeavor to create an automated system tailored to our needs. Initially, we integrated YOLOv4 to produce bounding boxes around traffic signs, facilitating the automatic identification of their positions within images, thus concluding the initial phase of our approach. Subsequently, we directed our efforts towards developing a specialized dataset to address our specific problem, as existing datasets did not fulfill our requirements. Building upon our initial findings, we sought to enhance our system by implementing two stabilization methods for predicted distances. The first method entails generating and utilizing depth maps

SYSYEM 2023: 9th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 3-6, 2023
leporoni.1944533@studenti.uniroma1.it (G. Leporoni); ponzi@diag.uniroma1.it (V. Ponzi); pro.1944191@studenti.uniroma1.it (F. Pro); cnapoli@diag.uniroma1.it (C. Napoli)

for each frame, enhancing the accuracy of distance measurements between signs located at the same depth. The second method capitalizes on temporal frame correlation, enhancing the smoothness and consistency of our system, and thereby augmenting its overall performance.

The use of depth maps helps us to get more accurate measurements between signs that are collocated at the same depth. Temporal frame correlation instead helps us to: Filtrate some false positive predictions keeping a bounding box if and only if it appears in the previous and the next frames and get more stable distance predictions for successive ones.

The major car manufacturers are at the forefront in this field. Taking Tesla as an example, it uses a huge amount of sensors and cameras mounted on its vehicles. This implies that the car must be produced in that way. With methods like ours, what you can do is simply mount a camera, such as a dash cam, inside the vehicle as a driving aid. Furthermore, what we have tried to do is to implement, as in the reference paper, a method that was not bound to the parameters of the camera used. For example, the IPM methods are bounded by the height of the camera from the ground, instead in this case the driver does not have to worry about the position in which the camera is mounted, which can easily be used on different vehicles. building a simple and portable system usable on any camera.

2. Related works

Inverse Perspective Mapping [5] consists of removing the perspective distortion from the road surface, taking as reference the lane lines to compute distances assuming they have a fixed size. In this method, a bird's eye view of the roadway is computed to carry out the correspondence between a pixel dimension and the lane line size. This correspondence is then used to count the pixel between an object and the vehicle getting the approximated distance. This method has problems in the presence of road curves or road signs not very visible or absent. In addition, it is very dependent on the camera parameters.

Stereo vision [6] This method foresees the use of a stereo camera that generates two images, a left and a right view. From these two images of the same environment is generated a disparity map with the use of epipolar geometry. Using a simple formula from the generated map it is possible to compute for each pixel of the 2D image the z coordinates that give us the depth of the object in that pixel in the real 3D world. The main problem with this method is the expensive cost of the stereo camera.

AI-based approach[7] This method is based on a deep learning approach to monocular images. Starting from labeled data train a neural network able to compute distances from objects bounding boxes (DisNet).

Geometry approach [8] Other papers are based on the assumption of fixed sizes for known objects, such as vehicles. In this way, knowing camera parameters can be used as a formula to compute distances [9, 10**?**].

3. Our approach

Our approach focused on the use of Italian road signs. In Italy, for each category of sign there is a most commonly used size, so once we classified the sign surveyed, we assumed that its size was the common one.

To approach the problem, we started creating our dataset from scratch. To accomplish this task, we used a dash cam mounted on our vehicle recording routes around the city to finally get more or less 3 hours of recordings. Then we filtered out all unsuitable videos, from the remaining videos we got about 1500 frames representing the roads around the city. We cut each frame on the vertical axis because of a visible portion of the vehicle interior, removing useless information.

For object detection, we needed a quick solution to avoid wasting time in the whole process. So, we chose YOLOv4 (You Only Look Once) [11] because it runs a lot faster than other methods as RCNN [12] or methods based on color segmentation [13]. We downloaded a pre-trained YOLO network on which we did transfer learning on a German Traffic Sign dataset training for 4000 iterations. During the transfer learning phase. Other attempts we made were to use some image pre-processing techniques, those in grayscale, or the images on which we used histogram equalization getting unfortunately bad results. In the end, the network reached an accuracy of about 91%.

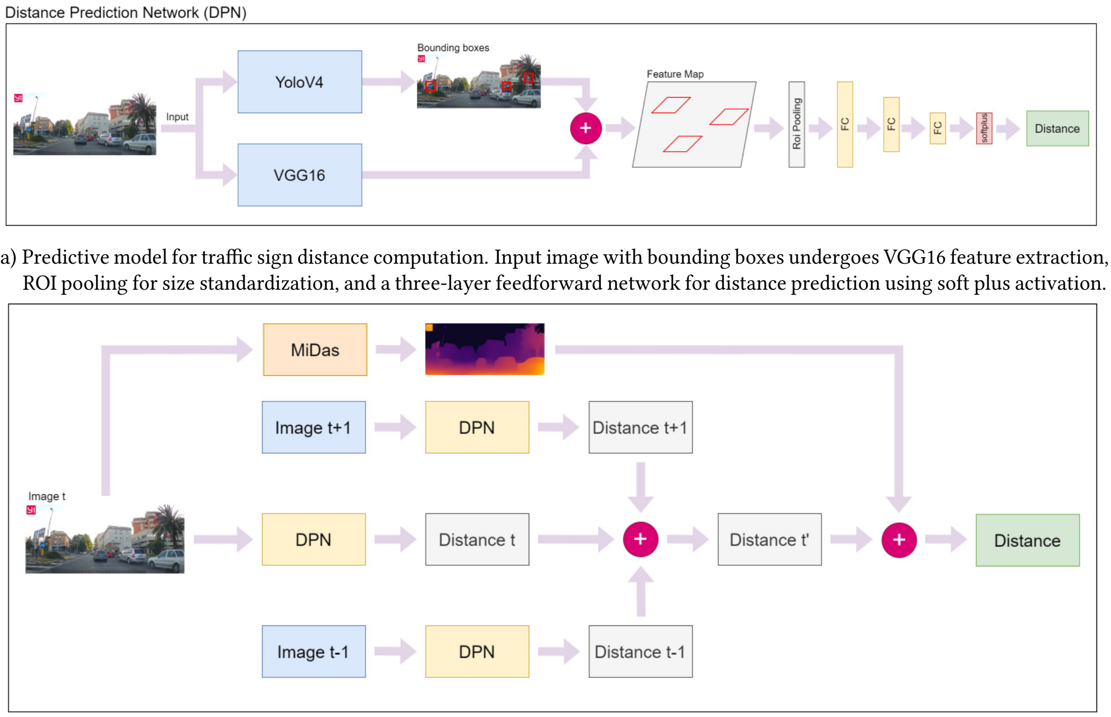
With the YOLO network, we got the bounding boxes of the traffic signs for each frame, discarding manually all the frames without detected objects or with the presence of wrong detections. To get the ground truth of each bounding box we use the following formula:

$$distance = \frac{Width_{cm} * Focal_{lenght}}{Width_{px}}$$

It is based on the focal length of the camera that we obtained by taking a picture of an object of known size placed at a known distance to count the pixels of which the object is composed within the image. This is the only parameter of the camera that was necessary to create the dataset.

In particular, the width of the triangular and octagonal signs used is 90 cm, while it is 60 cm for the square and circular ones.

Through this process, we built a dataset composed of 959 images. After the creation of the dataset, we focused



(a) Predictive model for traffic sign distance computation. Input image with bounding boxes undergoes VGG16 feature extraction, ROI pooling for size standardization, and a three-layer feedforward network for distance prediction using soft plus activation.

(b) Enhanced model integrating depth map information and temporal frame correlation for stabilized predictions. Input image with bounding boxes processed through VGG16, ROI pooling, and a modified three-layer feedforward network, leading to improved distance accuracy.

Figure 1: Schematic representations of the comprehensive distance computation system.

on the detection part. For this purpose, we used YOLOv4 as mentioned above.

After obtaining the bounding boxes for an image, it is passed to a specific network for the distance computation. This second network is composed of a CNN (VGG16) [14] for feature map extraction, and then this is combined with the information about bounding boxes passed through an ROI pooling layer [15]. This Layer is necessary because bounding boxes for a single image could be of different sizes, this layer aims to remove this difference in the dimension standardizing them. The output of the ROI pooling is finally passed to a feedforward network, composed of 3 layers (2048, 512, 1), that predicts distances using a soft plus activation function. The architecture of the network is shown in Figure 1a.

By testing the entire process on different videos, we noticed that for our cases this method was not stable in the predictions made between successive frames, in fact in some cases, it happened that there was a large variance between distances predicted for the same traffic sign in two or more successive frames. We tried to increase our results by adding the use of **depth map** information and exploiting the concept of **temporal frame correlation**.

Depth map [16]: The concept is that traffic signs at the same depth in the real world are more or less at the same distance from the vehicle. Based on this point we use a pre-trained network called MIDAS [17] [18] to get the depth map of the image under exam. Once bounding boxes are detected in the original image and distances are computed, we report the bounding boxes in the depth map image. For each traffic sign at the same depth, considering a small variance based on the maximum depth value inside the image, we computed an average of the distances in the original image to obtain a uniform value. At the moment, we used this method after the computation of the distances, but it could be used also in the creation of the dataset to get more detailed labels or in the training phase to directly stabilize results in the network.

Figure 2 shows a representation of this method, looking at the traffic signs in the image are now visible from the depth map coloration that they are at the same dis-

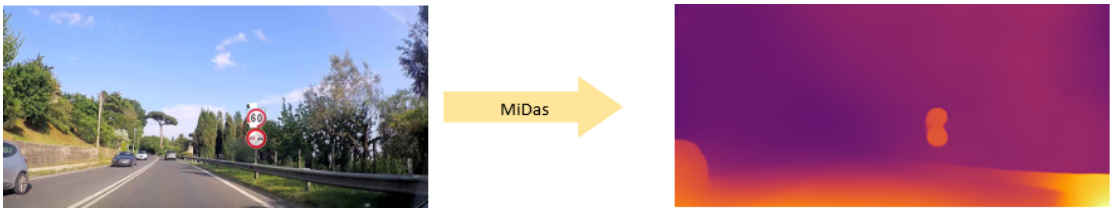


Figure 2: Example of depth map using MiDas network.

tance. So, thanks to this now the prediction for them is corrected at the same value.

Temporal frame correlation: We use this technique to give a linearity in predicting distances for the sequence of frames. Going through this method, we noticed that in some cases the network's predictions were much different for successive frames. To stabilize predictions, we thought that given a traffic sign in a frame at time t, if it is also present at time t-1 and t+1 it is a valid object to consider for time t and its distance is the average between the 3 frames in sequence. To verify if the same traffic sign is present in the 3 subsequence frames, we first find the center of its bounding box at time t and of all the traffic signs for the previous and forward frames. Then we compute the distances between points and if it is lower than a certain threshold, we are looking at the same traffic sign.

An example of this concept is given in Figure 3, in which there is a wrong detection at frame t (red circle on the top right image) and since this wrong prediction is not present at frame (t-1) and (t+1), it is also discarded at frame t.

The architecture of this modified network is represented in Figure 1b.

4. Training

About the training phase, due to time and resource issues, we were unable to train the networks for long sessions. We trained the YOLOv4 for about 4000 iterations using RGB frames from the German Traffic Sign Dataset. For the distance prediction network (DPN), all components composing the DPN network are trained together. We trained it with our dataset for 560 epochs using RGB frames. About the training parameters, we used a learning rate starting from 0,001 with ADAMS, minibatch size of 16, and loss the $Smooth_{L1}$.

5. Results

Talking about the detection part with the YOLO we reach an accuracy of around 91%.

For the distance prediction network instead, it is not possible to compute a true accuracy, but we reach a loss of more or less 130, visible in the graph in Figure 4.

It shows that the loss function has a trend that tends to improve if trained for more epochs.

As an evaluation metric, we used the ones provided by [7]. In particular, we use the RMSE on predictions divided by meters:

$$RMSE = \sqrt{\frac{1}{N}\sum_{d=1}^{n} \|d_i - d_i^*\|^2}$$

This is to see how the behavior of the network changes concerning the distance from the detected object. Results are represented in the graph in Figure 5, compared with the ones obtained by the reference paper. Visible predictions get worse as distances increase. We notice that bounding boxes of traffic signs at higher distances do not match perfectly their dimensions introducing an error. Another source of error is probably the fact that we have only a few samples of road signs at large distances.

Table 1 compares our results with the ones of the reference paper. As visible, results are similar, ours are a little bit better because lower values represent better predictions. This is because we make predictions only on traffic signs while they predict on cars, cyclists, and pedestrians, this means that they have a larger margin of error than us.

To show the method in action, we made some test video, available on YouTube, of the network works. In particular, we made videos with the following characteristics:

- Test video using the base network without depth map and temporal frame correlation (daylight conditions)
- Test video using depth map and temporal frame correlation (daylight conditions)
- Test video using the base network without depth map and temporal frame correlation, rounded on 5 meters (daylight conditions)
- Test video using depth map and temporal frame correlation, rounded on 5 meters (daylight conditions)

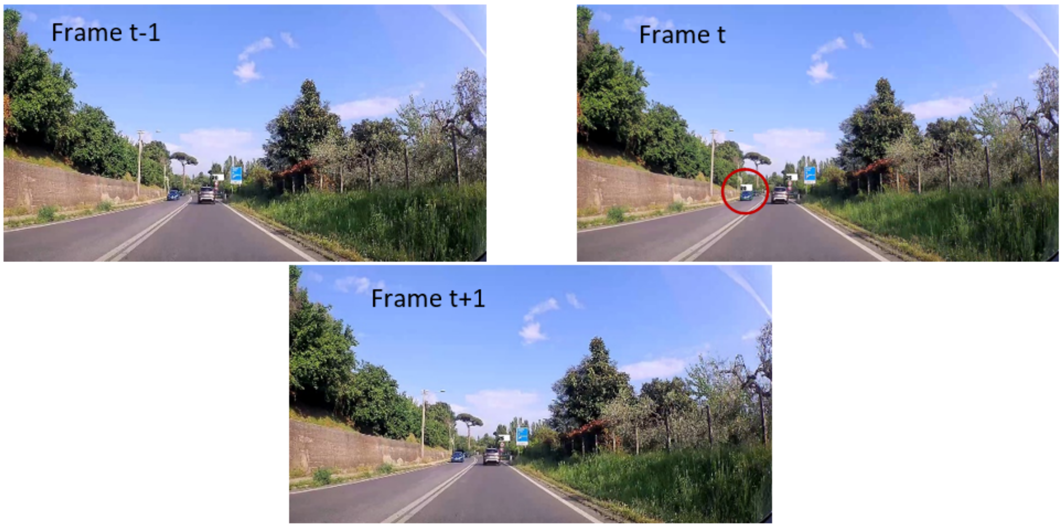


Figure 3: Example of temporal frame correlation in case of wrong predictions.

Table 1
Comparison of results between our implementation and the one of the paper we take as reference.

Method	Abs Rel	Squa Rel	RMSE	RMSE(log)
Our base model	0.131	0.468	3.126	0.173
Paper base model	0.251	1.844	6.870	0.314

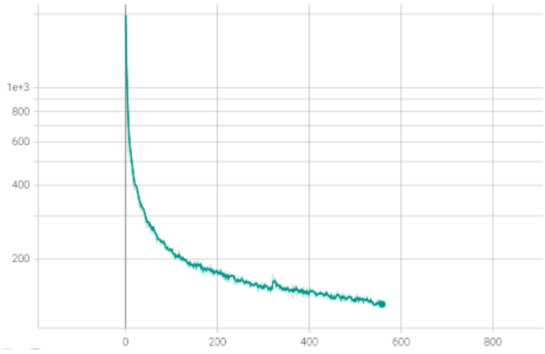


Figure 4: Graph of the loss function of the distances prediction network.

- Test video using depth map and temporal frame correlation, rounded on 5 meters (night conditions)

Rounded on 5 meters, means that we do an approximation on the predictions made to get more stable results. As, 12.4 meters is rounded to 10 meters, while 12.6 meters is rounded to 15 meters.

The formulas used in the table are the following:

$$RMSE(log) = \sqrt{\frac{1}{N}\sum_{d=1}^{n} \|log(d_i) - log(d_i^*)\|^2}$$

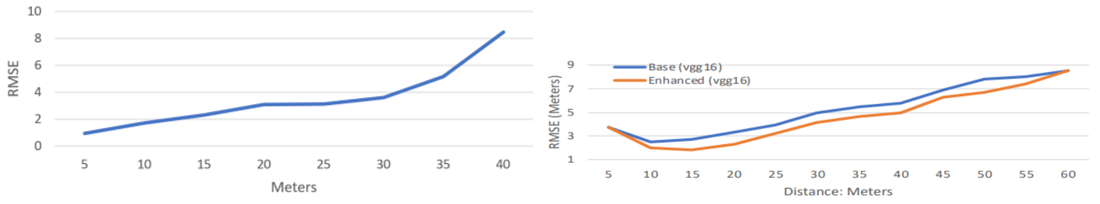
$$AbsRelativeDifference = \frac{1}{N}\sum_{d=1}^{n} \frac{|d_i - d_i^*|}{d^*}$$

$$SquaredRelativeDifference = \frac{1}{N}\sum_{d=1}^{n} \frac{\|d_i - d_i^*\|^2}{d^*}$$

Figure 6 shows two examples of predictions in images. The top image distances are predicted without the use of the depth map and temporal frame correlation, as the predictions do not seem reliable, they appear quite random. The bottom image instead, is done using our two variations. As visible all the detected signs are more or less at the same depth, this is not considered for the top image, while in this case thanks to the depth map their predictions are adjusted correctly.

6. Conclusion

The method seems to work well, there are errors introduced by the labels of our dataset that are not accurate,



(a) Our meters-RMSE predictions graph

(b) Reference paper meters-RMSE predictions graph

Figure 5: These are the graphs that are put in relation to the predictions at certain meters with the distance error from the true values.

caused by the possible different dimensions for each traffic sign on the road introducing a small error that then will propagate throughout the process, even if we tried to solve it using depth map and temporal frame correlation. So, the main future step could be using more accurate labels for the samples inside the dataset. The work is based on the objects detected and rounded by bounding boxes but is not always sure that their dimensions match perfectly the sizes of the traffic signs, so this point introduces errors in the predictions of the network. As said at the beginning, in Italy the same traffic signs could be used up to 3 different dimensions, so it could be useful to infer their dimensions to improve the predicted distances. As future improvement, there possible extension of the detected objects also to vehicles and pedestrians.

Acknowledgments

This work has been developed at is.Lab() Intelligent Systems Laboratory at the Department of Computer, Control, and Management Engineering, Sapienza University of Rome (https:// islab.diag.uniroma1.it). The work has also been partially supported from Italian Ministerial grant PRIN 2022 "ISIDE: Intelligent Systems for Infrastructural Diagnosis in smart-concretE", n. 2022S88WAY - CUP B53D2301318, and by the Age-It: Ageing Well in an ageing society project, task 9.4.1 work package 4 spoke 9, within topic 8 extended partnership 8, under the National Recovery and Resilience Plan (PNRR), Mission 4 Component 2 Investment 1.3—Call for tender No. 1557 of 11/10/2022 of Italian Ministry of University and Research funded by the European Union—NextGenerationEU, CUP B53C22004090006.

References

[1] V. Ponzi, S. Russo, A. Wajda, R. Brociek, C. Napoli, Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models, volume 3360, 2022, pp. 55 – 63.

[2] G. Capizzi, C. Napoli, L. Paternò, An innovative hybrid neuro-wavelet method for reconstruction of missing data in astronomical photometric surveys 7267 LNAI (2012) 21 – 29. doi:10.1007/978-3-642-29347-4_3.

[3] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[4] J. Zhu, Y. Fang, H. Abu-Haimed, K.-C. Lien, D. Fu, J. Gu, Learning Object-specific Distance from a Monocular Image, Technical Report, 2019. URL: http://arxiv.org/abs/1909.04182. doi:10.48550/arXiv.1909.04182, arXiv:1909.04182 [cs] type: article.

[5] S. Tuohy, D. O'Cualain, E. Jones, M. Glavin, Distance determination for an automobile environment using Inverse Perspective Mapping in OpenCV, in: IET Irish Signals and Systems Conference (ISSC 2010), 2010, pp. 100–105. doi:10.1049/cp.2010.0495.

[6] X. Sun, Y. Jiang, Y. Ji, W. Fu, S. Yan, Q. Chen, B. Yu, X. Gan, Distance Measurement System Based on Binocular Stereo Vision, IOP Conference Series: Earth and Environmental Science 252 (2019) 052051. URL: https://doi.org/10.1088/1755-1315/252/5/052051. doi:10.1088/1755-1315/252/5/052051.

[7] DisNet: A novel method for distance estimation from monocular camera, ???? URL: https://patrick-llgc.github.io/Learning-Deep-Learning/paper_notes/disnet.html.

[8] S. Saleh, S. Khwandah, A. Heller, W. Hardt, A. Mumtaz, Traffic Signs Recognition and Distance Estimation using a Monocular Camera, 2019.

[9] S. Russo, C. Napoli, A comprehensive solution for psychological treatment and therapeutic path planning based on knowledge base and expertise sharing, volume 2472, 2019, pp. 41 – 47.

[10] G. Lo Sciuto, S. Russo, C. Napoli, A cloud-based flexible solution for psychometric tests validation,

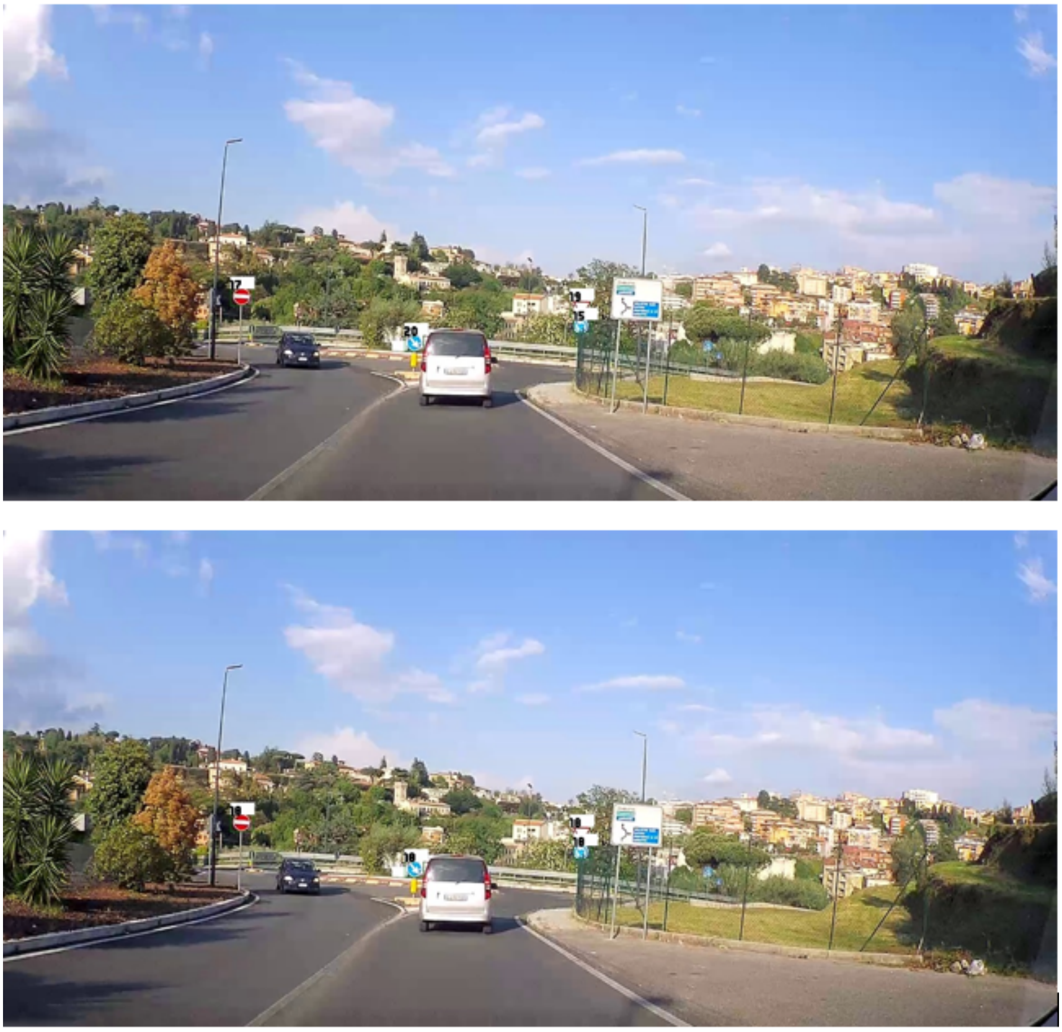


Figure 6: (Top image) example of predictions without depth map and frame correlation time. (Bottom image) example of predictions using depth map and frame correlation time

administration and evaluation, volume 2468, 2019, pp. 16 – 21.

[11] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, Technical Report, 2020. URL: http://arxiv.org/abs/2004.10934. doi:10.48550/arXiv.2004.10934, arXiv:2004.10934 [cs, eess] type: article.

[12] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169, iSSN: 2380-7504.

[13] A. Youssef, D. Albani, D. Nardi, D. Bloisi, Fast Traffic Sign Recognition Using Color Segmentation and Deep Convolutional Networks, volume 10016, 2016. doi:10.1007/978-3-319-48680-2_19.

[14] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Technical Report, 2015. URL: http://arxiv.org/abs/1409.1556. doi:10.48550/arXiv.1409.1556, arXiv:1409.1556 [cs] type: article.

[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Technical Report, 2014. URL: http://arxiv.org/abs/1311.2524. doi:10.48550/arXiv.1311.2524, arXiv:1311.2524 [cs] type: article.

[16] C. Godard, O. Mac Aodha, M. Firman, G. Brostow, Digging Into Self-Supervised Monocular Depth Es-

timation, Technical Report, 2019. URL: http://arxiv.org/abs/1806.01260. doi:10.48550/arXiv.1806.01260, arXiv:1806.01260 [cs, stat] type: article.

[17] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision Transformers for Dense Prediction, Technical Report, 2021. URL: http://arxiv.org/abs/2103.13413. doi:10.48550/arXiv.2103.13413, arXiv:2103.13413 [cs] type: article.

[18] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, V. Koltun, Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer, Technical Report, 2020. URL: http://arxiv.org/abs/1907.01341. doi:10.48550/arXiv.1907.01341, arXiv:1907.01341 [cs] type: article.