National University of Singapore

Final Report for Team 19

LIDAR's Intelligent Crowd Counting

Team 19 Members	Student ID
LI YIAN	A0101907N
PRASHANT CHAUDHARY	A0213485E
ANKEIT TAKSH	A0213496B
ANIRBAN KAR CHAUDHRI	A0108517H

Supervised by

Dr. Tian Jing

NUS-ISS

Thursday 30th Sep, 2021

1 Project Quick Links

- **GIT Repository**
  https://github.com/LidarSpecialists/SensetimeObjectDetection

- **System Demo - (Stage 1)** - LIDAR Crowd Pedestrian Detection Demo
  https://www.youtube.com/watch?v=vpy4CWgZYRM

- **System Demo - (Stage 2)** - LIDAR Crowd Data Augmentation Demo
  https://www.youtube.com/watch?v=ucfATfkJFZk

- **Presentation Link**
  https://github.com/aktaksh/Lidar3DobjectDetection/blob/master/Presentation/Capstone%20Phase%201%20presentation_v3.pptx

2 Sponsor Company Introduction

SenseTime is a leading global company focused on developing responsible AI technologies that advance the world's economies, society and humanity for a better tomorrow. They have made a number of technological breakthroughs, one of which is the first ever computer system in the world to achieve higher detection accuracy than the human eye.

The deep learning and computer vision technologies they have developed are already empowering industries spanning across education, healthcare, smart city, automotive, finance, retail, manufacturing, communications and entertainment. Today, our technologies are trusted by over 3,500 customers and partners around the world to help address real world challenges. Going forward, we strive to empower more industries with our AI platform and build a stronger AI ecosystem together with industry and academia.

They have presences in markets including Hong Kong, Mainland China, Japan, South Korea, Saudi Arabia, the United Arab Emirates, Taiwan and Macau. In 2018, we expanded our footprint and established SenseTime International in Singapore as our regional headquarters and a springboard to the rest of Southeast Asia, such as Malaysia, Thailand, The Philippines and Indonesia.

3 Business Problem Background

SenseTime has sponsored us with this project to create a benchmark dataset for LIDAR crowd detection by introducing a novel data augmentation method for LiDAR-only learning problems that can greatly increase the convergence speed and performance. This will be done using computer vision and other deep learning tools.

3.1 Business Applications:

Smart City - predict and prevent overcrowding and trample in popular attractions (e.g. new year countdown). This can help predict and prevent overcrowding and trample in popular attractions.

Tourism Industry - Compliance with STB people density requirement (1 person/10 sqm). Knowing occupancy rate of a venue can help operations to optimize manpower and deploy staff only if there is sufficient need. Historical data can be used to do projective analysis to make operations more effective and cut costs.

Retail Industry - Identify Hot zones to optimize store layout and maximize store revenue. Compliance with STB people density requirements for the tourism industry, e.g. swimming pool in a hotel or people in a restaurant.

4 Project Scope

4.1 Objectives & Deliverables

In this project, the team will be working with SenseTime with the following two STAGES of project:

4.1.1 Stage 1: To Augment Benchmark 3D Dataset Generation with Pedestrians

4.1.2 Stage 2: To develop a Crowd Pedestrian 3D Detection Model

4.2 Augmented 3D Benchmark Dataset

Aim is to augment the dataset from KITTI[14] / Waymo Open Dataset[30] for Dense scenarios. A Pedestrian to be placed densely on a terrain or street without collision with other points in the 3D space. The optimal target will be to have 7000+ training point cloud frames per attribute value.

This augmented dataset will be used for Capstone projects and future R&D projects in SenseTime. A good benchmark dataset is a cornerstone of model training. With a good benchmark dataset, researchers can significantly reduce the time spent on data collection and labelling required before training a model. For example, MNIST is one of the most popular deep learning datasets for handwritten digits recognition and has been widely used by data scientists to train and test new architectures or frameworks.

Such datasets have many benefits: they can be used to compare existing and new models, it saves other researchers time on the laborious task of data collection. It also demonstrates the difficulties of various tasks in one research field.

To summarize our first aim is to get the dataset and then train a detection model using the same and potentially release the dataset.

4.3 LIDAR Intelligent Crowd Detection System

LIDAR Intelligent Crowd Counting system will process the input data provided by SenseTime and will be able to run on Point Cloud data independently, process Point Cloud frames and detect pedestrians under dense sce
narios.

Developed Crowd Pedestrian 3D detection model will act as Baseline model for Multi LiDAR fusion Research. Moreover, it will be better than benchmark Point RCNN [64],[65] based models under dense scenarios.

It will also support business applications such as crossline counting and regional people counting. We can possibly generate SDK for a proof-of-concept (POC) & then demo the working solution.

4.4 Success Measurements

The benchmark dataset includes 7000+ training point cloud frames with Pedestrian randomly and densely on the street frames. After augmentation the model should perform better than the benchmark point RCNN model under dense scenarios, running on the augmented dataset. The mAP or recall of the proposed detection model should be higher than point RCNN under the dense scenario.

5 Literature Review

5.1 Sensor Data

Light Detection And Ranging (LiDAR) is well-known as a remote sensing method, mainly used for surveying and mapping large areas. The technology was then adapted for terrestrial mobile systems first through 2D, planar LiDARs, largely used for indoor applications. Recently, 3D LiDARs have seen increasing use, particularly for automotive applications.

In the context of automated driving, 3D LiDARs have shown promise for various exteroceptive tasks including localization, mapping and perception. They represent a viable alternative to cameras for object detection and tracking, addressing many of the shortcomings such as limited field of view (FOV), scale ambiguity and poor performance in dimly lit environments. 3D LiDARs are also often used in tandem with cameras as they provide complementary information in the form of accurate range measurements.

5.2 Dataset

The team has reviewed two categories of datasets. One is 3D LiDAR datasets and the other category is for crowd augmentation from available Pedestrian datasets.

5.2.1 3D LiDAR Datasets

Out of 8+ datasets, the team has identified the below 3 datasets which could be used as a benchmark dataset for augmentation purpose:

- Waymo Open dataset[30] - The Waymo Open Dataset consists of high-resolution sensor data collected by autonomous vehicles operated by the Waymo Driver in a wide variety of conditions. We're releasing this dataset publicly to aid the research community in making advancements in machine perception and self-driving technology.

- KITTI 3D dataset[14] - The 3D object detection benchmark consists of 7481 training images and 7518 test images as well as the corresponding point clouds, comprising a total of 80.256 labelled objects.

- SemanticKITTI dataset[15] – SemanticKITTI is based on the KITTI Vision Benchmark and provides semantic annotation for all sequences of the Odometry Benchmark. The dataset contains 28 classes including classes distinguishing non-moving and moving objects. Overall, our classes cover traffic participants, but also functional classes for ground, like parking areas, sidewalks.

We used KITTI[14] / SemanticKITTI[15] datasets but could not leverage Waymo Open dataset[30] because it is entirely in a different format than the KITTI and hard to convert to KITTI format as well.

Datasets for 3D Object Detection and Tracking							
Name & Reference	**Year**	**#Scenes**	**#Classes**	**#Annotated Frames**	**#3D Boxes**	**Scene Type**	**Sensors**
KITTI	2012	22	8	15K	200K	Urban (Driving)	RGB & LIDAR
SUN RGB-D	2015	47	37	5K	65K	Indoor	RGB-D
ScanNetV2	2018	1.5K	18	-	-	Indoor	RGB & Mesh
H3D	2019	160	8	27K	1.1M	Urban (Driving)	RGB & LIDAR
Argoverse	2019	113	15	44K	993K	Urban (Driving)	RGB & LIDAR
Lyft L5	2019	366	9	46K	1.3M	Urban (Driving)	RGB & LIDAR
A*3D	2019	-	7	39K	230K	Urban (Driving)	RGB & LIDAR
SemanticKITTI	2017	11	28	15K	Segmentation	Urban (Driving)	RGB & LIDAR
Waymo Open	2020	1K	4	200K	12M	Urban (Driving)	RGB & LIDAR
nuScenes	2020	1K	23	40K	1.4M	Urban (Driving)	RGB & LIDAR

5.2.2 Pedestrian Datasets

Literature review has been done on the pedestrian datasets, and 6 open source datasets were identified. However, all the datasets available are image based.
As we found there is a lack of point cloud based pedestrian dataset in the field, and hence, for this project we will generate a pedestrian dataset from KITTI object data.

The details of the 6 datasets are summarized in Table:

Datasets	#Cams	Scene	Annotation unit	#Samples	#Resolution	#Binary Attributes	View Point	Occlusion	Part location
VIPeR	2	outdoor	PID	1264	48 x 128	21	yes	no	no
PRID r9J	2	outdoor	PID	400	64 x 128	21	no	no	no
GRID	8	outdoor	PID	500	from 29 x 67 to 169 x 365	x	67	no	no
APiS	-	outdoor	PI	3661	48 x 128	x	365	no	no
PETA	-	mixture	PID	19,000	117 x 39 to 169 x 365	61	no	no	no
RAP	26	indoor	PI	41585	from 36 x 92 to 344 x 554	69	yes	yes	yes

5.2.3 Evaluation Metrics

Different evaluation metrics have been proposed to test these methods for various point cloud understanding tasks. For 3D shape classification, Overall Accuracy (OA) and mean class accuracy (mAcc) are the most frequently used performance criteria. 'OA' represents the mean accuracy for all test instances and 'mAcc' represents the mean accuracy for all shape classes. For 3D object detection, Average Precision (AP) is the most frequently used criterion. It is calculated as the area under the precision-recall curve. Precision and Success are commonly used to evaluate the overall performance of a 3D single object tracker. Average Multi-Object Tracking Accuracy (AMOTA) and Average Multi-Object Tracking Precision (AMOTP) are the most frequently used criteria for the evaluation of 3D multi-object tracking. For 3D point cloud segmentation, OA, mean Intersection over Union (mIoU) and mean class Accuracy (mAcc)are the most frequently used criteria for performance evaluation[55]. Recall and Mean Average Precision (mAP) are also used in instance segmentation of 3D point clouds. Higher the recall value means better is the detection performance.

5.3 Data Augmentation

Data augmentation has been an essential technique for boosting the performance of 3D image classification and object detection models. Augmentation methods typically include manually designed image transformations, to which the labels remain invariant, or distortions of the information present in the images. For Example, elastic distortions, scale transformations, translations, and rotations are beneficial on models.

5.3.1 Implementation Method

We formulate the problem of finding the right augmentation strategy as a special case of hyperparameter schedule learning. From the literature review, the typical method consists of two components: a specialized data augmentation search space for point cloud inputs and, second, a search algorithm for the optimization of data augmentation parameters.
Visualization of the augmentation operations in the proposed search space as shown below -

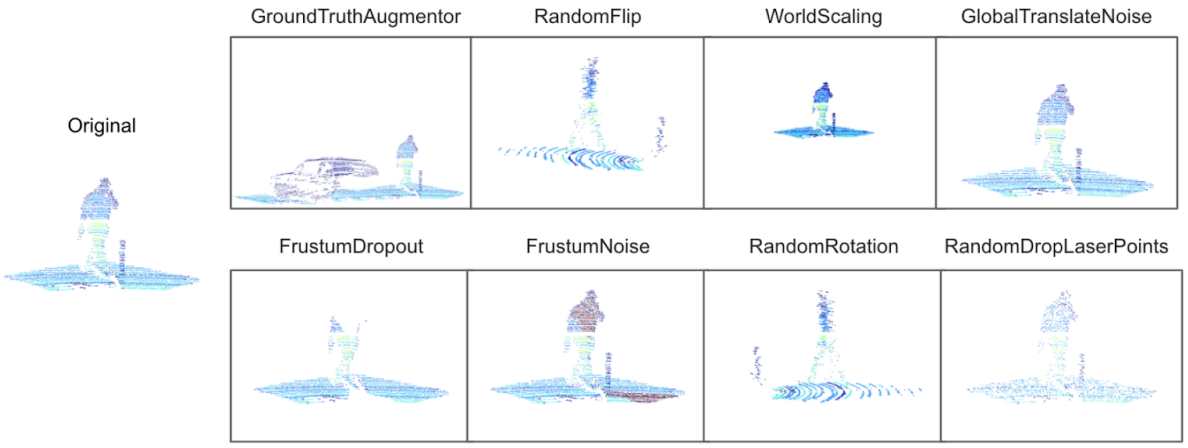


Image Courtesy: Improving 3D Object Detection through Progressive Population Based Augmentation (Shuyang Cheng1, Zhaoqi Leng?1, Ekin Dogus Cubuk2, Barret Zoph2, ChunyanBai1, Jiquan Ngiam2, Yang Song1, Benjamin Caine2, Vijay Vasudevan2, Congcong Li1, Quoc V. Le2, Jonathon Shlens2, and Dragomir Anguelov1)

In the proposed search space, an augmentation policy consists of N augmentation operations. Additionally, each operation is associated with a probability. An augmentation policy is defined by a list of distinct augmentation operations and the corresponding augmentation parameters. For example, the ground-truth augmentation operation has parameters denoting the probability for sampling vehicles, pedestrians, cyclists, etc.; the global translation noise operation has parameters for the distortion magnitude of the translation operation on x, y and z coordinates.

To reduce the size of the search space and increase the diversity of the training data, these different operations are always applied according to some learned probabilities in the same, predetermined order to point clouds during training. The basic augmentation operations in the proposed search space fall into two main categories: global operations, which are applied to all points in a frame (such as rotation along Z-axis, coordinate scaling, etc.), and local operations, which are applied to points locally (such as dropping out points within a frustum, pasting points within bounding boxes from other frames, etc.).

5.4 3D Object Detection - Model Pipelines & Approaches

The following approaches are reviewed in this section:

5.4.1 Projection / Pixel-based Approach

Projects the 3D point cloud into 2D, either Bird-Eye-View (BEV) or front view. Associated LiDAR points with image pixels by projecting 3D point clouds onto 2D images and exploited this association to fuse RGB information into 3D points. They also considered 3D semantic segmentations an auxiliary task to learn better representations. MV3D hierarchically fused the CNN features extracted from the front view, bird's eye view and camera view to jointly predict object class and regress the oriented 3D bounding boxes [60]. PIXOR[71] took bird's-eye-view representation as input alone and designed a proposal-free, single-stage detector to output pixel-wise predictions. It, however, assumes that all objects lie on the same ground and cannot handle indoor scenes where multiple objects often stack together in vertical space. BirdNet and RT3D generated region proposals from a bird's-eye-view but achieved weak results. VeloFCN is the first work to project point clouds onto a cylindrical surface. LMNet[72] took the front-view representation as input alone but got unsatisfactory results even on car detection, because of the loss of details.

5.4.2 Voxel-based Approach

Partition the 3D space into voxel grid and utilize neural networks to extract features. Voxel-based methods utilize a voxel grid representation for point clouds and involve different ways to extract features. Vote3D adopted sliding windows on sparse volumes to extract hand-crafted geometric

features for each volume. Vote3Deep introduced 3D convolutional neural networks to extract features for each volume. VoxelNet built three voxel feature encoding (VFE) layers to extract 3D features for the region proposal network. However, the main issues of voxel representation are computational efficiency and memory consumption[60].

5.4.3 3D Point-based Approach

Directly takes the raw point cloud as input in the form, complicated statistics operations not needed. F-Pointnet and F-ConvNet[64] directly operated on original point clouds after popping up RGB-D scans, without any loss of data, leading to precise detection. Their generation of object proposals depended largely on camera images. To get more accurate locations of objects, F-Pointnet adopted instance segmentation to classify the points in the view frustum while F-ConvNet[64] divided the view frustum into a sequence of frustums to extract frustum-level features. This approach also directly operates on raw point clouds, and our proposals are generated from point clouds only, without the need of camera images. It adopts the truncated distances to cut the view frustum for more accurate locations, without the need of point cloud segmentation.

5.4.4 Finalized Approach – [3D Point-based Approach]

We finalized the 3D Point-based approach to directly process LiDAR 3D point-based data using point net, it directly operates on 3D point clouds and achieves robust and accurate 3D detection performance.
It will be beneficial because there will be no information loss, unlike other two approaches namely Projection / Pixel based, and Voxel based methods. Also, this will be less sparse data and cause less wastage of computing resources while processing.

5.5 Some Existing Models:

5.5.1 PointNet++ (2017):

Proposes a multi-scale grouping (MSG) strategy to construct density adaptive PointNet[5] layers. This strategy extracts multiple scales of local patterns at each abstraction level and combines them to enhance the robustness of feature learning under non-uniform sampling density. However, it does not explicitly consider the influence of distance on the density (the same parameters are used in different regions), and the multi-scale grouping increases the computational complexity.

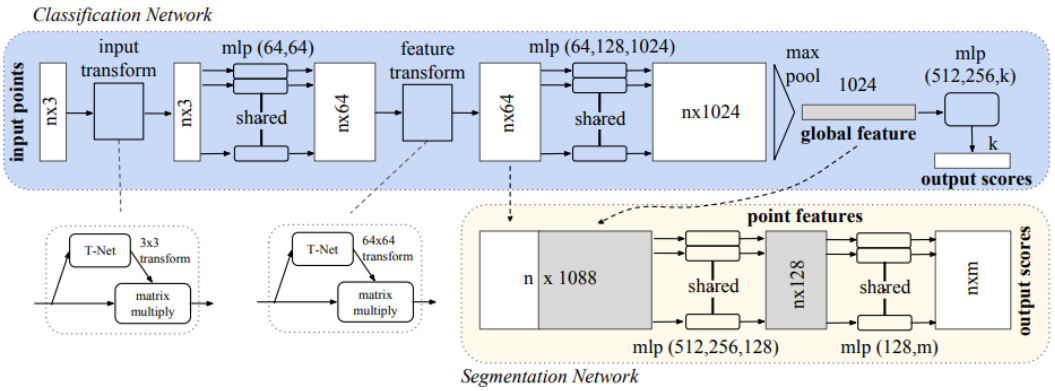


Image Courtesy: PointNet++: Deep hierarchical feature learning on point sets in a metric space, C. R. Qi, L. Yi, H. Su, and L. J. Guibas.

The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. “mlp” stands for multi-layer perceptron, numbers in brackets are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net

5.5.2 Frustum ConvNet (2019)

Extracts features at different distances by a sequence of frustums for each region proposal, and the frustums are used to group local points. This method has considered the distance, but it relies on 2D proposals in RGB images, and the large grouping granularity is not conducive to local feature extraction.

The whole framework of our F-ConvNet[64] is illustrated below -

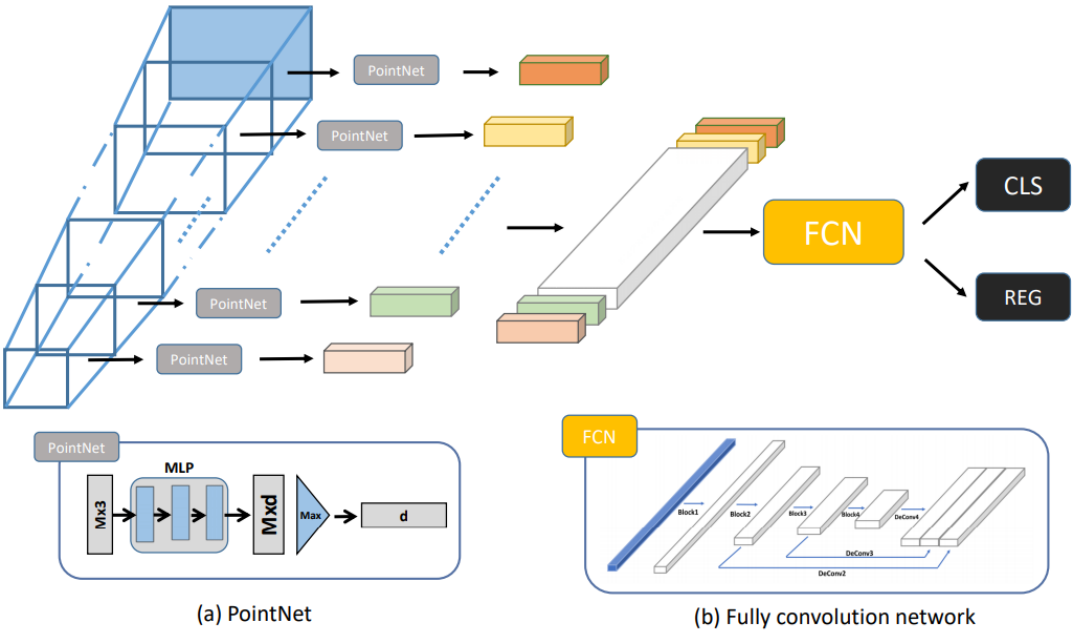


Image Courtesy: Frustum convnet [64]: Frustum convnet: Sliding frustums to aggregate local pointwise features, Z. Wang and K. Jia

5.5.3 Voxel-FPN [68] (2020):

Performs multi-resolution voxelization on the original point cloud, and then a FPN (Lin et al. 2017a) structure is adopted to fuse multi-resolution features, which is similar to the multi-scale grouping in PointNet++[54] and the factor of distance is not considered.

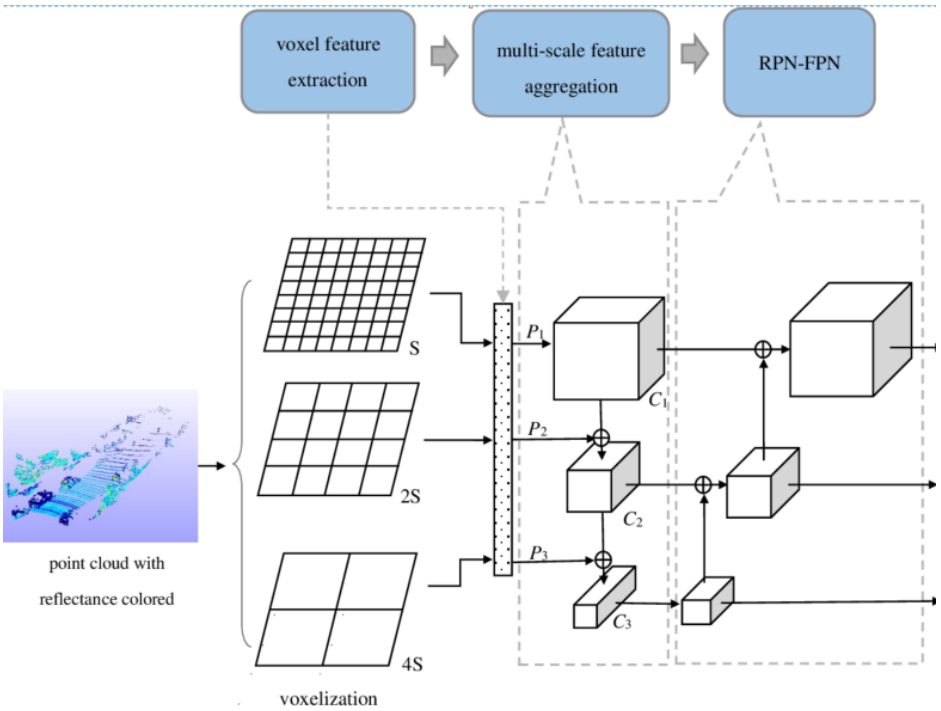


Image Courtesy: Voxel-FPN[68]: Voxel-FPN: multi-scale voxel feature aggregation, Bei Wang, Jianping An, Jiayan Cao

5.5.4 Point RCNN (2019):

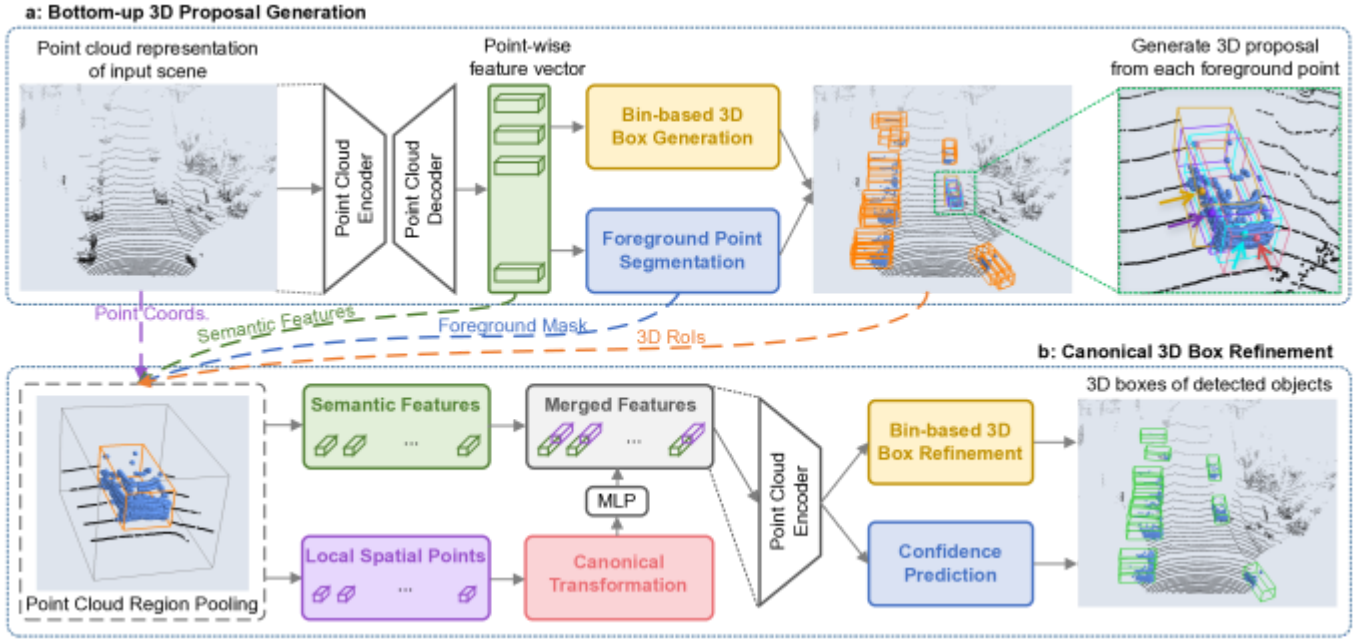


Image Courtesy: PointRCNN[64]: 3D Object Proposal Generation and Detection from Point Cloud (sshi, xg wang)

PointRCNN architecture for 3D object detection from point cloud is shown above. The whole network consists of two parts: (a) for generating 3D proposals from raw point clouds in a bottom-up manner. (b) for refining the 3D proposals in canonical coordinate.

Instead of generating proposals from RGB image or projecting point cloud to bird's view or voxels as Projection based methods do, Point RCNN stage-1 sub-network directly generates a small number of high-quality 3D proposals from point cloud in a bottom-up manner via segmenting the point cloud of whole scene into foreground points and background. The stage-2 sub-network transforms the pooled points of each proposal to canonical coordinates to learn better local spatial features, which is combined with global semantic features of each point learned in stage-1 for accurate box refinement and confidence prediction.

5.5.5 DA-Point RCNN (2020):

DA-Point RCNN[67] has the same basic framework as that of Point RCNN[63], which consists of two stages: stage 1 sub-network for generating 3D proposals from raw point cloud, and stage 2 sub-network for refining the proposals. We can extract separate features for objects with different point densities by a multi-branch backbone network.

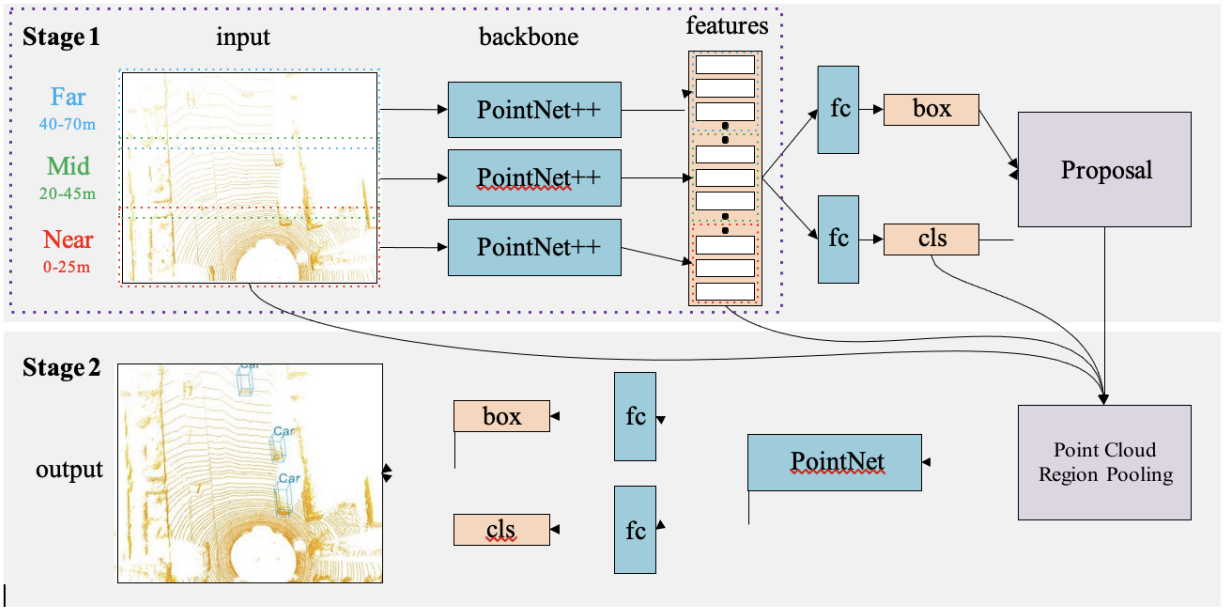


Image Courtesy: A Density-Aware PointRCNN for 3D Objection Detection in Point Clouds: Jie Li & Yu Hu [67]

5.5.6 Varifocal Loss – [VFNet, 2021]

Accurately ranking the vast number of candidate detections is crucial for dense object detectors to achieve high performance. In this work, we propose to learn IoU-aware classification scores (IACS) that simultaneously represent the object presence confidence and localization accuracy, to produce a more accurate ranking of detections in dense object detectors. In particular, we design a new loss function, named Varifocal Loss (VFL), for training a dense object detector to predict the IACS, and a new efficient star-shaped bounding box feature representation (the features at nine yellow sampling points) for estimating the IACS and refining coarse bounding boxes. Combining these two new components and a bounding box refinement branch, we build a new IoU-aware dense object detector based on the FCOS+ATSS architecture, what we call VarifocalNet or VFNet for short.

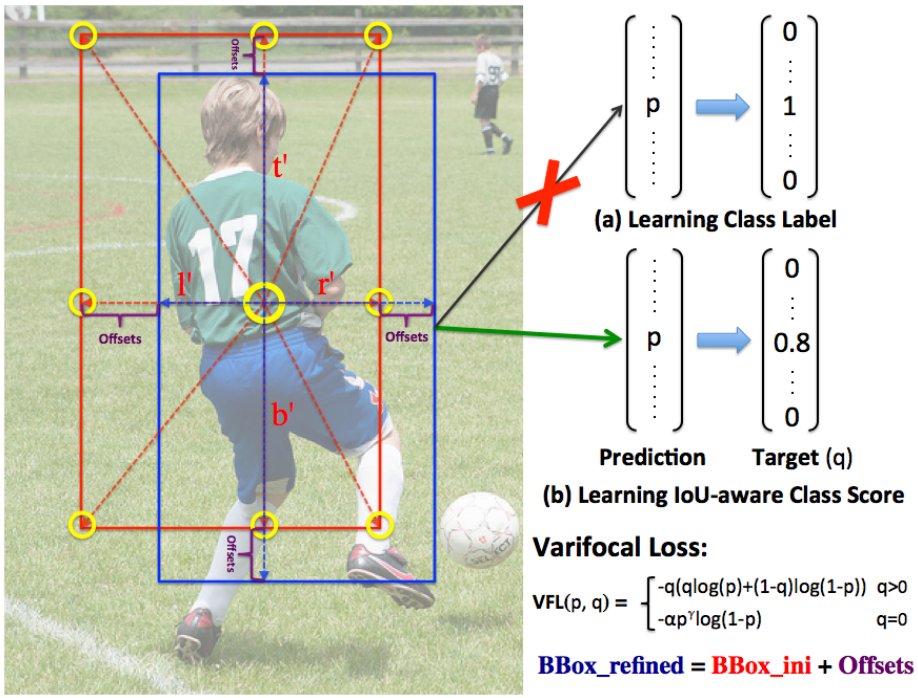


Image Courtesy: VFNet[73]: Learning to Predict the IoU-aware Classification Score.

5.5.7 Finalized Base Model – [Point RCNN]

We finalized Point RCNN[63] as the baseline and then we further trained and developed it at the stage 2 of this project to improve dense detection on augmented dataset and updating classification and/or regression loss functions at foreground point segmentation stage for better prediction accuracy and recall.

5.6 Task and Challenges

5.6.1 Tasks

In the perception module of autonomous vehicles, semantic segmentation, object detection, object localization, and object classification/recognition constitute the foundation for reliable navigation and accurate decision. These tasks are described as follows, respectively.

1) **3-D point cloud semantic segmentation:** Point cloud semantic segmentation is the process to cluster the input data into several homogeneous regions, where points in the same region have the identical attributes. Each input point is predicted with a semantic label, such as ground, tree, and building. This task can be concluded as given a set of ordered 3-D points X = {x1, x2, xi, . . . , xn} with xi $\in$ R3 and a candidate label set Y = {y1, y2, . . . , yk}, assign each input point xi with one of the k semantic labels. Segmentation results can further support object detection and classification.

2) **3-D object detection/localization:** Given an arbitrary point cloud data, the goal of 3-D object detection is to detect and locate the instances of predefined categories [e.g., cars, pedestrians, and cyclists, and return their geometric 3-D location, orientation, and semantic instance label. Such information can be represented coarsely using a 3-D bounding box which is tightly bounding the detected object. This box is commonly represented as (x, y, z, h,w, l, θ, c), where (x, y, z) denotes the object (bounding box) center position, (h,w, l) represents the bounding box size with width, length, and height, and θ is the object orientation. The orientation refers to the rigid transformation that aligns the detected object to its instance in the scene, which are the translations in each of the x-, y-, and z-directions and a rotation about each of these three axes. c represents the semantic label of this bounding box (object).

3) **3-D object classification/recognition:** Given several groups of point clouds, the objectiveness of classification/recognition is to determine the category [e.g., mug, table, or car] the group points belong to. The problem of 3-D object classification can be defined as: given a set of 3-D ordered points X = {x1, x2, xi, . . . , xn} with xi $\in$ R3 and a candidate label set Y = {y1, y2, . . . , yk }, assign the whole point set X with one of the k labels.

5.6.2 Challenges and Problems

In order to segment, detect, and classify the general objects using DL for AVs with robust and discriminative performance, several challenges and problems must be addressed. The variation of sensing conditions and unconstrained environments results in challenges on data.

The irregular data format and requirements for both accuracy and efficiency pose the problems that DL models need to solve.

Challenges on LiDAR Point Clouds: Changes in sensing conditions and unconstrained environments have dramatic impacts on the appearances of objects. In particular, the objects captured at different scenes or instances and even for the same scene, the scanning times, locations, weather conditions, sensor types, sensing distances, and backgrounds are all brought about by differences. All these conditions produce significant variations for both intraclass and extra-class objects in LiDAR point clouds [55].

1. **Diversified point density and reflective intensity:** Due to the scanning mode of LiDAR, the density and the intensity for objects vary a lot. The distribution of these two characteristics highly depends on the distances between objects and LiDAR sensors. Besides, the ability of the LiDAR sensors, the time constraints of scanning and needed resolution also affect their distribution and intensity.

2. **Noisy:** All sensors are noisy. There are a few types of noise that include point perturbations and outliers. It means that a point has some probability of being within a sphere of a certain radius around the place it was sampled (perturbations), or it may appear in a random position in space.

3. **Incompleteness:** Point clouds obtained by LiDAR are commonly incomplete. This mainly results from the occlusion between objects, cluttered background in urban scenes, and unsatisfactory material surface reflectivity. Such problems are severe in real time capturing of moving objects, which exist large gaping holes and severe undersampling.

4. **Confusion categories:** In a natural environment, shape similar or reflectance similar objects have interference in object detection and classification. For example, some man-made objects, such as commercial billboards, have similar shapes and reflectance with traffic signs.

Problems for 3-D DL Models: The irregular data format and the requirements for accuracy and efficiency from tasks bring some new challenges for DL models. A discriminate and general-purpose 3-D DL model should solve the following problems when designing and constructing its framework.

1. **Permutation and orientation invariance:** Compared with 2-D grid pixels, the LiDAR point clouds are a set of points with irregular order and no specific orientation. Within the same group of N points, the network should feed N! permutations in order to be invariant. Besides, the orientation of point sets is missing, which poses a great challenge for object pattern recognition.

2. **Rigid transformation challenge:** There exist various rigid transformations among point sets, such as 3-D rotations and 3-D translations. These transformations should not affect the performance of networks.

3. **Big data challenge:** LiDAR collects millions to billions of points in different urban or rural environments with natural scenes. For example, in the Kitti dataset[14], each frame captured by 3-D Velodyne laser scanners[61] contains 100k points. The smallest collected scene has 114 frames, which has more than 10 million points. Such amounts of data bring difficulties in data storage and processing.

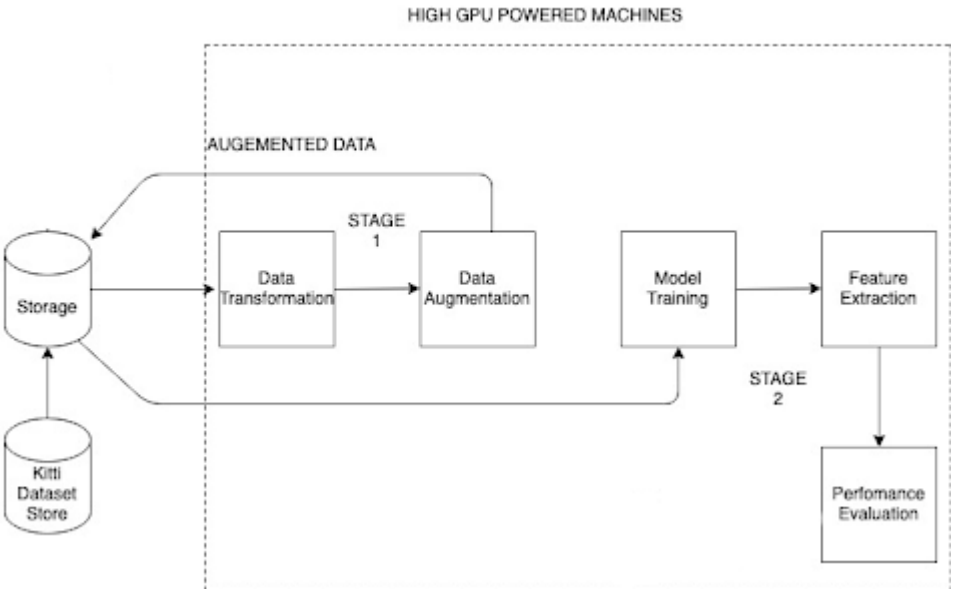
4. **Accuracy challenge:** Accurate perception of road objects is crucial for AVs. However, the variation for both intraclass and extra-class objects and the quality of data poses challenges for accuracy. For example, objects in the same category have a set of different instances, in terms of various material, shape, and size. Besides, the model should be robust to the unevenly distributed, sparse, and missing data.

5. **Efficiency challenge:** Compared with 2-D images, processing a large quantity of point clouds produces high computation complexity and time costs. Besides, the computation devices on AVs have limited computational capabilities and storage space. Thus, an efficient and scalable deep network model is critical.

6 System Design & Implementation

Stage 1 & Stage 2 Pipeline Architecture

The system architecture for Data Augmentation and Crowd Detection is given below as a complete pipeline. Effort wise Stage 1 and Stage 2 focus areas have been mentioned in the diagram as well.



LIDAR Input: LIDAR sensors from the AV will serve as input into the system

Distributed Storage: Various distributed locations will allow for more efficient collection and storage of video inputs. There will be a batch and real-time queue (as required).

Processing, Inference & Storage: The 3D point cloud Processing will be completed as per the optimized output from the research. The ROIs will be classified using the saved weights from the trained Neural Network Model

Final Storage: The output of the model together with relevant details will be stored in a centralized location for retrieval.

Surveillance Monitoring: There will be monitoring monitoring with visualization to calculate statistics, create reports and allow for searching of the data. Also, real-time alerts will be pumped into the monitoring system to flag suspicious cases.

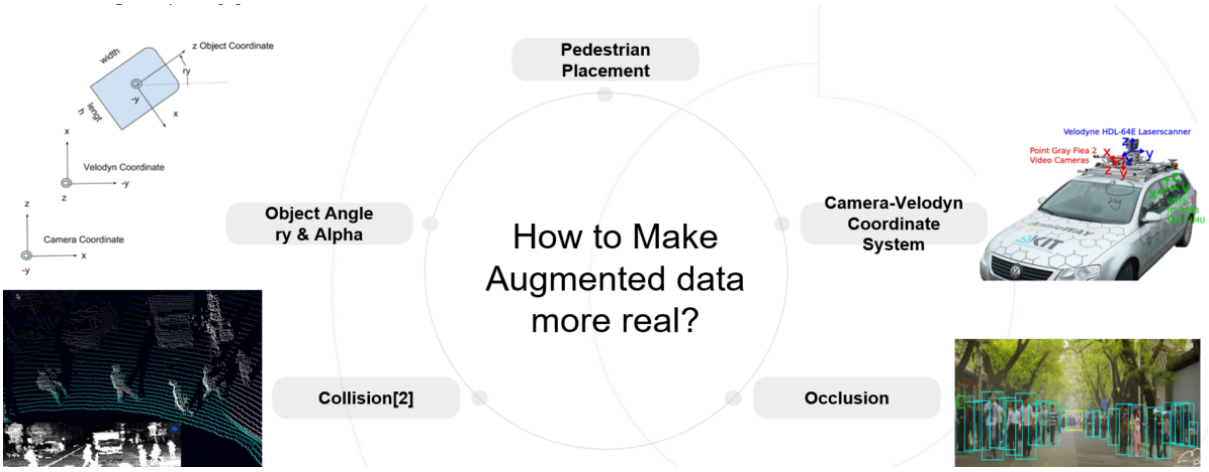
Data Augmentation

This Section is focussed on the current state of the project where we are building a data augmentation pipeline.

Key Considerations

Unlike abundant 2D image dataset, 3D datasets are very limited, and labelling of 3D data is much more time-consuming compared to 2D labelling. Limited training samples are always a challenge for research on 3D datasets. Therefore, data augmentation became very important in terms of boosting the performance of 3D models [69].

From literature review and discussion with the sponsor team, it is found out that there is a lack of a comprehensive methodology in the field. In Yan Yan's work, the collision is being considered but occlusion and object angle are not considered for the data augmentation [70]. Angle and Occlusion is very important in dense scenarios. Hence in this project, we propose an augmentation system considering 5 perspectives.



1. **Pedestrian Placement:** The Pedestrian should be placed on the road, hence, it is required to know where the road is in 3D environment.
2. **Camera-Velodyne Coordinate system**: In Kitti and SementicKitti dataset point cloud training data are all stored in 2 coordinate systems. Point cloud raw data are stored in Velodyne coordinate, whereas the labelling data are stored in camera coordinate[61]. There is a calibration file to transfer between these 2 coordinates. In this project, we transfer points and save point cloud and labelling file following this convention
3. **Object Angle**: In point cloud object data, there are 2 angles, ry and alpha. Ry is the angle of the object from the horizontal coordinate, whereas alpha is the relative location of the object from the origin of lidar. It is important to consider the angle during data augmentation, because the point from the front face of a pedestrian will be his face and chest, whereas the point from the back of a pedestrian will be his blackhead and shoulder.
4. **Collision**: object should not be colliding and exist in the same location. This is also applicable in 3D data augmentation
5. **Occlusion**: under a dense scenario, the object at back is badly occluded by the object at front. Data augmentation should take this into consideration as well, as point clouds won't be able to penetrate through objects. However, LiDAR data is easier to achieve multiple sensor fusion as the absolute distance could be computed and merged together. Hence the occlusion is of relatively less importance to simulate multiple sensor fused data.

In this project, we have taken consideration of the above 5 perspectives and designed a pipeline to tackle it stage by stage.

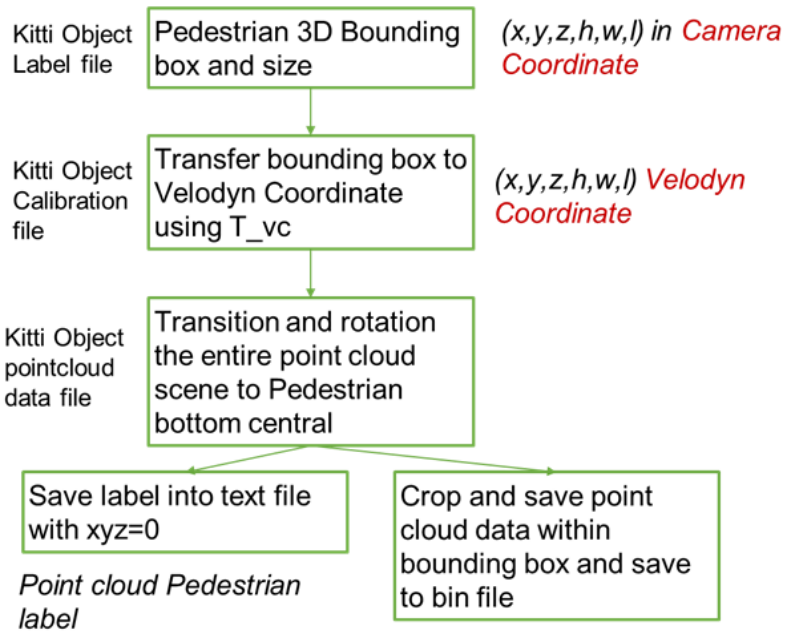
Dataset Augmentation Methodology and Pipeline design

The entire dataset augmentation has 2 big steps. We have split it into 4 iterations to achieve the end task.

1. Generate a dataset for 3D pedestrian sample
2. Densely dispense pedestrian to road to generate dense scenario frames

a) Object Sample Database Generation pipeline

For the sake of the lack of 3D pedestrian dataset in the industry, we created one from the KITTI dataset. Below is the pipeline we designed.



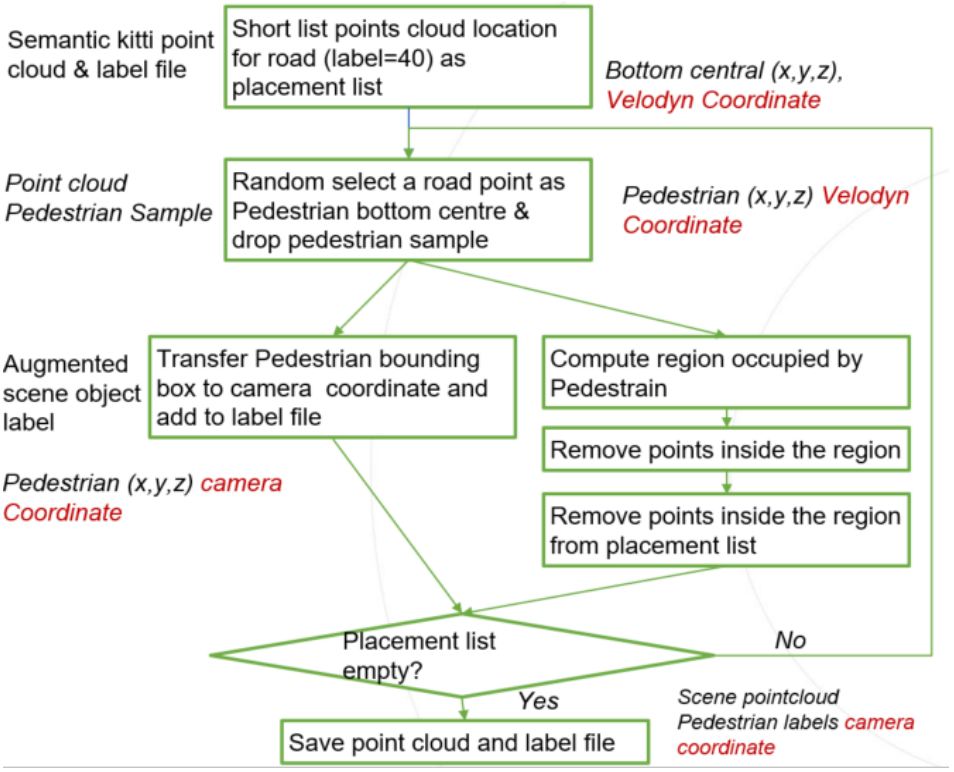
1. Kitti object label file is used to extract a pedestrian 3D bounding box in camera coordinate. Here we get its bottom centre coordinate (x, y,z) and height of the bounding box(h), width(w), and hight(h) all in camera coordinate and unit of meter
2. Use the calibration matrix and convert the above mention bottom centre coordinate to velodyne coordinate
3. Transit and rotate the entire frame to the object's bottom and centre point. Crop all the points inside the bounding box and save the point cloud data into a bin file.
4. Save the corresponding label file into text, with xyz=0

With this work stage, we have taken consideration of the camera-velodyne coordinate system and saving the data following industry convention. This makes it easier to be processed by other researchers and it can also be visualized using existing tools.

b) Densely dispense pedestrian to road to generate dense scenario frames

This task has been splitted into 3 iterations to tackle the above mentioned 5 perspectives one by one. In the first iteration we solved pedestrian placement, camera-velodyne coordinate, object collision. In the second iteration we are going to solve the Object Angle problem (ry & Alpha). In the last iteration, we are going to solve Occlusion.

c) Densely dispense Pedestrian to road including pedestrian placement, camera-velodyne coordinate, object collision

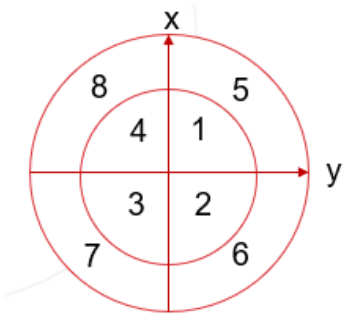


As shown in the flow chart above, both SemanticKITTI and KITTI dataset are used. SemanticKITTI are used as the frame background, kitti are used to get pedestrian dataset.

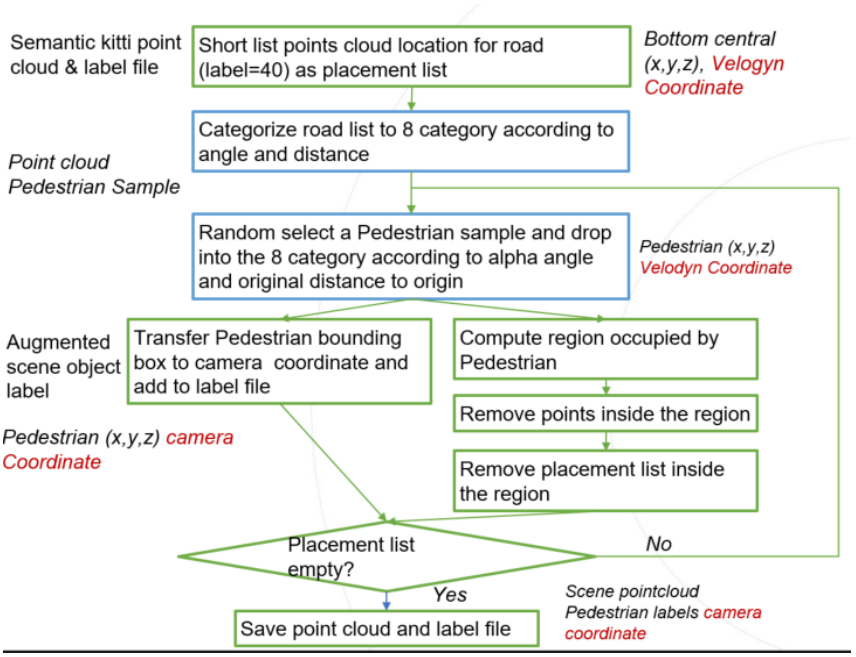
1. In SemanticKITTI point, shortlist all the locations for road, where SemanticKITTI label=40. All these location function as potential location for pedestrian placement, save it to a 'placement list'
2. Randomly select a pedestrian from the pedestrian dataset and randomly select a dropout location A from 'the placement list'. Add the coordinate of the A to the pedestrian's point cloud to transit the pedestrian to the dropout location.
3. Transfer the new bottom centre coordinate into camera coordinate and save it to the label file.
4. To avoid collision, compute the region occupied by the pedestrian and remove other points inside the region from the point cloud frame data.
5. Also remove the points from the 'placement list' which fall inside the occupied region.
6. Repeat above 2-5 steps until the placement list is empty.
7. Save the complete point cloud data and label file. Now we have 1 frame of dense scenario point cloud data.

d) Densely dispense Pedestrian to road including pedestrian placement, camera-velodyne coordinate, object collision, object angle (ry & alpha) & distance to origin

When LIDAR detects an object from a different angle, the feedback point would be different. When the object is far from the origin, its point cloud data will also be sparser as compared to it is near to origin. To tackle object angle variation, we decide to use the original info of the pedestrian angle and its original distance from the origin. We divide the SemanticKITTI background frame to 8 zones, according to angle and distance to origin. As shown in the picture below.



With this consideration, the pipeline updated to the following iteration 2 version. The extra parameter user can use to adjust the dataset generation are pedestrian occlusion ration, pedestrian transaction ratio –



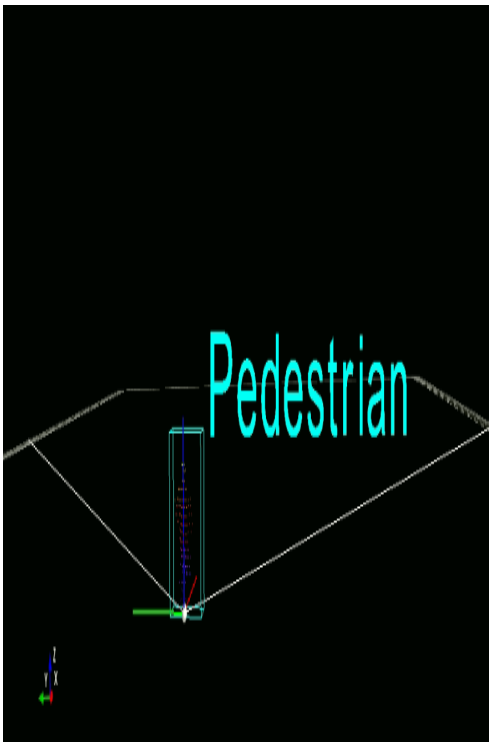
1. In SemanticKitti point, shortlist all the locations for road, where Semantic Kitty label=40. All these location function as potential location for pedestrian placement, save it to a ‘placement list’ and divide the placement list to 8 ‘placement zone’
2. Randomly select a pedestrian from the pedestrian dataset and compute its distance to the origin and alpha to the origin. Assign this pedestrian to 1 of the 8 zone correspondingly
3. and randomly select a dropout location A from the ‘placement zone’ . Add the coordinate of the A to the pedestrian’s point cloud to transit the pedestrian to the dropout location.
4. Transfer the new bottom centre coordinate into camera coordinate and save it to the label file.
5. To avoid collision, compute the region occupied by the pedestrian and remove other points inside the region from the point cloud frame data.
6. Also remove the points from 8 ‘placement zones’, which fall inside the occupied region.
7. Repeat above 2-6 steps until the placement list is empty.

Save the complete point cloud data and label file. Now we have 1 frame of dense scenario point cloud data.

The extra parameter users can use to adjust the dataset generation are pedestrian occlusion ration, pedestrian transaction ratio. Users can also select to dispense pedestrians in different densities at the 8 zones.

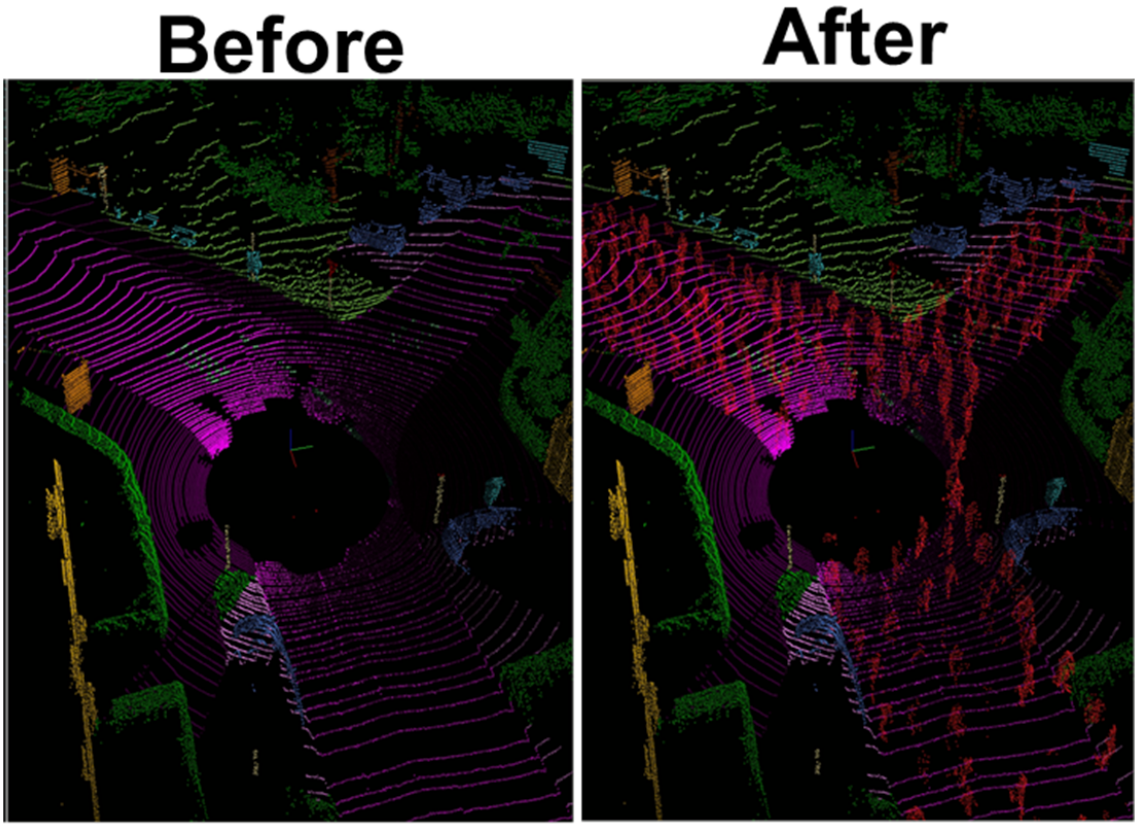
Data Augmentation Results

For task 1, we have completed and successfully generated a pedestrian sample dataset with 4487 pedestrian samples with complete labelling. The data could be processed and visualized using the mainstream tools.

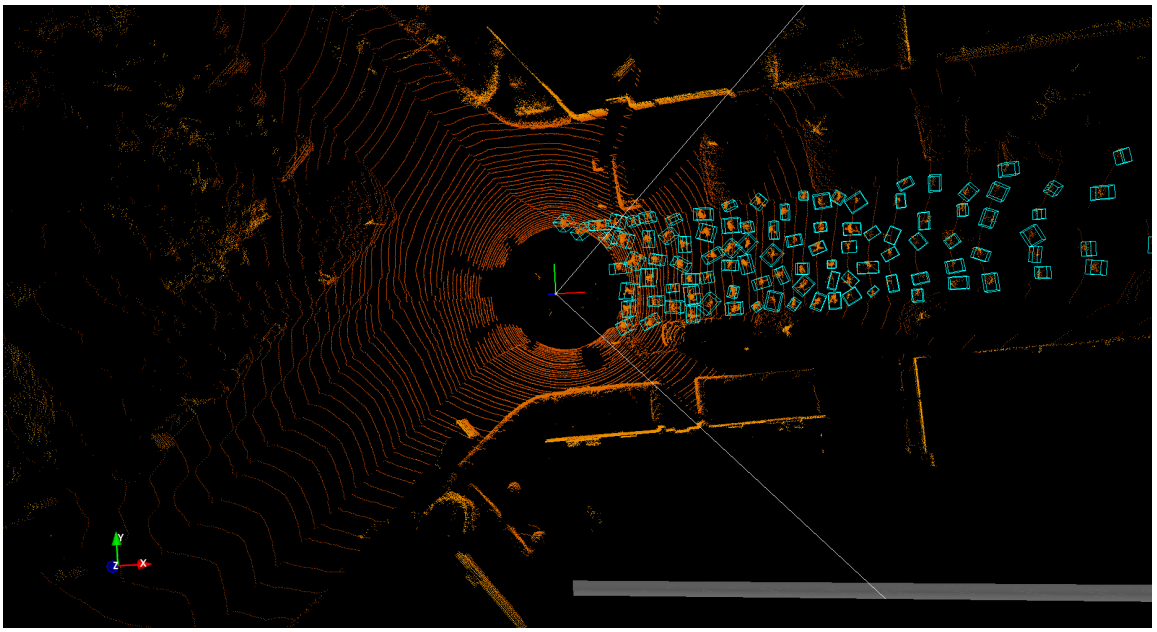


For task 2, we have completed iteration a, and successfully generated 7000+ frames with pedestrians densely dispensed on the road with both detection box labelling and semantic segmentation labelling.

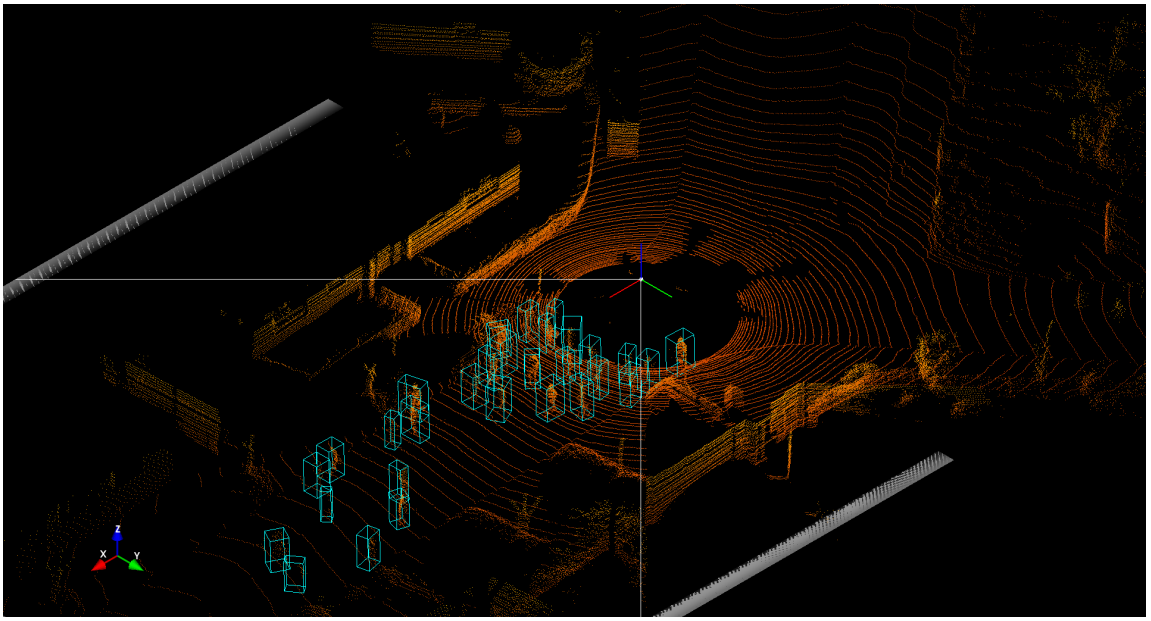
In the iteration a), Pedestrians were randomly dropped onto the road as densely as possible. As shown in the picture below, the red points are pedestrians on the road. Both bounding box and semantic segmentation labels are being generated to support future R&D in the field.



In the iteration b, angle and pedestrian facing direction are being considered and pedestrians were dropped into 8 different zones. Since the KITTI dataset mainly labels objects in the front direction, it was observed that the pedestrians in the augmented dataset are mainly dropped into zone 1, zone 2, zone 5 and zone6 as shown in the picture below.

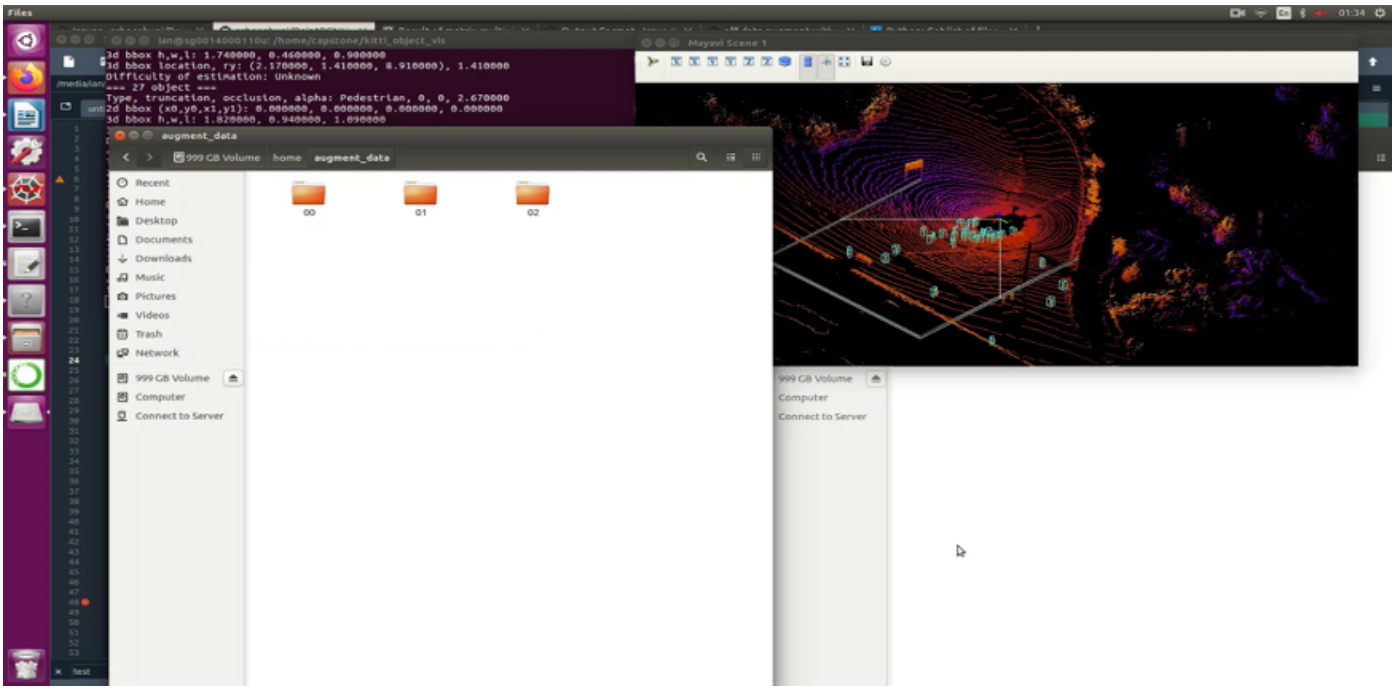


To future improve the realness of the augmented dataset, more random parameters were added. Adding randomness of pedestrian distance and number of pedestrians. more authenticate augment data scenes were generated as shown below. It could be seen that pedestrians were randomly grouped and placed, which is more ideal.



Data augmentation System Demo Screenshots

Here we have shown the system demo screenshot and various performance metrics obtained as a result of object detection:



Link - https://www.youtube.com/watch?v=ucfATfkJFZk (LIDAR Crowd Data Augmentation Demo video)

Baseline Model on Standard vs Crowd Augmented Datasets:

S. No.	Model	Training Dataset	Evaluation Dataset	Complexity	Result	Evaluation
1	Point RCNN Baseline	Standard KITTI	Standard KITTI	Easy	Best	Total bbox recall (thresh=0.500): 888 / 2280 = 0.389
2	Point RCNN Baseline	Standard KITTI	Crowd Augmented Dataset	Hard	Worse	Total bbox recall (thresh=0.500): 20425 / 83217 = 0.245
3	Dense Point RCNN	Crowd Augmented Dataset	Crowd Augmented Dataset	Hard	Avg	Total bbox recall (thresh=0.500): 26443 / 83217 = 0.318

Stage-2 : 3D Detection Model

6.1.1 Key Considerations

As mentioned in the literature review section, we finalized the Point RCNN model as the baseline model. Its main benefits include directly consuming only point-cloud as input and its 3D detection benchmark using KITTI dataset outperforms many existing state-of-the-art methods with remarkable margins.

To further get into details, instead of generating proposals from RGB image or projecting point cloud to bird's view or voxels as previous methods do, RCNN stage-1 sub-network can directly generate a

small number of high-quality 3D proposals directly from the point cloud in a bottom-up manner via segmenting the point cloud of the whole scene into foreground points and background.
The stage-2 sub-network transforms the pooled points of each proposal to canonical coordinates to learn better local spatial features, which is combined with global semantic features of each point learned in stage-1 for accurate box refinement and confidence prediction.

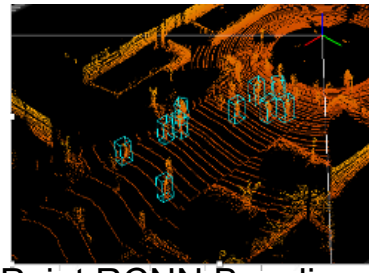
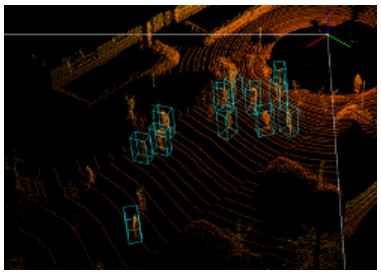
3D Detection Model - Our Methodology and Pipeline design

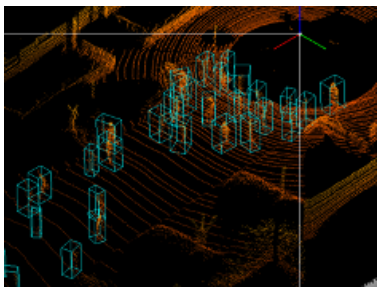
1) First we trained our base model using the standard KITTI dataset and also evaluated using the same. As the objects in the standard dataset are sparse hence it is a simpler task to detect objects in that. We noted the standard result as a control for the rest of the evaluation and experimentation.
2) In the second step, we used the baseline model again that was trained on standard Kitty dataset. But for the evaluation we used our own crowd augmented dataset. Since it was harder for the baseline model to do the dense detection it failed to perform well with a much lesser bounding box recall.
3) Next, in order to improve the results, we used the same baseline model but trained it with our own crowd augmented dataset and evaluation was also done using the dense crowd dataset. This time however, the results were superior to the scenario when the model was trained using standard Kitti Dataset (#2) but it was still lagging behind the scenario where the evaluation was done on the standard Kitti Dataset (#1).
4) Lastly, we studied the behaviour of Foreground point segmentation in the Point RCNN and experimented with the classification loss function (namely, the focal loss) and replaced it with the Varifocal loss function. After a few epoch of training, we got slightly better results than the baseline model trained with crowd augmented dataset (#3).

In summary, dense/crowd detection is a hard problem for a model especially that is trained on a sparse pedestrian dataset. However, we can improve the performance of the Object detection not only by training it on a dense object training dataset but also we can use better loss functions to further improve the recall results.

In the next sections we can look at the results and evaluation matrix for each of these scenarios.

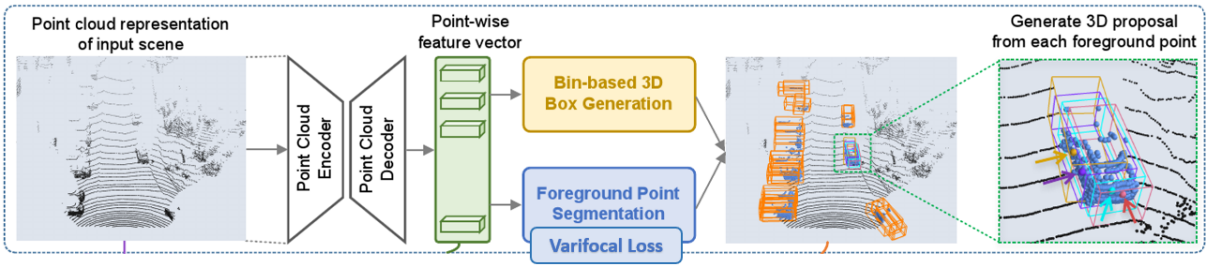
6.1.2 3D Detection Model - Primary results

 Point RCNN Baseline	This is the baseline model trained on standard Kitty dataset. Evaluation is done on Dense/crowd kitti dataset at hard level and it has failed to perform well with a bounding box recall rate of 0.245. Training: Standard Kitti dataset Evaluation: Crowd Augmented Kitti dataset
 Point RCNN Dense	This is the Dense PointRCNN model trained on the Crowd Augmented Kitti dataset. Evaluation is done on Crowd Augmented Kitti dataset as well at hard level and performance improved at 0.318 Training: Crowd Augmented Kitti dataset Evaluation: Crowd Augmented Kitti dataset

 Ground Truth	The given image shows the ground truth of the Crowd Augmented Kitti dataset which was generated.

6.1.3 3D Detection Model - Updated Loss Function

To improve on the model we adopted the varifocal loss instead of simple loss function.
As multiple objects within a small area, picking out the correct bounding boxes becomes difficult.Although correct but bounding boxes overlap too much hence simple loss functions such as IoU typically underperform.
Hence we are Applying "Varifocal loss" under Foreground Point Segmentation Stage which Helps in asymmetrical weighting, as it assigns More weightage to the high-quality positive samples and Less weightage to the negative samples



<u>Standard Scenario</u> - Point RCNN Implementation

- Foreground points provide rich information on predicting their associated objects' locations and orientations
- Model is forced to capture contextual information for making accurate point-wise prediction
- Hence **foreground segmentation** and **3D bounding box** proposal generation are **performed simultaneously**
- Default **focal loss function** is defined as per image below:

$$\mathcal{L}_{\text{focal}}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t),$$

$$\text{where } p_t = \begin{cases} p & \text{for forground point} \\ 1-p & \text{otherwise} \end{cases}$$

- During training point cloud segmentation, we keep the default settings $\alpha t = 0.25$ and $\gamma = 2$ as the original paper

<u>Dense Scenario</u> - "Variable" focal loss to improve detection accuracy

- **IOU-aware Classification Score (IACS) -** We have adopted a very recent work "varifocal loss" to predict IoU between the predicted bounding box and the ground-truth bounding box.
- Our classification **loss function** is defined as:

$$L_{\text{cls}} = \frac{1}{M}\sum_{i} \text{VFL}_i,$$

- Where VFL is **Varifocal Loss** is defined as:

$$\mathrm{VFL}(p,q) = \begin{cases} -q(q\log(p) + (1-q)\log(1-p)), & q > 0 \\ -\alpha p^{\gamma}\log(1-p), & q = 0, \end{cases}$$

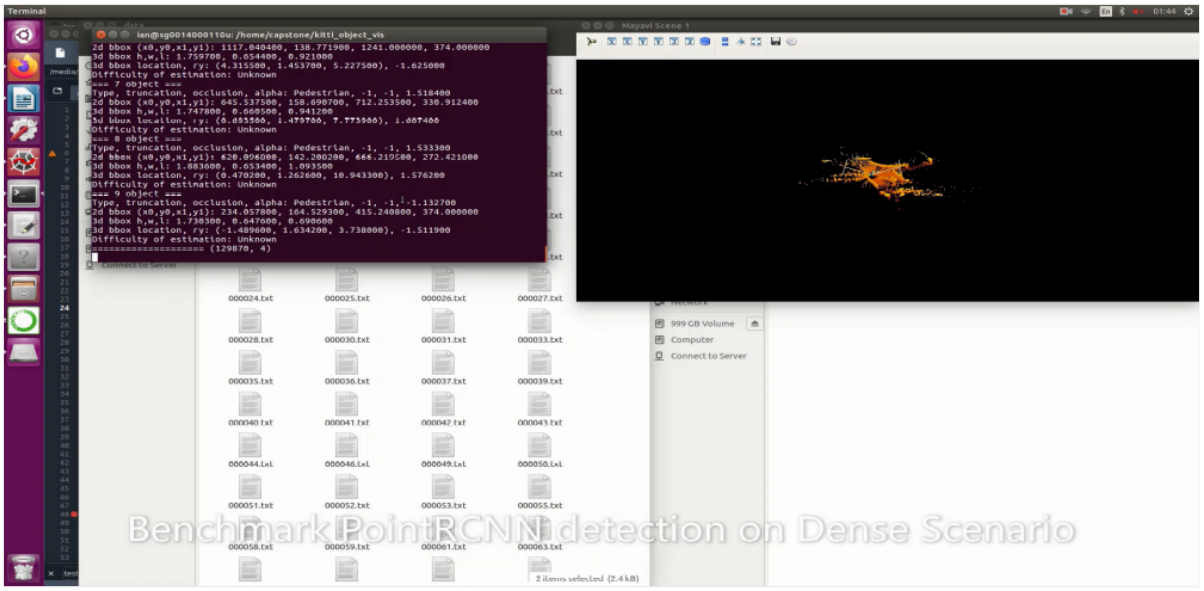
- **p** is the predicted score, and **q** is the IoU between the predicted bounding box and the ground-truth bounding box.
- (**α=0.75** and **γ=2** play a similar role as in focal loss,)

Baseline vs Updated Model on the Crowd Augmented Dataset

S. No.	**Model**	**Training Dataset**	**Evaluation Dataset**	**Complexity**	**Result**	**Evaluation**
1	Point RCNN Baseline	Standard KITTI	Standard KITTI	Easy	Best	total bbox recall (thresh=0.500): 888 / 2280 = 0.389
2	Point RCNN Baseline	Standard KITTI	Crowd Augmented Dataset	Hard	Worse	total bbox recall (thresh=0.500): 20425 / 83217 = 0.245
3	Dense Point RCNN	Crowd Augmented Dataset	Crowd Augmented Dataset	Hard	Avg	total bbox recall (thresh=0.500): 26443 / 83217 = 0.318
4	Dense Point RCNN (with Varifocal Loss)	Crowd Augmented Dataset	Crowd Augmented Dataset	Hard	Better	total bbox recall (thresh=0.500): 26818 / 83217 = 0.322

Dense Detection - System Demo Screenshots

Here we will elaborate the system demo and various performance metrics obtained as a result of Dense Detection.



Link - https://www.youtube.com/watch?v=vpy4CWgZYRM (LIDAR Crowd Pedestrian Detection Demo)

7 Resource Requirements

7.1 Computing Resources

As an 3D detection model, computing resources can largely affect the training speed. To balance the training speed and cost, a system with a powerful GPU is required.

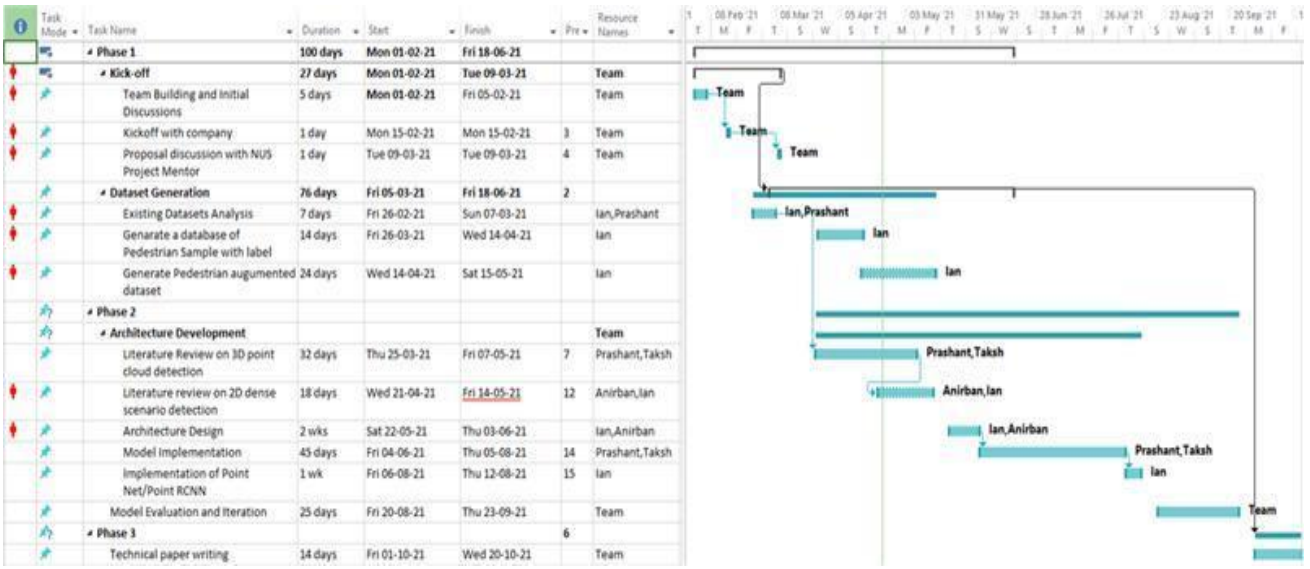
High-Performance Computing (HPC) provided by SenseTime will be used to train and test the model.

Here is the recommended technical specification:

- CPU: INTEL Xeon X5650 2.66GHz Hexa Core
- GPU: NVIDIA Tesla M2090 6GB GDDR5 512 cores
- RAM: 48GB

8 PROJECT TIMELINES

As listed below:



Task Mode	Task Name	Duration	Start	Finish	Pre	Resource Names
	Phase 1	100 days	Mon 01-02-21	Fri 18-06-21		
	Kick-off	27 days	Mon 01-02-21	Tue 09-03-21		Team
	Team Building and Initial Discussions	5 days	Mon 01-02-21	Fri 05-02-21		Team
	Kickoff with company	1 day	Mon 15-02-21	Mon 15-02-21	3	Team
	Proposal discussion with NUS Project Mentor	1 day	Tue 09-03-21	Tue 09-03-21	4	Team
	Dataset Generation	76 days	Fri 05-03-21	Fri 18-06-21	2	
	Existing Datasets Analysis	7 days	Fri 26-02-21	Sun 07-03-21		Ian,Prashant
	Genarate a database of Pedestrian Sample with label	14 days	Fri 26-03-21	Wed 14-04-21		Ian
	Generate Pedestrian augumented dataset	24 days	Wed 14-04-21	Sat 15-05-21		Ian
	Phase 2					
	Architecture Development					Team
	Literature Review on 3D point cloud detection	32 days	Thu 25-03-21	Fri 07-05-21	7	Prashant,Taksh
	Literature review on 2D dense scenario detection	18 days	Wed 21-04-21	Fri 14-05-21	12	Anirban,Ian
	Architecture Design	2 wks	Sat 22-05-21	Thu 03-06-21		Ian,Anirban
	Model Implementation	45 days	Fri 04-06-21	Thu 05-08-21	14	Prashant,Taksh
	Implementation of Point Net/Point RCNN	1 wk	Fri 06-08-21	Thu 12-08-21	15	Ian
	Model Evaluation and Iteration	25 days	Fri 20-08-21	Thu 23-09-21		Team
	Phase 3				6	
	Technical paper writing	14 days	Fri 01-10-21	Wed 20-10-21		Team

Project Journey

Our Project started in Jan 2021, when Li Ian, Taksh, Anirban and Prashant teamed up to work on Capstone. As Ian was already working in SenseTime (AI), he helped us to get the sponsorship of this project. He also helped to arrange a couple of meetings with R&D team architects and developers to get insight in LIDAR and Autonomous vehicles domain and various datasets that were available and various challenges that they had faced. Ian proposed the idea of Augmenting the existing Standard Kitti Dataset in order to improve dense detection performance using any of the available 3D object detection pipeline utilizing Point Cloud 3D data and/or fusion based networks. We all liked the main idea and understood the steps and challenges involved and started building our knowledge by finding out relevant material available online and going through different white papers on 3D point cloud object models implementations using LIDAR or sensor Fusion approaches. As these require a lot of computing power we all upgraded our Google Colab to Pro and Ian / Taksh also tried to set up better machines for our testing as well as parallel development. Ian primarily focussed on Data Augmentation strategies, implementation. Prashant and Taksh primarily looked at 3D models and started applying POC concepts to finalize the base model. Anirban helped in between in various areas including 2D and fusion based models. Ian, Prashant and Taksh also worked in the existing datasets analysis as a pre-step before dataset augmentation. Prashant also helped in Team collaboration, planning and milestones. In a nutshell, everyone contributed in various areas from literature review, to pipeline design, implementation and testing the scenarios. We worked in individual capacity, as a sub-group of two people and collaborated as a team together in certain areas like System design, implementation, testing & evaluation. Pairing sessions to focus on specific challenges and bottlenecks in smaller groups helped in faster/parallel development. For example, to train and test data with standard Kitti and augmented crowd data we assigned each subtask to individuals and got parallel results, but System architecture, brainstorming on choosing right loss function and other common activities like

report, presentation and technical writing, we also worked as a team. Overall it was a great learning experience for all of us. To summarize our individual areas of focus we have added below section -

Li Ian

- Proposed the Idea / Inputs on Data Augmentation strategies
- Helped connecting the team with Sense-time
- Procured Infra from Sense-time for running heavy tasks
- Existing Datasets Analysis
- Overall System Architecture Design / Evaluation / Testing
- Data Augmentation Pipeline - Generated DB, Algo, Implementation, testing
- Evaluation of scenario 2, standard model training and evaluation using augmented dataset
- System Demos and PPT development

Prashant Chaudhary

- 3D Literature Review including Dense Detection & Loss functions (Varifocal Loss)
- Verified various 3D implementations in public domain and created POC for selected models
- Overall System Architecture Design / Evaluation / Testing
- Standard Datasets Evaluation
- 3D Model - Implementation & Testing
- Evaluation of scenario 4 , updated model trained and evaluated using augmented dataset
- Report & PPT development
- Build the team, Project Coordination, Planning and milestones tracking

Ankeit Taksh

- 3D Literature Review including Dense Detection
- Verified various 3D implementations in public domain and created POC for selected models
- System Architecture Design / Evaluation / Testing
- Procured Infra for parallel development
- 3D Model - Implementation & Testing
- Evaluation of scenario 3, dense model training and evaluation using dense dataset
- Report and PPT development

Anirban Kar Chaudhuri

- 2D/Fusion Based Approaches Literature Review including Dense Detection
- Overall System Architecture Design / Evaluation / Testing
- Supported team in verifying models by writing quick utilities to support rapid development
- Evaluation of scenario 1, standard model training and standard using standard Kitti dataset
- Report and PPT development

9 Technical Challenges & Benefits to SenseTime

9.1 Technical Challenges

It is a requirement of SenseTime that the PyTorch framework is used to build and train the Pedestrian Attribute Recognition model. This is to fit in with other deep learning frameworks and modules used at SenseTime.

However, as PyTorch is new to the team, some ramp up might be required to understand the framework and build/train PyTorch based models effectively. Also, the level of technical depth available in PyTorch is more than in TensorFlow, whilst this allows for more fine-tuning and tweaking it is supposedly also harder.
Furthermore, there are few publicly known PyTorch based models for the Pedestrian Attribute Recognition research area which can be used as reference architectures for comparison and improvement.

9.2 Benefits to the sponsor company (SenseTime)

Augmented Data opens up a platform for the future R&D projects - This augmented dataset created for Capstone project will serve as a baseline for the future R&D projects in SenseTime. Depending upon the situation for which a model has to be applied, specific datasets can be prepared and help to train the models for better performance and accuracy. A good benchmark dataset is a cornerstone of model training. With a good

benchmark dataset, the R & D team can significantly reduce the time spent on data collection and labelling required before training a model.

Positive direction to explore possibilities - As the updated model with varifocal loss function trained on crowd augmented dataset performing well also shows a positive direction for SenseTime to continue exploring in this direction. With application of these data augmentation techniques, standard datasets can be altered on specific scenarios that may not be available in the real world or may be too specific for a use case, thereby giving a more strategic advantage in testing and evaluation of models.

10 References

[1] Z. Liang, Y. Guo, Y. Feng, W. Chen, L. Qiao, L. Zhou, J. Zhang, and H. Liu, "Stereo matching using multi-level cost volume and multi-scale feature constancy," IEEE TPAMI, 2019.
[2] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," IJCV, 2013.
[3] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: a survey," IEEE TPAMI, 2014.
[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in CVPR, 2017.
[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in CVPR, 2017.
[6] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapeNets: A deep representation for volumetric shapes," in CVPR, 2015.
[7] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in ICCV, 2019.
[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, and H. Su, "ShapeNet: An information-rich 3D model repository," arXiv preprint arXiv:1512.03012, 2015.
[9] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in CVPR, 2019.
[10] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in CVPR, 2016.
[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in CVPR, 2017.
[12] T. Hackel, N. Savinov, L. Ladicky, J. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.net: A new large-scale point cloud classification benchmark," ISPRS, 2017.
[13] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "Apollocar3D: A large 3D car instance understanding benchmark for autonomous driving," in CVPR, 2019.
[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving," in CVPR, 2012.
[15] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of lidar sequences," in ICCV, 2019.
[16] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in CVPR, 2017, pp. 4631–4640.
[17] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge," in ICRA, 2017, pp. 1386–1383.
[18] X. Han, H. Laga, and M. Bennamoun, "Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era," IEEE TPAMI, 2019.
[19] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3D data: A survey," ACM Computing Surveys, 2017.
[20] E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada, and B. Ottersten, "Deep learning advances on different 3D data representations: A survey," arXiv preprint arXiv:1808.01462, 2018.
[21] Y. Xie, J. Tian, and X. Zhu, "A review of point cloud semantic segmentation," IEEE GRSM, 2020.
[22] M. M. Rahman, Y. Tan, J. Xue, and K. Lu, "Recent advances in 3D object detection in the era of deep neural networks: A survey," IEEE TIP, 2019.

[23] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," Machine Vision and Applications, vol. 19, no. 4, pp. 261–275, 2008.
[24] M. De Deuge, B. Douillard, C. Hung, and A. Quadros, "Unsupervised feature learning for classification of outdoor 3D scans," in ACRA, 2013.
[25] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in CVPR, 2015.
[26] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," in ICRA, 2019.
[27] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan et al., "Argoverse: 3D tracking and forecasting with rich maps," in CVPR, 2019.
[28] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska et al., "Lyft level 5 av dataset 2019," 2019.
[29] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, "A*3D dataset: Towards autonomous driving in challenging environments," ICRA, 2020.
[30] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in CVPR, 2020.
[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in CVPR, 2020.
[32] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in CVPR, 2009, pp. 975–982.
[33] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The isprs benchmark on urban object classification and 3D building reconstruction," ISPRS, 2012.
[34] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, "Parisrue-madame database: a 3D mobile laser scanner dataset forbenchmarking urban detection, segmentation and classification methods," in ICRA, 2014.
[35] B. Vallet, M. Bredif, A. Serna, B. Marcotegui, and N. Paparoditis, ´ "Terramobilita/iqmulus urban point cloud analysis benchmark," Computers & Graphics, vol. 49, pp. 126–133, 2015.
[36] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," IJRR, 2018.
[37] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, K. Yang, and J. Li, "Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways," arXiv preprint arXiv:2003.08284, 2020.
[38] N. Varney, V. K. Asari, and Q. Graehling, "Dales: A large-scale aerial lidar data set for semantic segmentation," arXiv preprint arXiv:2004.11985, 2020.
[39] H. Lu, X. Chen, G. Zhang, Q. Zhou, Y. Ma, and Y. Zhao, "SCANet: Spatial-channel attention network for 3D object detection," in ICASSP, 2019.
[40] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multiview convolutional neural networks for 3D shape recognition," in ICCV, 2015.
[41] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in CVPR, 2018.
[42] Z. Yang and L. Wang, "Learning relationships for multi-view 3D object recognition," in ICCV, 2019.
[43] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in CVPR, 2016.
[44] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Groupview convolutional neural networks for 3D shape recognition," in CVPR, 2018.
[45] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," BMVC, 2017.
[46] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with LSTM for 3D shape recognition and retrieval," IEEE TMM, 2018.
[47] X. Wei, R. Yu, and J. Sun, "View-gcn: View-based graph convolutional network for 3D shape analysis," in CVPR, 2020.

[48] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in IROS, 2015.
[49] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in CVPR, 2017.
[50] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "OCNN: Octree-based convolutional neural networks for 3D shape analysis," ACM TOG, 2017.
[51] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in CVPR, 2018.
[52] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3D point cloud classification and segmentation using 3D modified fisher vector representation for convolutional neural networks," arXiv preprint arXiv:1711.08241, 2017.
[53] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in NeurIPS, 2017.
[54] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in NeurIPS, 2017.
[55] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, "Deep Learning for 3D Point Clouds: A Survey" arXiv preprint arXiv, 2020.
[56] J. Antonio G. Trejo, D. A. M.Ravell, "Dense Crowds Detection and Counting with a Lightweight Architecture" arXiv preprint arXiv, 2020.
[57] E. Goldman,R. Herzig, A.Eisenschtat, O. Ratzon, I. Levi, J. Goldberger, Tal H., "Precise Detection in Densely Packed Scenes" arXiv preprint arXiv, 2019.
[58] T-Yi Lin, Priya Goyal, Ross G., K. P. Dollar, "Focal Loss for Dense Object Detection" FAIR, 2018.
[59] Haoyang Z., Y. Wang, F. Dayoub, Niko S., "VarifocalNet: An IoU-aware Dense Object Detector" arXiv, 2021.
[60] Jie Zhou, Xin Tan, Z Shao, Lizhuang Ma., "FVNet: 3D Front-View Proposal Generation for Real-Time Object Detection from Point Clouds" arXiv, 2019.
[61] Inc. Velodyne LiDAR. 2018. HDL-64E Data Sheet. http://velodynelidar.com/docs/datasheet/63-9194_Rev-F_HDL64E_S3_DataSheet_Web.pdf
[62] Yu Feng, Alexander Schlichting, and Claus Brenner. 2016. 3D feature point extraction from LiDAR data using a neural network. ISPRS Archives 2016.
[63] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving" IEEE, 2016.
[64] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local pointwise features for amodal 3d object detection" arXiv:1903.01864, 2019.
[65] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "IPOD: Intensive point-based object detector for point cloud," arXiv preprint arXiv:1812.05276, 2018.
[66] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in CVPR, 2019.
[67] Jie Li, Yu Hu1, "A Density-Aware PointRCNN for 3D Object Detection in Point Clouds" in arXiv, 2021.
[68] Bei Wang, Jianping An, Jiayan Cao, "Voxel-FPN: multi-scale voxel feature aggregation in 3D object detection from point clouds" in arXiv, 2019
[69] Jaeseok Choi, Yeji Song and Nojun Kwak.Part-Aware Data Augmentation for 3D Object Detection in Point Cloud, 2018
[70] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 2018
[71] Bin Yang, Wenjie Luo, Raquel Urtasun, PIXOR: Real-time 3D Object Detection from Point Clouds, 2019
[72] Kazuki Minemura, Hengfui Liau, Abraham Monrroy, Shinpei Kato, LMNet: Real-time Multiclass Object Detection on CPU using 3D LiDAR
[73] Haoyang Zhang, Ying Wang, Feras Dayoub, Niko Sünderhauf, VarifocalNet: An IoU-aware Dense Object Detector, arXiv:2008.13367