3D Object Detection With Stereo Vision and Transformer

Authors

Abid Ahsan Samin, 170041024

Abdullah Hassan, 170041019

Md. Rakib Hossain khan, 170041006

Supervisor

Dr. Md. Kamrul Hasan

Professor, Systems and Software Lab (SSL)

Department of Computer Science and Engineering

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc.

Islamic University of Technology (IUT)

Department of Computer Science and Engineering (CSE)

Academic Year: 2020-2021

April, 2022

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and simulations carried out by **Abdullah Hassan**, **Abid Ahsan Samin**, **Rakib Hossain Khan** under the supervision of **Dr. Md. Kamrul Hasan**, Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Abid Ahsan

Abid Ahsan Samin, 170041024

Abdullah Hassan

Abdullah Hassan, 170041019

Rakib

Md.Rakib Hossain Khan, 170041006

Supervisor:

Md. Kamrul Hasan

Dr. Md. Kamrul Hasan

Professor,Systems and Software Lab (SSL)

Department of Computer Science and Engineering

Islamic University of Technology (IUT)

Acknowledgements

We would like to express our grateful appreciation for **Dr. Md. Kamrul Hasan, PhD**, Professor, Department of Computer Science & Engineering, IUT for being our adviser and mentor. Without his support and proper guidance this research would never have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him. We are also grateful to **Dr. Hasan Mahmud, PhD**,Assistant Professor, Department of Computer Science Engineering, IUT for his valuable inspection and suggestions on our work. His motivation, suggestions and insights for this research have been invaluable. Without his guidelines this Research wasn't possible.

Abstract

Identifying 3D objects with computer vision in a precise manner has been a challenging task in the field of autonomous driving. Partly because it requires proper depth estimation. Until now, Li-DAR technology has been used to achieve this task which is precise but also expensive. The introduction of pseudo Li-DAR promises an alternative approach which is cheaper with fairly good precision. However, pseudo Li-DAR can be replaced with 2D image representation with similar precision. Transformer is another technology which is widely used to process sequential data. Recent studies show that transformer can also be used for object detection purposes. In this literature, we look into the concept of pseudo Li-DAR, image representation of depth and detection transformer(DETR). Later, we introduce a new approach of using image based depth output with DETR to achieve accurate object detection. Finally, we compare our results with other available methods used for object detection in order to establish a benchmark.

Contents

1 Introduction

1.1 Overview

Self-driving cars with stereo vision are becoming increasingly popular these days. Over the last decade, the discipline of Computer Vision has exploded, particularly in the areas of obstacle detection and Computer Vision using Deep Learning.

Obstacle detection techniques like YOLO[12] and RetinaNet[7] provide 2D Bounding Boxes that show where the obstacles are in the image. Most object detection algorithms today use monocular RGB cameras and are unable to provide the distance between each obstacle.
Engineers combine the camera with LiDAR (Light Detection And Ranging) sensors, which employ lasers to return depth information, to return the distance of each barrier. Sensor Fusion is used to combine the results of computer vision and LiDAR.
The usage of LiDAR, which is costly, is a flaw in this strategy. Engineers utilize a technique in which they coordinate two cameras and use geometry to define the distance between each obstacle: That new configuration is known as a Pseudo-LiDAR. Accurate 3D identification and localization of cars, pedestrians, and other objects are required for safe autonomous driving. This, in turn, necessitates precise depth data, which LiDAR (Light Detection And Ranging) sensors can provide. LiDAR sensors are famously expensive, even though they are highly precise and reliable: a 64-beam type can cost up to $75,000. (USD).Another option is to use affordable consumer cameras to determine depth.

Despite recent spectacular progress in stereo-based 3D object identification enabled by pseudo-LiDAR (Wang et al., 2019)[16], A moral dilemma arises as a result of the trade-off between cost and safety.

1.2 Problem Statement

The problem is to find a cost effective and efficient way of detecting 3D objects with the help of stereo vision images that can perform on par if not better with available 3D object detection sensor such as Li-DAR.

1.3 Motivation Scopes

We begin by looking at the depth estimate technique that is at the basis of today's stereo based 3D detection methods (Wang et al., 2019a)[16]. The fact that depth is rarely computed directly contributes significantly to systematic depth bias. Instead, one should make an initial assessment of the situation. Discrepancy a pixel's horizontal shift between the left and right pictures and inverts it. to get pixel-by-pixel depth While deep neural networks have made a significant improvement in disparity,an estimate ,Because of the reciprocal transformation, establishing and learning networks to increase the accuracy of disparity estimation overemphasizes nearby items.

A unit disparity error (in pixels) equals a 25cm depth error for a 15-meter-away object: the length of a side mirror. On the other hand, the same disparity mistake for a 60 meter away object becomes a 6.7-meter depth error: a objects length. Because both errors are penalized equally, the network spends more effort correcting subtle errors on nearby items than massive errors on faraway objects, resulting in

deteriorated depth estimations and, as a result, poor identification and localization for faraway objects.

We suggest modifying the stereo network architecture and loss function for direct depth estimation. But this approach has some drawbacks, we cant use contextual information from point cloud. so either we need some approach to fuse coordinate information with image or a architecture which can fuse. But requires a huge amount of effort and experiments to use any new approach. Using the fundamental notion of convolutions as a jumping-off point.[**?**]

Rethinking Pseudo LiDAR representation (Ma et al. 2020) [8] shows in the importance of coordinate system transform. The following are the contributions of this paper: First, they established by substantial experimental proof that it is the coordinate system transformation, not the data representation, that makes the pseudo-LiDAR representation effective[8]. Second, they discovered that using pseudo-LiDAR representation to boost detection performance isn't essential. Image representation-based algorithms can also achieve competitive, if not higher, performance when incorporating spatial coordinates. They obtained state-of-the-art performance and demonstrate the promise of image representation based 3D detectors owing to increasingly powerful picture-based deep learning algorithms.[8]

1.4 Research Challenges

There's some challenges in this task of 3D object detection. Every research brings it's challenges with it. In 3D object detection localizing a object in global coordinate in hard task to do when you don't have spatial information. For this reason 2D detectors, which performs amazingly in localizing 2D objects in image, performs poorly. for this reason researchers introduced Pseudo LiDAR[16] which uses stereo images to generate spatial information but CNN or 2D object detectors can use contextual information which can't only be done using.

But this approach has some drawbacks, we cant use contextual information from point cloud. so either we need some approach to fuse coordinate information with image or a architecture which can fuse. But requires a huge amount of effort and exeperiments to use any new approach.

- **Incompleteness in data:**
  - Caused by occlusions between objects and cluttered background.
- **Day and night lighting condition**
  - LiDAR performs way better in night condition compared to image based methods.
- **Limited dataset**
  - 3D detection required labeled orientation and bounding box,for precise detection other sensor data is also needed.

2 Background Studies

2.1 Depth Estimation

We can calculate an object's distance using two cameras. This is the fundamental geometry of stereo vision, and it is based on the notion of triangulation. It works like this:

- **Stereo Calibration:** We have retrieve camera matrix from calibration files:

$$\lambda. \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_{c0} \\ 0 & d & y_{c0} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{1}$$

- **Epipolar Geometry:** Some constructive maths is used to get depth,width and heigh in real world coordinates.

$$z = D(u.v)(depth) \tag{2}$$

$$x = \frac{(u - c_U).z}{f_U}(width) \tag{3}$$

$$y = \frac{(v - c_V).z}{f_V}(height) \tag{4}$$

- **Disparity Mapping:** Compute the Disparity Map

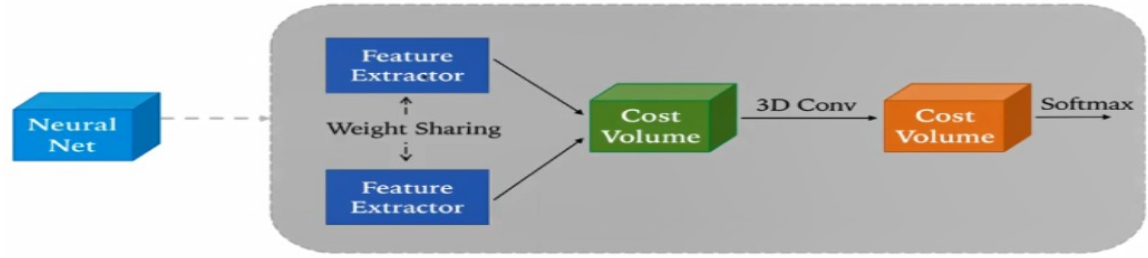
$$d = x_L - x_R \tag{5}$$

here disparity d is the x axis difference between same point projected on right image and left image.

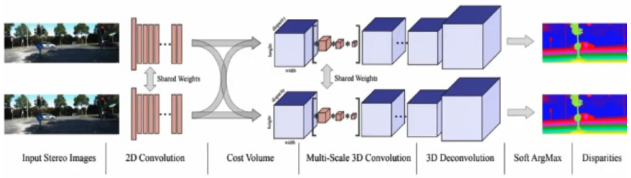
- **Depth Mapping:** from the last step we found disparities , now if we use focal distances we can derive the depth in the following equation.

$$Z = \frac{f.b}{x_L - x_R} = \frac{f.b}{d} \tag{6}$$

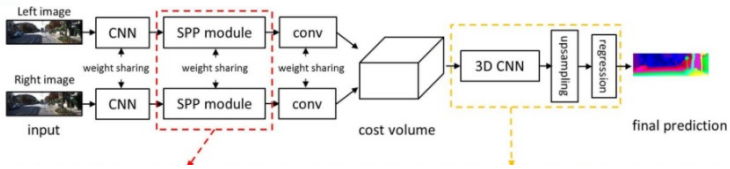
- **Obstacle Distance Estimation:** Find objects in 3D, and Match with the Depth Map,The x axis difference between the identical point projected on the right picture and the left image is called disparity d.



(a) Depth estimation network pattern



(b) GC-net[5]



(c) Psm-net[3]

2.2 3D object detection

Specifically, we experiment on AVOD [16] and frustum-PointNet[10] , the two top ranked algorithms with open-sourced code on the KITTI benchmark. In general, we distinguish between two different setups:

- In the first setup we treat the pseudo-LiDAR information as a 3D point cloud. Here, we use frustum Point-Net [10], which projects 2D object detections into a frustum in 3D, and then applies PointNet[10] to extract point-set features at each 3D frustum.
- From a top-down perspective, the 3D data is turned into a 2D image: depth and width become the geometric dimensions, while height is stored in the channels. Visual features and BEV LiDAR data are connected to 3D box suggestions by AVOD, which then merges the two to conduct box classification and regression.

2.3 Deep Learning Methods

2.3.1 Pseudo Li-DAR

3D Object detection on an image requires accurate depth information to superimpose bounding boxes. This depth information is usually obtained by two different ways. First method is from optical imagery either from mono or stereo camera system. Another method is from physical Li-DAR sensor. Li-DAR uses a 64 or 128 sparse rotating laser beams to create a 3D dot projection of the spatial environment. Studies[16],[4],[3] shown that depth information obtained from stereo images are more error prone than that of physical Li-DAR sensors.

Especially when there are objects far away. Stereo depth imagery groups far away object pixels together. It makes depth estimation harder for further objects. When 2D convolution is applied to these images the pixel representing the actual object gets distributed in a wider space. The result is detection far away from the actual position of the object (Fig.1). It is estimated that the poor performance of the stereo image based depth estimation comes from poor information quality of the images. However, Wang[16] argues that it is not the poor information quality but poor representation of the information that causes this poor performance. Wang[16] in their paper proposes a two step approach (Fig.2).

First, the depth map with a lot of pixels from the stereo images are back projected into point cloud representation. This representation is named as pseudo Li-DAR as it tries to mimic to output of a physical Li-DAR sensor. Then, this point cloud information is used in available 3D object detection algorithms. This

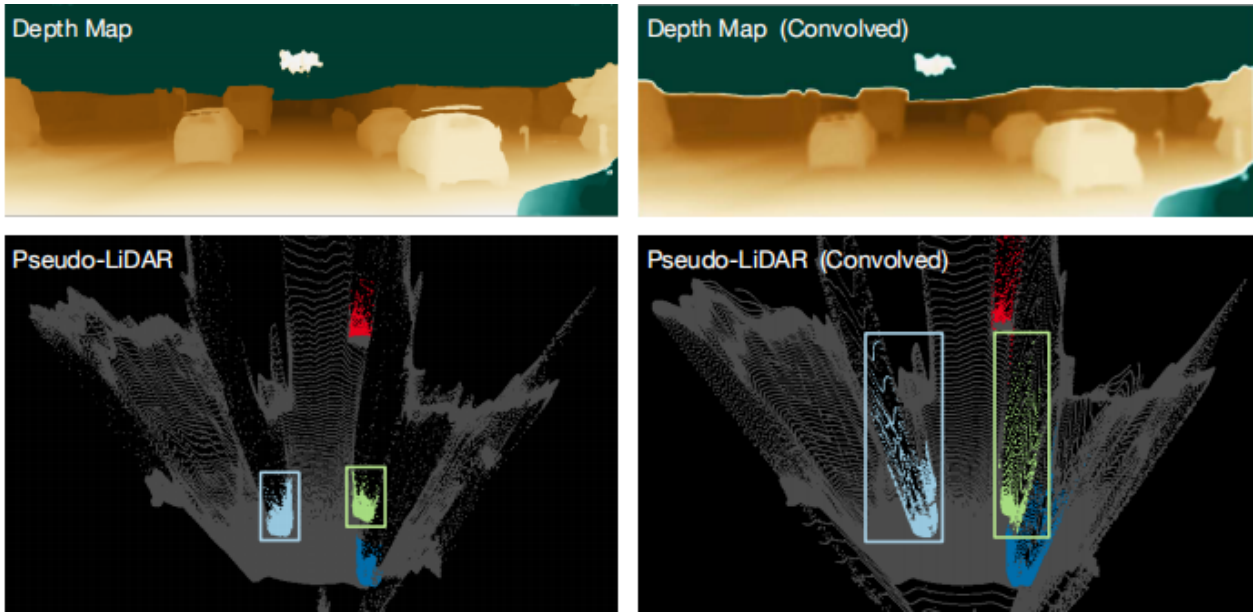


Figure 2: The pixels representing far away objects gets flattened after 2D convolution is applied.[16]

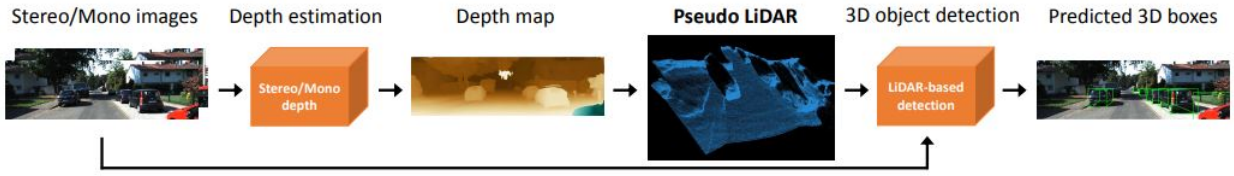


Figure 3: The model proposed for pseudo Li-DAR implementation. A depth map is generated from stereo/mono images. Then back projecting 3D co-ordination for each pixel a point cloud is generated. Finally, the point cloud is used for 3D object detection model.[16]

approach shows a phenomenal result. A 74% accuracy over the then state-of-the-art 22% on KITTI object detection benchmark. However, this method still struggles with far away objects. Further work has been done to mitigate this weakness which we will see in later part of this literature.

The approach is to take image output from two cameras. Let's say they are I_l and I_r. There is a horizontal offset for both cameras which is b. If we assume I_l as reference then we record the amount of pixel shift on I_r on Y. This is the disparity map. The horizontal focal length f_U is known. With all these information

we can calculate the depth map from the equation given below:

$$D(u_{distance}, v_{distance}) = \frac{f_U.b}{Y(u_{distance}, v_{distance})} \tag{7}$$

Up until pseudo Li-DAR this depth information is concatenated as additional channels with the input image. On the contrary, pseudo Li-DAR uses this information to find out precise 3D co-ordination for each pixel. The 3D co-ordination can be found by the following equations:

$$z = D(u.v)(depth) \tag{8}$$

$$x = \frac{(u - c_U).z}{f_U}(width) \tag{9}$$

$$y = \frac{(v - c_V).z}{f_V}(height) \tag{10}$$

A combination of depth estimation algorithm is applied to generate pseudo Li-DAR point cloud. Among them, Pyramid Stereo Matching Network (PSMNet)[3] proved to be most accurate. Then the point cloud is used with Frustum PointNet[10] and AVOD[6] 3D object detection algorithm.

These two algorithm is designed to use physical Li-DAR sensor data. Intersection over Union (IoU) is used to validate the output. With an IoU threshold of 0.7 and moderate on car category of KITTI object detection benchmark pseudo Li-DAR gives AP_{3D} of 56.8% on AVOD and 51.8% on F-POINTNET. This is a better result from stereo only approach such as MLF-Stereo[17] which only gives AP_{3D} of 19.5%. However, using physical Li-DAR still yeilds better result on AVOD and F-POINTNET.

2.3.2 Pseudo Li-DAR++

Although pseudo Li-DAR shows promising performance in easy and moderate level of difficulty in KITTI object detection benchmark, it struggles with the hard difficulty.

This means pseudo Li-DAR can not accurately identify far away or occluded objects. The authors of pseudo Li-DAR++[18] shows that this weakness can be overcome by introducing a stereo depth network (SDN) which takes account depth correction instead of disparity correction for depth estimation.

A single pixel error in disparity indicates just a 0.1m error in depth at a depth of 5 meters, but a 5.8m error at a depth of 50 meters when utilizing the KITTI dataset's parameters(Fig. 3). The SDN focuses less time on correcting nearby object depth error and shifts attention to far away objects which are more critical.

In addition to SDN, the authors also introduce a 4-beam physical Li-DAR sensor. This sensor itself can not provide sufficient data for reconstructing a 3D object in point cloud.

However, concrete depth measurement of certain points helps the stereo based depth estimation algorithm fine tune its prediction. A graph assisted depth correction (GDC) algorithm is introduced to diffuse the sensor data with the stereo vision depth estimation.

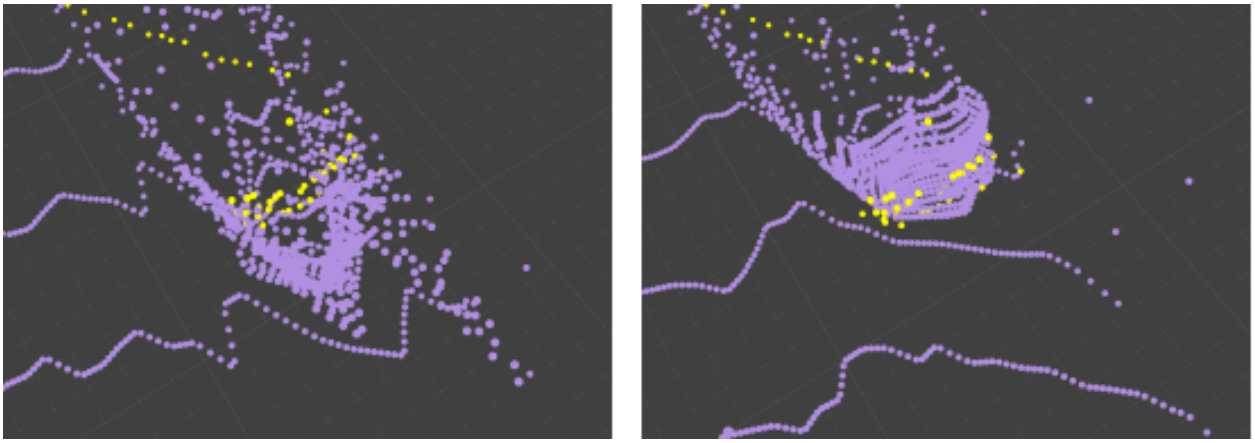


Figure 4: Cost volume of disparity (left) vs. cost volume of depth (right). From a bird's eye view, the 3D points generated from LiDAR (yellow) and stereo (pruple) corresponding to an automobile in KITTI are shown in the figure (BEV). The disparity cost volume's points are stretched out and noisy, but the depth cost volume's points correctly represent the automobile contour.

Error in depth increases in quadratic order in proportion to disparity. To counter this issue, the stereo network directly optimizes the depth loss.

$$\sum_{(u,v)\epsilon A} l(Z_{distance}(u_{distance}, v_{distance}) - {Z^*_{distance}}^2(u_{distance}, v_{distance})) \tag{11}$$

Here, Z represents the estimated depth and Z^* represents ground-truth depth. Z and Z^* can be obtained from $Y(u_{distance}, v_{distance})$ shown in Equation 1. This is necessary but still insufficient to overcome the problem. Next step is to construct a depth cost volume C_{depth} which will encode features describing how likely z from $Z(u, v)$ is the depth of $ImagePixel(u_{distance}, v_{distance})$. A new tensor $Standard_{depth}$ is used to figure out the pixel depth with this information.

$$Z_{distance}(u_{distance}, v_{distance}) = \sum_{z} softmax(-S_{depth}(u_{distance}, v_{distance}, z_{distance}))*z \tag{12}$$

The obtained depth is then corrected with graph assisted depth corection (GDN). Both L(Li-DAR point cloud) and PL(pseudo Li-DAR) point cloud is taken. Then,

the characteristic of local objects is determined through a K-Nearest Neighbour (KNN) graph. The L points are projected onto the pixel location $(u_{distance}, v_{distance})$ and then the predicted distance values $Z_{distance}(u_{distance}, v_{distance})$ from stereo network is optimized.

2.3.3 Rethinking Pseudo Li-DAR

It was assumed that the good performance of pseudo Li-DAR was due to the data representation. Particularly, the point cloud generated from input images. Ma[8] argues that the performance improvement from pseudo Li-DAR is not from data representation but from co-ordinate transformation. When the image co-ordinates are transformed into world co-ordinates they implicitly encode camera parameter information which then contributes to proper depth estimation.

To prove this, they have built a new model named PatchNet-vanilla. The inner workings of PatchNet-vanilla are same as pseudo Li-DAR. However, it refrains from back projects depth information in a point cloud. rather, the authors chose a image based representation technique. This approach allows them to use PatchNet output on existing 3D Li-DAR based algorithms as well as extracting deep features using 2D CNN algorithms (Fig. 4).

PachNet-vanilla takes monocular or stereo image as input and predicts depth map for each pixel (u, v) with a stand alone CNN. Another CNN is used to generate 2D region proposals. According to this proposal Region of Interests (RoI) are cropped from the dpeth map. 3D co-ordinate for each pixel of RoIs in the depth map can obtained by Equation 2,3 and 4. Instead of using this 3D co-ordination data as point cloud PatchNet-vanilla organizes the (x, y, z) values as image representation.

Given the (x, y, z) values a CNN backbone extracts deep features from the image and then filters with mask global pooling and generated mask (Fig. 5). Most

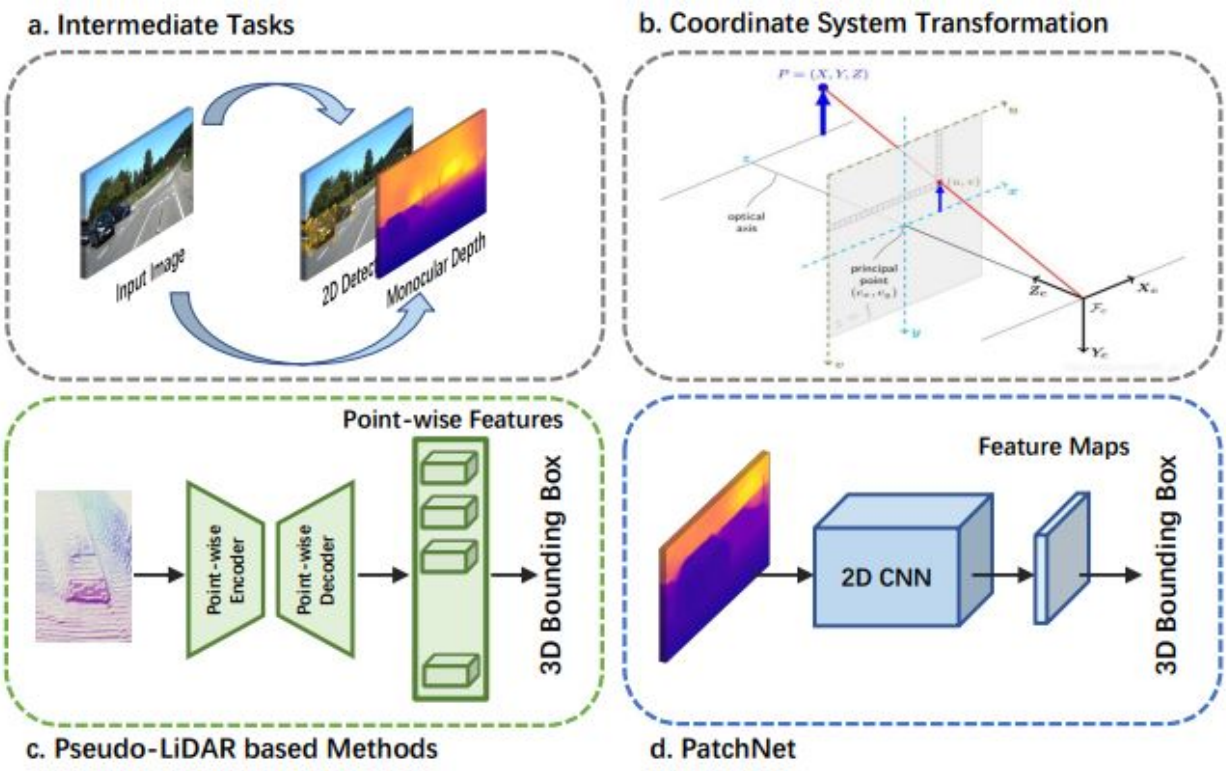


Figure 5: PatchNet and pseudo-LiDAR-based approaches are compared. They both use off-the-shelf models to produce intermediate tasks (a) and project the picture coordinates to world coordinates (b). Pseudo-LiDAR algorithms treat these data as LiDAR signals and anticipate results using a point-wise network (c). PatchNet, on the other hand, organizes them as visual representations for later analysis (d).

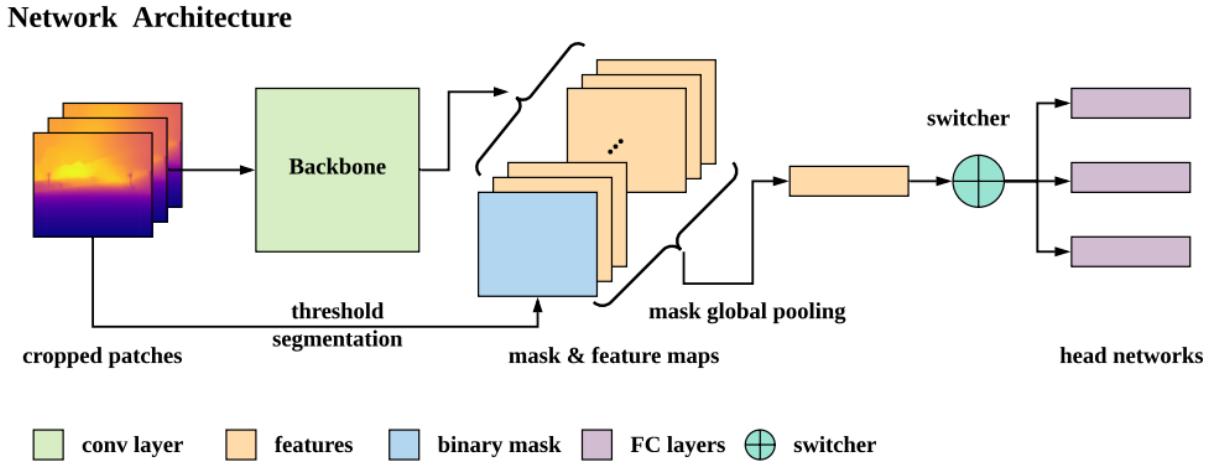


Figure 6: The network architecture is depicted in this diagram. We first build a binary mask based on mean depth from an input patch with x, y, and z channels, and then utilize it to guide the pooling layer to extract the features related to the foreground item. Then, depending on how difficult it is to anticipate, we assign examples to different head networks.

strong CNN backbone can be used for this purpose. The authors used ResNet-18 with Squeeze-and-Excitation (SE)[4] block. The pooling layers of SE-ResNet-18 has been removed so that the output is the same size as input image patches. Then mask global pooling is applied. However, unlike usual global polling only pixels in the RoIs are considered to construct the feature vector. There are 3 branches on this network according to difficulty. These 3 branches share the same network architecture.

Thus, this does not improve any accuracy but provides scope to parallel process multiple images at the cost of extra GPU power and memory. The performance of PatchNet-vanilla is on par and sometimes better when using a strong backbone than that of pseudo Li-DAR. The $AP_{3D}|_{R11}$ values of PatchNet-vanilla on KITTI object detection benchmark are 28.7, 18.4 and 16.4 respectively on easy, moderate and hard difficulty. On the contrary, pseudo Li-DAR scores 28.2, 18.5 and 16.4 on easy, moderate and hard difficulty respectively. It proves that PatchNet-vanilla can perform on par with pseudo Li-DAR solidifying the authors statement.

2.3.4 Detection Transformer (DETR)

We now look into a new type of object detection architecture called detection transformer[2]. Detection transformer considers the object detection problem as a direct set prediction problem.

Unlike popular object detection methods, it does not require any post-processing steps such as proposal or anchor generation based on known information. It uses a small set of ground truth objects and classifies objects using bi-partite matching. Direct set predictions in detection require two ingredients: (1) a set prediction loss that forces unique matching between predicted and ground truth boxes; (2) an architecture that predicts a collection of items and models their relationship in a single run. The artchitecture of DETR comprises of

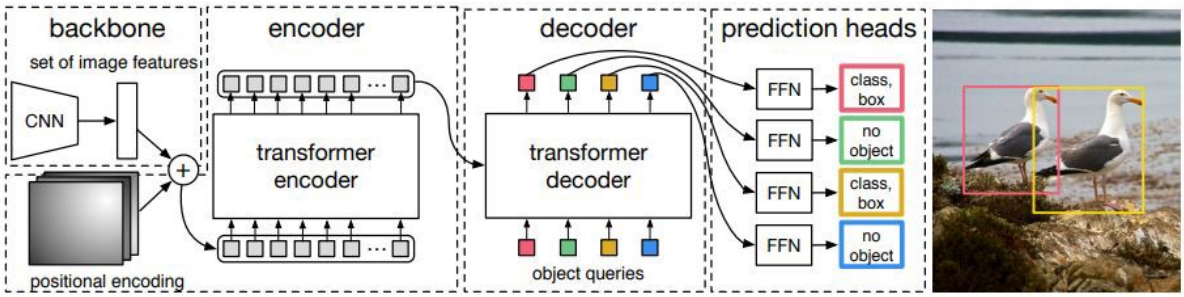


Figure 7: DETR learns a 2D representation of an input image using a traditional CNN backbone. Before feeding data into a transformer encoder, the model flattens it and adds a positional encoding. The encoder output is also attended to by a transformer decoder, which accepts as input a tiny fixed number of learnt positional embeddings, which we call object queries. We feed each decoder's output embedding into a shared feed forward network (FFN), which predicts a detection (class and bounding box) or a "no object" class.

three major components. A CNN backbone to extract compact feature vector from input image. A transformer encoder and decoder. Finally, a simple Feed Forward Network (FFN) that matches prediction boxes with ground truth boxes.

The CNN backbones generates a feature map $f \in \mathbb{R}^{C \times H \times W}$ from the initial image input. Typically, $C = 2048$ and $H, W = \frac{H_0}{32}, \frac{W_0}{32}$ where H_0 and W_0 represents initial height and width of the image. In the encoder the channel dimension of the feature map is reduced from C to a smaller value d. Then, we collapse the spatial dimensions into one dimension resulting in a $d \times HW$ feature map.

Fixed positional encoding are used at each layer of self attention module of the encoder. At the decoder, N positional embedding are processed in parallel at each attention. These are learnt embedding called object queries. The work of the FFN is to transform these N embedding into box co-ordinates and classification label. DETR performs really well against popular detection method such as Faster-RCNN.

3 Proposed Method

Objective of our method is to develop an end to end neural network that can detect 3d objects more accurately without using point cloud representation. 2D object detectors came way ahead and their research field is vast. Surely, point cloud based detectors works way better but to perform head to head using image based detector we are using a different experimental representation of data on which we can use somewhat tweaked 2D object detector. We have chosen DETR for our experiments because it has more usable cases for traffic scenarios.

3.1 Why Transformer in 3D detection

For NLP tasks, we prefer Transformer. First, we'll look at RNN. Vanishing/exploding gradients are a well-known issue, implying that the model is biased.In the present phase, the most recent inputs in the sequence, or in other words, earlier inputs, have almost no effect on the outcome.

LSTMs/GRUs primarily attempt to address this issue by incorporating a separate memory (cell) and/or additional gates to learn when to let go of past/current information.

Given this, information from previous steps must still transit through a series of computations, and we must rely on these new gate/memory mechanisms to transfer data from previous steps to the present one.

One of the key advantages of the transformer architecture is that we have direct access to all other stages (self-attention) at each step, effectively eliminating information loss in the message passing process. Furthermore, we can examine both

future and past elements at the same time, a feature of bidirectional RNNs that eliminates the requirement for 2x processing. And, of course, everything happens at the same time (non-recurrent), making both training and inference significantly faster.

Because every other token in the input requires self-attention, the processing will be on the order of $O(N2)$ (glossing over specifics), which indicates that applying transformers to long sequences will be more expensive than using RNNs. RNNs probably still have an edge over transformers in this area.

3.2 Data Representation

Our main experiment lies here,we are modifying the input data for DETR[] . From Stereo Depth Estimation network (SDN) we get a Depth map. In Pseudo LiDAR (Wang et al.)[] pipeline it was stated that that they converted the depth map using the camera calibration parameter to a set of point cloud. which only consists of (x,y,z) coordinates of the environment. But, in our experiment, we convert the depth map to (x,y,z) coordinates. We also map the pixel position to its pixels coordinate in global space.

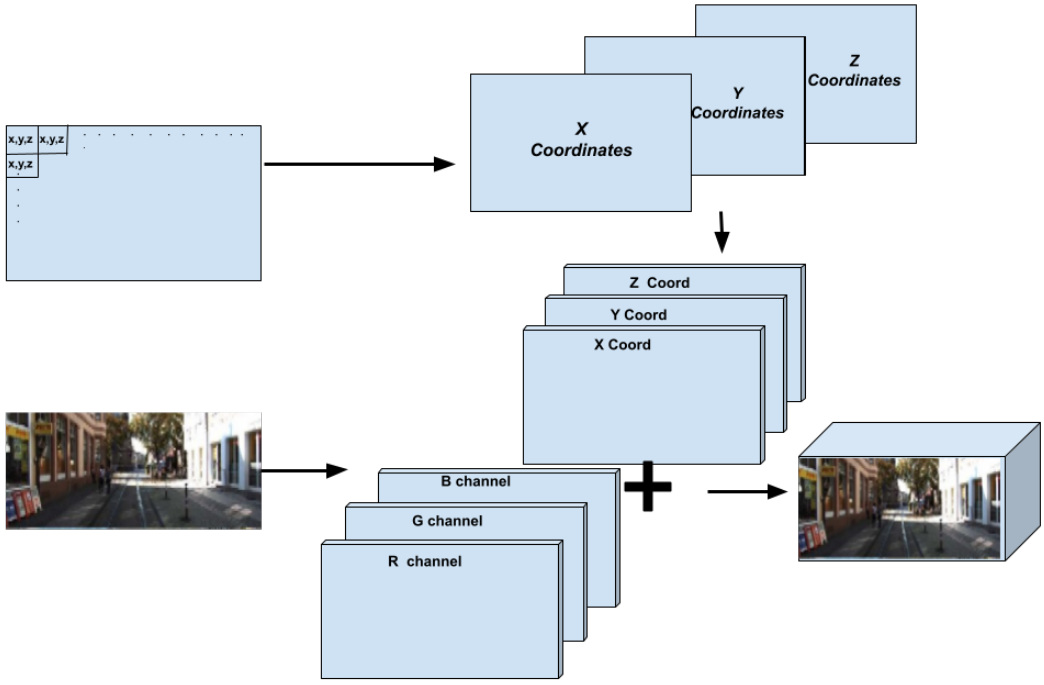


Figure 8: Pixel coordinate fusion strategy we are using here. from Depth estimation network we will get every pixels global coordinates (x,y,z). then we are separating the them into 3 channels like x channel,y channel, z cannel and concatinating with the RGB image.

3.3 Pre-processing

Data pre-processing is one of the most important steps in any network. In deep learning, data augmentation is crucial, and picture augmentation, as an integral aspect of target identification and image categorization, enhances the algorithm's performance greatly.

3.3.1 Fog and Rain Augmentaion

The data augmentation pipeline for camera pictures provided by Tremblay et al. [14] is used to create augmented data. They utilize complex mathematical models to simulate photo-realistic fog and rain on camera photos, attempting to capture the effect of rain and fog in the actual world.

Tremblay et al.citetremblay2020 rain integrate the PBR(Physics Based Rendering) and Rain is generated with a GAN-based rain generator by first translating the picture to its rainy form with the GAN, then synthesizing the rain overlay onto the resulting image with the P..[14]

3.3.2 Mosaic Augmentation

The Mosaic data augmentation algorithm in YOLOv4 picks four photographs at random from the train set and merges their contents into a synthetic picture that may be utilized for training. The model's capacity to recognize complicated backdrops can be improved using this data augmentation strategy.

The CutMix data augmentation algorithm, which is a further evolution of the CutMix data augmentation method, is referred to as the Mosaic data augmentation algorithm in YOLOv4[1]. Image flipping, scaling, and other operations on an

Figure 9: Fog and rain augmentation from the paper implementation "Robustness of Object Detectors in Degrading Weather Conditions" [9]

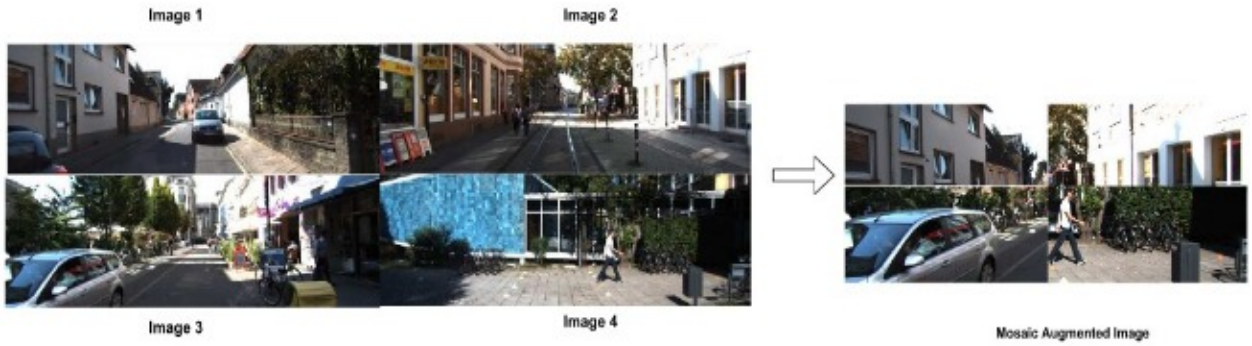


Figure 10: Mosaic Augmentation of 4 images, the number of images can vary

image are common ways of data augmentation, whereas CutMix data augmentation involves splicing two images and transferring the spliced images straight to the

neural network for training.

The Ground Truth denoting the target point will change as the size of the original image changes, such as zooming in or out. As a result, if the object changes, the real box will change as well. This algorithm's adjustment on the real box will alter in tandem with the target. The Mosaic data augmentation algorithm's coordinate processing constraint requirements are divided into the following inequality groups:

$$\begin{aligned} y_{min} &> cut_y \\ x_{min} &> cut_x \\ y_{max} - y_{min} &< m \\ x_{max} - x_{min} &< n \end{aligned} \tag{13}$$

usually this augmentation is little tricky to use. If we cutmix multiple images it is unlikely to have vehicles in the upper regions of an image. thus it won't help in that cases. We need to cautious whenever we use this augmentation strategy.

3.3.3 Bag of Freebies

Bag of Freebies is a collection of approaches or methods for improving model accuracy by changing the training strategy or cost.

These are many strategies that may be taken while offline training to improve overall accuracy without raising overall inference cost.

- Photometric Distortions Eg: Brightness, Contrast, Hue, Saturation, Noise
- Geometric Distortions Eg: random scaling, cropping, flipping, and rotation

Figure 11: Automod libary to simulate different augmentation method mentioned in bag of freebies

basically we are not using all the augmentation techniques. The augmentaion which may hamper our data representation are avoided.

3.4 Proposed Network

- Transformer[15] can work with whole context at a time in parallel, so the whole data can be processed at once just like 2D detection in DETR[2].
- Occlusions is dealt in a good way in DETR[2] paper which can be usefull for 3D detection as well.

the model we sketched is just to replaced point cloud based detector with DETR. with a transformer encoder decoder and use Bipartite loss function for training. We are using pre-trained PSMNet[3] model for depth estimation.

depth estimation network we get Dispariy map, we convert that map to depth map. After that, instead of converting the depth map to point cloud representation we are calculating the calculating corresponding (x, y, z) global points and storing it in a format like map.

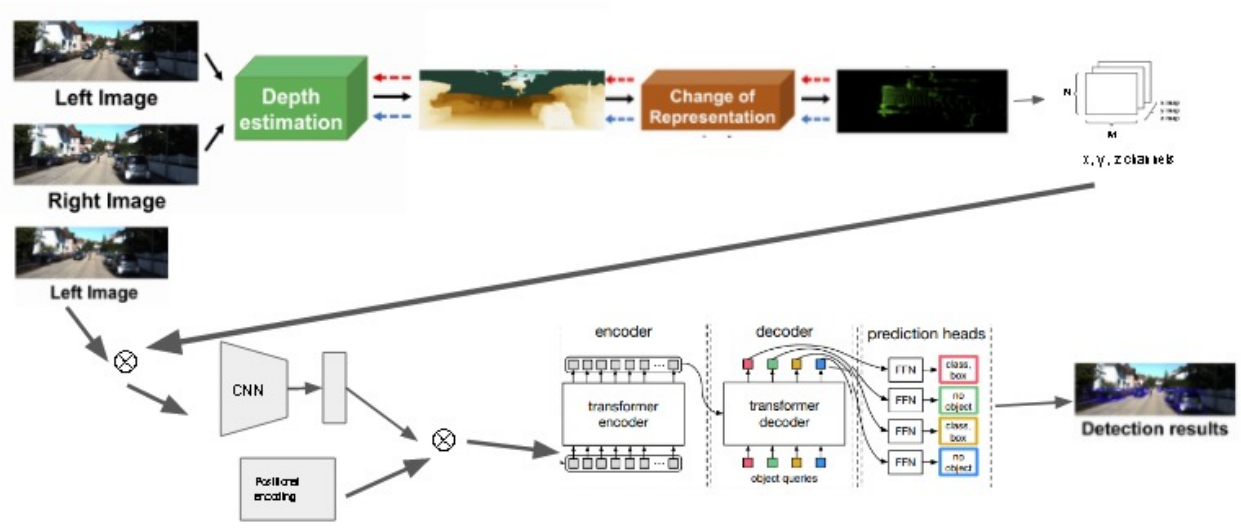


Figure 12: our proposed network DETR-pixCoord which has a PSMNet[3] as SDN(Stereo Depth Network and DETR[2] as Object Detector)

we implemented **Generalized IOU** for this task especially. which is essential for calculating 3D bipartite matching loss.

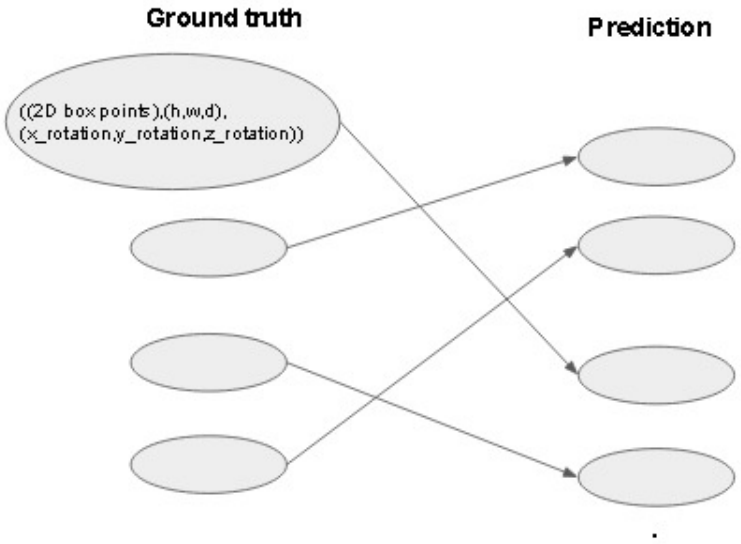


Figure 13: Bipartite matching in our proposed network,slightly modified

our proposed method used a DETR model for 3D detection. Usually DETR is used for 2d object detection but we are attempting to use it on 3D detection task,first of all the DETR model uses full image during training and inference at a sigle time pass. so, the whole image context is preserved and attention mechanism can focus on a section using the whole image as context.

4 Experiments and Result Analysis

4.1 Datasets

In our research we are using **KITTI 3D Object Detection Evaluation 2017**. A total of 80.256 labeled items are included in the 3D object detection benchmark, which comprises of 7481 training photos and 7518 test images, as well as the accompanying point clouds. Precision-recall curves are used to evaluate the system. They compute average precision to rank the approaches. For all test pairings, we require that all methods utilize the same parameter set.

They employ the PASCAL criteria, which are also used for 2D object detection, to assess 3D object detection performance. As a result, far objects are filtered based on their picture plane bounding box height. They point out that because only items visible on the picture plane are labeled, the assessment fails to account for detections that are not visible on the image plane, which might lead to false positives. They want a 70 percentage 3D bounding box overlap for autos, and a 50 percentage 3D bounding box overlap for cyclists and pedestrians.

The dataset they provide consists of:

- Pair of color images
- 3 temporarily proceeding frames.
- Velodyne point cloud if we want laser information
- Camera calibration matrix.

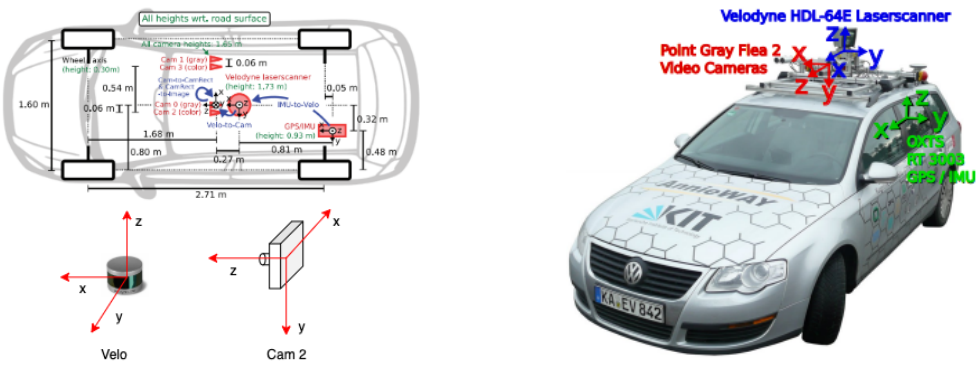
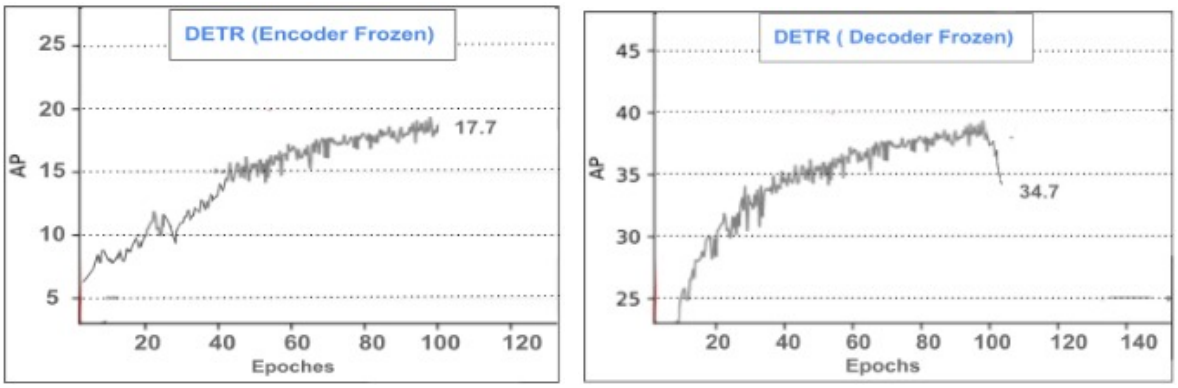


Figure 14: Automod libary to simulate different augmentation method mentioned in bag of freebies

- Training label of object data.

4.2 Training

Transfer Learning: We used pre-trained weights for our experiment. But results are not satisfactory using pre-trained weights with transfer learning. The maximum AP we got was 17.7 for frozen encoder and got 34.7. which is far from our expected outcome.



(a) Frozen dencoder training (b) Frozen encoder training

Figure 15: In this figure we are experimenting with frozen layer,we are evaluating the average precision on 120 epochs to see if the training is working.

so, we can see from the graph, transfer learning won't work in our work. As, he model was previously trained on Ms COCO dataset. It is not relevant to our work.

Image data format is different and the image objects are different as well. Which cause the feature extraction sub optimal. Freezing layers won't work for this reason

Training from scratch:

We trained the model with default parameter of DETR for 200 epochs. We set the learning rate 10^{-5} . By this we got some satisfactory results. we have trained

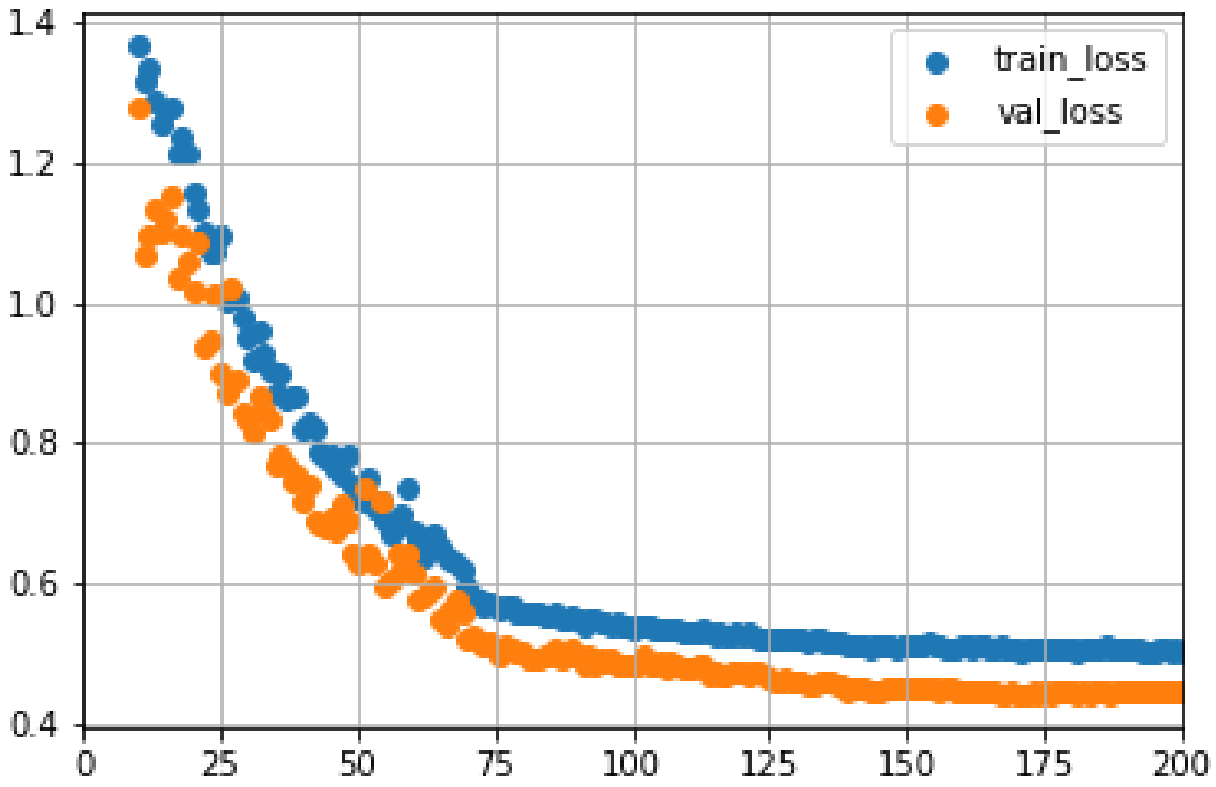


Figure 16: Training loss and validation loss is shown in the grpah. after 200 epochs results seems to be near 0.5 which is good for our model

the model 6 days to get the results. We could've done better and the graph shows there's a good change of getting better results. IN figure 16

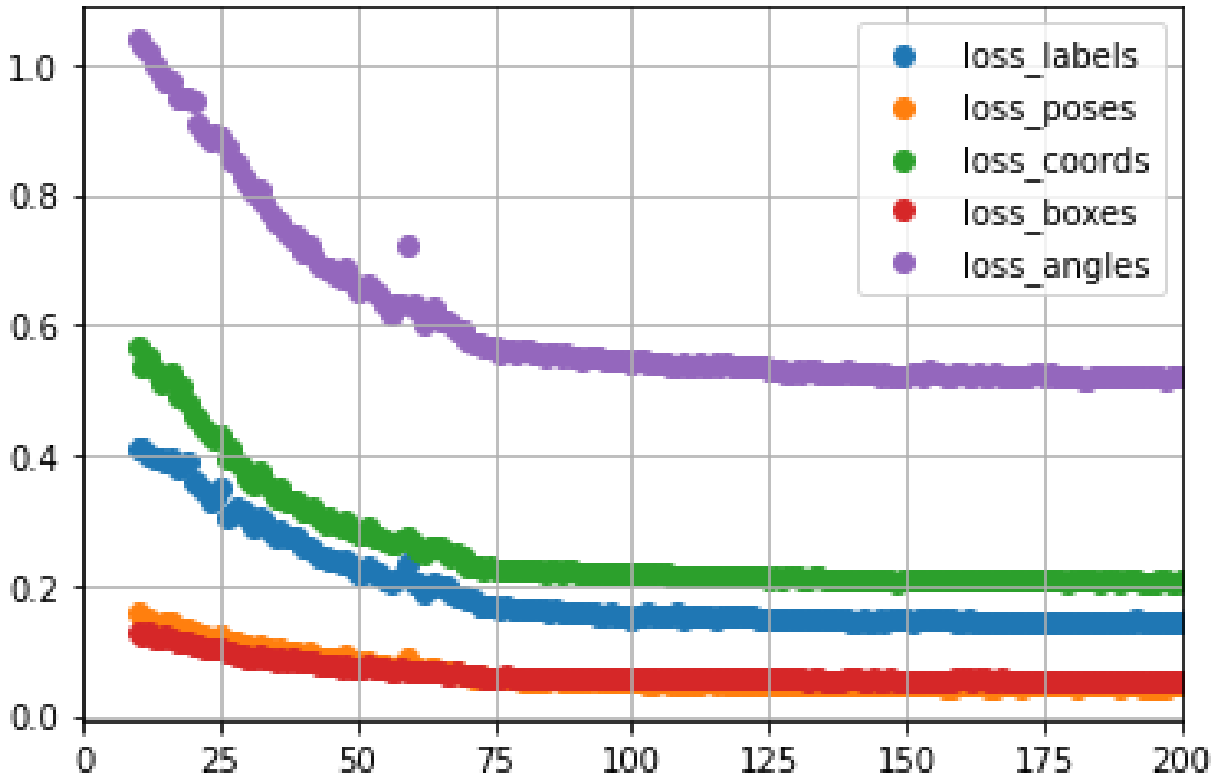


Figure 17: in the graph label,pose,coordinate,box and angle loss are shown. All of these are decreasing.

4.3 Experiment on Standard Benchmark Datasets

PSEUDO-LIDAR (PL in short) is the name given to our combined technology for 3D object identification (SDN and GDC). We assess P-RCNN alone and together in many circumstances to determine the contribution of each component.

Network with stereo depth (SDN): The backbone of our stereo depth estimation network is PSMNET (Chang Chen, 2018)[3]. (SDN). By projecting the associated LiDAR points onto photographs, we may determine the depth ground truth. We also train a PSMNET for comparison in the same way, which reduces disparity error.

3D object Detection: All of them make use of data from LiDAR and/or monocular

pictures. Using all of the foregoing, we found that P-RCNN[13] produced the best results. As a result, we're using P-RCNN[13] to evaluate all pseudo LIDAR approaches.

		IoU=0.5			IoU=0.7		
Detection algo	**input**	**Easy**	**Moderate**	**Hard**	**Easy**	**Moderate**	**Hard**
3DOP	s	46.0	43.6	30.1	6.6	5.2	4.1
OC-Stereo	s + l	87.7	80.0	70.3	64.1	48.3	40.4
S-RCNN	s + l	85.8	66.3	57.2	54.1	36.7	31.1
PL:P-RCNN	s	88.0	73.7	67.8	62.3	44.9	41.6
PL:++:P-RCNN	s	89.7	78.6	75.1	67.9	50.1	45.3
E2E-PL:P-RCNN	s	90.4	79.2	75.9	71.1	51.7	46.7
DETR-pixCoord(Frozen layers)	s	25.4	21.3	15.8	11.2	9.1	6.6
DETR-pixCoord	s	85.0	73.7	67.8	59.3	41.4	39.6

Table 1: On the KITTI validation set, 3D vehicle detection results were obtained. We give the AP3D (%) of the average precision of 3D vehicle detection. Methods are organized based on the input signals: S for stereo pictures, M for monocular images. PSEUDO-LIDAR is the acronym for PL. Our DETR-pixCoord results are shown in blue.

Now as we know car detection set is main criteria for evaluation so, we are showing the results in table 1(a). we can see performing quite similar to PL:P-RCNN. but comparing to others the result is not that much satisfactory. this might be the becuse the DETR model was not converged properly. We needed to train it more.

Model	Easy	Moderate	Hard
PL:P-RCNN	60.1	41.7	53.2
PL++:P-RCNN	60.4	44.6	53.4
E2E-PL: P-RCNN	64.8	43.9	66.5
LiDAR:P-RCNN	85.9	75.8	68.3
DETR-pixCoord	56.3	40.4	49.3

(a) Evaluation on validation only on Car class in AP%

Model	Easy	Moderate	Hard
PL:P-RCNN	54.5	34.1	28.3
PL++:P-RCNN	61.1	42.4	37.0
E2E-PL: P-RCNN	64.8	43.9	38.1
LiDAR:P-RCNN	71.2	75.8	68.3
DETR-pixCoord	45.3	37.4	25.6

(b) Results on test set

Table 2: The results of the KITTI test set are displayed. We get the same consistent performance.

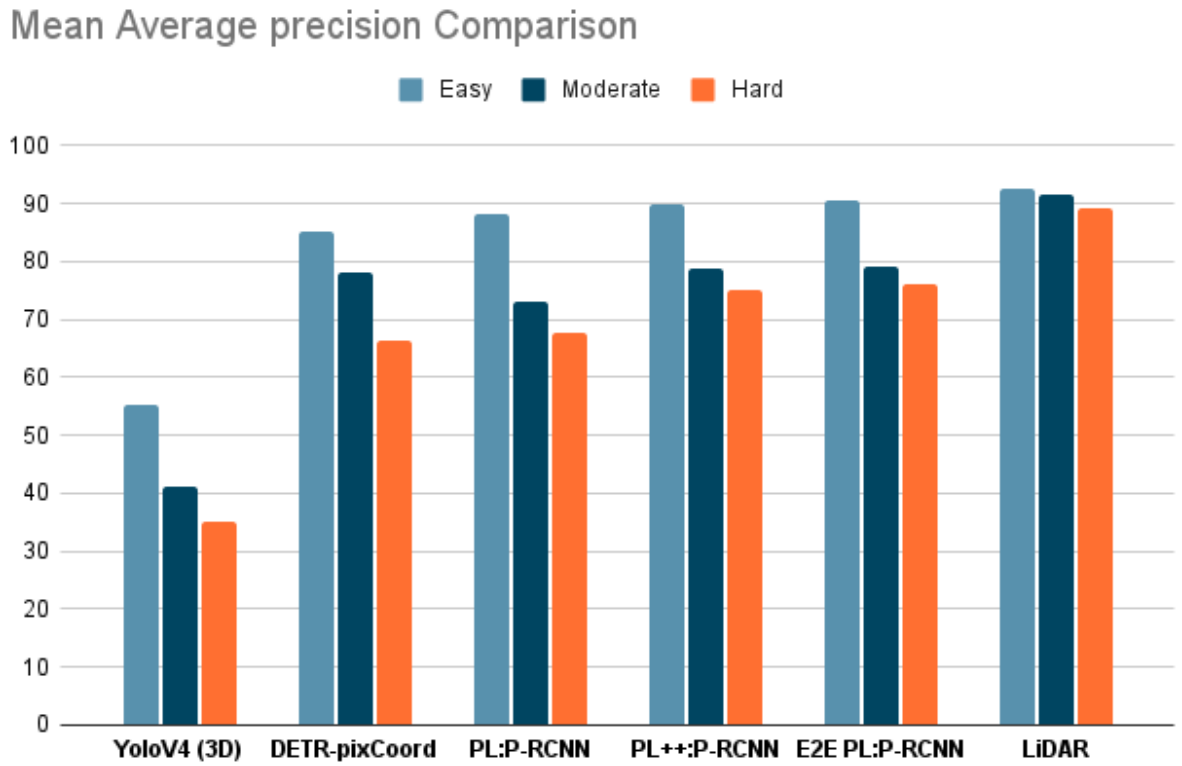


Figure 18: side by side comparison of AP% of the models YOLOv4,DETR-pixCoord(our),PL++,E2E PL, LIDAR:P-RCNN. The Results are devided into 3 categories Easy,Medium,Hard.

On the test set, we didn't see better result than the PL:R-CNN. But the results we can see we get similar results in medium cases.

4.4 Evaluation of Efficiency

Model Name	Number of parameters	Epochs	Inference time (ms)
PL:P-RCNN	61M	120	1.25
E2E PL:P-RCNN	71M	120	0.97
PL++:P-RCNN	65M	120	1.50
DETR-pixCoord	41M	200	0.81

Table 3: The table shows model's, (PL,PL++,E2E PL and DETR pixCoord(our)) number of parameter and Inference time.

4.5 Qualitative Analysis

The best example of our model success we can see is to detect an occluded car which can't be detected using PL approach. DETR with it's contextual information can detect Occluded object as original paper suggests. that's why in table 1 we can see better results for medium difficulty detect ions. with better training we can get better results than that.

(a) Inference Result of DETR pixCoord (Our) (b) Inference Result of Pseudo LiDAR

Figure 19: Inference result shown for DETR pixCoord which is detecting all the cars in image. Especially the occluded car on the right.

5 Future Works

In Our work so far, we just tried to experiment and prove the fact that we don't need point-cloud base architecture or point-cloud representation to get LiDAR near level accuracy. We now want to work on following sections:

- We will try to make the training process End to End just like the paper E2E Pseudo LiDAR[11] which will make the training process from sub optimal to optimal.
- Planning to use different fusing techniques to improve the accuracy of the mode
- Will test on various other benchmarks like Nuscences and Cityscapes. Which further solidify the validity of our approach.

6 References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. 03 2017.

[6] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *European Conference on Computer Vision*, pages 311–327. Springer, 2020.

[9] Muhammad Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Bischof. Robustness of object detectors in degrading weather conditions. 06 2021.

[10] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[11] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5880–5889, 2020.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[13] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Maxime Tremblay, Shirsendu S. Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 2020.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[16] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[17] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018.

[18] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *ICLR*, 2020.