Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth

Jamshidi Seikavandi, Meisam; Nasrollahi, Kamal; Moeslund, Thomas B.

Published in:
International Conference on Machine Vision

DOI (link to publication from Publisher):
10.1117/12.2586908

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth

Meisam Jamshidi Seikavandi[1], Kamal Nasrolahi [2,3], Thomas B. Moeslund[2]

[1]Khaje Nasir University of Technology, Tehran, Iran;
[2]Visual Analysis of People (VAP) Laboratory, Aalborg University, Aalborg, Denmark;
[3]Research Department of Milestone Systems A/S, Copenhagen, Denmark

ABSTRACT

Recent advances have shown sensor-fusion's vital role in accurate detection, especially for advanced driver assistance systems. We introduce a novel procedure for depth upsampling and sensor-fusion that together lead to an improved detection performance, compared to state-of-the-art results for detecting cars. Upsampling is generally based on combining data from an image to compensate for the low resolution of a LiDAR (Light Detector and Ranging). This paper, on the other hand, presents a framework to obtain dense depth map solely from a single LiDAR point cloud that makes it possible to use just one deep network for both LiDAR and image modalities. The produced full-depth map is added to the grayscale version of the image to produce a two-channel input for a deep neural network. The simple preprocessing structure is efficiently competent in filing cars' shapes, which helps the fusion framework to outperforms the state-of-the-art on the KITTI object detection for the Car class. Additionally, the combination of depth and image makes it easier for the network to discriminate highly occluded and truncated vehicles.

Keywords: Sensor Fusion, Deep Learning, Object Detection, Autonomous Driving, Multimodal Fusion, Depth Perception, LiDAR

1. INTRODUCTION

Nowadays, accurate object detection is a crucial problem in computer vision for applications such as autonomous driving, obstacle detection, and path planning. Although researchers have studied it for many years, it is still hard to reach a reliable and stable object detection result in general road scenarios considering the variation of road scenes. Besides, the complex overlapping situations can also bring difficulties to this task.

Recent progress in convolutional neural networks (CNN) for object detection has made these networks an integral part of many computer vision systems. There is a wide variety of structures from simple structures like Alexnet [1] and VGG16 [2] to complex networks like YOLOs [3–5] and RCNN [1]. CNNs are producing state-of-the-art object detection results on both camera data [6] as well as LiDAR point cloud data [7].

Many types of object detection methods have been proposed based on different sensors, e.g., monocular camera, stereo camera, and 3D LiDAR, respectively. The most widely used one is the camera for its low price, with the ability to obtain rich color information. Many recent end-to-end deep learning-based techniques using single-camera solutions achieve state-of-the-art performance, but they are challenged in occluded scenes. 3D LiDAR sensor can, however, help to handle occlusion scenes, but it only provides a sparse point cloud and can be valid only within a certain distance [8].

To compensate for the weakness of different (camera and LiDAR-based) methods, fusion schemes are devised to utilize the advantages of each method. There are different types of data fusion, depending on the types of sensors, data formats, and the level of fusion (abstraction, data/feature, and decision level fusion).

The issue with LiDAR point clouds is that they are sparse, and hence not necessarily good enough for applications like depth perception and multi-modal object detection. On the other hand, for LiDAR and camera fusion-based methods, upsampling LiDAR sparse data make it easier to fuse both modalities. An upsampled depth map can be treated as an image, and this brings an opportunity to use many pre-trained CNN networks as well. Some upsampling works have only used LiDAR date [9, 10], while others like [11, 12] have used images in guided approaches. Although guided structures, which use image information to fill the sparse depth, look better in appearance, we prefer to upsample LiDAR depth just using LiDAR data.

In this paper, we propose a LiDAR-camera fusion framework to improve object detection results, where we mainly exploit the full-depth of LiDAR's point cloud with introducing a novel upsampling scheme. Our contributions of this paper lie in the following three parts; First, we propose a new algorithm for filling the sparse depth map just using LiDAR projected point clouds. Then, with concatenating the full-depth and grayscale image, we introduce a new fusion-based input for our object detection network. Finally, these two-channel inputs train a YOLOV3 network to outperform state-of-the-art object detection systems.

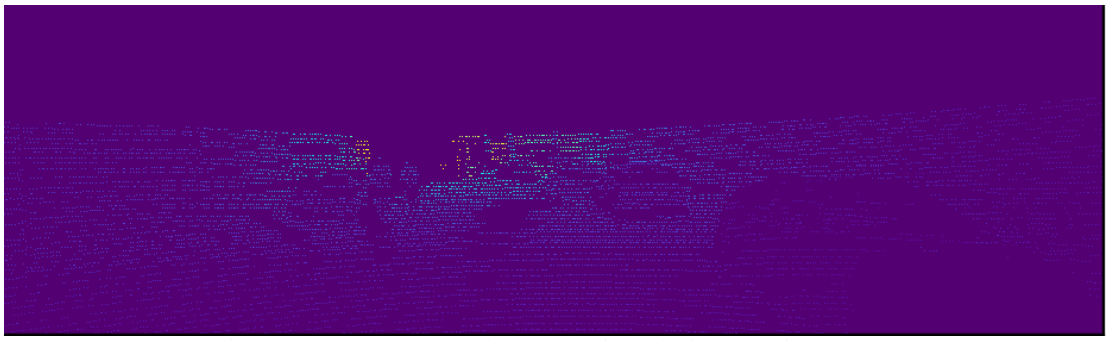


Figure 1: The sparse depth map (projected LiDAR point clouds)

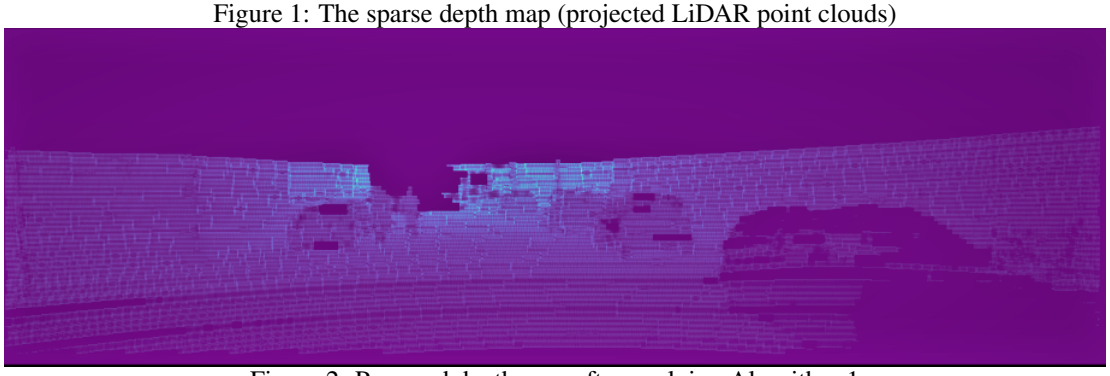


Figure 2: Prepared depth map after applying Algorithm 1

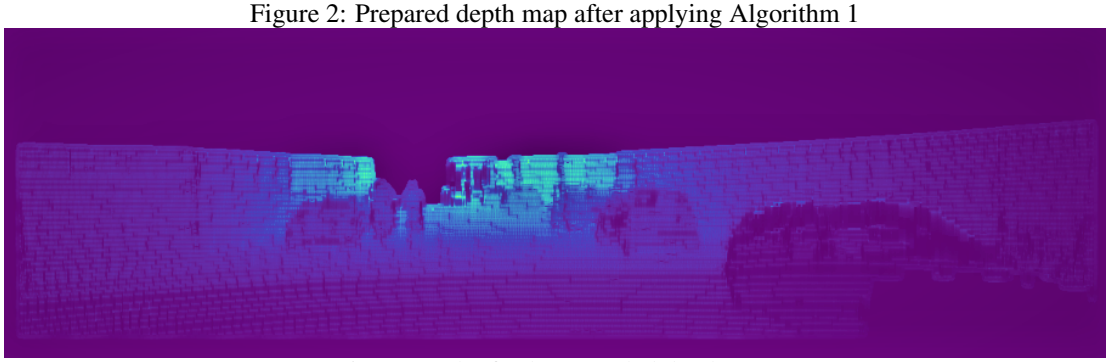


Figure 3: The final upsampled depth map

2. RELATED WORK

A considerable number of works are done focusing on object detection using depth perception. They vary from works with just LiDAR or image modality to works with combinations of these modalities. When it comes to upsampling, some methods use sparse LiDAR data without upsampling, while others increase the resolution of the LiDAR point clouds before using it for detection. Some of these methods are reviewed in this section.

2.1 RGB-D methods

Although some RGB-D works do not use LiDAR data, yet they are inspiring for LiDAR and camera fusion tasks. There are a variety of works that use depth and RGB combinations for object detection purposes. In [13], they adjust the scale of objects based on their depth to apply different actions at different distances. In [14], they adapt a faster R-CNN model for RGB and depth fusion. The fusion takes place in different layers of the model. It is shown that mid-layer fusion has the best performance. [15] uses late fusion structure for depth and RGB by encoding depth values as colorized surface normals to improve the performance of depth through pre-trained CNN. The depth and RGB fusion are also used for facial landmark detection [16], indoor semantic maps [17], human pose estimation [18], person re-identification [19, 20]. Moreover, RGB-D-T (RGB-D+thermal) is used for facial recognition [21].

2.2 LiDAR object detection

LiDAR object detection is more effective for 3D approaches as in [22], where point clouds are used after a preprocessing as an input for a CNN called PIXOR to obtain a more robust object detection for autonomous driving. In [23], an end to end CNN based method has served to provide a 3D object detection from LiDAR point clouds. They present a deep learning, 3D object detection model based on the LiDAR point cloud in which the YOLOv2 network is used [7]. [24]

Patch refinement is composed of two independently trained voxel-based network for 3D object localizing. [25] introduces a method to provide Pseudo-LiDAR data from cameras using the stereo depth network guided by LiDAR point clouds. In [26], uses 3D FCN for LiDAR point clouds. features maps are first downsampled by convolution operations and then upsampled by two parallel deconvolutions to yield objectness map and bounding box map.

2.3 Camera and LiDAR data fusion

Recently, LiDAR and camera fusion has occupied many interests to itself [27–31]. This popularity is mainly because of the diverse structures of these sensors, which prepare a notable mutual cover for the sensors. Because of the 3D semantic of LiDAR data, some works concentrated on depth perception of LiDAR and mixing it with an image to obtain a 3D object detection [32, 33]. In [34–36], we can see that hand-crafted features helped to resolve the multi-modality problem.

2.3.1 LiDAR data upsampling

The 3D LiDAR point cloud and camera image have different modalities. As we mentioned before, LiDAR point clouds are sparse, and combining them with high-resolution images needs complicated and specially-made structures. Upsampling is a facilitator solution to resolve this problem. In [10], an interpolation using Bilateral Filter is used to obtain a dense depth map solely from the LiDAR point cloud. In contrast, some of the works use auxiliary data such as scene labeling data [11, 12], and RGB image and the anisotropic diffusion tensor to guide upsampling [9].

2.3.2 Fusion structures

LiDAR and camera fusion can occur in different stages of the network structure. Early fusion is a standard method that needs at least up-sampling for LiDAR data [36]. [37] provide spatiotemporal alignment for LiDAR and video streams. RGB frames, optical flow, and LiDAR dense depth are concatenated and fed into a CNN. Indeed LiDAR data will be adjusted to add to RGB as extra channels [34]. [35] produces HHA (horizontal disparity, height above ground, and angel) from LiDAR point cloud and fuses it with RGB as input for a CNN. In [38], three groups of bounding boxes are provided form LiDAR BEV, LiDAR front views, and RGB image; then, they are fed into a classifier.

Late fusion is a method that combines features of LiDAR and camera at the end [39]. Middle-fusion is the most challenging approach. It includes an unlimited number of schemes; [40] uses multi-scale fusion by using features of mage extracted from different layers of a deep network with a bird's eye view (BEV) LiDAR data. In [32], by adopting two-stage CNN, a flexible framework is proposed to utilize any 2D detection network and fuse it with a 3D point clouds to achieve 3D detection. Similarly, [41] first generates a 2D object region proposal using CNN and then extends this 2D region to a 3D bounding box using LiDAR point clouds. Another work [42] combines Image and BEV feature maps through two sub-networks to yield 3D object proposals. In [43], they implement LiDAR and camera fusion for pedestrian classification. Both late and early fusion are utilized, and the better result is being evident in the early fusion CNN model. Another LiDAR and camera fusion presented in [44], YOLO models for depth, and RGB data produce features independently. Then the features are combined and increase the accuracy of the result of vehicle detection in comparison with separate LiDAR and camera results. In our work, instead of YOLO, we use YOLOv3, and our fusion is early fusion. Due to a variety of methods, it is still an open problem to work in LiDAR and camera fusion.

3. PROPOSED MODEL

3.1 Full-depth image preparation

Given the LiDAR's 3D cloud map, we obtain its corresponding 2D representation by projecting this data to the camera surface. For this projection, we use calibration data in the employed dataset. The dataset is described in the next section. We use the way they used in [45].

- $\mathbf{R}_{velo}^{cam} \in \Re^{3\times3}$ ... Rotation matrix: Velodyne $\Rightarrow$ camera
- $\mathbf{t}_{velo}^{cam} \in \Re^{1\times3}$ ... Translator matrix: Velodyne $\Rightarrow$ camera
- $\mathbf{P}_{rect}^{i} \in \Re^{3\times4}$ ... Projection matrix after rectification
- $\mathbf{R}_{rect}^{i} \in \Re^{3\times3}$ ... rectifying rotation matrix using

$$\mathbf{T}_{velo}^{cam} = \begin{pmatrix} \mathbf{R}_{velo}^{cam} & \mathbf{t}_{velo}^{cam} \\ 0 & 1 \end{pmatrix} \tag{1}$$

A 3D point x get projected to point y in the surface of the ith camera.

$$y = \mathbf{P}_{rect}^{0}\mathbf{R}_{rect}^{i}\mathbf{T}_{velo}^{cam}x \tag{2}$$

After projection, we have a sparse 2D depth map, which is used in the next step. The LiDAR depth map is sparse, thus, we need to upsample it. We apply weighted depth filling algorithm (Algorithm 1) to prepare a full-depth image. As is

evident in Figure 1, just a minority of pixels have non-zero values. We call these pixels as valued-pixels and another pixels as empty-pixels.

In this step, we try to fill in the empty-pixels. Concerning the structure of LiDAR data, as we can see Figure 1, the sparsity of the LiDAR data in the vertical direction is more than the horizontal one. Thus we employ a function that fills all vacant pixels with searching all the neighborhood of the pixel with longer vertical footstep rather than horizontal vicinity to achieve a monotonous depth map. For each empty-pixel, our algorithm explores backward and forward for the pixel to find all valued-pixels, and based on this findings, if the number of detected valued-pixels is more than zero for both directions, it assigns a value equal to the weighted average of all detected valued-pixels unless it goes for next pixel without giving it a value.

In the sparse depth map, the distance between to valued-pixels is approximately 4 - 6 pixels and 8-12 pixels, respectively, for vertical and horizontal directions. Moreover, due to the weighted structure in Algorithm 1, our method is not sensitive to selecting a large footstep. Thus we have to choose footsteps at least 6 and 12 for the vertical and horizontal axes. We chose 6 and 12 to minimize the computation load.

After filling the vacant points, we use a complementary algorithm (Algorithm 2) to make the shapes in the depth image more cohesive and fill the small vacant (see Figure 3). This algorithm checks a rectangular region around an empty-pixel, and if the number of valued-pixels is more than a threshold, the empty-pixel gets a value equal to the weighted average of all valued-pixels. The threshold has to be chosen in a way that every interior pixel of a shape gets filled. It is subordinated to the size of the square's side, which is three here. Similar to Algorithm 1, the weighted structure aims the Algorithm 2 to work with even large footsteps without problem but computational difficulties.

Algorithm 1 Weighted Depth Filling Algorithm

$S_x =$ step length for X-axis
$S_y =$ step length for Y-axis
for Each pixel in depth map **do**
  if $Pixelvalue = 0$ **then**
    $S = 0$
    $S_c = 0$
    for searched pixels forward with S-x footstep **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d) \times$(searched pixel value)
        $S_c1 \leftarrow S_c1 + (1/d)$
      end if
    end for
    for searched pixels backward with S-x footstep **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d) \times$(searched pixel value)
        $S_c2 \leftarrow S_c2 + (1/d)$
      end if
    end for
    $i \leftarrow i + 2$
  end if
  if $Sc1 > 0$ & $Sc2 > 0$ **then**
    Pixel value$\leftarrow S/(Sc1 + Sc2)$
  end if
end for
Do the same for Y-axis

3.2 channels calculation

In this step, we have a full-depth image, which includes depth detail of the scene, helping to better detection in addition to the camera image. For simplicity and using the structure and training weights of the ordinary RGB-trained YOLOv3

Algorithm 2 Complementary Weighted Depth Filling Algorithm

$Ss =$ step length for Searching
for Each pixel in the depth map **do**
  if $Pixelvalue = 0$ **then**
    $S = 0$
    $S_c = 0$
    for search a sqaure around the pixel Ss×Ss **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d)\times$(searched pixel value)
        $S_c \leftarrow S_c + (1/d)$
      end if
    end for
    $i \leftarrow i + 2$
  end if
  if $Sc/(Ss \times Ss) > Thresold$ **then**
    Pixel value$\leftarrow S/Sc$
  end if
end for

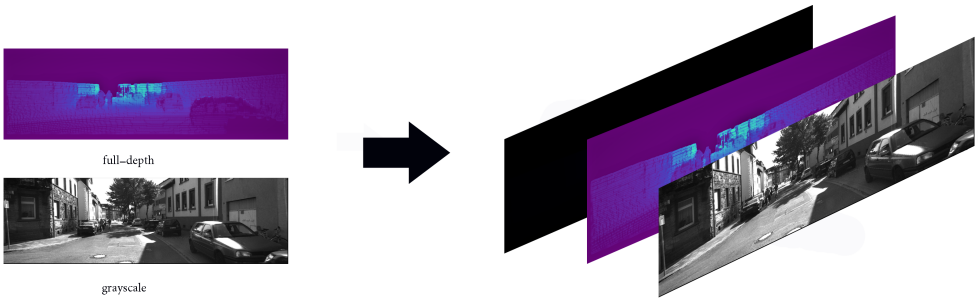


Figure 4: Channels calculation: after preparing the full-depth channel,another all zero channel is added to it to combine with the grayscale image to make the final three-channel input

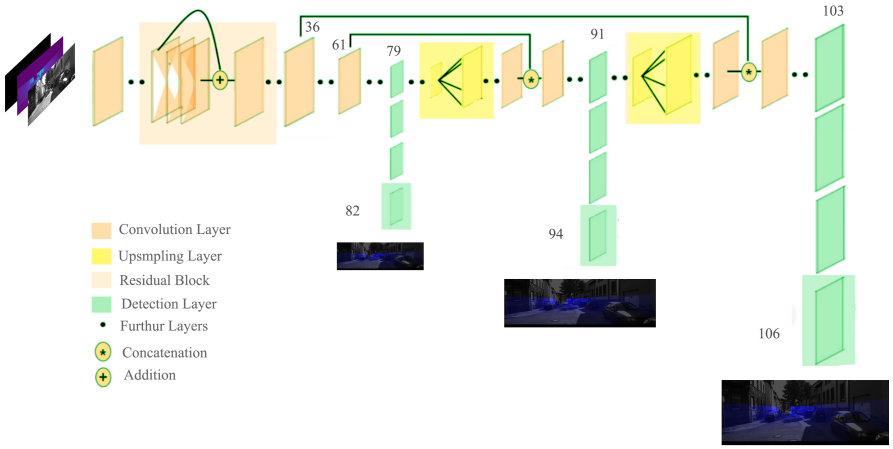
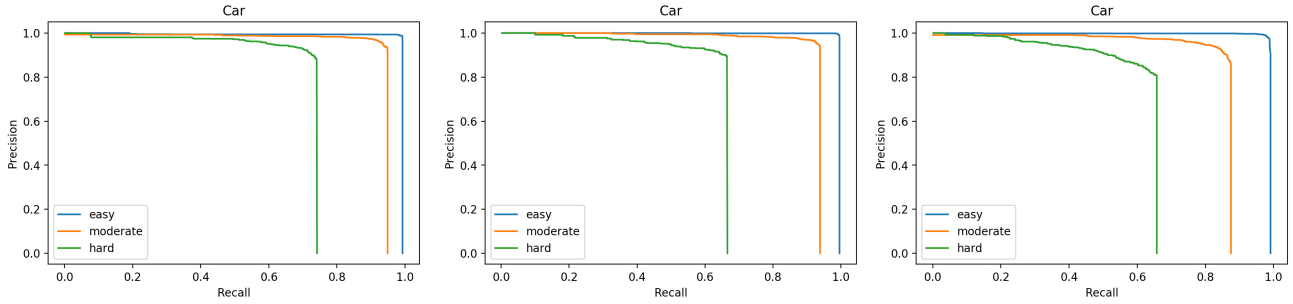


Figure 5: YOLOv3 structure shows how three different scales of feature maps used in the network.

network, we do not change the three-channel structure. As it is shown in Figure 4, the input of the YOLOv3 network is a three-channel image with one all-zero channel.

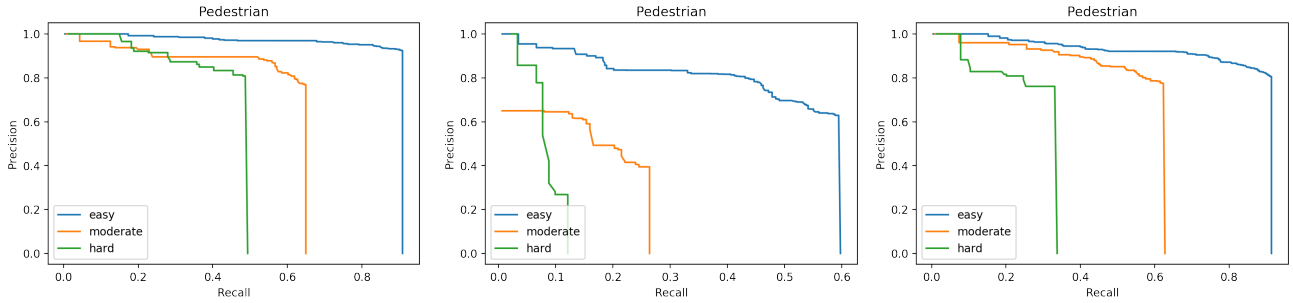
3.3 YOLO-V3

YOLO, published by Redmon at al. [3]. YOLO is a convolutional neural network that divides an image to grids and predicts multiple boxes simultaneously. It means that the speed increased considerably, but the accuracy drops. The next



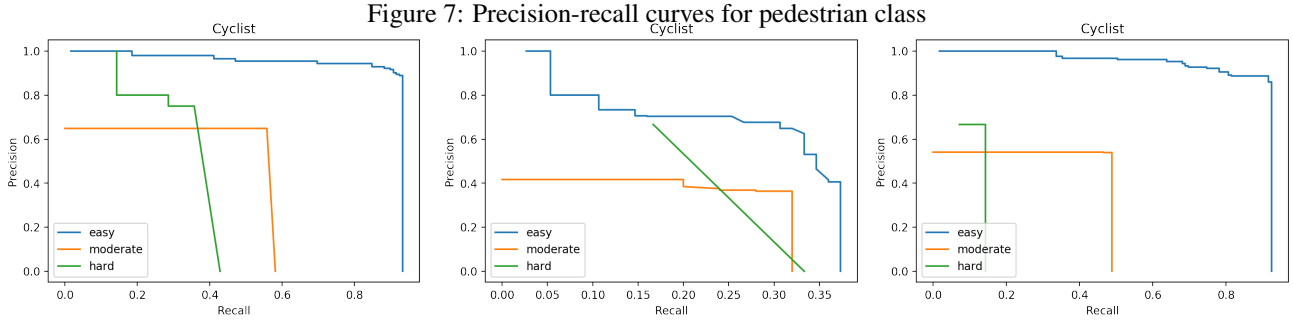
(a) **Proposed method** (b) RGB-YOLOv3 (c) LiDAR depth

Figure 6: Precision-recall curves for car class



(a) **Proposed method** (b) RGB-YOLOv3 (c) LiDAR depth

Figure 7: Precision-recall curves for pedestrian class



(a) **Proposed method** (b) RGB-YOLOv3 (c) LiDAR depth

Figure 8: Precision-recall curves for cyclist class

version of YOLO called YOLO9000 succeeded to keep the speed advantages and improve the accuracy of YOLO by introducing anchor boxes [4]. YOLOv3 proposed as an incremental improvement by Redmon et al [5]. First, it used the Darknet-53 backbone, which in comparison to Draknet-11 in YOLOv2, has better accuracy and efficiency in terms of classification. Secondly, instead of having a softmax and using binary cross-entropy, it uses logistic regression to calculate the score for each bounding box. Finally, YOLOv3 has excellent performance for small objects due to prediction from three different scale feature maps (see Figure 5). As anchor boxes are highly effective on the performance of the YOLO network, we used a K-means algorithm [4] to calculate the optimum anchor boxes based on KITTI labels.

4. EXPERIMENT

In this work, the network is trained just for three classes: car, pedestrian, and bicycle, as these are the most common objects in the context of interest of this paper, autonomous driving.

4.1 KITTI dataset

KITTI dataset uses a variety of sensor modalities such as high-resolution color and grayscale stereo cameras and Velodyne 3D laser scanner. It has plenty of real-world images with their corresponded LiDAR data. They support different categories such as 2D/3D object detection, path planning, depth completion, and some other categories. Calibration and labels data are available for each category [46]. We use object detection KITTI dataset with 7840 images and divide it to 80% for training and 20% for validation sets.

Table 1: Performance comparison of 2D object detection methods on the KITTI 2D object detection benchmark for the class car. Values are average precision (AP) scores on the official test set.

Car (AP)						
Methods	***Modality***	***Depth Upsampling***	***Easy***	***Moderate***	***Hard***	***Runtime***
Proposed method	LiDAR + Image	Solely Upsampled	**98.82%**	**94.82%**	75.16%	0.133 s
MMLab PV-RCNN [47]	LiDAR	–	98.17 %	94.70 %	**92.04** %	0.08 s
TuSimple [48]	Image	–	95.12 %	94.47 %	86.45 %	1.6 s
UberATG-MMF [24]	LiDAR + Image	Guided with Image	97.41 %	94.25 %	89.87 %	0.08 s
Patches - EMP [49]	LiDAR	–	97.91 %	93.75 %	90.56 %	0.5 s
Deep MANTA [50]	Image	–	**98.89** %	93.50 %	87.37 %	0.7 s
RRC [51]	Image	–	95.68 %	93.40 %	87.37 %	3.6 s
STD [52]	LiDAR	–	96.14 %	93.22 %	90.53 %	0.08 s
RGB-YOLO-v3	Image	–	97.23%	93.01%	64.02%	0.134 s
LiDAR-depth-YOLO-v3	LiDAR	Solely Upsampled	96.12%	85.05%	61.21%	0.129 s
MV3D (FV) [25]	LiDAR	Guided with Image	93.08 %	84.39 %	79.27 %	0.24 s
Complexer-YOLO [53]	LiDAR + Image	–	91.92 %	84.16 %	79.62 %	0.06 s
Pseudo-LiDAR++ [54]	LiDAR	Guided with Image	94.46 %	82.90 %	75.45 %	0.4 s
RefineNet [55]	Image	–	91.91 %	81.01 %	65.67 %	0.20 s
MV3D (RGB+FV) [25]	LiDAR + Image	Guided with Image	86.34 %	79.47 %	74.80 %	0.24 s
A3DODWTDA [26]	LiDAR	Solely Upsampled	82.98 %	79.15 %	68.30 %	0.08 s
MV3D(RGB) [25]	Image	–	83.86 %	76.45 %	73.42 %	-
3D FCN [56]	LiDAR	–	86.74 %	74.65 %	67.85 %	>5 s
MV-RGBD-RF [57]	LiDAR + Image	Guided with Image	77.89 %	70.70 %	57.41 %	4 s

Table 2: Calculated AP for pedestrian class

Pedestrian (AP)			
Methods	***Easy***	***Moderate***	***Hard***
Proposed method	**88.37%**	58.38%	**44.55%**
RGB-YOLOv3	88.29%	**58.50%**	38.19%
LiDAR depth	85.19%	56.64%	28.06%

Table 3: Calculated AP for bicycle class

Bicycle (AP)			
Methods	***Easy***	***Moderate***	***Hard***
Proposed method	89.18%	37.69%	**29.28%**
RGB-YOLOv3	**94.61%**	**79.30%**	4.76%
LiDAR depth	87.98%	26.39%	61.21%

4.2 Training setup

We trained three YOLOv3 networks with RGB, LiDAR depth, and depth+grayscale inputs. The networks are trained with 10,000 iterations. We use Google Colab for training YOLOv3. Google Colab is an online processor that provides considerable hardware, as mentioned below. GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM

4.3 Evaluation metrics

We follow the evaluation method of [46], which divides objects into three categories: easy, moderate, and hard. This is due to the size, occlusion, and truncation flags of each object. Based on the instruction discussed below, the situation of each object is clarified.

- **Easy**: box height $> 40px$, occlusion=0, truncation $< 15\%$
- **Moderate**: box height $> 25px$, occlusion=1, truncation $< 30\%$
- **Hard**: box height $> 40px$, occlusion=2, truncation $< 50\%$

we calculated the AP which is introduced in [58] as the area under precision-recall curve by numerical integration. this is highlighted in [59] that for the car class the Intersection over Union (IoU) threshold has to be 0.7 instead of 0.5 for other classes.

4.4 results and discussion

In the KITTI object detection benchmark, methods are ordered based on moderately difficult results. Thus, in Table 1, we focus on the moderate section, and the methods are sorted based on that. We can see that our proposed algorithm outnumbers all the mentioned schemes for the moderate difficulty in car detection. In an easy category, the proposed method has a 98.82% AP which is the highest amount between methods that utilize depth upsampling.
As mentioned in section 3.1, we propose a simple upsampling algorithm that converts the sparse depth map just based on LiDAR data to a dense depth map. Looking at Table 1, the LiDAR depth method, which used the preprocessing outcome, has achieved the best performance among all methods with LiDAR modality that use preprocessing [25, 26, 54]. The thing that has to be noticed here is that our LiDAR depth not only succeed solely but also the fusion of this full-depth map with image has ranked best among single modality-based methods as well as fusion-based ones. It shows that our preprocessing process prepares data that is more effective for fusion approaches.

Precision-recall curves in Figure 6 imply that the proposed method is trained suitably for car detection. Also, in comparison to the other two schemes, it has better performance. One factor of these results could refer to the structure of the depth filling algorithm that follows triangular shapes better than other shapes (see Algorithm 1 and Algorithm 2). The results reported in Table 2 and Table 3, demonstrate that the proposed method yields better performance in easy and hard sections for pedestrian class. Moreover, looking at Figure 8 and Figure 7, precision-recall curves for pedestrian class and cyclist class, have better patterns for the proposed method rather than RGB-YOLOv3 and LiDAR-depth, which have zigzag curves.

5. CONCLUSION

In this paper, we proposed an uncomplicated depth upsampling and sensor-fusion method that altogether achieves state-of-the-art performance on car class in the KITTI object detection benchmark. We prepared a full-depth map that is capable of filling simple shapes like cars efficiently. We fused the depth with the grayscale image as an input for the YOLO-v3 neural network. Results show that the proposed method outperforms single modality-based methods as well as fusion-based ones.

REFERENCES

[1] J. E. Espinosa, S. A. Velastin, and J. W. Branch, "Vehicle detection using alex net and faster r-cnn deep learning models: a comparative study," in International Visual Informatics Conference, 3–15, Springer (2017).

[2] M. Menikdiwela, C. Nguyen, H. Li, and M. Shaw, "Cnn-based small object detection and visualization with feature activation mapping," in 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ), 1–5, IEEE (2017).

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788 (2016).

[4] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in Proceedings of the IEEE conference on computer vision and pattern recognition, 7263–7271 (2017).

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767 (2018).

[6] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6647–6655 (2017).

[7] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in European Conference on Computer Vision, 197–209, Springer (2018).

[8] J. Hecht, "Lidar for self-driving cars," Optics and Photonics News **29**(1), 26–33 (2018).

[9] L. Chen, Y. He, J. Chen, Q. Li, and Q. Zou, "Transforming a 3-d lidar point cloud into a 2-d dense depth map through a parameter self-adaptive framework," IEEE Transactions on Intelligent Transportation Systems **18**(1), 165–176 (2016).

[10] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution lidar-based depth mapping using bilateral filter," in 2016 IEEE 19th international conference on intelligent transportation systems (ITSC), 2469–2474, IEEE (2016).

[11] M. Dimitrievski, P. Veelaert, and W. Philips, "Semantically aware multilateral filter for depth upsampling in automotive lidar point clouds," in 2017 IEEE Intelligent Vehicles Symposium (IV), 1058–1063, IEEE (2017).

[12] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically guided depth upsampling," in German conference on pattern recognition, 37–48, Springer (2016).

[13] H. Schulz, N. Höft, and S. Behnke, "Depth and height aware semantic rgb-d perception with convolutional neural networks," in Proc. Eur. Conf. Neural Netw.(ESANN), 463–468 (2015).

[14] C. Ertler, H. Posseger, M. Optiz, and H. Bischof, "Pedestrian detection in rgb-d images from an elevated viewpoint," in 22nd Computer Vision Winter Workshop, (2017).

[15] A. Aakerberg, K. Nasrollahi, C. B. Rasmussen, and T. B. Moeslund, "Depth value pre-processing for accurate transfer learning based rgb-d object recognition.," in IJCCI, 121–128 (2017).

[16] M. Quintana, S. Karaoglu, F. Alvarez, J. M. Menendez, and T. Gevers, "Three-d wide faces (3dwf): Facial landmark detection and 3d reconstruction over a new rgb–d multi-camera dataset," Sensors **19**(5), 1103 (2019).

[17] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in 2017 IEEE International Conference on Robotics and automation (ICRA), 4628–4635, IEEE (2017).

[18] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 1986–1992, IEEE (2018).

[19] F. M. Hafner, A. Bhuiyan, J. F. Kooij, and E. Granger, "Rgb-depth cross-modal person re-identification," in 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1–8, IEEE (2019).

[20] A. Møgelmose, T. B. Moeslund, and K. Nasrollahi, "Multimodal person re-identification using rgb-d sensors and a transient identification database," in 2013 International Workshop on Biometrics and Forensics (IWBF), 1–4, IEEE (2013).

[21] M. O. Simón, C. Corneanu, K. Nasrollahi, O. Nikisins, S. Escalera, Y. Sun, H. Li, Z. Sun, T. B. Moeslund, and M. Greitans, "Improved rgb-dt based face recognition," Iet Biometrics **5**(4), 297–303 (2016).

[22] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 7652–7660 (2018).

[23] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4490–4499 (2018).

[24] M. Liang*, B. Yang*, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in CVPR, (2019).

[25] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in CVPR, (2017).

[26] F. Gustafsson and E. Linder-Norén, Automotive 3D Object Detection Without Target Domain Annotations, Master's thesis, Linköping University (2018).

[27] S. Gu, T. Lu, Y. Zhang, J. M. Alvarez, J. Yang, and H. Kong, "3-d lidar+ monocular camera: An inverse-depth-induced fusion framework for urban road detection," IEEE Transactions on Intelligent Vehicles **3**(3), 351–360 (2018).

[28] J.-r. Xue, D. Wang, S.-y. Du, D.-x. Cui, Y. Huang, and N.-n. Zheng, "A vision-centered multi-sensor fusing approach to self-localization and obstacle perception for robotic cars," Frontiers of Information Technology & Electronic Engineering **18**(1), 122–138 (2017).

[29] Z. Ouyang, C. Wang, Y. Liu, and J. Niu, "Multiview cnn model for sensor fusion based vehicle detection," in Pacific Rim Conference on Multimedia, 459–470, Springer (2018).

[30] S. Budzan and J. Kasprzyk, "Fusion of 3d laser scanner and depth images for obstacle recognition in mobile applications," Optics and Lasers in Engineering **77**, 230–240 (2016).

[31] Y. Liu, S. Piramanayagam, S. T. Monteiro, and E. Saber, "Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order crfs," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 76–85 (2017).

[32] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in Proceedings of the European Conference on Computer Vision (ECCV), 641–656 (2018).

[33] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 8445–8453 (2019).

[34] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," IEEE Transactions on Industrial Informatics **14**(9), 4224–4231 (2018).

[35] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing lidar and images for pedestrian detection using convolutional neural networks," in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2198–2205, IEEE (2016).

[36] X. Du, M. H. Ang, and D. Rus, "Car detection for autonomous vehicle: Lidar and vision fusion approach through deep learning framework," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 749–754, IEEE (2017).

[37] M. Giering, V. Venugopalan, and K. Reddy, "Multi-modal sensor registration for vehicle perception via deep neural networks," in 2015 IEEE High Performance Extreme Computing Conference (HPEC), 1–6, IEEE (2015).

[38] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1907–1915 (2017).

[39] Z. Wang, W. Zhan, and M. Tomizuka, "Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection," in 2018 IEEE Intelligent Vehicles Symposium (IV), 1–6, IEEE (2018).

[40] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 3194–3200, IEEE (2018).

[41] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 918–927 (2018).

[42] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1–8, IEEE (2018).

[43] G. Melotti, C. Premebida, N. M. d. S. Goncalves, U. J. Nunes, and D. R. Faria, "Multimodal cnn pedestrian classification: A study on combining lidar and camera data," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 3138–3143, IEEE (2018).

[44] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," Pattern Recognition Letters **115**, 20–29 (2018).

[45] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, and others, "Gradient-based learning applied to document recognition," Proceedings of the IEEE **86**(11), 2278–2324 (1998).

[46] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," The International Journal of Robotics Research **32**(11), 1231–1237 (2013).

[47] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," arXiv preprint arXiv:1912.13192 (2019).

[48] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2129–2137 (2016).

[49] J. Lehner, A. Mitterecker, T. Adler, M. Hofmarcher, B. Nessler, and S. Hochreiter, "Patch refinement: Localized 3d object detection," arXiv preprint arXiv:1910.04093 (2019).

[50] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in CVPR, (2017).

[51] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," in CVPR, (2017).

[52] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: sparse-to-dense 3d object detector for point cloud," ICCV (2019).

[53] M. Simon, K. Amende, A. Kraus, J. Honer, T. Samann, H. Kaulbersch, S. Milz, and H. Michael Gross, "Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, (June 2019).

[54] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," in International Conference on Learning Representations, (2020).

[55] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "Refinenet: Refining object detectors for autonomous driving," IEEE Transactions on Intelligent Vehicles (Dec 2016).

[56] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in IROS, (2017).

[57] A. Gonzalez, G. Villalonga, J. Xu, D. Vazquez, J. Amores, and A. Lopez, "Multiview random forest of local experts combining rgb and lidar data for pedestrian detection," in IEEE Intelligent Vehicles Symposium (IV), (2015).

[58] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep (2011).

[59] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International journal of computer vision **88**(2), 303–338 (2010).

AUTHORS' BACKGROUND

Name	Title	Research Field	Personal website
Meisam J. Seikavandi	MSc.	Computer vision, Machine Learning	researchgate.net/profile/Meisam_Jamshid
Kamal Nasrolahi	Full Professor	Computer vision, Machine Learning	http://www.create.aau.dk/kn/
Thomas B. Moeslund	Full Professor	Computer vision, AI	vbn.aau.dk/en/persons/103282