Orthographic Feature Transform for Monocular 3D Object Detection

Group 17:
Zehnaseeb Ali | Shreya Krishna | Neel Macwan | Pranav Chougule

1 Introduction

The most common methods for detecting 3-D objects up to this point have heavily relied on LIDAR point clouds. However, approaches that solely use images have fallen significantly behind LIDAR in terms of accurately determining object depth. Due to the high expense of current LIDAR equipment and the subpar results of image-only methods, there is a demand for a 3D object detection method that utilizes monocular images.

To address the issues mentioned above, a 3-D object detection algorithm is employed that utilizes a single monocular RGB image as input and generates a 3-D bounding box. This approach achieves superior results on the Kitti Benchmark. To achieve this, we rely on an orthographic feature transform (OFT), which maps a set of features extracted from a perspective RGB image to an orthographic birds-eye view feature map. Hence, the OFT does not depend on depth information. It creates such a representation that is able to identify the relevant features of the image in the bird-eye-view.

In this study, the authors argue that being able to reason about the world in 3D is crucial for successful 3D object detection. The orthographic feature transform is used, which allows them to move beyond the image domain by mapping image-based features to a 3D space. This enables holistic reasoning about the spatial layout of the scene in a domain where object scale is consistent and object distances have a meaningful interpretation. To summarize, the primary concept being conveyed is that it is more advantageous to conduct the majority of reasoning using the orthographic space, which represents words and letters, as opposed to the pixel-based image domain that represents visual information and pictures.

2 Related work

3D object detection from LiDAR: The variations arise from the method used to encode the LiDAR point clouds. The Frustrum-PointNet of Qi et al. [1] and the work of Du etal. [2] operate directly on the point clouds themselves. Minemura et al. [3] and Li et al. [4] instead project the point cloud onto the image plane. Other methods, such as TopNet [5], BirdNet [6] and Yu et al. [7], discretize the point cloud into some birds-eye-view (BEV) representation. In this piece of work we convert perspective view to orthogonal view and no point cloud is involved.

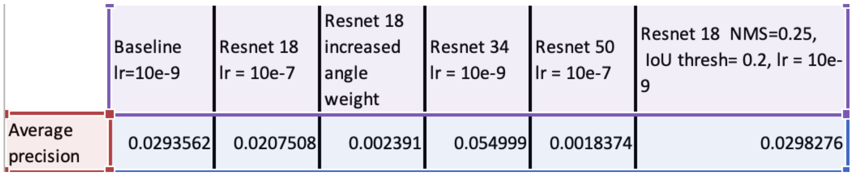
3D object detection from images: One study closely related to the research is Mono3D [8], which densely covers the 3D space with 3D bounding box proposals and then assesses each proposal's value using various image-based features.The main constraint of all the mentioned research is that every bounding box or region proposal is regarded as a distinct entity, preventing any collective analysis of the scene's 3D arrangement.

Single-Stage Monocular 3D Object Detection via Keypoint Estimation [SMOKE][9] :The proposed approach in the paper utilizes a combination of a keypoint estimation and regressed 3D variables to forecast and consolidate a 3D bounding box for all recognized objects.

3 Baseline

The algorithm consists of five main parts. First, a ResNet feature extractor is used to extract feature maps from the input image at multiple scales. Second, an orthographic feature transform is used to convert the image-based feature maps at each scale into an orthographic birds-eye-view representation. Third, a top-down network, which is made up of several ResNet residual units, processes the birds-eye-view feature maps in a way that is invariant to the perspective effects seen in the image. Fourth, a set of output heads generates a confidence score, position offset, dimension offset, and orientation vector for each object class and location on the ground plane. Finally, a non-maximum suppression and decoding stage is used to identify confidence map peaks and generate discrete bounding box predictions.

This experiment uses the KITTI 3D object benchmark dataset, which comprises 15000 images. The split for training images, validation images and test images is 3712, 3769, and 7518 respectively. The model was trained for 600 epoch with a learning rate of 10e-9 using ResNet 18 as the frontend network. The experiment metrics used was masked L1 loss and heatmap loss. The accuracy metric used was Average precision using Intersection over Union(IoU) of the bounding boxes. The training took 3 days on full dataset.



	Baseline lr=10e-9	Resnet 18 lr = 10e-7	Resnet 18 increased angle weight	Resnet 34 lr = 10e-9	Resnet 50 lr = 10e-7	Resnet 18 NMS=0.25, IoU thresh= 0.2, lr = 10e-9
Average precision	0.0293562	0.0207508	0.002391	0.054999	0.0018374	0.0298276

Figure 1: Testing Results of different approachs

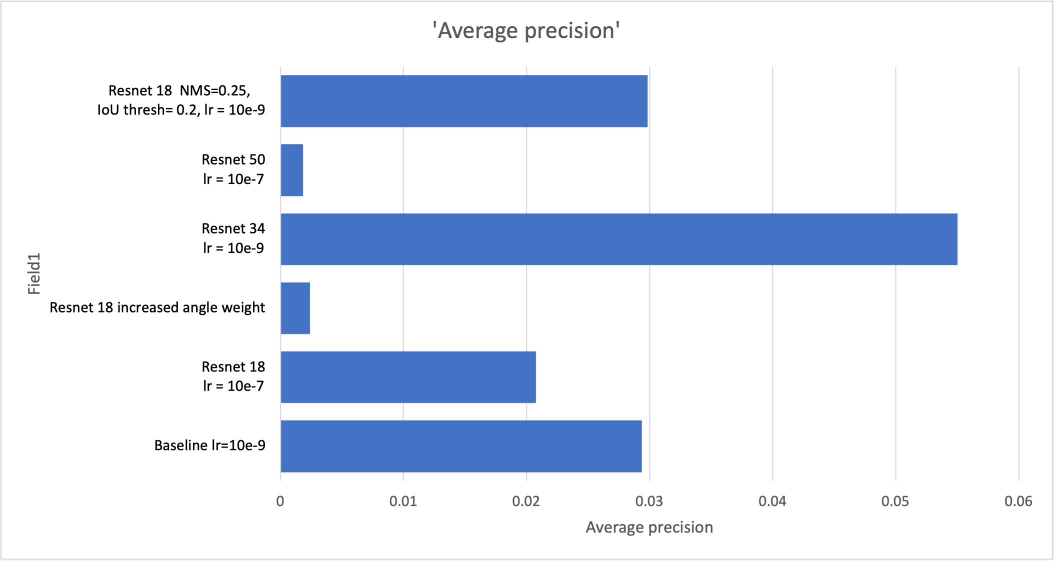


Figure 2: Visualization of the above given table

4 Approach

During our object detection experiments using the KITTI dataset, we initially trained the model for just 100 epochs, but observed many inaccurate bounding boxes with incorrect orientation. We then tried training on half of the dataset, but the results were even worse with many irrelevant bounding boxes. To address this issue, we increased the number of training epochs to 600, which resulted in a decrease in the number of irrelevant bounding boxes by approximately 10%. However, the highest IoU we were able to achieve was only 26%, and most test images had an IoU of 0, indicating poor overall performance. Although we observed some object detection, there were significant problems with the orientation, dimensions, and overall logic of the detected objects.

After observing the unsatisfactory performance of our baseline model, we focused on improving feature extraction. We speculated that utilizing a superior network could potentially lead to better results, so we tried training our model using resnet34. Although, we observed almost the same loss convergence as before (figure 3), the testing data resulted in a much better average precision numbers. The bounding boxes were still a little disoriented in the visuals. For this testing we introduced non-maximum suppression to reduce the number of irrelevant bounding boxes and improve the model's overall reasoning. We further refined our results by adding another threshold to the IoU. We then attempted to further improve the model performance by utilizing resnet50 and increasing the number of layers, but this resulted in worse numbers than ResNet18. To address this issue, we experimented with decreasing the learning rate by 100 times, and during testing. But this also did not give a better result.

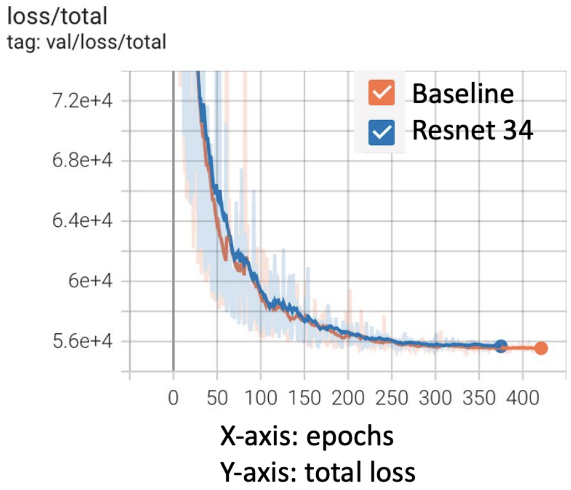


Figure 3: Loss plot comparison

Our testing visuals exhibited two problems. Firstly, images captured in dim light were not being detected accurately, and sometimes not at all. We concluded that the training dataset lacked enough dim light images, and therefore trained more on dim lighted images. This improved the detection accuracy in dim light conditions. Secondly, we observed that the orientation of bounding boxes was not as accurate as the ground truth. To address this, we trained the data using tweaked angle weights. However, this slowed down the convergence of angle loss further, as shown in the figure 4.

Out of all the approaches we tried, the Resnet34 model proved to be the most effective for our task. It demonstrated exceptional car detection capabilities, even at greater depths, outperforming all other models we trained. By analyzing the visual comparisons of the baseline testing results and the improved testing results, shown in figures 5 and 6 respectively in the appendix, we can clearly see the remarkable difference in performance between the Resnet34 model and the other models we tested.

We attribute the superior performance of the Resnet34 model to its relatively shallow architecture, which allows it to better learn reasoning during training. Compared to deeper models, such as Resnet50 or Resnet101, the Resnet34 model has fewer layers, making it less prone to overfitting and allowing it to extract meaningful features more efficiently.

Overall, our findings suggest that the Resnet34 model is an excellent choice for car detection tasks, especially when dealing with large datasets or complex environments where detecting cars at greater depths is crucial. We believe that this model will continue to be a valuable asset in the field of computer vision and image processing.

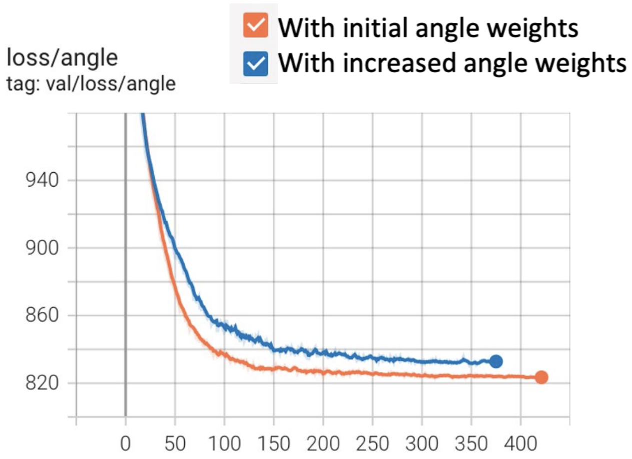


Figure 4: Angle Loss plot

5 Conclusion and Future Work

The performance of our model has significantly improved after implementing the necessary changes. It can now accurately detect cars, and even produce detections at a greater depth than the ground truth using the orthographic feature transform, which is a remarkable improvement compared to general monocular 3D object detection using Lidar. In the future, we plan to modify the encoder code to enable our model to classify multiple classes of objects, leading to better precision. Although our model can detect objects accurately, the orientation of bounding boxes is still not perfect. We believe that normalizing the angle weights may improve results. Additionally, we want to explore feature extraction using models with a similar structure to resnet34 but better performance. Once we refine these aspects, we aim to test our data on recorded videos and assess the model's performance.

6 References

[1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[2] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3D object detection from RGBD data. arXiv preprint arXiv:1711.08488, 2017.

[3] X. Du, M. H. Ang Jr, S. Karaman, and D. Rus. A general pipeline for 3D detection of vehicles. arXiv preprint arXiv:1803.00387, 2018.

[4] K. Minemura, H. Liau, A. Monrroy, and S. Kato. Lmnet: Real-time multiclass object detection on CPU using 3D LiDARs. arXiv preprint arXiv:1805.04902, 2018.

[5] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3D lidar using fully convolutional network. arXiv preprint arXiv:1608.07916, 2016.

[6] S.Wirges, T. Fischer, J. B. Frias, and C. Stiller. Object detection and classification in occupancy grid maps using deep convolutional networks. arXiv preprint arXiv:1805.08689, 2018.

[7] S.-L. Yu, T. Westfechtel, R. Hamada, K. Ohno, and S. Tadokoro. Vehicle detection and localization on birds eye view elevation images using convolutional neural network. In IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), volume 5, 2017.

[8] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3D object detection for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2147–2156, 2016.

[9] .Zechen Liu, Zizhang Wu, Roland Toth; SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020.

7 Appendix

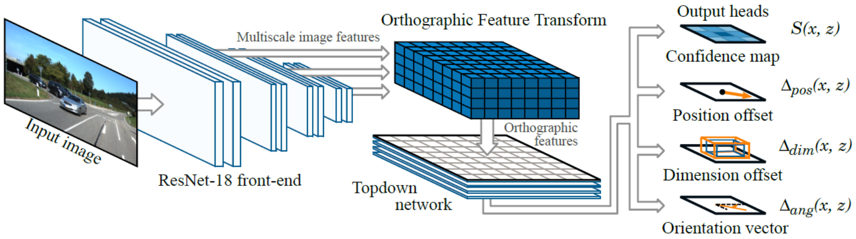


Figure 5: Testing Results of Baseline

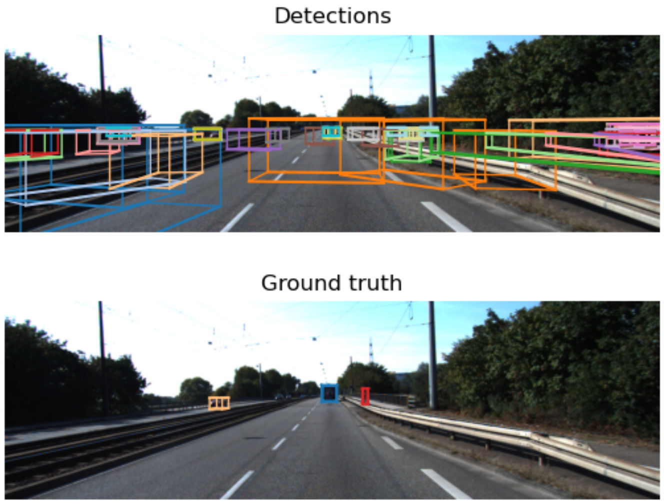


Figure 6: Testing Results of Baseline

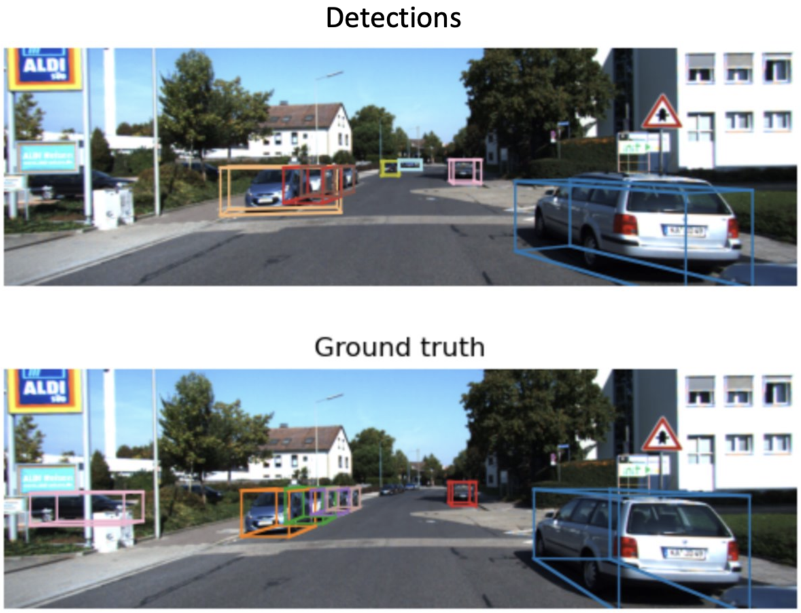


Figure 7: Improved Testing Results on Resnet 34