Enhanced Approach for On-road Obstacles Detection using Level-Set-YOLOv3 Combination

1st Djazila Souhila KORTI
Department of Telecommunication
Belhadj Bouchaib University
Ain-Temouchent, Algeria
souhilakorti@gmail.com

2nd Foued DERRAZ
Department of Telecommunication
Abou Bekr Belkaid University
Tlemcen, Algeria
fdrz70@gmail.com

3rd Zohra SLIMANE
Department of Telecommunication
Belhadj Bouchaib University
Ain-Temouchent, Algeria
zoh_slimani@yahoo.fr

4th Kheira LAKHDARI
Department of Telecommunication
Abou Bekr Belkaid University
Tlemcen, Algeria
kblakhdari@gmail.com

***Abstract*—Automatic detection and classification of on-road obstacles is an integral part of Intelligent Transport Systems (ITS). It increases road safety by providing valuable information about the environment where the vehicle navigates using on-board sensors, and can intervene in risky situations. However, it is a difficult task to achieve due to multiple variability conditions presenting in the real urban scenes. These latter can perform partial occlusion, multi-view angles, multi-scale objects, different lighting conditions, etc. In order to solve these problems, we describes in this paper a system for automatic detection and classification of on-road obstacles. The system is achieved by combining the Level-Set segmentation technique and the YOLOv3 algorithm that uses Darknet-53. The YOLOv3 used in our work was pre-trained with COCO dataset, on 80 categories of common objects. We started by fine tuning the model, to detect three types of on-road obstacles: Car, Person, and Bycicle, with KITTI dataset, which contains scenes from public roads. Then we used the segmented RGB images from KITTI, for training and testing the performance of the model. The proposed model with combination of Level-Set and YOLOv3 is compared to a simple YOLOv3 model without any preprocessing. The results demonstrate the effectiveness of the model with combination in detecting small size, overlapping objects with high precision of about 87%.**

***Index Terms*—ITS, on-road obstacles, object detection, object classification, Level-Set, YOLOv3.**

I.I NTRODUCTION

With the increasing number of accidents and fatalities on the roads each year, the development of Intelligent Transport Systems (ITS) has received a great deal of attention [1][2][3][4]. ITS should maintain a driver's safety and comfort during the navigation to the required destination, by integrating different systems that can react to any external environmental changes and make automatic decisions. However, a detection and recognition system of on-road obstacles such as pedestrian, cyclist, roadsides, traffic sign, etc. is an essential element for the development of ITS[5][6]. Such a system will enable providing real time information about the external environment where the vehicle navigates from on-board sensors, and intervene in risky situations. Several types of sensors have been used, the most known are RADAR[7], LiDAR[8], and cameras[9]. With the high complexity and dynamism of the external environment, many issues are encountered and must be overcome[10]. For example:

- Road obstacles are subject to partial occlusion.
- Road obstacles have many representations in the geometric and textual sense, which make their identification difficult.
- Road obstacles can be detected from different angles and at different distances, which usually causes confusion for the recognition system.

To handle these issues, several methods were proposed using deep learning (DL), as a powerful technique for extracting and learning feature representations automatically from data. Prabhakar *et al.*[10] used Faster R-CNN for the detection and classification of road obstacles. The results show that the proposed system is invariant to obstacle's shape and view. Guan *et al.*[11] Proposed a real time decision fusion framework for the detection and recognition of vehicles. LiDAR's and camera's data were fused and fed to the You Only Look Once version 3 (YOLOv3) for training. The results shows that the system achieves high accuracy precision for both day and night driving scenes. Bouti *et al.*[12] proposed a two-stage method for the detection and recognition of road signs captured by a camera. For the first stage, they used Histogram of Oriented Gradient (HOG) and Support Vector machine (SVM) for the detection part. The second stage consist of using a modified LeNet model for the classification part. Zhou *et al.*[13] Proposed to use a set of detectors that can capture partial occlusion patterns from different body parts.

The results shows that training jointly these part detectors can improve the performance for detecting occluded pedestrians. However, the final decision is still made by integrating multiple part scores, which makes the whole procedure more complex and hard to train. Malbog *et al.*[14] applied a Mask Region-Based CNN and instance segmentation to detect pedestrian's crosswalk. The model was trained using image data gathered from camera, and an accuracy of 97% was achieved. Heng *et al.*[15] Proposed a real time segmentation method to road signs recognition by using You Only Look at Coefficient (YOLACT). However, YOLACT performance is deteriorating for different situations such as lighting, weather, and different angle on the traffic signs. Chang *et al.*[16] Proposed to use a simple segmentation algorithm and a CNN model to classify occluded vehicles. For the occlusions with more than two vehicles, the performance of the proposed method deteriorates in terms of precision.

Aiming at the above problems, we used a two-stage method for the detection and the classification of on-road obstacles. We first realized a segmentation using level-set technique. Then the segmented data was fed to the pre-trained neural network YOLOV3, which will predict for the segmented objects their corresponding labels using rectangular bounding boxes. The proposed method is able to overcome the topology problems, allowing the detection of non-rigid objects such as human with different postures, occluded, and different scales objects.

The following contents are arranged as follows: section II describes the entire process of the proposed on-road obstacles detection and classification system. Experimental results to assess the performance of the proposed system are presented in Section III. Section IV concludes the paper and present an outlook on further possible future works and improvements.

II. METHODOLOGY

A. System Design

The system flow chart for our detection and classification of on-road obstacles, is depicted in Fig. 1. Our system uses a pre-trained YOLOv3 with COCO (Common Objects in Context) dataset to initialize the weights. We have realized two experiences. For the first one, all layer weights were trained with KITTI dataset, to tune the model to detect only three object classes. It includes Car, Person, and Bicycle. For the second one, a segmentation using the Level-Set technique was first realized, to identify every pixel, belonging to each object. The segmentation step allowed us to overcome the overlapping and confusion issue and detect each object distinctly. Finaly, the model was trained and tested on these segmented data.

B. Datasets

- COCO dataset : COCO, is a large-scale object detection, segmentation, and captioning dataset[18]. COCO includes images of complex everyday scenes. Therefore, it is based on 200.000 images, and 330.000 labeled images of 80 object categories.
- KITTI dataset: KITTI dataset[19], contains 7481 finely annotated images of driving scenes, with 80 object classes. Among these classes, only frequent objects (car, truck, pedestrian, cyclist, bicycles and seated people) are labeled independently, all other classes are labeled as "Others" or "Don't Care".

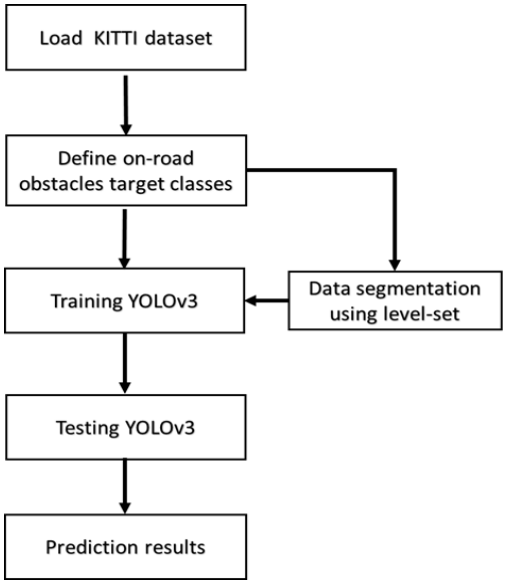

Fig. 1. System flow chart.

C. Level-Set method

The Level-Set method was proposed by Osher *et al.*[20]. It is a powerful numerical technique for image segmentation and analysis. The basic idea is to represent a curve $\boldsymbol{C}$ as the zero level of a higher dimensional level set function (LSF) $\phi(x, y)$ as mentioned in (1).

$$C = \{x \in \Omega | \phi(x, y) = 0\} \tag{1}$$

With the curve as the boundary, the whole surface can be divided into an internal region and an external region of the curve. Define a Signed Distance Function (SDF) on the surface as shown in (2).

$$\phi(x, y) = \pm dist(x, C) \tag{2}$$

Where, the value of ***dist*** is the shortest distance between the point of x on the surface and the curve.ϕ is positive on the inside and negative on the outside of the curve.

The curve evolution can then be formally defined as (3)

$$\frac{\partial C(t)}{\partial t} = V\overrightarrow{N} \tag{3}$$

This can also be expressed by the evolution of the level set function $\phi(x, y)$ as shown in (4).

$$\frac{\partial \phi}{\partial t} = V|\nabla_{\phi}| \tag{4}$$

Where $\overrightarrow{N}$ is the unit vector in the inward normal direction of the curve, and V the speed function that controls the motion of the curve along its normal direction.

The evolution of the LSF is performed iteratively. For $t \in \{0, 1, ..., T-1\}$, the T-step iterative update is represented in (5).

$$\phi_{t+1}(x,y) = \phi_t(x,y) + \Delta t \frac{\partial \phi_t}{\partial t} \quad (5)$$

Where Δt is the time step, $\frac{\partial \phi_t}{\partial t}$ is the update term, $\phi_0(x,y)$ is the initial LSF, and $\phi_T(x,y)$ is the corresponding output after T evolution steps.

D. YOLOv3 architecture

YOLOv3[21], is an improved version of YOLO[22], and YOLOv2[23]. The defining feature of this network is its ability to combine three phases in one-step: (i) Object detection, (ii) Classification, and (iii) Localization.

Compared to previous versions, YOLOv3 is a multi-label classification, and a multi-scale detection tool, based on the principle of regression instead of classification. The whole detection process is performed in a single evaluation of the image, which make the model very fast. The network structure of YOLOv3 consists of two main blocs, a deeper and robust feature extractor, called Darknet-53, and a multi-scale detector.

- Darknet53[21]: Darknet53 network is formed by 53 convolutional layers for features extraction and dimension reduction as shown in Fig.2. The architecture was improved by adding five residual blocs. These blocs are built of shortcut connections that are necessary for the gradient descent. The feature extraction process uses the Feature Pyramid Network (FPN) mechanism[24] to extract features maps at three different scale.
- Multi-scale detector: is a convolutional network, which uses the extracted multi-scale feature maps. Each map is divided into grids, and different bounding boxes are predict for each grid. For each box YOLOv3 predicts four attributes: Width (w), Height (h) and Cartesian position (x and y) of the box inside the image. A score confidence $\boldsymbol{c}$ indicates whether this box contains an object, and a class probability $\boldsymbol{p}$ to indicate to which class this box belongs. If $\boldsymbol{c}<\alpha$, where α is a given confidence threshold, then the bounding boxes with the least confidence score are removed. On the other hand, if $\boldsymbol{c} \geq \alpha$, then the box is associated with the object type. Finally the Non-Maximum Suppression (NMS) algorithm[25], is used to eliminate the overlapping boxes for the same object, and only the box with the high score confidence is kept.

E. Implementation

- System specifications :
  The algorithm used for our obstacle detection and classification system was written using Python. The model training and testing were implemented on a computer with the following specifications: Ubuntu 18.04, i7-7700 64-bit, 16 GB RAM, 256 GO SSD. The implementation extensively makes use of the libraries: OpenCV[26], Pytorch[27] and TensorFlow[28]. In our implementation, the dataset labels were modified so it includes only three classes, Car, Person, and Bicycle. KITTI labeling format was formatted, so it matches with YOLOv3 as following: [<object-class><x><y><w><h>], where a class takes value between 0 to 2, which means Car, Person, Bicycle, respectively.
  The experimentations were executed using two model's architectures: the YOLOv3 model on its own and also combined with a first step of segmentation using the Level-Set method.

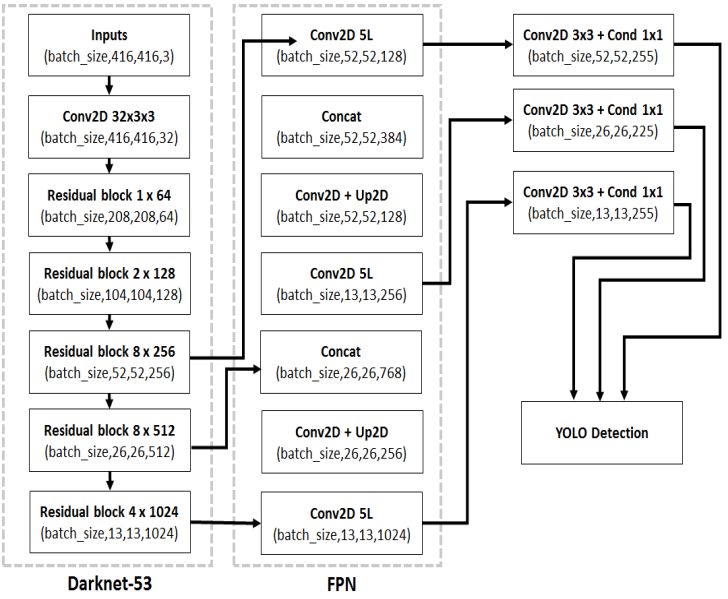

Fig. 2. YOLOv3 architecture.

- YOLOv3 model :
  The pre-trained weights of YOLOv3 with COCO dataset were uploaded as initial weights for this first model (Fig. 3). We adapted the weights of this pre-trained model to our tree-class detection problem, by training all the layers using KITTI dataset. The model receives inputs images of 416×416 pixels (RGB images), where each one is divided to 13×13 cells. 9 bounding boxes are generated from each cell. The score predicted for each box takes a value from 0 to 1, which indicates whether the box contains an object or not. The learning rate was chosen to be 0.001. The momentum and weight decay were set to 0.9 and 0.0005, respectively.

- Level-Set-YOLOv3 model: For this second model (Fig. 4), the input images were first segmented using the Level-Set method. We started by setting an initial contour on the different objects of interest automatically. Next, we have initialized the Level-Set function to the signed distance function of the initial contour. Finally we used T=10 iterations, to update the contour until convergence. The training and the testing process remain the same as the first experience.

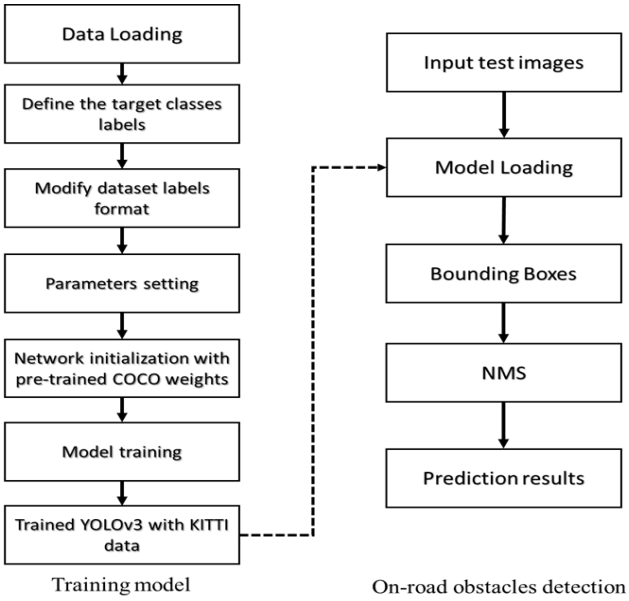

Fig. 3. The bloc diagram of the YOLOv3 model.

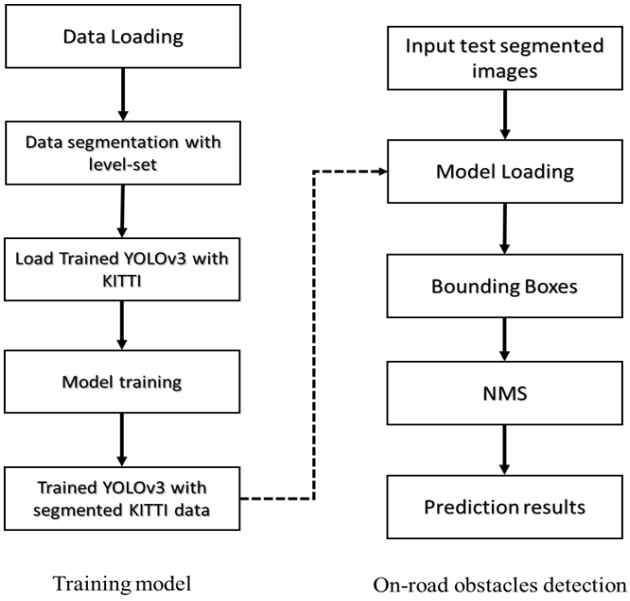

Fig. 4. The bloc diagram of the Levelset-YOLOv3 model.

III. RESULTS

The system was tested using 300 images from KITTI test dataset. The scoring threshold for classification was set 0.5, and the NMS for the detection boxes overlapping was set 0.5. To verify the effectiveness of the conducted experiments on the trained YOLOv3, Precision, is used as evaluation parameter as shown in (6).

$$Precision = \frac{T_p}{T_p + F_p} \tag{6}$$

Where, T_p and F_p refers to True Positive (correct detections), and False Positive (incorrect detections), respectively. For the first experience, the model achieves a precision of 68.01% with a detection speed of 30 seconds per frame (CPU computation time). One of the limitations observed during the experiments is the low accuracy for the detection of low contrast objects when only the YOLOv3 is used. The system failed to recognize the distant persons, with different lighting conditions.

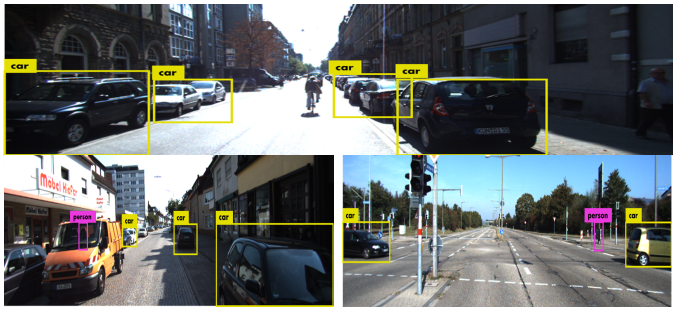

Fig. 5. The bloc diagram of the Levelset-YOLOv3 model.

With the combination of Level-Set with YOLOv3, a precision of 87% was achieved, with a detection speed of about 35 seconds per image (CPU computing time). The results show the ability of the Level-set to adapt to different topologies with no requirement of any knowledge of their shapes. This advantage allowed the model to recognize person's with different postures at different scales. The model was able to detect distinctly the overlapping objects with accuracy.

Despite its simplicity, the results show the feasibility of combining a fast YOLO detector with a segmentation technique, in order to enhance the model's performance to detect small, overlapping objects in order to handle complicated recognition tasks.

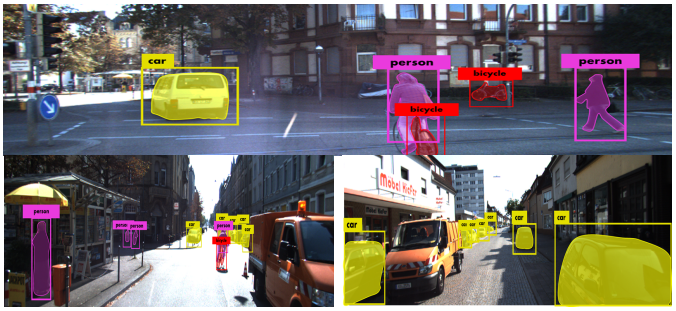

Fig. 6. Prediction results with Level-Set-YOLOv3 on testing images.

IV. CONCLUSION

Due to the increasing numbers of road accidents, the development of intelligent systems is required for making driving safer and more reliable. Facing the complex traffic scenes, such as occluhaved, different lighting conditions, multi-view angles and far distances, a vision-based system for on-road obstacles detection and classification was implemented in this paper. We have realized two experiences. For the first one, the pre-trained YOLOv3 model with COCO dataset was fine

tuned to detect only three object classes with KITTI dataset. However, the results have shown that the algorithm produced more false negatives in small objects detection. To enhance the model's performance we have realized a second experience, during which we have added a segmentation step. The data were first segmented using the Level-Set method, then fed to the tuned YOLOv3 model for training and testing. The Level-Set method showed its ability to represent contours of complex topologies, and handle topological changes, such as persons with different postures.

Despite its simplicity, the experimental results have shown that the proposed Level-Set-YOLOv3 combination has achieved substantial performance improvement, which makes the model able to detect small, overlapping objects with high precision. As perspectives, we propose to integrate the segmentation part into the YOLOv3 model directly. Which means train the model to do both segmentation and prediction simultaneously, with a wider number of target classes. We also propose to create our own dataset, run the model on a GPU, and test its performance for real-life applications.

REFERENCES

[1] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A survey of autonomous vehicles: Enabling communication technologies and challenges," Sensors (Switzerland), vol. 21, no. 3, pp. 1–33, 2021, doi: 10.3390/s21030706.

[2] V. Sharma, L. Kumar, and S. Sergeyev, Recent Developments and Challenges in Intelligent Transportation Systems (ITS)—A Survey. Springer Singapore, 2021.

[3] S. Telang, A. Chel, A. Nemade, and G. Kaushik, Intelligent transport system for a smart city, vol. 308. Springer International Publishing, 2021.

[4] T. K. Chan and C. S. Chin, "Review of autonomous intelligent vehicles for urban driving and parking," Electron., vol. 10, no. 9, 2021, doi: 10.3390/electronics10091021.

[5] D. Gangwani and P. Gangwani, Applications of Machine Learning and Artificial Intelligence in Intelligent Transportation System: A Review, no. January. Springer Singapore, 2021.

[6] L. S. Iyer, "AI enabled applications towards intelligent transportation," Transp. Eng., vol. 5, p. 100083, 2021, doi: 10.1016/j.treng.2021.100083.

[7] A. Palffy, J. Dong, J. F. P. Kooij, and D. M. Gavrila, "CNN Based Road User Detection Using the 3D Radar Cube," IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 1263–1270, 2020, doi: 10.1109/LRA.2020.2967272

[8] V. Ponnaganti, M. Moh, and T. S. Moh, "Utilizing CNNs for Object Detection with LiDAR Data for Autonomous Driving," Proc. 2021 15th Int. Conf. Ubiquitous Inf. Manag. Commun. IMCOM 2021, 2021, doi: 10.1109/IMCOM51814.2021.9377361.

[9] R. D. Brehar, M. P. Muresan, T. Marita, C. C. Vancea, M. Negru, and S. Nedevschi, "Pedestrian Street-Cross Action Recognition in Monocular Far Infrared Sequences," IEEE Access, vol. 9, pp. 74302–74324, 2021, doi: 10.1109/ACCESS.2021.3080822.

[10] X. Yu and M. Marinov, "A study on recent developments and issues with obstacle detection systems for automated vehicles," Sustain., vol. 12, no. 8, 2020, doi: 10.3390/SU12083281.

[11] G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," TENSYMP 2017 - IEEE Int. Symp. Technol. Smart Cities, pp. 3–8, 2017, doi: 10.1109/TENCONSpring.2017.8069972.

[12] L. Guan, Y. Chen, G. Wang, and X. Lei, "Real-time vehicle detection framework based on the fusion of lidar and camera," Electron., vol. 9, no. 3, 2020, doi: 10.3390/electronics9030451.

[13] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi, "A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network," Soft Comput., vol. 24, no. 9, pp. 6721–6733, 2020, doi: 10.1007/s00500-019-04307-6.

[14] C. Zhou and J. Yuan, "Multi-label Learning of Part Detectors for Heavily Occluded Pedestrian Detection," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-October, pp. 3506–3515, 2017, doi: 10.1109/ICCV.2017.377.

[15] M. A. Malbog, "MASK R-CNN for Pedestrian Crosswalk Detection and Instance Segmentation," ICETAS 2019 - 2019 6th IEEE Int. Conf. Eng. Technol. Appl. Sci., pp. 2–6, 2019, doi: 10.1109/ICETAS48360.2019.9117217.

[16] S. S. Heng, A. U. bin Shamsudin, and T. M. M. Said Mohamed, "Road Sign Instance Segmentation By Using YOLACT For Semi-Autonomous Vehicle In Malaysia," pp. 406–410, 2021, doi: 10.1109/icce50029.2021.9467206.

[17] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Vision-based occlusion handling and vehicle classification for traffic surveillance systems," IEEE Intell. Transp. Syst. Mag., vol. 10, no. 2, pp. 80–92, 2018, doi: 10.1109/MITS.2018.2806619.

[18] G. T. U. A. Colleges et al., "Microsoft COCO," Eccv, no. June, pp. 740–755, 2014.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," Int. J. Rob. Res., vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.

[20] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," J. Comput. Phys., vol. 79, no. 1, pp. 12–49, 1988, doi: 10.1016/0021-9991(88)90002-2.

[21] J. Redmon and A. Farhadi, "YOLO v.3," Tech Rep., pp. 1–6, 2018, [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[23] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[24] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen, "Weighted feature pyramid networks for object detection," Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019, pp. 1500–1504, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.

[25] N. O. Salscheider, "Featurenms: Non-maximum suppression by learning feature embeddings," Proc. - Int. Conf. Pattern Recognit., pp. 7848–7854, 2020, doi: 10.1109/ICPR48806.2021.9412930.

[26] https://opencv.org/about/

[27] https://pytorch.org/

[28] https://www.tensorflow.org/