2020 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 15-16, Ponta Delgada, Portugal

Multimodal Deep-Learning for Object Recognition Combining Camera and LIDAR Data

Gledson Melotti[1,2] Cristiano Premebida[2] Nuno Gonçalves[2]

Abstract—**Object detection and recognition is a key component of autonomous robotic vehicles, as evidenced by the continuous efforts made by the robotic community on areas related to object detection and sensory perception systems. This paper presents a study on multisensor (camera and LIDAR) late fusion strategies for object recognition. In this work, LIDAR data is processed as 3D points and also by means of a 2D representation in the form of depth map (DM), which is obtained by projecting the LIDAR 3D point cloud into a 2D image plane followed by an upsampling strategy which generates a high-resolution 2D range view. A CNN network (Inception V3) is used as classification method on the RGB images, and on the DMs (LIDAR modality). A 3D-network (the PointNet), which directly performs classification on the 3D point clouds, is also considered in the experiments. One of the motivations of this work consists of incorporating the distance to the objects, as measured by the LIDAR, as a relevant cue to improve the classification performance. A new range-based average weighting strategy is proposed, which considers the relationship between the deep-models' performance and the distance of objects. A classification dataset, based on the KITTI database, is used to evaluate the deep-models, and to support the experimental part. We report extensive results in terms of single modality *i.e.*, using RGB and LIDAR models individually, and late fusion multimodality approaches.**

Index Terms—**Robotic perception, LIDAR, classifiers fusion, machine learning**

I. Introduction

The field of robotic-perception has been developed considerably in terms of detecting and recognizing objects in the environment [1], [2], which strongly contributes to the technological progress in advanced robotics and autonomous vehicles. The growing field related to artificial perception for intelligent/autonomous vehicles (IV/AV), aggregating knowledge from several areas, such as electrical engineering, mechatronics, computing, statistics, and machine learning/artificial intelligent (ML/AI), has been achieving very promising and encouraging results on multisensor perception for autonomous vehicles [3].

Robotic perception can be understood as the process in which an autonomous robot (or vehicle) interprets sensor data collected on-board, in order to understand the world around it, thus allowing decision-making in an optimized and secure way. As pointed out by [3]–[5], sensory perception is not a trivial task. When it comes to object detection in real-world conditions, many difficulties are posed. For example, pedestrian detection is known to be such a quite challenging task for a AI-based perception system since people appearance depends on clothing, body articulation, and it may suffer influence of occlusion and lighting [6].

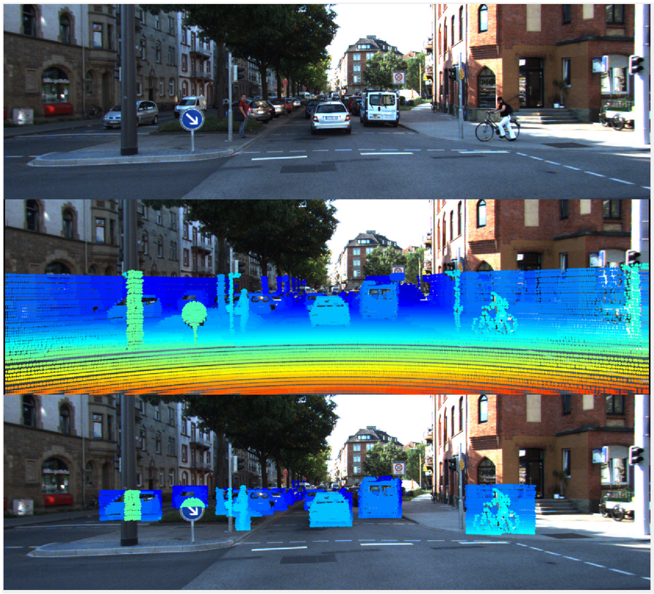


Fig. 1: This is an example obtained from KITTI $2D$ Object Detection Dataset showing the environment as "observed" by a robotic-vehicle. The $3D$ point cloud is coloured proportionally to the measured range. In the last row there are the projected point sets in the neighbour region of pedestrians, vehicles, and a cyclist.

Several sensors capture data from the environment in different ways, such as cameras, radars, stereo systems, $2D$ lasers, and LIDARs [3]. The image data processing has been researched not only in the past, but also in recent years, achieving very significant results in image recognition tasks, especially with the progress of modern machine learning techniques and deep learning [7], [8]. Therefore, the relevance of using cameras in a perception system is unanimous. However, cameras have considerable sensibility to varying ambient lighting and may need further lighting to capture images at night [9]. LIDAR sensors, on the other hand, have attracted interest in many applications, particularly those involving perception systems because they provide "physical" and direct

[1] G.Melotti is with Federal Institute of Espirito Santo, Brazil. Email: {gledson}@ifes.edu.br

[2] Authors are with the Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Portugal. Emails: {cpremebida , nunogon}@isr.uc.pt

978-1-7281-7078-7/20/$31.00 ©2020 IEEE

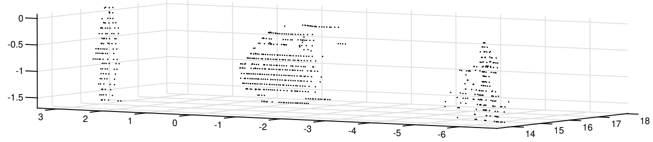


Fig. 2: Clustered $3D$ data points belonging to a pedestrian, a car, and a cyclist. A dataset including clustered objects were used to train a 3D deep-neural network (PointNet).

information regarding detected objects *i.e.*, LIDAR provides three-dimensional representations (usually in the form of point clouds) of the surrounding, it can operate at poor-illumination conditions, and provides intensity data as well. However, LIDAR sensors have some disadvantages such as the limited range and sparseness of points at a large distance [9].

One way to improve autonomous robotic systems, and autonomous driving, is by sensor fusion strategies, such as combining camera (RGB) and LIDARs (in the form of $3D$ point clouds, or range view, or bird-view representations) data. In the context of combining data from multiple sensors via a CNN model, three strategies can be considered: *early*, *intermediate* or *late* fusion [10]–[13]. The scope of this paper encompasses late fusion techniques.

Figure 1 illustrates an example of camera and LIDAR combination for object detection, by the projection of calibrated LIDAR data ($3D$ points) into the image-frame. The corresponding $3D$ point clouds of a car, pedestrian, and a cyclist can be seen in Fig. 2. It is possible to note that the set of LIDAR data points were segmented/clustered *i.e.*, the backgrounds and foregrounds points were removed.

Regarding algorithms, the convolutional neural networks (CNN) are the state of the art to process images and have obtained highly satisfactory results for image classification of objects [7], [14]–[18]. For point clouds, there are algorithms that perform the detection, extraction and removal of outliers, such as 3DmFV, Multi-resolution Surface Variation, PointNet, PointNet++ [2], [19]–[21]. Although technology wises, cameras and LIDARs have both strengths and weaknesses. Therefore, combining information from different sensors might contribute to enhance the performance of a perception system. The improvement of such a system does not depend exclusively on the sensors, hence new algorithms and fusion techniques should be developed [22], [23].

This work intends to contribute to the advances of multi-sensor perception for autonomous vehicle systems by focusing on multimodal late fusion strategies for object classification. This study addresses deep learning algorithms to process LIDAR ($3D$ point clouds) and camera data (RGB images). Additionally, we propose a new weighting strategy named Average Weighting Range (AW_R) which uses the relationship between the classification performance and the distance to the objects; despite its simple nature, the AW_R technique achieved promising results. This paper presents extensive experiments using single and multimodalities, and reports comparisons using state-of-the-art late fusion techniques; additionally, we propose distance-based learning approaches (using SVM and Genetic Algorithm) to combine multimodality models.

II. Related Work

Robotic perception systems usually take into account methods capable of extracting useful characteristics (knowledge) of the data being analyzed, such as deep learning methods using convolution concepts to process images [8] and $3D$ data [19], [21]. The concept of image classification is well defined through networks that employ layers of convolutions. However, many network architectures are extensive and require large quantities of time and memory. An alternative to reduce time and memory is transfer learning or ‘neural implants’, which are layers attached to a trained network, allowing the network to learn new tasks with few examples [24].

$3D$ point clouds can be used directly in neural networks, *i.e.*, with no need to project them on a $2D$ plane, such as the PointNet method used for detection/classification and segmentation of static point clouds [20], [21]. In contrast, the networks FlowNet3D [25] and PointFlowNet [26] are able to estimate scene flow, that is to say, they estimate $3D$ motions of point clouds from a time-evolving environment. The application of networks by employing point clouds should consider the robustness of identifying which points are part of the object. This means that the network must be able to recognize adversarial point clouds, in other words, to verify the robustness of network against adversarial attack [27].

To ensure safe driving on roads and highways, for both drivers and non-drivers, the technologies embedded in autonomous vehicles have to take into account the estimation of the position and orientation of the vehicle itself. In this way, the research developed by [28] presented an architecture with several deep neural networks and point clouds to localization for autonomous driving by calculating eigenvalues using PointNet [29], and $3D$ CNN. First, it extracts the keypoints defined by means of the neighbors eigenvalues of a $3D$ point. The PointNet extracts features, which are the inputs of the $3D$ convolutional neural networks ($3D$ CNN). The $3D$ CNN regularizes the volume over the dimensions. In addition, recurrent neural networks are used to process temporal motion dynamics.

An alternative strategy for LIDAR data processing consists of transforming the $3D$ data in a $2D$ representation, what could facilitate and simplify the utilization of state-of-the-art deep-CNN models. By projecting depth (distance/range view) and reflectance (intensity return) data, the resulting 2D-LIDAR “images” can be directly processed by off-the-shelf CNNs. Nonetheless, the point clouds generated by the LIDAR sensor are sparse and, therefore, such points must be sampled to obtain high-resolution range maps. Such maps can be obtained with different size of sliding windows and upsampling techniques such as Bilateral Filter, Inverse Distance Weighting, Ordinary Kriging, Delaunay triangulation, horizontal disparity processing [30], [31].

In early and late fusion systems, the input and output data, respectively, are combined for the purpose of obtaining a

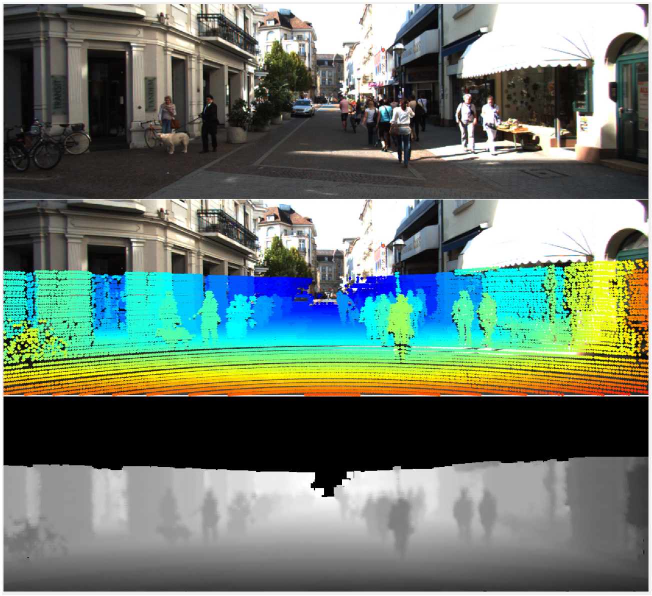


Fig. 3: An example of a depth map (DM), the last image, obtained by applying Bilateral Filter on the LIDAR points that have been transformed from $3D$ to image (pixel) coordinates, as shown in the 2nd image.

better and more robust classification result. Early fusion is the fusion of information/data at the input level of the classifiers; for example, images obtained from different modalities (*e.g.*, RGB images and depth maps) can be inputted into a multi-channel single CNN model. On the other hand, late fusion combines the scores (or confident level) from more than one learning model at the decision level [11], [13], [32]. Fusion schemes can be also carried out by machine learning algorithms, such as Support Vector Machines, Artificial Neural Network, Genetic Algorithms, among other techniques [33], [34]. In fact, the classification models can be processed "in parallel" and, at a certain stage, the resulting individual outputs can then be combined to perform the late fusion [10], [12], [31], [35].

III. Methodology and Techniques

In this section, the process of generating depth maps *i.e.*, 2D range view representation, out of a 3D point cloud is described, followed by the dataset description, and the late fusion techniques.

A. Depth Maps from LIDAR Data

When the LIDAR is calibrated with respect to a camera, it is simple to find the correspondence between the $3D$ coordinates of a point and the pixel values in the image plane, that is, each LIDAR-point will contain the position in pixels coordinates $(u, v)_i$ and, associated to that, the range/distance value r_i, with $i = 1, \dots n$ [36]–[38]. For the purpose of obtaining a high-resolution $2D$ representation of the $3D$ point cloud PC, the $2D - PC$ projections are upsampled within the image plane, resulting in a depth map as shown in Fig. 3. We have used a tailored version of the Bilateral Filter (a spatial filter method) implemented by using a sliding window with a mask $\mathbf{M} = 13 \times 13$ in size. In this work, the depth map (DM) contains range LIDAR data; which means the camera images were considered for calibration and visualization purposes only. In order to estimate the desired 'depth' value of the central-pixel of the mask, the implemented Bilateral Filter uses a bespoke weighting solution.

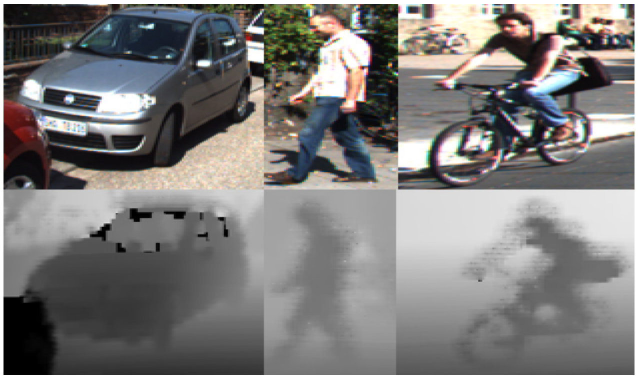


Fig. 4: Labeled car, pedestrian, and cyclist examples on RGB (colour images) and DM representation.

B. Dataset and Objects Distance Distribution

We manually cropped objects out of RGB images, depth maps (DM) and point clouds (PC), and then built a classification dataset containing three classes: vehicles (cars, vans, and trucks), pedestrians, and cyclists. Figure 4 shows some examples of the object classes. The number of objects (examples) on the training (vehicles=20632, pedestrian=2827 and cyclists=1025), validation (vehicles=2293, pedestrian=314 and cyclists=114) and testing (vehicles=9825, pedestrian=1346 and cyclists=488) sets is shown in Fig. 5, and also the distribution of the number of objects as function of the distance. Since one of the objectives of this work is to evaluate late fusion techniques that incorporate the distance to the objects as a relevant feature, the distance-distribution is an important factor. Figure 5 presents the percentage of objects, per class, w.r.t. the distance in meters as measured by the LIDAR sensor.

The distance of each object was obtained through the LIDAR projections and by considering the 'unbiased' average distance of it (each point on the depth map corresponds to a distance value), after eliminating their respective maximum and minimum values.

Before we calculate distances from point clouds, the $3D$ training dataset was studied by increasing the number of points belonging to the objects' set of points, as an optimal way to train the PointNet model, which requires a fixed input size. Then, we have performed upsampling and downsampling to guarantee the input dimension is constant *i.e.*, every point set belonging to an object has the same number of points. Along these lines, we got the datasets with 64, 128, 256, 512 and 1024 points to train the PointNet. The best classification result on the training was achieved with the 256 points, according to

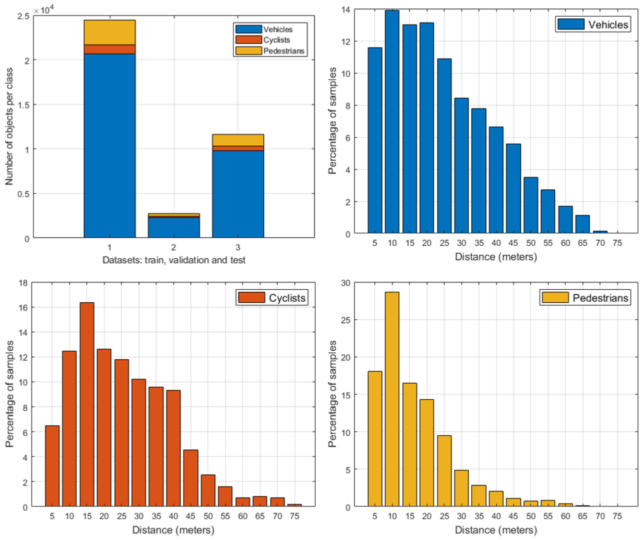


Fig. 5: The bar-graph (top-left) shows the number of examples per class (vehicles, cyclists, and pedestrians) on the training (24484 objects), validation (2721 objects) and testing (11659 objects) datasets, respectively. The other graphs show the distribution of examples separated by the categories and by the distance in meters.

TABLE I: Classification results on the training, in terms of F-score (in %), for the $3D$ point clouds using the PointNet model.

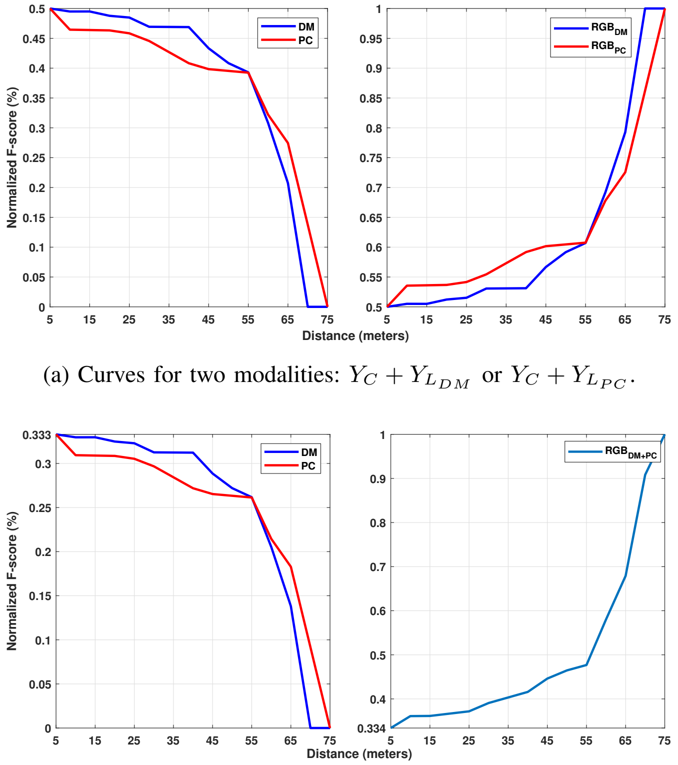
Classes	PC				
	64	128	256	512	1024
Ped.	75.65	86.85	**95.70**	91.93	86.10
Veh.	96.45	98.42	**99.41**	99.01	98.04
Cyc.	16.90	72.63	**91.74**	82.41	70.43
Ave.	63.00	85.97	**96.62**	91.12	84.86

Table I, where ‘Ped.’, ‘Veh.’, ‘Cyc.’ and ‘Ave.’ denote pedestrian, vehicles, cyclists, and the simple average, respectively. All the learning models were trained from scratch.

C. Weighted Object Distance Fusion

A new late fusion study is proposed in this paper, which takes the form of a weighted average (w), where the weights were obtained from distances of the DMs and PCs with the normalized classifiers F-scores on the training and validation dataset. The motivation comes from the fact that the LIDAR deep-models performance drops as the distance to the objects increases. On the other hand, the RGB model classification performance is relatively uniform with respect to distance from objects.

The average F-score, used as performance measure, were calculated considering the number of objects by increasing distance on the training and validation set, as per measured by the LIDAR, according to Figure 6. As a consequence, the weighting strategy can be interpreted as a function of the objects's distances and the classifiers (considering the DM-CNN and PC-PointNet models) performance (F-score × Distance).



(a) Curves for two modalities: $Y_C + Y_{L_{DM}}$ or $Y_C + Y_{L_{PC}}$.

(b) Curves for three modalities: $Y_C + Y_{L_{DM}} + Y_{L_{PC}}$.

Fig. 6: These curves show the normalized average F-scores obtained from the LIDAR-based model (DMs and PCs) for increasing distance of objects. The curves on the left represent the weights for the DMs and PCs modalities, while the curves on the right are the weights for the RGB modality, *i.e.*, the weight w_i.

Fig. 6a shows the normalized average F-score with maximum value of 0.5, therefore we guarantee that the RGB model outputs (y_C) will have a maximum weight equal to 0.5, while the Fig. 6b was normalized with a maximum value of 0.333.

Basically, the weights w depend on the models using PCs and DMs representations, and their performance on the training and validation set measured by the F-score for increasing objects distance values. The output (y) of this late fusion method (denoted by AW_R) is formulated according to the expression as follows

$$y = \Big(1 - \sum_i w_i\Big) y_C + \sum_i w_i y_{L_i} \tag{1}$$

where y is the final score (output) after the fusion, y_C is the classification score from the camera (RGB) model, y_{L_i} is the output from the LIDAR model, and i is the index denoting the LIDAR-based classifier that can be DM, PC, or both. The weight w_i, for a given LIDAR-classifier, follows a F-score curve in the Fig. 6, which also depends on the distance to the object.

TABLE II: F-score for single modalities on the test.

Modalities	RGB	DM	PC
F-score	96.24	89.55	88.46

D. Late Fusion Techniques

The late fusion methods usually assume independence on the classifiers outputs [11], [32]. We report comparative results using the following deterministic late fusion strategies: maximum, minimum, average, and normalized product (2).

$$\begin{aligned} \mathcal{S}_{max} &= max_i(\mathcal{S}_i) \\ \mathcal{S}_{mim} &= min_i(\mathcal{S}_i) \\ \mathcal{S}_{aver} &= \frac{1}{n}\sum_{i=1}^{n}\mathcal{S}_i \\ \mathcal{S}_{prod} &= \frac{\prod_{i=1}^{n}\mathcal{S}_i}{\prod_{i=1}^{n}\mathcal{S}_i + \prod_{i=1}^{n}(1-\mathcal{S}_i)} \end{aligned} \tag{2}$$

where n is the number of models, and S_i is the confidence score (output or 'likelihood') from a given model *i.e.*, a CNN or PointNet network.

Learning strategies based on a SVM (support vector machine) [39] and a GA (genetic algorithm) [40], [41] have been implemented as well. Additionally, the later methods were also incorporated to the object range/distance, obtained through the PCs and/or DMs representations, as an additional feature together with the scores from the single models of CNNs and PointNet. In this case, the methods are designated by GA_R and SVM_R.

The fitness function of the genetic algorithm is defined by Equation 3, aiming at maximizing the average F-score.

$$y = I_1\Big(1 - \sum_i w_i\Big)y_C + I_2\sum_i w_i y_{L_i} \tag{3}$$

where I_1 and I_2 are individuals ("chromosomes"). The other parameters are the same as in Equation (1). If the genetic algorithm does not consider the distance in the calculations, then the Equation (3) does not have the weighting terms w_i.

IV. Evaluation and Results

We have considered three types of datasets for evaluation purposes: RGB, DM, and PC. These single modalities classification result on the testing set, measured by F-score, using the Inception V3 CNN (RGB images and DMs), as well as PointNet (PCs), which are shown in Table II *i.e.*, without fusion strategy. Likewise, the results using late fusion techniques are shown in Table III, where the overall classification performance surpassed the single modalities.

The traditional methods of late fusion, as maximum, minimum, average, product, SVM and GA, without considering the values of the distances of the objects, have presented satisfactory performance, mainly for the fusion modalities $Y_C + Y_{L_{PC}}$ and $Y_C + Y_{L_{DM}} + Y_{L_{PC}}$ using N-Product and GA, respectively, which have represented the best overall performance for those two modalities (as shown in Table III). When considering distance/range in fusion strategies, such as SVM_R, GA_R and WA_R, the performance has been equivalent to the previous cases. However, for the combination of $Y_C + Y_{L_{DM}}$ the proposed method (WA_R) has achieved the best result among all models.

TABLE III: Average F-score, using late fusion and multi-modality representations on the testing set.

Late	Modalities		
Fusion	$Y_C + Y_{L_{DM}}$	$Y_C + Y_{L_{PC}}$	$Y_C + Y_{L_{DM}} + Y_{L_{PC}}$
Max.	96.88	96.91	96.75
Min.	97.03	96.95	96.97
Ave.	96.88	97.00	96.52
NProd.	97.14	**97.10**	97.01
GA	97.05	97.03	**97.27**
SVM	96.46	96.59	96.24
GA_R	96.34	96.27	96.26
SVM_R	96.52	96.60	96.57
AW_R	**97.22**	96.98	96.26

V. Conclusions and Remarks

This paper presents a thorough study on multiple classifiers combination, based on late fusion strategy, for object classification in robotic-perception environment. The classification techniques, using deep CNNs, were performed on three sensor data representations modalities: RGB images (using monocular camera), depth maps (or range view) and $3D$ point clouds obtained by a $3D$ LIDAR sensor. To evaluate the techniques a 3-class object classification has been created, whose classes are: vehicles (cars, vans, and trucks), cyclists, and pedestrians.

One of the key contributions of the paper was to show the importance of considering the object distance as an additional cue to be incorporated in a perception system. This work focus on late fusion strategies to combine/fuse the output (*likelihoods* or confident level) from the neural networks. A tailored distance-based method (designated by WA_R) has been proposed as a weighting function of the CNNs performance on the training set with respect to the object distance as measured by the LIDAR sensor.

In terms of performance on the testing set, the best result for the $Y_C + Y_{L_{DM}}$ modality was achieved by the proposed WA_R method, while the best results for $Y_C + Y_{L_{PC}}$ and $Y_C + Y_{L_{DM}} + Y_{L_{PC}}$ modalities were achieved by the normalized product and the genetic algorithm, respectively. The present study is promising and is worth of more attention, particularly on the idea of a performance measure regarding object distances which can be taken into consideration in multi-classifiers combination.

Finally, based on the results related to object distances and the fusion strategy presented in this paper, LIDAR and camera sensors are complementary, that is to say, the fusion between the two modalities has improved the overall performance, and therefore is relevant to multisensor perception systems.

References

[1] C. Wang, J. Yuan, and L. Xie, "Non-iterative slam," in *2017 18th International Conference on Advanced Robotics (ICAR)*, July 2017, pp. 83–90.

[2] P. Siritanawan, M. Diluka Prasanjith, and D. Wang, "3D feature points detection on sparse and non-uniform pointcloud for slam," in *2017 18th International Conference on Advanced Robotics (ICAR)*, July 2017, pp. 112–117.

[3] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *CoRR*, vol. abs/1704.05519, 2017.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[6] S. Aly, "Partially occluded pedestrian classification using histogram of oriented gradients and local weighted linear kernel support vector machine," *IET Computer Vision*, vol. 8, no. 6, pp. 620–628, 2014.

[7] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, June 2018.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] A. Miron, A. Rogozan, S. Ainouz, A. Bensrhair, and A. Broggi, "An evaluation of the pedestrian classification in a multi-domain multi-modality setup," *Sensors*, vol. 15, no. 6, pp. 13 851–13 873, 2015.

[10] N. Kapinski, J. M. Nowosielski, M. E. Marchwiany, J. Zielinski, B. Ciszkowska-Lyson, B. A. Borucki, T. Trzcinski, and K. S. Nowinski, "Late fusion of deep learning and hand-crafted features for Achilles tendon healing monitoring," *arXiv e-prints*, p. arXiv:1909.05687, Sep 2019.

[11] N.-B. Chang and K. Bai, *Multisensor Data Fusion and Machine Learning for Environmental Remote Sensing*. Boca Raton London New York: CRC Press, 2018.

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *CoRR*, vol. abs/1604.06573, 2016. [Online]. Available: http://arxiv.org/abs/1604.06573

[13] E. Morvant, A. Habrard, and S. Ayache, "Majority vote of diverse classifiers for late fusion," in *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 153–162.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*. San Diego, California, USA: ICLR, May 2015.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 1–9.

[17] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, USA, 2012, pp. 1097–1105.

[19] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3145–3152, 2018.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5099–5108. [Online]. Available: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[22] T. Akilan, Q. M. J. Wu, A. Safaei, and W. Jiang, "A late fusion approach for Harnessing multi-cnn model high-level features," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Banff, AB, Canada: IEEE, Oct 2017, pp. 566–571.

[23] D. O. Pop, A. Rogozan, F. Nashashibi, and A. Bensrhair, "Incremental cross-modality deep learning for pedestrian recognition," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, June 2017, pp. 523–528.

[24] Y. Lifchitz, Y. Avrithis, S. Picard, and A. Bursuc, "Dense classification and implanting for few-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2019.

[25] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] A. Behl, D. Paschalidou, S. Donne, and A. Geiger, "Pointflownet: Learning representations for rigid motion estimation from point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA: IEEE, June 2019.

[28] W. Lu, Y. Zhou, and G. W. S. H. S. Song, "L^3-net: Towards learning based LiDAR localization for autonomous driving," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA: IEEE, June 2019.

[29] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 77–85.

[30] S. Gu, Y. Zhang, X. Yuan, J. Yang, T. Wu, and H. Kong, "Histograms of the normalized inverse depth and line scanning for urban road detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3070–3080, Aug 2019.

[31] 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)*. Stockholm, Sweden: IEEE, May 2016, pp. 2198–2205.

[32] A. Mi, L. Wang, and J. Qi, "A multiple classifier fusion algorithm using weighted decision templates," *Scientific Programming*, vol. 2016, pp. 1–10, January 2016.

[33] G. Greffenstette, P.-A. Moëllic, and C. Millet, "Object/background scene joint classification in photographs using linguistic statistics from the web," in *OntoImage*, 2008.

[34] D. Liu, K. Lai, G. Ye, M. Chen, and S. Chang, "Sample-specific late fusion for visual category recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 803–810.

[35] Z. Guo, W. Liao, Y. Xiao, P. Veelaert, and W. Philips, "Deep learning fusion of RGB and depth images for pedestrian detection," in *30th British Machine Vision Conference (BMVC)*, Cardiff, UK, September 2019.

[36] G. Melotti, C. Premebida, N. M. M. da S. Gonçalves, U. J. C. Nunes, and D. R. Faria, "Multimodal CNN pedestrian classification: a study on combining LIDAR and camera data," in *21st IEEE Int. Conference on Intelligent Transportation Systems (ITSC)*, USA, Nov 2018.

[37] G. Melotti, A. Asvadi, and C. Premebida, "CNN-LIDAR pedestrian classification: combining range and reflectance data," in *ICVES*. IEEE, 2018, pp. 1–6.

[38] 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)*, Nov 2016, pp. 2469–2474.

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[40] G. Bakrl, D. Birant, and A. Kut, "An incremental genetic algorithm for classification and sensitivity analysis of its parameters," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2609 – 2620, 2011.

[41] R. Robu and H. Stefan, "A genetic algorithm for classification," in *Recent Researches in Computers and Computing - International Conference on Computers and Computing, ICCC'11*, 05 2011, pp. 52–56.