LiDAR and Camera Based 3D Object Classification in Unknown Environments Using Weakly Supervised Learning

Siva Prasad Raju Bairaju[a], Srinivas Yalagam[b] and Krishna Reddy Konda[c]
ZF Technology Center, India

Keywords: Sensor Fusion, Object Detection/Classification, Late Fusion, Pointcloud, Weakly Supervised Learning.

Abstract: Sensor redundancy is often relied upon the method in various applications to ensure robust and secure operation. Autonomous Driving (AD) and Advanced Driver Assistance Systems (ADAS) are no exceptions. camera and LiDAR are the principle sensors that are used in both applications. LiDAR is primarily used for object localization due to its active nature. A camera on the other hand is used for object classification owing to its dense response. In this paper, we present a novel neural network and training methodology for camera-based reinforcement of LiDAR object classification. The proposed method is also useful as a domain adaptation framework in an unknown environment. A pre-trained LiDAR-based object classification network is iteratively trained based on camera classification output to achieve continual improvement while in operation. The proposed system has been tested on benchmark datasets and performs well when compared with the state of the art.

1 INTRODUCTION

AD application uses multiple sensors to detect on-road objects such as pedestrians, cars, buses, etc as part of the perception stack. The most widely used sensors for this purpose are LiDAR, RADAR, and camera. Each sensor has its advantages and disadvantages. Multi-sensor fusion is used for improving the accuracy of detection and classification. Given the dense sensor response, the camera has always been preferred over LiDAR for object classification. However, LiDAR is preferred for object localization owing to its three-dimensional and active response. Moreover, the usage of multiple sensors adds redundancy to environment perception and guards against sensor failures.

In recent years, huge progress has been made in object detection and classification using deep learning-based methods. Deep Learning-based object detection on LiDAR point cloud has excellent localization but modest classification performance. There have also been several deep learning-based camera and LiDAR fusion-based methods for the classification and detection of objects in ADAS and AD scenarios.

However, deep learning-based methods require a huge amount of labelled data for training the neural network-based models. Moreover, any pre-trained standalone or fusion-based algorithm requires domain-specific training for deploying in a particular environment, which again is a time-consuming process as it requires data recording and ground truth generation in that particular domain. It is in this context, we propose our method, a semi-supervised algorithm for domain adaptation and speedy deployment of camera and LiDAR-based perception algorithms for AD and ADAS applications.

In our method, we take advantage of both sensors and try to improve 3D object classification. Usage of 3D object classification instead of 2D classification helps in accurate localization of classified objects. Further special filtering of YOLO based labels based on accuracy is undertaken to eliminate error propagation to LIDAR based 3D classification. By performing online training of the LiDAR-based object classification model, the proposed method also captures time-limited temporary features/markers of the target object. Consequently increasing the overall classification accuracy.

For this, we propose a new method called **On-TheGo weakly supervised Learning** which improves the classification of LiDAR detected objects without manually labelled data. We take the state-of-the-art algorithm in terms of speed and accuracy

[a] https://orcid.org/0000-0001-5039-1802
[b] https://orcid.org/0000-0003-4176-339X
[c] https://orcid.org/0000-0002-1727-9832

Bairaju, S., Yalagam, S. and Konda, K.
LiDAR and Camera Based 3D Object Classification in Unknown Environments Using Weakly Supervised Learning.
DOI: 10.5220/0011657700003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 304-311
ISBN: 978-989-758-626-2; ISSN: 2184-4313

for 3D object detection called PointPillars(Lang et al., 2019) and the best state-of-the-art 2D object recognition algorithm in terms of speed and accuracy on image called YOLO(Redmon et al., 2016). We use a branch network based on PointNet(Qi et al., 2017) which is trained iteratively to improve the classification of LiDAR-based 3d detected objects while in operation. Original PointNet network is used to classify indoor 3D objects, however, in this context we use it for outdoor object classification in AD/ADAS scenarios. Here PointPillars, YOLO, and PointNet are optional networks that can be used as plug and play with alternative best algorithms to improve the classification of 3D objects.

This research paper is organized as follows: A detailed review of semi-supervised or weakly supervised algorithms involving LiDAR and camera for AD and ADAS applications is presented in section 2. Section 3 describes the contribution of current work. In section 4, we have presented the methodology of the proposed algorithm. In section 6 we have discussed network architecture and implementation details. Data preparation details are presented in section 7. Experimentation and results on the nuScenes dataset are discussed in section 8 and applications are proposed in 9.1. This paper concludes with future scope in section 10.

2 RELATED WORK

LiDAR camera fusion has been a preferred way of object detection and classification for active ADAS and AD systems. As mentioned in the earlier sections, LiDAR and camera have better localization and classification respectively when compared with each other. Hence, fusion not only increases the accuracy of detection and classification but also increases the redundancy of the sensor setup. There have been several deep learning-based methods for LiDAR and camera fusion in recent years. They can be classified as early and late fusion. Early fusion-based methods, combine the sensor data at the initial stages (Qi et al., 2018; Chen et al., 2017; Ku et al., 2018).

Similarly, late fusion-based methods(Song and Xiao, 2016; Hoffman et al., 2016) process the sensor data separately to arrive at individual predictions. These predictions are further combined using various models to arrive at detection and classification.

Of particular interest for us in the context of current work is the late fusion category of methods. Since LiDAR and camera data are processed separately, there exists two separate detection/classification models which are completely independent of each other. In general, both models are trained separately with a separate set of ground truth which has been marked separately. However, in this work, we would like to explore the possibility of exploiting predictions of one of the sensor modalities to generate classification labels eliminating pre-training of one of the sensor models. Further such a method also helps in domain adaptation for unknown environments.

There have been very few works in this direction, which exploit the redundancy across sensor domains to train the sensor models for detection and classification.

In (Kuznietsov et al., 2017), the authors predicted depth from a single image using sparse LiDAR depth as ground truth and unsupervised depth measurements from a stereo pair. Here, LiDAR depth acts as ground truth for image-based depth estimation. Similarly, in (Caltagirone et al., 2019), the authors propose two classifiers acting on different views of the data cooperatively and iteratively improve each other's performance by using unlabelled examples. This method is among the top performers while using only a small amount of labelled data.

In (Buhler et al., 2020), The authors proposed two architectures to learn common representations of LiDAR and camera data, in the form of a 2D image. It is useful in feature matching algorithms. In (Yan et al., 2018), the human classifier can be learnt directly from the deployment environment, removing the dependence on labelled data. This method tracks people by detection of legs extracted from a 2D LiDAR and fusing this with the faces or the upper bodies detected with a camera using a sequential implementation of the Unscented Kalman Filter (UKF). Depth estimation from a single mono camera is proposed in (Kumar et al., 2018). This method trains using sparse LiDAR data as ground truth for depth estimation for fisheye camera. In (Teichman and Thrun, 2012) using limited training data, a classifier is trained, and the predicted label is propagated across frames, which are again used for training.

As can be seen from the discussion, weakly supervised online training of detection and classification models have been used for dense depth estimation and object detection to a certain extent.

3 CONTRIBUTION

As can be seen from the previous section, there exist a series of methods that exploit redundancy across sensor domains for proposing a weakly supervised detection/classification algorithm for various applications.

However, there is a lack of methods that can perform continual improvement of detection/classification accuracy in unknown environments. In this work, we try to address this by proposing a unique network architecture and weakly supervised training methodology for 3D object classification.

Contribution of the paper can be summarized as follows:

- State of the Neural network-based architecture for accurate object detection and classification using LiDAR point cloud as an input.
- Training methodology for domain adaptation of LiDAR classification model with output of camera-based object classification being used as a labeling mechanism .
- Overall methodology for adaptation of LiDAR-based object classification network in an unknown environment.

4 METHODOLOGY

As discussed in the previous sections aim of the current proposal is to use camera-based classification output as a label to improve the accuracy of LiDAR-based classification iteratively during inference/run time. The proposed framework consists of two independent standalone pre-trained networks for both camera and LiDAR-based detection and classification. While PointPillars(Lang et al., 2019) is used for LiDAR, YOLO is used as a pre-trained network for the camera. Bounding box output of Pointpillars network is further given as an input to PointNet(Qi et al., 2017) for 3D object classification. PointNet network used for classification is iteratively trained during the inference/run time using YOLO output as a ground truth.

In our approach, synchronization between LiDAR and camera sensor data is a prerequisite, along with correct calibration for each sensor with respect to vehicle coordinates. PointPillars network is pre-trained with existing data to detect the objects in a 3D environment with a 3D point cloud as input, while the YOLO network is pre-trained to detect and classify the objects in a 2D image.

PointPillar network is implemented in Keras and trained on KITTI(Geiger et al., 2013) and nuScenes(Caesar et al., 2019) datasets independently, YOLO pre-trained network is taken which is trained on autonomous driving scenarios. We take these two pre-trained models and Objects detected by the PointPillar network are given as input to the branch network based on PointNet for 3D object classification.

3D object classification model which corresponds to a branch network is trained recursively in an unknown environment that is completely non-identical with the training data. During inference, we modify only the weights of the PointNet network, while pre-trained weights of PointPillars and YOLO are fixed.

PointNet(Qi et al., 2017) is based on the principle of continuous functional approximation. The Skeleton of the object is roughly estimated using a sparse set of key points sampled from the input point cloud. PointNet is highly robust to small perturbation of input points, as well as to corruption through point insertion (outliers) or deletion (missing data). PointNet is used as a classification network owing to its unique function approximation approach and low complexity.

PointPillars(Lang et al., 2019) is a fast encoding network for point cloud which uses PointNet to learn the representation of point cloud arranged in vertical columns. This network gives the best results for 3D object detection using a point cloud. Hence the decision to use it as a base detection network for the LiDAR point cloud.

5 PROBLEM FORMULATION

Let the camera image be denoted by I_t where t represents the time instance. Similarly, the point cloud of LiDAR output is represented by P_t. I_t is given as an input to a pre-trained model of YOLO to generate a 2D bounding box prediction called $Y(I_t)$. Let the Object localization output of the PointPillars network be denoted by $O(P_t)$. $O(P_t)$ is then projected onto the 2D image space of the camera plane using calibration parameters and compared with I_t. 2D bounding boxes are matched based on greater than 50 percent IOU(Intersection over Union) criteria. Consequently, classification labels of matched boxes are identified as classification labels G_t. Generated G_t is used for training and updating the LiDAR classification model $C_t(P_i)$ where P_i is a cropped 3D bounding box from PointPillar Network detection. This process is repeated iteratively to continually update the classification model $C_t(P_i)$. While the Object localization model $O_(P_t)$ and $Y(I_t)$ are pre-trained, the classification model is trained from the scratch. Given the efficiency of the camera in object classification, the usage of camera predictions as labels is a valid assumption. By following this approach, we eliminate the need for generating LiDAR object classification ground truth. Moreover, the proposed setup adapts very well to a dynamic environment, given the continual up-gradation of the classification model.

In Figure 2 sample case of matching 2D box and projected 3D box is presented. A projected 3D box is represented by a blue box, the green box represents the converted 2D box from the 3D projected box for IOU calculation, and the red box represents YOLO output. We only consider the bounding boxes which are having more than 50% IOU.

6 NETWORK ARCHITECTURE

The Block diagram of the proposed network architecture is visualized in Figure 1. as the 1 shows, the PointPillar network and YOLO a base networks. PointPillars accepts point clouds as input and gives oriented 3D boxes of cars, trucks, and cyclists. It consists of three main stages, as mentioned in (Lang et al., 2019). These are (1) Feature encoding network which converts a point cloud to a sparse pseudo-image; (2) 2D convolutional backbone which processes the pseudo-image into a high-level representation; (3) detection head which detects and regresses boxes. YOLO network has 24 convolutional layers followed by 2 fully connected layers as mentioned in (Redmon et al., 2016). The first convolutional layers of the network extract features from the image, and the fully connected layers predict the output coordinates and probabilities. As mentioned in (Qi et al., 2017), PointNet has three main modules: a max-pooling layer, a local and global information combination structure, and two joint alignment networks which aligns both input points and point features. The main advantage of PointNet is that it directly consumes unordered point sets as inputs.

As we can see from the block diagram in Figure 1, pre-trained YOLO acts as a camera-based detection and classification network. Similarly, a pre-trained PointPillars network is used as an object localization network. Finally, PointNet is used as a 3D classification network. The classification network is iteratively updated via training, using the labels generated from camera-based classification.

6.1 Implementation Details

In this section, we describe our network implementation details. We consider the base PointPillars architecture which has been trained on KITTI and nuScenes independently and YOLO model which has been trained using pascal VOC dataset (Everingham et al., 2010). In the iterative training process, we project the output of PointPillars onto the image to find the best match (we considered 50% IOU) with the output of YOLO. The detailed explanation of projection onto the image is described in section 6.2. After finding the best IOU, boxes that have less than 100 points in the point cloud are rejected to eliminate predictions that are based on sparse data. This is especially important in outdoor scenarios. Classification labels of matched objects are used for iterative training of the classification network. To reduce volatility, a training batch size of 12 was used with categorical cross-entropy as a loss function(Rusiecki, 2019).

6.2 Projection onto Image

Since we are using KITTI dataset as a reference for our method, we follow KITTI format for representing equations. The projection of a 3D point $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ in camera coordinates to point $\mathbf{y} = (u, v, 1)^{\mathrm{T}}$ in the i'th camera image is given as

$$\mathbf{y} = \mathbf{P}\ \mathbf{x} \tag{1}$$

We need to consider the rectifying rotation matrix of reference camera $\mathbf{R}_{rect}$. So,

$$\mathbf{y} = \mathbf{P}_{rect}\ \ \mathbf{R}_{rect}\ \ x \tag{2}$$

The rigid body transformation from Velodyne coordinates to camera coordinates

$\mathbf{R}_{velo}^{cam}\ \ \varepsilon\ \ \mathbb{R}^{3x3}$ ... rotation matrix
$\mathbf{t}_{velo}^{cam}\ \ \varepsilon\ \ \mathbb{R}^{1x3}$ ... translation vector

Using

$$\mathbf{T}_{velo}^{cam} = \begin{pmatrix} \mathbf{R}_{velo}^{cam} & \mathbf{t}_{velo}^{cam} \\ 0 & 1 \end{pmatrix} \tag{3}$$

a 3D point x in velodyne coordinates gets projected to a point y in i'th camera image as

$$\mathbf{y} = \mathbf{P}_{rect}\ \ \mathbf{R}_{rect}\ \ \mathbf{T}_{velo}^{cam}\ \ \mathbf{x} \tag{4}$$

7 DATASET PREPARATION

To perform rigorous evaluation, iterative training of the classification network is carried out from the scratch. Two different datasets representing different environments, namely KITTI(Geiger et al., 2013) and nuScenes(Caesar et al., 2019) are used for evaluation of the proposed setup. Labels for the pre-trained version of the classification network is generated by cropping all the 3D bounding boxes from point cloud

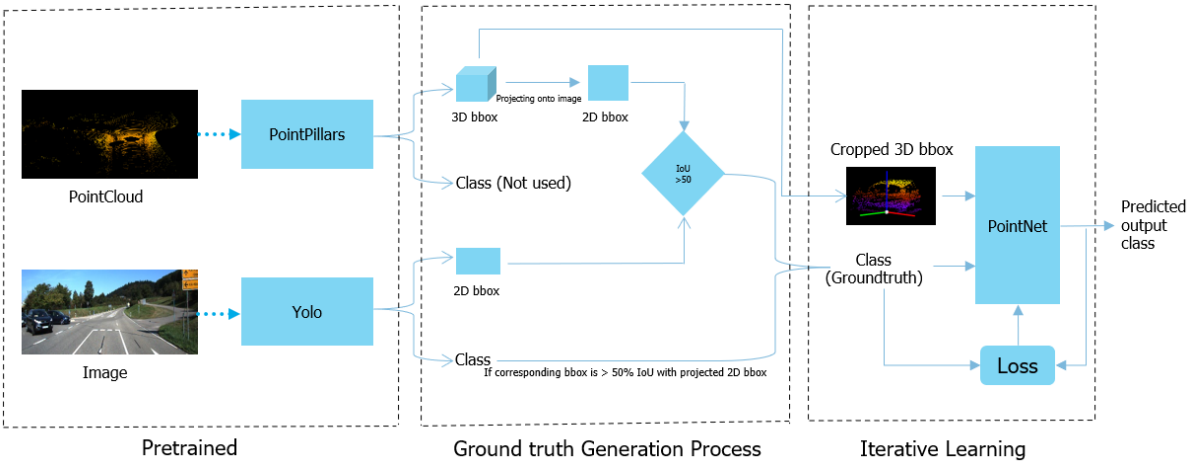
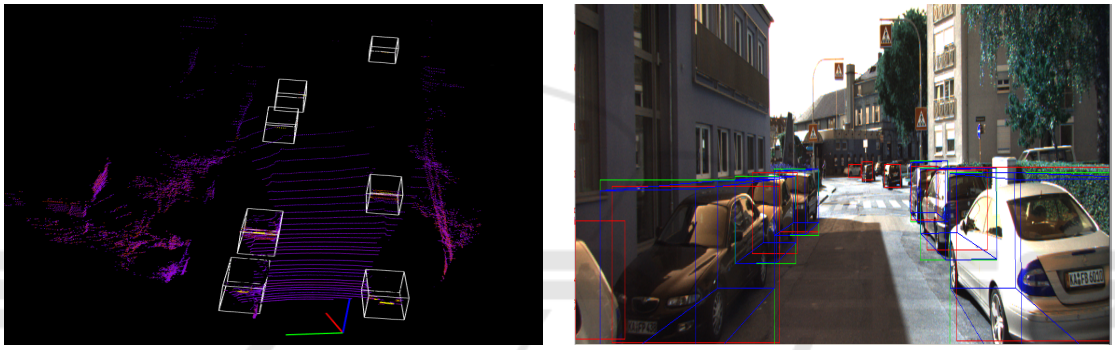


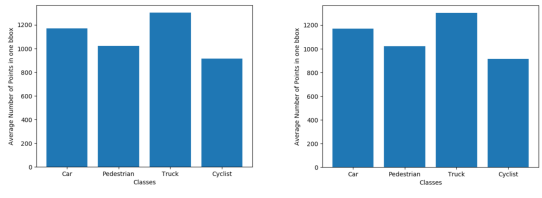
Figure 1: Block Diagram.



(a) (b)

Figure 2: KITTI dataset: Projected onto the image and generated labels(a) Point cloud view ; (b) Image view.

data and storing labels for each of the cropped objects. While pre-training localization network, we divide this data into training, validation, and testing in the ratio of 70, 10, and 20 percent. Before cropping the dataset, we first converted the nuScenes dataset into KITTI format. KITTI and nuScenes test data is used as an unknown environment for evaluating the performance of the classification network. The statistical distribution of points in the context of various classes is shown in Figure 3



(a) (b)

Figure 3: Average number of points in one 3D bounding boxes vs Classes (a) nuScenes Dataset ; (b) KITTI Dataset.

Since a high-definition LiDAR point cloud is composed of ∼100k points, cropped 3D boxes will also have highly variable point density which is shown in Figure 3, which causes bias for classification. For this purpose, we randomly sampled a fixed number of T points from those boxes containing more than T points. This sampling is done to decrease the imbalance of points between the boxes, which reduces the sampling bias and adds more variation to training. While feeding to the branch network also, we sample to a fixed number of T points and train the PointNet.

8 EXPERIMENTATION SETUP

In order to rigorously test the proposed framework, we have used KITTI and nuScenes datasets. Both datasets are widely used for evaluating LiDAR and camera fusion-based perception methods since these datasets consist of highly time synchronized LiDAR point clouds and images. Furthermore, the nuScenes dataset consists of 11000 successive data frames

which could be used for testing the OnTheGo updation of the LiDAR classification model. In order to overcome class imbalance we have merged the classes of pedestrian and cyclist into one human class along with merger of truck and car into vehicle class.

KITTI sensor setup contains 2-point greyscale and color cameras and 1 Velodyne HDL-64E rotating LiDAR sensor. So, it has 7481 training and 7518 testing samples of both images and point clouds. Similarly, nuScenes has six 1600 x 900 resolution cameras, one 32 beams LiDAR, and five RADAR sensors. It contains 33000 frames of highly synchronized camera and LiDAR data with ground truth. Out of 33000 frames in the nuScenes dataset 15000 frames are used for training, 2150 frames for validation, and 4290 for testing respectively. 11550 consecutive sets of frames are reserved exclusively for testing iterative OnTheGo classification model update. In the case of the KITTI dataset split are 6359 frames for training, 1122 frames for validation, and 7891 frames for testing.

In order to avoid loss of generality two sets of experiments have been performed namely

Experiment 1. In this iteration the camera detection and classification model $Y(I_i)$ is based on a pretrained YOLO model on the PASCAL VOC dataset, while PointPillars-based LiDAR object detection is pre-trained on nuScenes dataset. To demonstrate camera-reinforced domain adaptation of the framework, 3D classification model $C_t(O(P_t))$ is updated iteratively using consecutive frames from the nuScenes dataset. Comparison is made between classification accuracy on both nuScenes and KITTI test sets, before and after the OnTheGo training. Results have been summarized in Tables 1 and 2.

Experiment 2. In the second iteration, while camera based pre-trained model on the PASCAL VOC data set is retained, LiDAR object detection is pretrained on the KITTI dataset. OnTheGo training is again performed on consecutive frames reserved in the nuScenes dataset. Comparison of classification accuracy $C_t(O(P_t))$ before and after OnTheGo is tabulated in Tables 3 and 4.

9 RESULTS AND DISCUSSION

As can be seen from the Table 1 mean average precision has improved for both KITTI and nuScenes test sets by about 6 and 8 percent respectively. This is after iterative training of classification model on consecutive frames of nuScenes dataset with camera-based classification as labels as discussed in section 4. Such an observation reinforces that camera-based classification can be used as reinforcement for LiDAR-based classification. We can also infer that the proposed classification network after the iterative training on nuScenes data also improves its performance even on KITTI test data. This shows that features learnt during the iterative learning are generic and not domain specific. Table 2 further elaborates on class specifics of mean average precision.

Experiment 2 demonstrates the domain adaptability of the network. As we can see from the results of Table 3, even though the network is pre-trained on the KITTI data set, it can adapt very well to a new environment represented by the nuScenes dataset. Performance of classification is bettered for nuScenes by about 18 percent while also improving by 7 percent on the KITTI test set. Such an observation proved the domain adaptability of the proposed framework while also retaining robust performance in the previous domain. Table 4 details the class-specific mean average precision.

To gauge the iterative performance of the algorithm, we have also plotted the instantaneous accuracy of the algorithm for batches of 12 objects spanning Frame 1 to Frame 2443 in one particular scene of the nuScenes dataset in Figure 4. A batch of 12 objects is selected in order to maintain uniformity of training iterations. Since each frame may or may not contain a required number of objects for calculation of loss. We can infer from the figure 4 that classification accuracy starts from zero and steadily increases to a saturation value. Instantaneous dips in accuracy corresponding to the entering of new objects and large variance in the scene. We also infer that our method is learning local temporal features. Irrespective of the dips, we can see the continual rise in classification accuracy over a period. Such an observation reiterates our claim of continuous adaptation of the network with respect to a given environment under the guidance of a camera-based object classification network.

Table 1: PointPillars pretrained on nuScenes dataset and tested on KITTI & nuScenes testsets(before and after OnTheGo training).

Dataset	PointPillars (mAP) (Before OnTheGo) $O(P_t)$	Proposed network (After OnTheGo) $C_t(O(P_t))$
nuScenes	64.82	70.12
KITTI	61.32	69.25

Table 2: Class wise results for before and after OnTheGo training on both nuScenes and KITTI test datasets (PointPillars Pretrained on nuScenes).

Classes	nuScenes Before — After	KITTI Before — After
Vehicle Class	76.44 — 79.64	72.63 — 80.4
Human Class	53.2 — 60.6	50.01 — 58.24

Table 3: PointPillars pretrained on KITTI dataset and tested on KITTI & nuScenes testsets(before and after OnTheGo training).

Dataset	PointPillars (mAP) (Before OnTheGo) $O(P_t)$	Proposed network (After OnTheGo) $C_t(O(P_t))$
KITTI	69.31	76.14
nuScenes	43.875	61.84

Table 4: Class wise results for before and after OnTheGo training on both nuScenes and KITTI datasets (PointPillars Pretrained on KITTI).

Classes	nuScenes Before — After	KITTI Before — After
Vehicle Class	50.21 — 65.48	75.66 — 81.65
Human Class	37.54 — 58.2	62.96 — 70.63

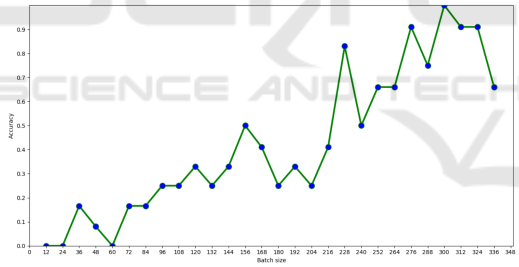


Figure 4: Object classification accuracy variation over iterative training steps.

9.1 Observations

As an outcome of these results we can deduce the following observations:

1. The proposed framework is very conducive for domain adaptation.
2. Has a capability to learn the localized temporal features (distinct features like shapes, distribution of objects, etc).
3. Can also handle the occlusions better.

10 SUMMARY AND FUTUREWORK

In this paper, we proposed a method to improve 3D object classification iteratively in an unknown environment without any ground truth. We have described a method to use existing state-of-the-art methods for 3D object detection and 2D object recognition on images to generate custom ground truth for iterative training. The proposed method is evaluated using publicly available datasets and performs very well regarding intended objectives. In the future, we would like to extend the proposed method to dynamic camera calibration using LiDAR-based localization as ground truth for iterative domain adaptation

REFERENCES

Buhler, A., Vödisch, N., Bürki, M., and Schaupp, L. (2020). Deep unsupervised common representation learning for lidar and camera data using double siamese networks. arXiv preprint arXiv:2001.00762.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2019). nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027,.

Caltagirone, L., Svensson, L., Wahde, M., and Sanfridson, M. (2019). Lidar-camera co-training for semi-supervised road detection. arXiv preprint arXiv:1911.12597.

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

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International Journal of Computer Vision, vol. 88, pages 303–338, June.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, vol.32, num. 11, pp. 1231–1237, publisher. Sage Publications Sage UK: London, England.

Hoffman, J., Gupta, S., and Darrell, T. (2016). Learning with side information through modality hallucination. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, page 826–834.

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

Kumar, V. R., Milz, S., Witt, C., Simon, M., Amende, K., Petzold, J., Yogamani, S., and Pech, T. (2018).

Monocular fisheye camera depth estimation using sparse lidar supervision. 2018 21st IEEE International Conference on Intelligent Transportation Systems (ITSC), page 2853–2858.

Kuznietsov, Y., Stuckler, J., and Leibe, B. (2017). Semi-supervised deep learning for monocular depth map prediction. Proceedings of the IEEE conference on computer vision and pattern recognition, , page 6647–6655.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., and Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12697–12705.

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

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652–660.

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

Rusiecki, A. (2019). Trimmed categorical cross-entropy for deep learning with label noise. journal. Electronics Letters, vol. 55, num. 6, pages. 319–320, publisher IET.

Song, S. and Xiao, J. (2016). Deep sliding shapes for amodal 3d object detection in rgb-d images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 808–816.

Teichman, A. and Thrun, S. (2012). Tracking-based semi-supervised learning. The International Journal of Robotics Research, Volume 31, number 7, page 804–818, publisher SAGE Publications Sage UK: London, England.

Yan, Z., Sun, L., Duckctr, T., and Bellotto, Nicola, M. (2018). online transfer learning for 3d lidar-based human detection with a mobile robot. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), page 7635–7640.