Monocular 3D Object Detection in Adverse Weather Conditions

1st Kanishka gabel
Automotive Engineering
University of Michigan - Ann Arbor
Ann Arbor, US
kgabel@umich.edu

2nd Srushti Hippargi
Automotive Engineering
University of Michigan - Ann Arbor
Ann Arbor, US
shipparg@umich.edu

3rd Siddhik Reddy Kurapati
Mechanical Engineering
University of Michigan - Ann Arbor
Ann Arbor, US
siddhikr@umich.edu

***Abstract*—Proposed Monocular 3-D object detection in [1] and modified monocular algorithm was implemented to calculate and localize 3D location of cars and their respective 3D bounding boxes in the input raw image data from camera. This project was a challenging problem as achieving a high accuracy required detecting vehicles in adversarial weather. To account for adverse weather, not only monocular's configuration/paramters were modified but filters(nighttime and fog) were used to modify our train dataset to achieve AP score on testing sets.**

***Index Terms*—Monocular, Synthetic Adverse Weather, KITTI**

I. Introduction

Monocular 3D object detection is used to detect and locate objects in 3D by only using raw camera data, this project is highly challenging problem due its ill-posed nature, especially without any information about depth, lidar and/or multi frames. The proposed method is a critical component in many computer vision application, especially robotic and autonomous systems and useful in low-cost system setups without using multiple sensors like Lidar, making it a prominent research topic in autonomous systems.

For understanding problem statement, mainly two papers have been referred for understanding and implementation monocular 3D object detection. Liu, Zheng and Cheng propose an approach utilizing the Perceiver I/O model to produce an unified feature embedding,furthermore using an self-attention mechanism to deliver a highly- accurate 3D box prediction. Proposed MonoXiver[1] is processing module to significantly improve the accuracy of object detection requiring only 90 minutes of training on a single GPU card for the Kitti dataset. Liu, Xue,and propose a simple effective algorithm for monocular 3D object detection without any extra information from other sensors like LiDAR sensors.The proposed MonoCon had utilized a ConVNet feature backbone and a list of regression heads that learns auxiliary monocular contexts that are projected from 3-D bounding boxes identified from an imageset consisting raw image camera.

Achieving high accuracy in detecting vehicles in adversarial weather conditions poses several challenges like (i)Limited Visibility - Adverse weather conditions such as heavy rain, fog, snow, or dust can significantly make it difficult for traditional vision-based systems to accurately detect and recognize vehicles, (ii) Ambiguous Visual Cues - Weather conditions can introduce visual cues that are ambiguous and challenging to interpret. For instance, raindrops on a camera lens may be mistakenly identified as objects, leading to false positives, (iii)Altered Image Features - Adverse weather conditions can alter the appearance of vehicles in images. Snow accumulation on vehicles, wet surfaces, or foggy atmospheres can change the way light interacts with objects, making it harder for algorithms to extract relevant features for accurate detection,(iv) Sensor Limitations - The sensors commonly used for vehicle detection, such as cameras and LiDAR, may have limitations in adverse weather. For example, rain or snow accumulation on the sensor itself can degrade its performance. Other factors are Dynamic Environmental Changes, Reduced Contrast, Data Variability, and Limited Annotated Data. Adverse weather conditions such as heavy rain, fog, snow, or dust can significantly make it difficult for traditional vision-based systems to accurately detect and recognize vehicles.

II. Methodology

This project proposes modified Monocular 3D object detection based on [2] for Self-Driving application. The main objective of the project is to localize 3D bounding boxes in a input single 2D image. It uses KITTI as monocular image dataset with object bounding box annotations. Fig 1. (b) and Fig 2 have been used to establish groundtruth. Other monocular image dataset like COCO, Pascal VOC that contains 2D images with object bounding box annotations. For the project, three images per frame, including the Raw camera image, Semantic segmentation and Instance segmentation have been used for training purpose, and later have been converted to KITTI format. One of main challenge of monocular 3D object detection is the accurate determination or localization of 3D center.

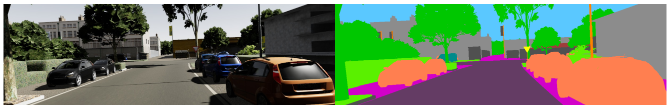

Fig. 1. (a) Original Raw Camera Image, (b) Semantic segmentation

In [1], their propose an approach is with a empirical upper-bound analysis with SOTA bottom-up monocular 3D object detectors. In the KITTI format[1], a 3D bounding box is by: (i)Bounding 3D structure center location (x, y, z). (ii)

Fig. 2. Instance Segmentation

observation angle of the required detectable object with respect to the camera based on the vector joining the camera center to the 3D object center, and (iii) the shape dimensions, i.e height, width and length. Monocular Contexts as auxiliary learning tasks in training to improve the performance of the MonoCon algorithm in detecting the objects. In [2] proposed method MonoCon utilizes three components: (i) Deep Neural Network(DNN) based feature backbone, (ii) number of regression branches for essential parameters, and (iii)number of regression branches for learning auxiliary contexts.

A. Method to modify the training dataset to compensate for the adversarial weather

To deal with the adversarial weather during the object detection, we propose a filter algorithm to adapt existing training data set. Applying a foggy and droplet filter to our dataset to simulate adverse weather conditions, like fog, is a reasonable approach to augmenting your data for training a model that needs to perform well in such conditions. The method used simulates adverse weather conditions, specifically fog and rain droplets, in an input image. It begins by reading the original image and creating a fog layer based on a specified intensity level. The fog layer is generated by blending the original image with a white fog overlay, and this is adjusted based on the intensity parameter. To simulate rain droplets, random positions on the image are selected using a binary mask generated with a given droplet intensity. Intensity values for the rain droplets are then randomly assigned in each channel (RGB) to simulate variations in droplet appearance. The fog layer is updated with the simulated rain droplets at the corresponding positions, and the resulting foggy image is obtained by blending the original image with the modified fog layer using the cv2.addWeighted function. The final image is then saved to the specified output path, effectively augmenting the original dataset with synthetic adverse weather conditions. Similarly, filter which accounts for nighttime and fog can used to modify our existing training dataset to account for adverse weather condition in test dataset.

This proposed method improves the model's robustness to different scenarios. By introducing synthetic fog to our images, we are essentially providing our model with additional variations in the data. This can help the model learn to recognize objects and features under adverse weather conditions, even if the training data does not originally contain such examples.

Important consideration to keep in mind are: (i) Realism of Augmentation: While adding a simple white filter simulates fog to some extent, real fog has more complex effects on images, including changes in lighting, visibility, and color. Depending on the requirements of our application, we may need more sophisticated fog simulation techniques, (ii) Model Architecture and Training: The success of the model in foggy conditions is not solely dependent on data augmentation. The choice of the model architecture, hyper parameters, and training strategy also plays a crucial role, (iii) Monitor Performance: During training, monitor the model's performance on both the training and validation sets. If the model is overfitting or not generalizing well, there may need to adjust your augmentation strategy or other aspects of your training process.

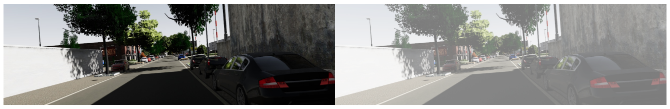

Fig. 3. (a) Original Image, (b) Synthetic Adverse Weather condition - Only Foggy(contrast=0.7)

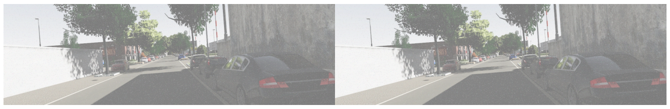

Fig. 4. (a) Synthetic Adverse Weather condition - Foggy(contrast = 0.7) and Water Droplets(contrast =), (b) Synthetic Adverse Weather condition - Foggy(contrast = 0.7) and Water Droplets(contrast = 0.7)

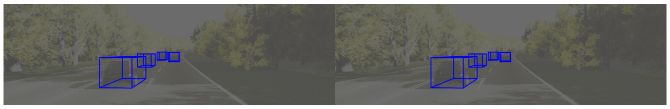

Fig. 5. (a) Synthetic Adverse Weather condition - Foggy(contrast factor = 0.6) and night(nighttime intensity factor = 0.35), (b) Synthetic Adverse Weather condition - Foggy(contrast factor = 0.6) and night(nighttime intensity factor = 0.35)

In summary, Fig. 3, 4 and 5 shows the output by passing the required filter to get Synthetic Adverse Weather Condition. Fig. 4 shows varying droplet parameter with foggy synthetic Adverse weather condition. Augmenting your dataset with a foggy filter is a good strategy, but it's just one part of the overall approach to improve model's performance in adverse weather conditions. Regular evaluation and iteration on both data and model architecture will contribute to better results. But, for our dataset, in order to achieve a high AP score on adverse weather test dataset, only night-time and fog filter is sufficient to get a high score. Rainwater droplets filter can be used to train our network if our test cases included any rain weather scenario.

B. Training a Model

We converted our Dataset to KITTI format, Google Cloud and VM instance has been used for the Training our Model. We created Virtual Machine to use GPU(GPU Type: Nvidia T4) to train our model and established SSH connection through gcloud CLI and also used it with VSCode. As described in [2],proposed MonoCon method is designed by three main components:

1. Deep Neural Network(DLA-34 [2]) as feature backbone.This is used to compute a output feature map F of dimensions D X h X w, where D is the output feature map dimension, h and w are evaluated by the stride in backbone.

2. 3D Bounding Box Regression Heads, Liu and Xue [2] proposes to determine the projected 3D bounding box center in the image plane using anchor-offset formulation.The regression head for computing the offset vector between 2D bounding box center and 3D box center, evaluation of depth, shape dimension, observation angle ,and the prediction 3D bounding box proposed by Liu have been implemented in our code.

3. The Number of regression head branches for learning auxiliary contexts.

Five loss functions have been used which are: (i) Gaussian kernel weighted focal loss for heatmaps, where Gaussian kernels model ground truth points. It helps focus on hard foreground examples, (ii)Laplacian aleatoric uncertainty loss for depth estimation using a Laplace distribution. It incorporates the predicted depth uncertainty, (iii)Dimension-aware L1 loss for shape dimensions redistributes the standard L1 loss based on the predicted dimensions to be more IoU-oriented, (iv)Standard cross-entropy loss for bin index in observation angles, (v)Standard L1 loss for various outputs (offset vectors, angle residuals, 2D box sizes, quantization residual). The overall loss is a weighted sum of the individual losses(weight = 1), with the 2D size L1 loss(weight = 0.095) given a smaller weight than the rest.

Table 1 shows the configurations and parameters that we have taken to achieve high accuracy for monocular 3D object detection.

TABLE I
CONFIGURATIONS

Configurations	Parameters
Batch Size	8
Num_Workers	4
DNN Layers	34
IMAGENET_PRETRAINED	True
Number of Classes	3
Maximum number of Objects	50
Learning rate	2.25E-04
Weight decay for Lr	1E-05
Number of Epochs	150
Schedule	True
Type of Norm for gradient	2.0
Maximum norm value for gradient	35
FILTER.MIN_HEIGHT	25
FILTER.MIN_DEPTH	2
FILTER.MAX_DEPTH	65
FILTER.MAX_TRUNCATION	0.5
FILTER.MAX_OCCLUSION	2
Global Configuration GPU ID	0
Global Configuration Cuda Benchmark	True
Global Configuration Seed	-1

C. Results

We initially ran our MonoCon with 50 epochs(time taken: 2.5 hours) without changing any parameter/configurations, achieving low accuracy. Next, we ran same with 150(8.5 hours) epochs, while achieving a high accuracy on normal test dataset whereas securing low AP(6.2673) on test dataset with adverse weather conditions. Finally for our proposed method i.e applying filters to our training dataset to train our network to account for the adverse weather condition secured comparatively high AP(15.2455) score(150 epochs - 8.5 hours) for the test dataset with adverse weather conditions and similar AP score to previous method.

Table 1 shows the configurations and parameters that we have taken to achieve high accuracy for monocular 3D object detection. Table 2 shows results for the proposed method.

TABLE II
RESULTS(PROPOSED METHOD)

Overall	AP40@	easy	moderate	hard
bbox	AP40:	26.5682	26.8746	25.9696
3d	AP40:	0.5069	0.2026	0.1796
aos	AP40:	26.31	26.32	25.39

III. CONCLUSION

This project proposes Monocular 3D object detection for Adverse Weather Conditions for Self-Driving application without using extra information from other sensors like LiDAR. We have used a modified version of MonoCon method which uses auxiliary monocular contexts to estimate 3D bounding box for each object instance for example car in a given 2D raw camera image. Furthermore, we propose another method to transform our dataset using foggy and Droplet filter which extends the MonoCon method to successfully deal with Adverse weather like fogs, or rain or pitch dark conditions. In future, we can explore different methods to improve our dataset as the dataset for adverse weather conditions is rather very limited. We can design a number filters accounting for rain, fog, nighttime, contrast difference to modify the existing dataset to train our network.

IV. GITHUB LINK

https://github.com/Kgabel/Final-Project-ROB535-MONOCON

REFERENCES

[1] Liu, X., Zheng, C., Cheng, K. B., Xue, N., Qi, G. J., & Wu, T. (2023). Monocular 3D Object Detection with Bounding Box Denoising in 3D by Perceiver. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6436-6446).

[2] Liu, X., Xue, N., & Wu, T. (2022, June). Learning auxiliary monocular contexts helps monocular 3D object detection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 2, pp. 1810-1818).