Monocular 3D Object Detection

Soham Shirish Phanse
Department of Mechanical Engineering
University of Michigan
Ann Arbor, USA
ssphanse@umich.edu

Soham Sachin Purohit
Department of Robotics
University of Michigan
Ann Arbor, USA
sohamsp@umich.edu

Jayaprakash Harshavardhan
Department of Mechanical Engineering
University of Michigan
Ann Arbor, USA
jharsh@umich.edu

***Abstract*—Object detection is a critical component to support autonomous driving. Autonomous vehicles rely on the perception of the surrounding environment to ensure robust driving performance and crew safety. We wish to develop affordable and accessible technology in autonomous vehicles. Monocular 3D object detection is a cost-effective solution to 3D object detection that eliminates the need for expensive sensors and computational machines. We tackle this problem using MonoCon, which learns *Monocular Contexts* using annotated 3D bounding boxes as auxiliary tasks. We provide a step-by-step approach for using this model for 3D object detection. We train this network on simulated data and derive inferences on the performance for normal and adverse weather conditions.**

***Index Terms*—Monocular 3D Object Detection, Auxiliary contexts, Deep Learning, Computer Vision**

I. Introduction

Object detection is essential in practical computer vision applications, such as autonomous driving and navigational robots. There are a lot of proposed methods for online object detection [9]. In these contexts, it is essential to realize 3D poses of objects in the environment. A variety of methods exist for tackling this problem.

LiDAR sensors are used for accurate depth measurements [4], and stereo cameras are used for stereo depth estimates, [5] which, however, are computationally intensive and costly. Our proposed approach uses Monocular 3D object detection that localizes 3D bounding boxes from a single 2D RGB input image and has emerged as a promising alternative with much lesser computational intensity. In the process, it also unravels one of the fundamental questions in computer vision: whether it is possible to reconstruct 3D objects from 2D images wherein a lot of data has been lost. For this task, we utilize the *MonoCon* method [1], which learns *Monocular Contexts* using annotated 3D bounding boxes as auxiliary tasks in training to accomplish the task of monocular 3D object detection.

The key idea is that in the annotated 3D bounding boxes of objects in an image, a rich set of projected 2D supervision signals, such as projected corner key points and associated offset vectors relative to the anchor. These signals harnessed in training induce more expressive representations for monocular 3D object detection. The 3D center of the object can be inferred with high accuracy in two prominent ways: when extra information is available, such as multi-frames, LiDAR depth measurements, etc, or monocular depth estimation results, which can be used for training or inference. However, in the absence of such information, this study is devoted to the *Monocon* approach, which first estimates the projected 3D center on the image plane (x_c, y_c) and the object depth z, which are usually separated from the 3D center location (x, y, z). Now, based on the inferred projected 3D center location and the object depth, the 3D location of the object center can be retrieved. It is assumed that the camera intrinsic matrix is known in the training and inference phases for this computation to be feasible. The model parameterizes a 3D bounding box in the KITTI benchmark [7] by:

1) the location of the center of the 3D object (x, y, z) (in meters) in the camera coordinates
2) the observation angle of the object α with respect to the camera frame origin
3) the shape dimensions (h, w, l), that is, height, width, and length (in meters) based on the position vector of the 3D object center from the camera center

The model has a simple end-to-end architecture with three components: a *Deep Neural Network* (DNN)-based feature backbone, regression head branches to learn the parameters used in 3D bounding box prediction, and others to learn auxiliary contexts.

KITTI is a popular computer vision dataset designed for autonomous driving research. The KITTI dataset includes different modalities and sensors, such as LiDAR, stereo cameras, and GPS/INS sensors, and it provides a comprehensive view of the surrounding environment around the vehicle. The dataset is divided into several classes, each with its challenges. These classes include object detection, tracking, scene understanding, visual odometry, and road/lane detection.

The contributions of this project are as follows-

1) We provide a step-by-step method for implementing MonoCon, the state-of-the-art in cost-effective monocular 3D object detection
2) We determine ideal training parameters for obtaining a reasonable accuracy on test data and derive inferences from the results of the training

The paper is organized as follows. Section II provides the MonoCon model with details about its structure. Section III provides the computing environment setup needed for training the model, including all the dependencies the model requires. Section IV details our model's training and testing, including

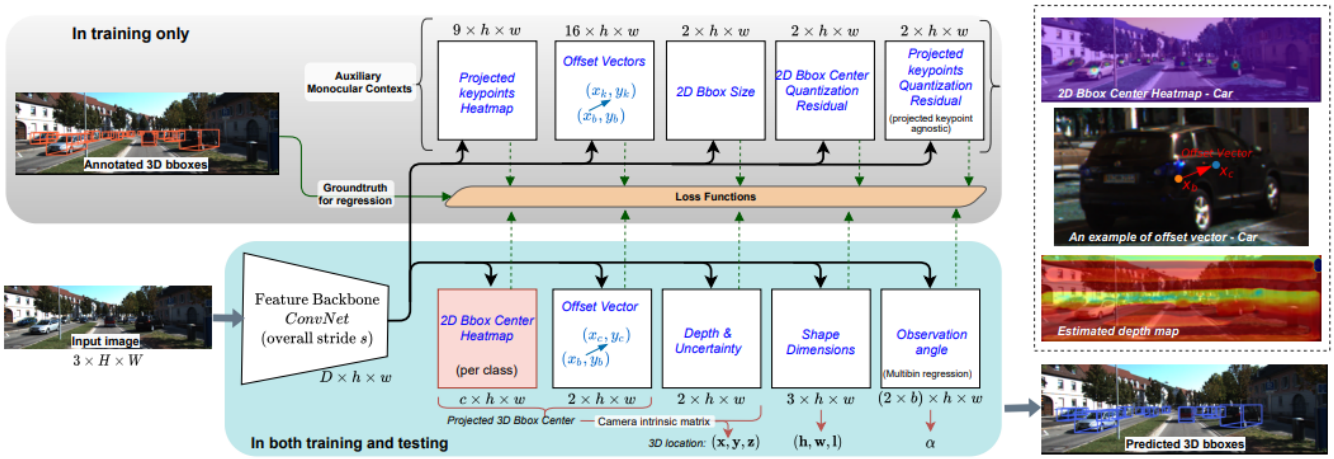

Fig. 1. MonoCon approach for 3D object recognition [1]

our model's results. We finally conclude and provide future directions in Section V.

II. Model Description

The model (Fig. 1) architecture is fundamental design - and is combined with a Deep Neural Network augmented with primary and auxiliary regression branches. The Convolutional Neural Network (CNN), which acts as the model backbone, when given an input of an RGB image of dimensions $3 \times H \times W$, after processing, converts it into an output feature map of sizes $D \times h \times w$. Here, D is the output feature map dimension, $h = H/s$, and $w = W/s$, with s being the feature backbone's overall stride/sub-sampling rate (e.g., $s = 4$). Next, a simple and direct method uses regression head branches.

To infer the three-dimensional bounding box, one set of these branches is used to estimate critical 3D parameters — three-dimensional locations, shape dimensions, and observation angles. In contrast, the other set of the regression branch is used for auxiliary contexts. In the intermediate results, the 2D bounding box center heatmap is class-specific (i.e., which depends on the object's class under consideration), whereas the others are not. The auxiliary branches are only included in the model architecture during the training phase; they are eliminated during testing since the suggested *MonoCon* is trained end-to-end and also because it improves testing efficiency. The intermediate results for the three regression branches are displayed at the right-top in Fig. (1) (notice that the depth map will only be used sparingly based on the 2D bounding box centers found).

III. Computing Environment Setup

We explore the possibility of using a VM (Virtual Machine) on Google Cloud with one GPU (Graphics Processing Unit) to improve the speed of our simulations, as there are resource constraints when using Google Colaboratory for large computations. The virtual machine recommended for this model would be a deep learning VM connected via SSH (Secure Shell) on the Debian operating system. We utilized Microsoft Visual Studio Code (VSCode™), which was connected to the Deep learning VM via an SSH connection, which improved ease of use by eliminating the need to use the VM terminal and bash scripting. The specifications we used for our virtual machine were 16GB RAM (Random-access memory), 100 GB ROM (Read-only memory), 2 Intelx86-64™bit core CPUs, and a GPU. Essential prerequisites like jupyter-notebook, Python, NVIDIA driver for one T4 GPU, etc., were downloaded and installed as per the requirements using standard web browsers like Chrome and Firefox.

IV. Solution Approach

Let Λ be an image defined on the lattice, and I be the image lattice (for example, 384 $\times$ 1280 as provided and following the KITTI benchmark). As previously indicated, monocular 3D object detection aims to deduce the 3D bounding box and the label (such as car, pedestrian, or bike) for each instance of an object in image lattice $I\Lambda$. The measurement of the 3D center position (x, y, z) in meters, the shape dimensions (h, w, l) in meters, and the observation angle ($\alpha \in [-\pi, \pi)$ in radians).

The observation angle has a deeper underlying relationship with the image appearance because it directly relates the pose of the camera with the object and can be used to deduce the relative orientation between the object's body frame and the camera frame, and hence is used in the prediction of the bounding boxes. The camera intrinsic matrix is assumed to be known in training and inference.

Typically, shape dimensions and orientation are directly regressed using features computed by a feature backbone such as a Convolutional Neural Network (CNN). Additionally, it is found that each of the direct regression approaches performs well in isolation. The overall performance of the 3D bounding box prediction can be measured by using a metric termed *Average Precision* (AP), which is based on the IOU metric, which is a ratio of Intersections over unions of image predictions and labels. It is known that AP is less sensitive to the dimensions and orientation of the image shape. If the

Fig. 2. 3D Bounding Boxes (left) || 2D Bounding Squares (right)

3D center location can be accurately inferred, the AP will not significantly decrease, even if the dimensions and orientation are not accurately predicted.

We have a 3-tiered approach to solving the Monocular Object Detection problem, namely: pre-process the dataset and convert it to the KITTI format, create data-loaders to load the data into batches for training, train the model, and generate the predictions.

A. Training

We first convert the dataset into the compatible KITTI format using the provided helper scripts, and create dataloaders to create batches of the training data. We set the following training parameters and train the model over the dataset using `train.py`:

1) `batch_size` = 8
2) `lr` = 2.25E-04 : Learning rate for the optimizer
3) `weight_decay` = 1E-05 : model weights decay rate
4) `num_epochs` = 200 : number of epochs for training
5) `num_classes` = 3: Number of distinct classes of objects: Car, cyclist, person

An interesting feature incorporated into the code by the authors includes a model-saving step - i.e., saving the model's state and weights after every pre-set number of epochs. Not only does this allow training to be completed in distinct sessions, but it also allows the user to choose different epochs of the same model to try and experiment and obtain the best possible predictions. It also bypasses the extra computation to track the validation loss for early stopping to prevent model over-fitting.

B. Testing and Inference

In the next phase, we test our model over the testing data using '`inference.py`', derived from analyzing the workflow in '`test.py`' and '`monoconengine.py`' scripts. The '`inference.py`' script loads the test data and passes it through the model to create the prediction outputs. However, an important caveat is that the outputs are not in the KITTI format. Hence, we make use of the helper script, '`kitti_3d_to_file()`', to convert the output into the standard KITTI format, which contains a '`.txt`' file with the object class label, bounding box parameters, observation angle, and object dimensions. An important thing to note is that the `inference.py` only produces the outputs of a single sample at a time and saves it to the desired path. We use the `merge.py` file to merge all the outputs into a single file, which contains the prediction outputs of all the test samples and which will finally be used in the submission.

V. Results

We tested our model on 423 simulated images with clear weather and 414 images with foggy conditions. For the normal weather images, we obtained an AP score of 15.5671; for the adverse weather images, we achieved an AP score of 8.5965. As expected, the model performs significantly worse for images with foggy weather.

VI. Conclusion and Further Work

The project presents a straightforward yet efficient approach for 3D object recognition in monocular vision without utilizing any additional data. We obtain reasonable results over images that do not have adverse weather, indicating robust performance when the weather is clear. However, improvements are needed for adverse weather conditions, which may or may not work through parameter tuning in MonoCon. For these situations, [10] proposes a monocular 3D detection model designed to perceive twin depth in adverse scenes, termed MonoTDP, which mitigates the reduction of detection performance in harsh environments. They introduce an adaptive learning strategy that addresses the depth/content loss in adverse regions and simultaneously estimates scene and object depth, integrating scene-level and object-level features.

References

[1] X. Liu, N. Xue, and T. Wu, "Learning Auxiliary Monocular Contexts Helps Monocular 3D Object Detection," in 36th AAAI Conference on Artificial Intelligence (AAAI), February 2022.

[2] Ekaba Bisong. "An Overview of Google Cloud Platform Services." In *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Springer, 2019, pp. 7-10.

[3] https://github.com/Armanasq/kitti-dataset-tutorial

[4] Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. Sensors, 18(10): 3337.

[5] Li, P.; Chen, X.; and Shen, S. 2019. Stereo r-cnn based 3d object detection for autonomous driving. In CVPR, 7644– 7652

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

[7] Ma, X.; Liu, S.; Xia, Z.; Zhang, H.; Zeng, X.; and Ouyang, W. 2020. Rethinking pseudo-lidar representation. In ECCV, 311–327. Springer.

[8] Cramer, H.; and Wold, H. 1936. Some theorems on distribution functions. Journal of the London Mathematical Society, 1(4): 290–294.

[9] Balasubramaniam, A., & Pasricha, S. (2022). Object Detection in Autonomous Vehicles: Status and Open Challenges. arXiv preprint arXiv:2201.07706.

[10] Li, X., Liu, J., Lei, Y., Ma, L., Fan, X., & Liu, R. (2023). "MonoTDP: Twin Depth Perception for Monocular 3D Object Detection in Adverse Scenes." arXiv preprint, arXiv:2305.10974.