Learned clustering for 3D object segmentation

Sriram Natarajan, *sriramnatarajan@student.utwente.nl*
Julien Vijverberg, *JVijverberg@cyclomedia.com*
Syed Zille Hussnain, *ZeHussnain@cyclomedia.com*
Luuk Spreeuwers, *l.j.spreeuwers@utwente.nl*

***Abstract*—Applications related to autonomous driving, urban planning and asset monitoring rely on accurate information about the objects and their location in real world coordinates. Identifying stationary objects is one such application that finds importance in urban planning and asset monitoring, for instance: detection of roadside billboards, lamp posts etc. With the availability of point cloud representations of the environment, several approaches have been proposed for detection and segmentation of stationary objects in 3D. The detection of billboards is one such application which is challenging because of its incoherent visibility in multi-view images and absence of depth information due to its shape. This paper proposes Joint SPLATNet3D for semantic-instance segmentation of stationary objects in the scene. The proposed network performs two tasks: predicts a semantic label and generates an instance embedding for every 3D point. The multi-task loss function enables the network to jointly optimize the two tasks. This paper describes the dataset generation and feasibility study of semantic and instance segmentation for billboards. The paper gives a comparative analysis of Joint SPLATNet3D and MT-PNet for both the tasks. Preliminary experiments on semantic segmentation show that SPLATNet3D gives an average IoU of 75% in comparison with MT-PNet which gives an IoU of 46%. Experiments on joint training show that Joint SPLATNet3D gives an IoU of 68% in comparison with MT-PNet which gives an IoU of 48% for semantic segmentation. The results of instance segmentation for both the networks do not show good improvements for this dataset.**

***Index Terms*—Urban planning, billboard detection, SPLATNet3D, MT-PNet, Joint SPLATNet3D, Multi-task loss function.**

I. Introduction

The accurate perception of visual information about streets and roads, play a crucial role in applications related to urban planning. The emergence of street view imagery has been beneficial for development of algorithms related to object detection in real world scenarios [1], [2]. Street view datasets contain plethora of images covering a large stretch of geography, depicting street views captured at regular intervals. Recent developments in lidar and camera sensors have been useful for creating point cloud representations of the environment [3]. This enables us to address different object detection and segmentation problems in 3D. Several approaches have been proposed in literature for learning features from images and lidar [4], [5], [6], [7], [8].

Applications related to urban planning rely on geographic information of objects in the environment and billboard detection is one such application. Advertisement billboards are used for marketing a brand, company, product, service etc. They are used as a tool to create awareness among people or broadcast information about a specific product or service. Billboards are usually placed in areas with high traffic, for example along highways and cities so that they are seen by a large group of people. Traditional approaches in urban planning involved capturing images of the streets and manually annotating the location of every billboard. The availability of street view images and lidar data have made it possible to automatically detect billboards in 3D. There are number of data acquisition setups used for capturing information from camera and lidar. The data acquisition setup used for experimentation in this paper is described in Fig 1 and is proprietary to Cyclomedia Technology [9]. The setup includes sensors mounted on a car with four cameras front(F), back(B), right(R), left(L) and a lidar placed at the center. The images capture information of several objects in the scene and billboards are one such objects of interest for this study. A point cloud representation of this data is generated by projecting every pixel in the image to 3D using camera parameters and depth information from lidar. The lidar is slightly slanted backwards so depth information may not be available for the whole image.

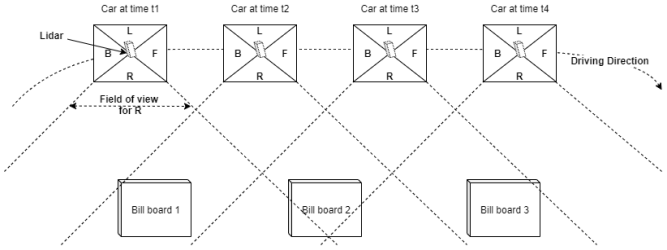

Fig. 1: Street view image and point cloud acquisition setup [10]. The setup describes four cameras (F,B,R,L) and a lidar that is slightly slanted backwards

The images contain billboards of varying aspect ratios, shapes and colors. This leads to several challenges in detecting billboards. First, billboards come in arbitrary shape and running 2D object detection algorithms [11], [12] on these images would generate loosely fit bounding boxes as in Fig 2. As a result the pixels apart from actual billboard result in noise when projected to 3D. This motivates the choice of segmentation based approaches because it is possible to segment regions of arbitrary shapes. Second, billboards that are placed close to each other as in Fig 3 are incoherently visible in successive frames and projecting the same information to 3D leads to false positives. Hence it is important for an algorithm to learn these distinct attributes in 3D and point

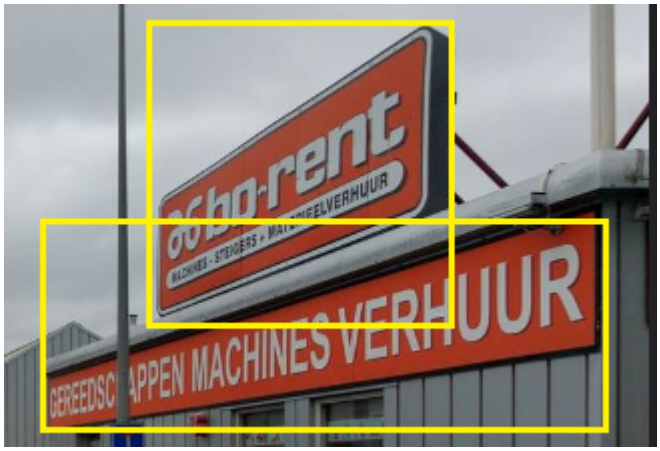

Fig. 2: Instance of an image from billboards dataset[10] showing loosely fit bounding boxes when viewed as 2D object detection problem with images.

Fig. 3: Illustration of multiple billboards of the same category where each billboard has to be detected as a separate instance after projecting it to 3D [10]

cloud representations could be leveraged to overcome this problem.

This paper proposes Joint SPLATNet3D for semantic-instance segmentation of stationary objects in the scene. The network is trained on 3D point clouds to jointly optimise the two tasks using a multi-task loss function. The key contributions of the work are: 1. Dataset generation for training and evaluation with semantic and instance labels. 2. Feasibility study on semantic segmentation of billboards with SPLATNet3D[13] and MT-PNet[14]. 3. Comparative study of Joint SPLATNet3D and MT-PNet. The feasibility study shows that SPLATNet3D gives an IoU of 75% whereas MT-PNet gives 46%. The experiments on joint training show that Joint SPLATNet3D gives an IoU of 68% and MT-PNet gives an IoU of 48% for semantic segmentation on billboards. Both the networks do not show good improvements on instance segmentation for this dataset.

The paper is organized as follows: Section II describes the related research in this area. Section III describes the complete methodology adopted to realise the approach. Section IV describes the dataset used for analysis, performance evaluation metrics and results. Section V presents the inferences made from the study and discusses the possible shortcomings in the proposed idea. Section VI presents the overall learning from the work and also lists directions for further research.

II. Related work

The availability of lidar data combined with the power of deep learning has drawn enormous attention of several researchers. Object detection with lidar, requires techniques different from the ones used for detection in images. The lidar data is sparse in nature and this makes it difficult for algorithms to detect object that are far away. On the other hand, images give dense representation of the regions but lack depth information [15]. This has led to development of several approaches that learn features from lidar and camera.

A. *Approaches based on multi-view images*

The class of techniques that use multi-view images and 2D convolutional neural networks(CNN) to tackle the problem of detection/segmentation fall under this category. One of the most promising approaches in this direction was MVCNN [16] where 2D CNNs were used to generate region proposals and the results were projected to 3D. Nassar et.al [17] proposed a joint learning technique for robust object detection in panoramic images. The approaches based on multi-view images have shown promising results but have inherent shortcomings. The geometric information with multi-view images is not as accurate as that of lidar point clouds. In scenarios where complex structures are involved it is difficult to choose the number of viewpoints that capture all the details of the scene [18].

B. *Approaches based on transformed lidar maps*

Transformed lidar maps are of two categories: range view (front view) and bird's eye view (top view). The work in VeloFCN [19] describes the approach to generate front view map and performs 3D object detection with fully convolutional networks. 3DFCN [20], PIXOR [21] use the bird's eye view map to train a fully convolutional network. The lidar maps used in the above works are generated using 2D projections and this results in loss of surface and depth information. The choice of viewpoint is based on heuristics and is specific to the object categories under study [13]. Voxel based approaches convert the unstructured point cloud to a 3D volumetric grid and thereby applying 3D convolutions on voxelized shapes [22]. This approach is computationally expensive because it involves 3D convolutions and volumetric representation is constrained by its resolution.

C. *Joint lidar and camera based approaches*

The approaches in this class could be described in two categories: cascading approaches and parallel approaches.

*1) **Cascading approaches**:* The approaches in this category generate region proposals on images using 2D object detection and use lidar to generate final predictions in 3D. Frustum PointNet[4] and Frustum ConvNet [5] use lidar depth map to project region proposals from images to a frustum like structure. The projected points are clustered using PointNets [6]. The performance of this class of techniques are bounded by the precision of 2D object detection. The region proposals of true detections are only projected to 3D and false negatives cannot be recovered.

*2) **Parallel approaches**:* AVOD [7] and MV3D [8] are approaches that use transformed lidar maps. These approaches use 2D convolutional networks as feature extractors to train separately on images and lidar maps. The feature vectors are fused to generate 3D detections. Contfuse [15] proposes deep continuous fusion to aggregate features from images and lidar maps. This class of approaches would be useful for objects like car, pedestrian and bicycle which have evident height, width and depth attributes. This would not be useful for objects like billboards because: 1. the height attribute of the billboard would be lost with top view image and 2. front view projection would lead to loss of depth information.

D. Approaches based on lidar point clouds

Point clouds are defined as unordered set of vectors with irregular structure. They represent geometric data of the environment which is very useful for 3D object detection. PointNets [6] have been one of the pioneering works for learning features directly on point clouds. The PointNets tend to neglect the geometric relationship between a point and its neighbours so improvements were proposed in PointNet++[23]. Alternate approaches use graph based structures in neural networks to learn the local structure of the objects. Some of the notable works are [24], [25], [26], [27]. Sparse Lattice Networks (SPLATNet) were proposed in [13] for point cloud segmentation and image labelling. The paper proposes two sub-networks 1. SPLATNet3D that processes 3D point clouds and 2. SPLATNet2D-3D that processes multi-view images and 3D point clouds. The works described above are related to semantic and part segmentation of point clouds. Instance segmentation is another important aspect in 3D segmentation and different techniques have been proposed. Semantic segmentation aims to predict a class label whereas instance segmentation clusters the semantic labels into instances. Instance segmentation is seen as a post processing on semantic segmentation [4], [28]. SGPN [29] has been one of the pioneering works that defined the similarity matrix for clustering instances. This comes with a drawback that the size of the similarity matrix increases quadratically with the number of points in the point cloud. Multi-task loss functions have been used for jointly learning multiple tasks like detection and segmentation [30]. Multi-task Pointwise Networks(MT-PNet) [14] proposes joint semantic-instance segmentation with multi-task loss function. The network is trained to simultaneously predict a semantic label and generate an instance embedding for every 3D point. The SPLATNet3D and MT-PNet are considered for analysis and Joint SPLATNet3D is proposed based on the discriminative loss function [31].

III. Methodology

The primary step involves feasibility study of semantic segmentation for billboards dataset. This includes dataset generation and training the two categories of networks: SPLATNet3D and MT-PNet. The details of dataset generation is described in this section and the experiments are detailed in Section IV.

The proposed approach for semantic-instance segmentation (Joint SPLATNet3D) is shown in Fig 4. Joint SPLATNet3D uses SPLATNet3D as a feature extractor and then diverges to two tasks: predicting a semantic label and creating an instance embedding for every 3D point. The network is trained on 3D point clouds where each point cloud $P \in \mathbb{R}^{n \times d}$, here n is the number of points and d is the number of features. The features in a point cloud can be point locations (XYZ), color information (RGB), surface normal etc. This paper uses point clouds with locations and color features where the color features are obtained from corresponding image pixels. The network is trained using a multi-task loss function defined as the sum of prediction loss and embedding loss. The prediction loss is based on softmax loss [32] and the embedding loss is based on the discriminative loss function [31]. The technique of using multi-task loss function is inspired from MT-PNet which uses PointNet as a feature extractor instead. Hence this study involves a comparative analysis of Joint SPLATNet3D and MT-PNet. The building blocks of Joint SPLATNet3D are Bilateral Convolutional Layer [33] and multi-task loss function.

A. Bilateral Convolutional Layer (BCL)

BCL describes a way to include sparse high dimensional filtering in neural networks. The approach in SPLATNet3D builds upon this idea to learn features from high dimensional, sparse point clouds. The network uses a stack of BCL's and 1x1 convolutional layers to generate per-point predictions. The BCL takes an input point cloud and performs three steps: Splat, Convolve and Slice as described in Fig 5

- **Splat:** Let $F \in \mathbb{R}^{n \times d_f}$ be the input features given to a BCL, where n is the number of points and d_f is the number of features (XYZRGB). BCL takes input features F and projects it onto a lattice space using barycentric interpolation [13]. A lattice space is defined using permutohedral lattice [34] with flexible representation of features. The choice of interpolation and lattice structure has been motivated by the authors in [13]. The grid spacing in the lattice structure is controlled using the Λ parameter called the scaling matrix.
- **Convolve:** BCL performs convolution on the splatted signal with learnable filter kernels.
- **Slice:** the convolved signal is mapped back to input signal using barycentric interpolation and the output could be used for further processing.

B. Multi-task loss function

The features extracted from SPLATNet3D are used to learn semantic and instance labels using a multi-task loss function. The output of the final 1×1 convolution layer in Fig 4 is given

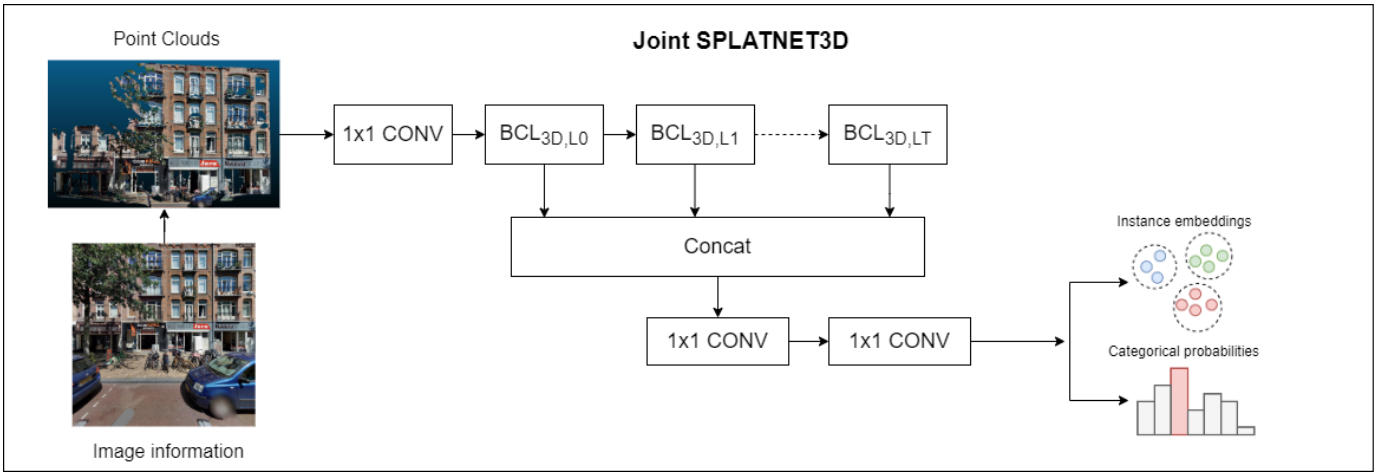

Fig. 4: Joint SPLATNet3D - proposed network for joint semantic-instance segmentation of point clouds. The network is trained with point clouds and the output of the final convolutional layer is given to two tasks: predict semantic labels and generate instance embeddings

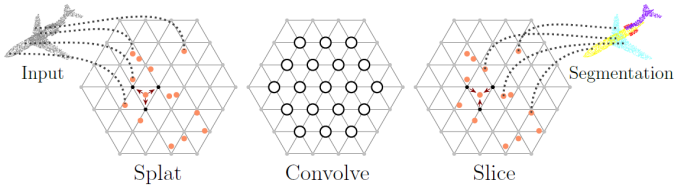

Fig. 5: Processing steps in BCL as described in [13]. Splat: interpolates the features of the input point cloud onto a permutohedral lattice. Convolve: performs sparse filtering on the splatted signal with learnable kernels. Slice: interpolates the convolved signal back to the input signal. Output is mapped as segmentation labels.

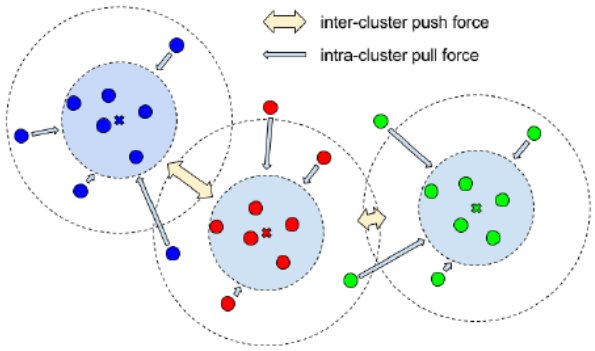

Fig. 6: The forces defined in the discriminative loss function [31]. The pull force draws the embeddings towards the mean of the cluster and push force tries to maintain the cluster centers far apart. The margins highlighted as dotted circles describe the active region of the forces.

to two tasks: predicting semantic labels and creating instance embeddings. The total loss (L) is defined as sum of two losses as defined in equation 1.

$$L = L_{prediction} + L_{embedding} \tag{1}$$

The prediction loss ($L_{prediction}$) is defined by the softmax loss [32] and embedding loss ($L_{embedding}$) is defined by the discriminative loss [31]. The intuition behind discriminative loss function is that embeddings with same instance label would end up close together and the ones that belong to different instances would lie apart as seen in Fig 6. The loss function described in equation 2 is based on pull and push forces among the clusters. A cluster is defined as group of point embeddings that belong to the same instance [14].

$$L_{embedding} = \alpha L_{pull} + \beta L_{push} + \gamma L_{reg} \tag{2}$$

The parameters defined in equation 2 namely α, β, γ are hyperparameters and the loss terms are defined below. The L_{pull}, L_{push} and L_{reg} functions are adapted from [14].

- L_{pull} or variance is the force within the cluster that pulls an embedding towards its center. The active region of the force is controlled by the parameter δ_v as seen by the dotted circle in Fig 6.
- L_{push} or distance is the force that pushes the cluster centers away from each other. The active region of push force is controlled by the parameter δ_d as seen by the dotted circle in Fig 6.
- L_{reg} or regularization is described as a pulling force that draws all clusters close to the origin.

C. Network Architecture

The architecture has been adapted from [13] and [14]. The input point cloud is passed through a 1×1 convolutional layer followed by a stack of 5 BCLs. The BCLs operate on permutohedral lattice defined by the lattice features (XYZRGB). The lattice scales(λ) for 5 BCLs are set as: (λ_0, $\lambda_0/2$, $\lambda_0/4$, $\lambda_0/8$, $\lambda_0/16$) respectively. The output of 5 BCLs are concatenated and passed through two 1×1 convolutional

layers. The concatenation of outputs from different layers has been motivated by the work in [35]. The output is then passed to a softmax layer and an embedding layer. The softmax layer generates class-wise semantic probability for each point. The embedding layer generates a vector of dimension $(n \times 32)$, where n is the number of points. The parameters of the embedding loss are defined as: $\alpha = 1, \beta = 1, \gamma = 0.001$ based on the intuition that L_{pull} and L_{push} are equally weighted. The parameters δ_d, δ_v are hyperparameters for training and setting $\delta_d > 2\delta_v$ ensures each embedding is closer to its own cluster center[14] .

D. Dataset generation

The billboards dataset [10] contains camera images and depth images from lidar, captured using the setup described in Section I. This dataset is proprietary to Cyclomedia Technology and is used for experimental purposes in this paper. A point cloud representation of this dataset is created by projecting the color information of every pixel in the image to 3D using the depth information from lidar. The information in the point clouds is represented by 3D coordinates (XYZ) and color information (RGB). Thus point clouds are generated for every image and are called as full image point clouds as shown in Fig 7.

Fig. 7: Snapshot of a full image point cloud from billboards dataset. The regions highlighted in the point cloud belong to one category of billboard called sign_facade. The highlighted regions are to be segmented as area point clouds.

This study aims to segment the regions of point cloud that contain a billboard in it. An area point cloud is defined as the region or area of billboard extracted from the full image point cloud. The dataset contains four classes of billboards: sign_facade, sign_ground, flag_facade, flag_ground as seen in Fig 8. The billboards come in different shape, background, aspect ratio and exhibit wide range of intra-class variations. Hence the experiments are conducted on a single category of billboard i.e. sign_facade. The networks are trained on two classes: a billboard and a background. The dataset generation is a two step process:

- Manual segmentation of area point clouds from full image point clouds.
- Generation of a semantic and an instance label for every 3D point. This is achieved using K-Nearest neighbour [36] algorithm. The area point cloud is the first nearest neighbor of its respective full image point cloud.

The datasets were generated in two stages: 1. Dataset for semantic segmentation and 2. Dataset for joint semantic-instance segmentation.

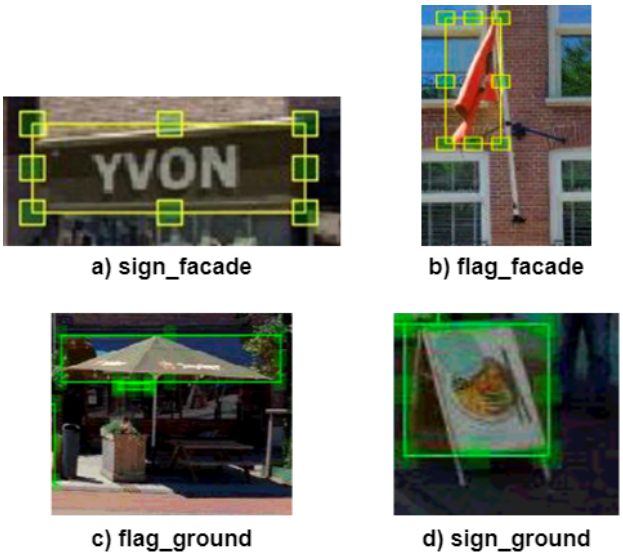

Fig. 8: The billboards in the dataset are categorized into 4 classes. There are several variants within each class and one instance of each class is seen in this image.

*1) **Dataset - Semantic Segmentation**:* The first version of dataset (V1) contains point clouds with one instance of billboard class(sign_facade) in a full image point cloud. It contains 75 point clouds for training and 25 for testing. This dataset suffered from class imbalance problem because the billboards constitute a small region of the point cloud with respect to background. In order to overcome this problem, networks were trained with weighted softmax loss function[37]. The weight for each class ($weight_{class}$) is computed as the ratio of number of points of the class over the total number of points(N_p) as in equation 3. The experiments are described in Section C 1 and based on the inferences, a revised version of the dataset (V2) was created. This dataset contains point clouds with one billboard and small region of background around it as seen in Fig 9. It contains 108 point clouds for training and 47 point clouds for testing.

$$weight_{class} = \frac{N_p[label == class]}{N_p} \tag{3}$$

*2) **Dataset - Joint Semantic and Instance Segmentation**:* The datasets used for analysis of instance segmentation has to contain multiple billboards in every point cloud. Similarly two versions of dataset have been created for analysis: dataset V3 and V4. Dataset V3 contains multiple instances of billboard with full background as seen in Fig 10. This dataset contains 84 point clouds for training and 22 point clouds for testing. Dataset V4 is a simplified version which contains multiple billboards with small background as shown in Fig 11. This

Fig. 9: Snapshots of point clouds from billboards dataset V2, describing a billboard and small region of background around it

Fig. 10: Snapshot of a point cloud from billboards dataset V3, describing multiple instances of billboard on a full background.

dataset contains 84 point clouds for training and 22 point clouds for testing.

IV. Experimental Results

The experiments are divided into two stages: semantic segmentation and joint semantic-instance segmentation. The initial experiments are based on semantic segmentation with SPLATNet3D and MT-PNet with billboards datasets V1 and V2. SPLATNet3D is based on Caffe[38] and MT-PNet is based on Pytorch[39]. The second set of experiments include joint training of semantic-instance segmentation with billboards datasets V3 and V4. The Joint SPLATNet3D has been implemented in Caffe and experiments were conducted on NVIDIA Tesla P100 graphics card.

A. Dataset

Billboards dataset: This dataset contains four versions as described in Section III D. Datasets V1 and V2 are used for feasibility study on semantic segmentation of billboards. Datasets V3 and V4 are used to study the performance of proposed the joint semantic-instance segmentation algorithm.

B. Performance Evaluation

Intersection over Union (IoU)[40] and accuracy metrics are used to evaluate semantic segmentation. The segmentation

Fig. 11: Snapshot of a point cloud from billboards dataset V4, describing multiple instances of billboards with smaller background.

network generates per-point predictions and this could be plotted as the elements of a confusion matrix namely True positives(TP), False positives(FP), True negatives(TN) and False negatives(FN). The IoU metric is defined in equation 4 and the accuracy is defined in equation 4.

$$IoU = \frac{TP}{TP + FP + FN} \tag{4}$$

$$Accuracy = \frac{TP}{Totalpredictions} \tag{5}$$

Instance segmentation is considered as an object detection task and is evaluated with mean average precision at IoU threshold of 0.5(mAP@0.5) as in [14].

C. Results

*1) **Semantic Segmentation**:* The initial experiment involved training SPLATNet3D with dataset V1. Training configuration: batch size = 2; sample size = 40000; base learning rate = 0.0001; learning rate decay = 0.5; decay rate = 10000; epochs = 50000. The network couldn't learn any useful features of billboard instead treated them as noise. Training the network with softmax loss function did not converge and we could infer that the problem was due to class imbalance in the dataset. This motivated the experiment with weighted softmax loss function instead of softmax loss function with a similar training configuration given above. The weighted softmax loss is useful when certain classes in the dataset are over or under represented. The network is trained with class weights: billboard = 50 and background = 1. The loss function has been implemented in Caffe, trained with SPLATNet3D and the results are shown in Fig 12. The plot shows that the accuracy for billboard class saturates close to 1.0 and that of background class saturates to 0.1.

This behavior of the network could be attributed to the nature of training data samples. The batches of training data couldn't represent a proper distribution of samples from both the classes. This would have caused the weighted softmax loss function to just learn the class with a higher weight. The idea of using a dataset with smaller background region around the billboard was motivated by this experiment. Hence billboards dataset V2 was used for further analysis. The SPLATNet3D was trained with weighted softmax loss function and dataset V2. The class weights were: billboard = 8 and background = 1. The weights were adjusted based on the proportion of the background with respect to billboard in the dataset. The results

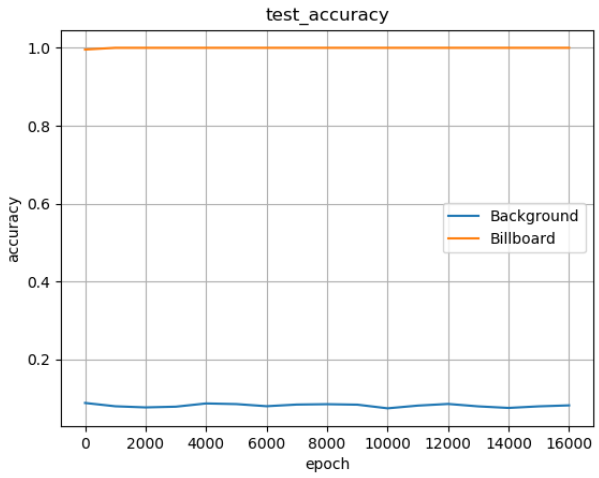

Fig. 12: Plot showing class-wise test accuracy for SPLATNet3D with weighted softmax loss function trained on dataset V1. The accuracy of billboard class saturates to 1.0 and that of background class saturates to 0.1

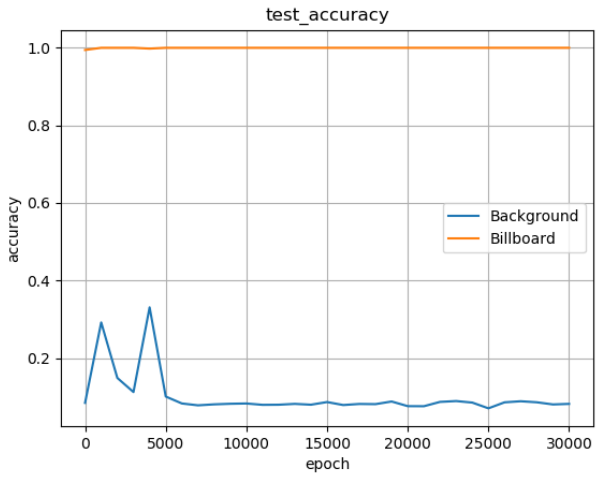

Fig. 13: Plot showing class-wise test accuracy for SPLATNet3D with weighted softmax loss function trained on dataset V2. The accuracy of billboard and background tends to saturate to 1.0 and 0.1 respectively

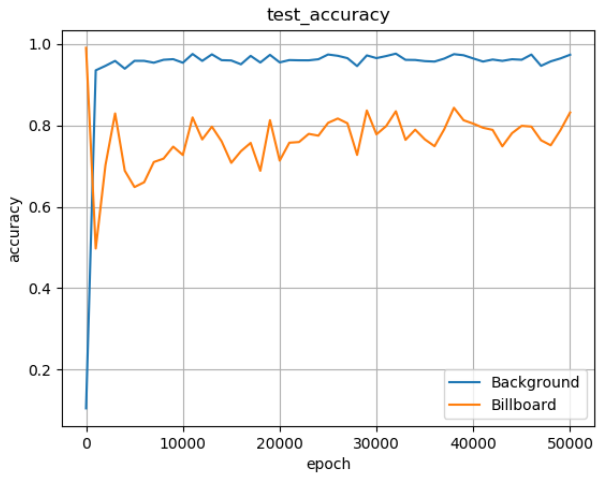

Fig. 14: Plot showing class-wise test accuracy for SPLATNet3D with softmax loss function and dataset V2. The plot shows improvement in accuracy for both the classes

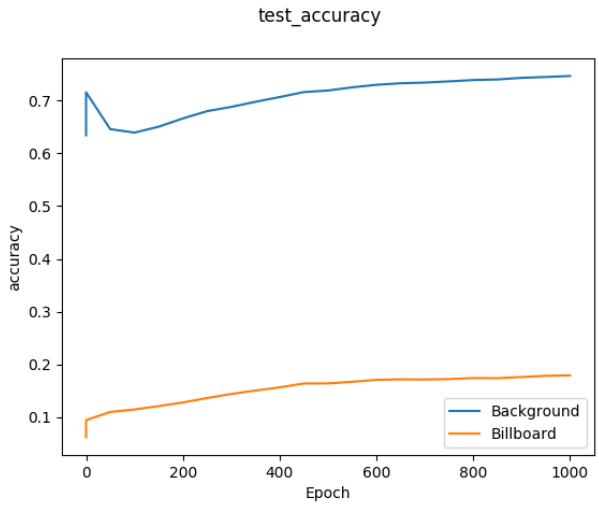

Fig. 15: Plot showing class-wise test accuracy for MT-PNet with softmax loss function and dataset V2. The accuracy of billboard is lower than that of background class

are plotted in Fig 13 and it is observed that the accuracy of both the classes are still saturating like the previous case.

The dataset V2 was better balanced than dataset V1 but still the network couldn't learn class wise features. This could have been because the weighted loss function tries to priortize the class with higher weight and it might not be suitable for this category of dataset. This led to experimentation of SPLATNet3D with softmax loss function and dataset V2. The results are shown in Fig 14 and the plot shows average accuracy of 0.78 for billboard and 0.93 for background. This shows that the network is able to learn features of background and billboard with dataset V2 and softmax loss function.

The next set of experiments were conducted with MT-PNet and dataset V2 was used because it showed promising results with SPLATNet3D. The original implementation of MT-PNet contains multi task loss function. The MT-PNet was trained with weighted softmax, softmax and joint loss functions. The training with softmax loss function gave better results. The training configuration of MT-PNet: batch size = 2; sample size = 40000; base learning rate = 0.001; learning rate decay = 0.5; decay rate = 50; epochs 1000. The plot in Fig 15 shows gradual increase in accuracy for both the classes but the final accuracy for billboard class is still low at 0.18. The background class shows better performance when compared to the billboard class. The loss of the network tends to saturate even with further decrease in learning rate and shows no further improvement in accuracy.

A comparative study of SPLATNet3D and MT-PNet for

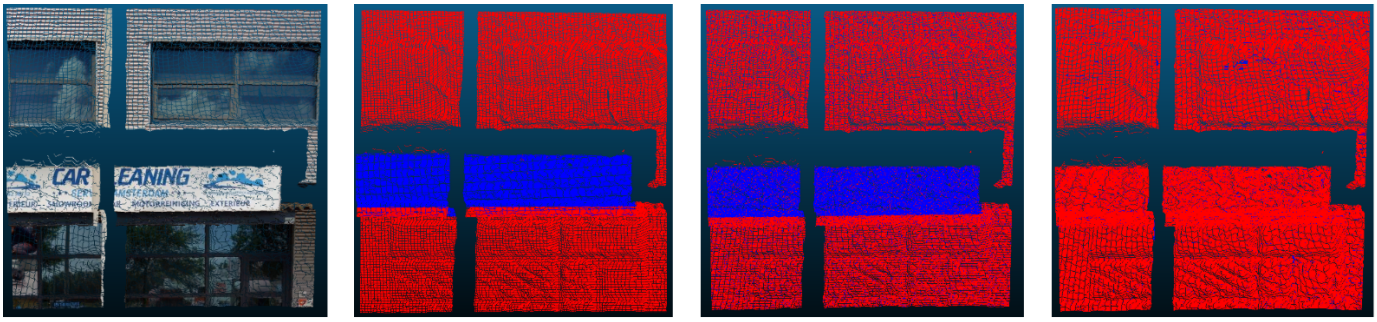

Fig. 16: Illustration of input point cloud, ground truth, prediction - SPLATNet3D and prediction - MT-PNet starting from left to right. SPLATNet3D does have many false positives but MT-PNet predicts most of the parts of point cloud as background

semantic segmentation with dataset V2 is presented TABLE I. The table describes classwise and overall average IoU for both the networks. It could be inferred from table that SPLATNet3D shows good improvement over MT-PNet for billboard class. The average IoU for background class does not vary much for both the networks. It is important to note that the performance of the networks could only be judged based on how well it classified points of the billboard and not that of the background. The prediction result for a test point cloud is shown in Fig 16. It could be inferred from Fig 16 c that SPLATNet3D shows better performance for billboard but many points in the background are misclassified. The prediction of MT-PNet in Fig 16 d shows that almost all of the points are classified as background with few random patches of points being classified as billboard.

TABLE I: Comparison of semantic segmentation with SPLATNet3D and MT-PNet for billboards dataset V2

	Semantic segmentation(IoU)		
Method	Billboard	Background	Overall
SPLATNet3D	**0.69**	**0.8**	**0.75**
MT-PNet	0.18	0.75	0.46

*2) **Joint Semantic and Instance Segmentation**:* The experiments in this section describe the training and evaluation of joint semantic-instance segmentation for Joint SPLATNet3D and MT-PNet. The results are presented in three sections: 1. Results Semantic segmentation 2. Results Instance segmentation. 3. Results overall comparison

Results - Semantic segmentation: The first experiment in this regard is carried out on Joint SPLATNet3D with softmax loss function trained on dataset V3. Training configuration: batch size = 2; sample size = 40000; base learning rate = 0.0001; learning rate decay = 0.5; decay rate = 10000; epochs = 50000; δ_d = 1.5; δ_v = 0.5. The results are shown in Fig 17 and the plot shows an average accuracy of 0.38 for billboard category. The performance of the network is very good for background but not for billboard class.

The experiments in Section IV C 1 show that dataset with smaller background gives better results for billboard. Therefore Joint SPLATNet3D was trained on dataset V4 with

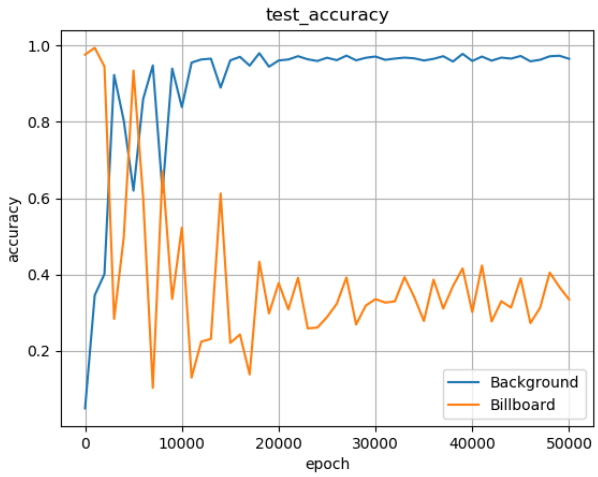

Fig. 17: Plot showing class-wise test accuracy for Joint SPLATNet3D with softmax loss function and dataset V3. The results in the plot show the accuracy for semantic segmentation trained in a joint setting

the same training configuration as in the previous experiment. The evaluation of Joint SPLATNet3D with dataset V4 is shown in Fig 18. The results show improvement in accuracy for billboard category with reduction for background. The improvement is attributed to the fact that dataset V4 contains smaller background than dataset V3. This observation is consistent with the feasibility study described in Section IV C 1. The average accuracy for billboard category is 0.58 which is better than the previous experiment but it also provides scope to further improve the performance.

The network is trained on homogeneous batches of data defined by the sample size and batch size. The data points are randomly sampled from every point cloud in the training set. The billboards represent small region of the point cloud when compared to the background. Therefore the probability of sampling points from background class is higher than that of the billboard class. This provides scope for experimentation with alternate sampling techniques which could be useful for this dataset. Sampling points equally from both the classes will generate a balanced distribution of points for every batch of

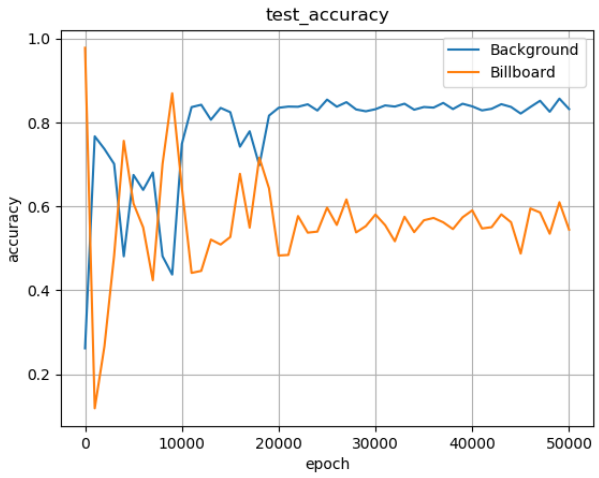

Fig. 18: Plot showing class-wise test accuracy for Joint SPLATNet3D with softmax loss function and dataset V4. The dataset with smaller background leads to improvement in the accuracy of billboard category

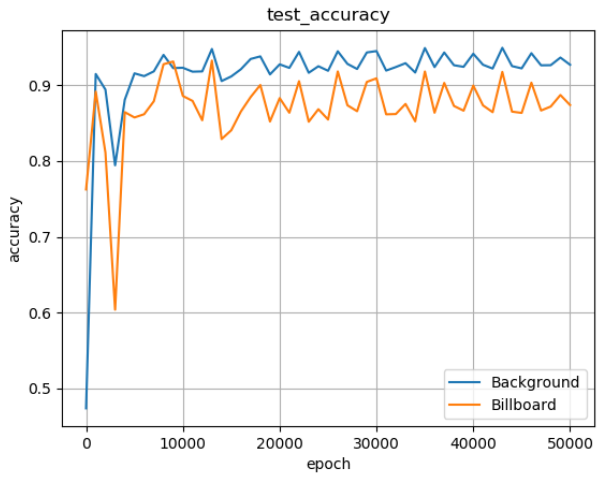

Fig. 19: Plot showing class-wise test accuracy for Joint SPLATNet3D with equal sampling and softmax loss function trained on dataset V4. The equal sampling shows improvement in the accuracy of both the categories

training. Therefore Joint SPLATNet3D with equal sampling and softmax loss function is trained on dataset V4 using a similar configuration setting as used in previous experiment. The result is shown in Fig 19 and it shows improvement in the accuracy of both the classes. This shows that training on smaller background with softmax loss function and equal sampling gives better performance.

The above experiment provides scope for analyzing the effect of weighted softmax loss function on sampling. This experiment aims to compare the performance of softmax and weighted softmax loss functions on equal sampling. The Joint SPLATNet3D is trained on dataset V4 with weighted loss function (weights: billboard = 2; background = 1). and equal sampling. The training configuration is similar to the experiments described above. The results are shown in Fig 20 and it could be inferred from the plot that accuracy of billboard and background category does not vary much with in comparison to the experiment with softmax loss function. Therefore using weighted loss function with equal sampling does not affect the performance of this network given the dataset contains small background with respect to billboards.

The next set of experiments were conducted on MT-PNet for joint semantic-instance segmentation. Similar experiments of MT-PNet on dataset V3 and V4 with random sampling does not show any improvement in the accuracy of billboard class. Therefore the MT-PNet was trained on dataset V4 with equal sampling and softmax loss function. Training configuration: batch size = 2; sample size = 40000; base learning rate = 0.0001; learning rate decay = 0.5; decay rate = 50; epochs = 2000; δ_d = 1.5; δ_v = 0.5. The results are shown in Fig 21 and the plot shows an average accuracy of 0.22 for billboard and 0.53 for background. The accuracy of both the class could be further improved.

The effect of loss function on sampling is also studied with MT-PNet. The network is trained on dataset V4 with

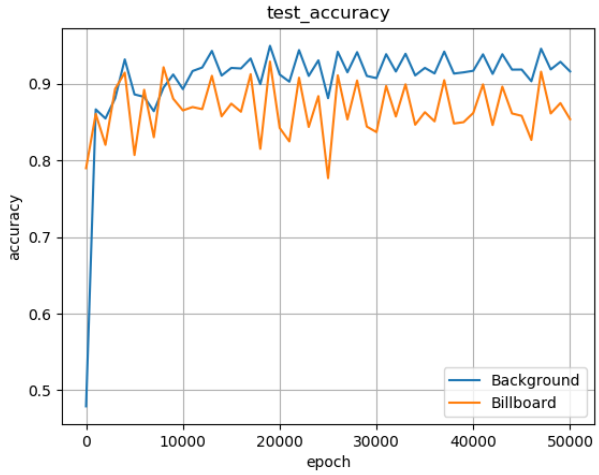

Fig. 20: Plot showing class-wise test accuracy for Joint SPLATNet3D with weighted softmax loss function, and equal sampling on dataset V4. The accuracy of both the classes show similar trend

weighted loss function (weights: billboard = 2; background = 1) and equal sampling. The results in Fig 22 show that using the weighted softmax loss function further improves the accuracy of both the categories in comparison to the softmax loss function. The best performance of MT-PNet is seen with equal sampling and weighted loss function trained on dataset with smaller background. Further experiments on MT-PNet with dataset V3 shows no improvement in accuracy of billboard class.

Results - Instance segmentation: The multi task loss function enables the network to learn instance embeddings for every point in the training set. The instance predictions

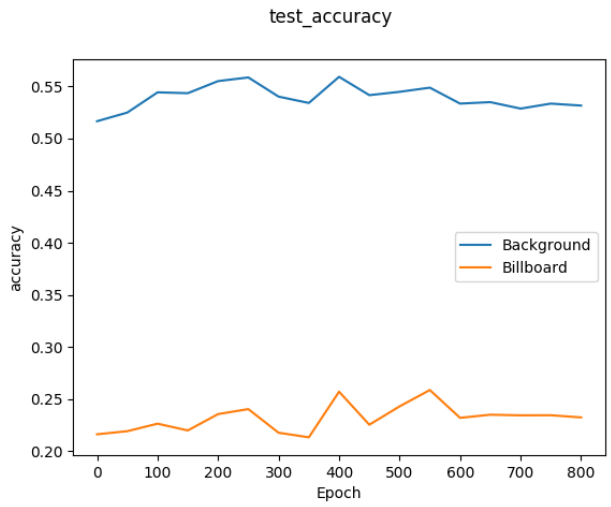

Fig. 21: Plot showing class-wise test accuracy for MT-PNet with equal sampling and softmax loss function on dataset V4. The accuracy of billboard is far lower than that of background class

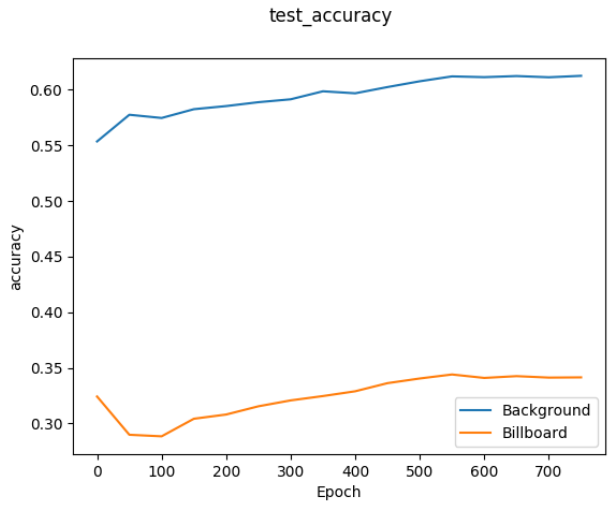

Fig. 22: Plot showing class-wise test accuracy for MT-PNet with equal sampling and weighted softmax loss function on dataset V4. The accuracy of billboard shows improvement but is still lower than that of background class

are determined by applying mean shift algorithm[41] on the embeddings of the trained model. The bandwidth for mean shift algorithm is set to the push force margin(δ_d) as defined in [14]. The choice of bandwidth is crucial and it varies for Joint SPLATNet3D and MT-PNet when trained on billboards dataset. The analysis presented in this section describe the training and evaluation with dataset V4 as it shows promising results for semantic segmentation. The initial setting for bandwidth was 1.5 and the discriminative loss was trained on labels of both the classes. This resulted in $mAP@0.5_{background}$ = 1 and $mAP@0.5_{billboard}$ = nan. This behavior of the network could be attributed to two reasons: 1. the mean shift algorithm clusters all the feature vectors as a single instance which is labelled as background. 2. a very high setting for bandwidth which would have clustered all the features as one instance. A closer inspection of the feature vectors would be more intuitive for further analysis and this is achieved using the t-SNE plot. The t-SNE plot for Joint SPLATNet3D and MT-PNet with points from two classes and one class is shown in Fig 23. It could be inferred from Fig 23 a, c that both the networks show the background points as a single large cluster. This is due to the labelling of the dataset because billboard contains multiple instances and the whole background is considered as a single instance. The t-SNE plot shown in Fig 23 b, d shows multiple groups of points and this is because they are plotted on samples of billboard class only. This gives us two options for further analysis: 1. introduce instances in background category or 2. learn the instances of the billboard category alone. The second option has been chosen for further analysis because the main goal of the work is to segment billboards effectively. Therefore the discriminative loss is trained and evaluated on samples of billboard category for both the networks.

The bandwidth(δ_d) parameter is crucial for training the discriminative loss and evaluating the trained model with mean shift algorithm. The effect of bandwidth on the number of instances and final mAP@0.5 has been analysed for both the networks. The analysis for Joint SPLATNet3D is shown in TABLE II. The table shows that training Joint SPLATNet3D with δ_d values ranging from 15 to 10e-20 gives no unique instances on evaluation. This is not the expected prediction because every train and test data contains multiple instances of billboard. Considering one of the trained models and estimating the bandwidth on the predicted feature vectors gives bandwidth values in the range of 10e-38 to 13e-38. Evaluating the mean shift algorithm with these values gives multiple instances as seen in TABLE II. The variations in number of instances and bandwidth in this range is random and cannot be trusted. It shows that the discriminative loss function aggressively decays the feature vectors of Joint SPLATNet3D and training with bandwidth values as low as 10e-38 would not be useful.

TABLE II: Joint SPLATNet3D - effect of varying bandwidth(δ_d) on the number of instances and final $mAP@0.5_{billboard}$ with billboards dataset V4

Bandwidth(δ_d)	Average no of instances	$mAP@0.5_{billboard}$
15	1	0.67
1.5	1	0.67
0.15	1	0.67
10e-10	1	0.67
10e-20	1	0.67
13e-38	4	0.13
12e-38	6	0.12
10e-38	4	0.14

Similar experiments are conducted with MT-PNet and the results are shown in TABLE III. The table shows that varying bandwidth in the range of 1.5 to 1.0 cluster the output into a single instance. Further investigation on bandwidth values in range of 0.6 to 0.39 does show multiple instances but the mAP@0.5 is very low.

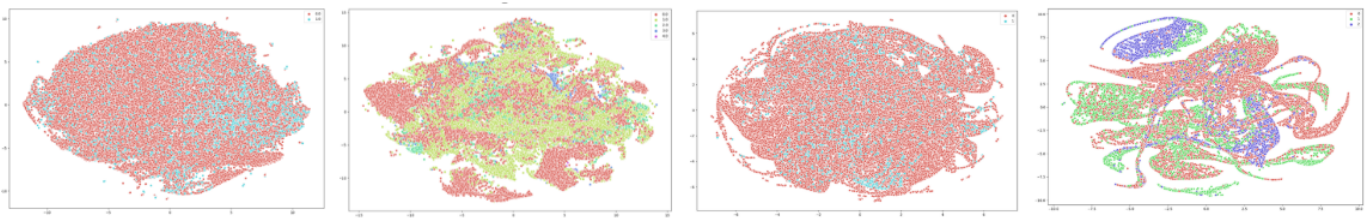

Fig. 23: t-SNE plot of both the networks with labels of both the classes and only billboard. From left to right: Joint SPLATNet3D with two classes and one class, MT-PNet with two classes and one class. t-SNE plot for two classes shows single large cluster and plot for one class shows multiple clusters.

TABLE III: MT-PNet - effect of varying bandwidth(δ_d) on the number of instances and final $mAP@0.5_{billboard}$ with billboards dataset V4

Bandwidth(δ_d)	Average no of instances	$mAP@0.5_{billboard}$
1.5	1	0.17
1.0	1	0.17
0.7	3	0.03
0.6	3	0.09
0.5	3	0.04
0.45	2	0.03
0.39	3	0.02

Results - Overall comparison: The best performing models of Joint SPLATNet3D and MT-PNet are considered and the results are shown in TABLE IV. The table does not include the results of instance segmentation because both the networks do show promising results. It could be inferred from the table that Joint SPLATNet3D shows good improvement over MT-PNet for billboard category and both the network show similar results for background. It is also interesting to compare the IoU for background in TABLE I and TABLE IV. This shows that the IoU for background is lower than that of billboard in TABLE IV. This behavior could be attributed to the nature of sampling used for training both the networks.

TABLE IV: Comparison of joint semantic-instance segmentation with Joint SPLATNet3D and MT-PNet for billboards dataset V4

	Semantic segmentation(IoU)		
Method	Billboard	Background	Overall
Joint SPLATNet3D	**0.73**	**0.62**	**0.68**
MT-PNet	0.34	0.61	0.48

The prediction result for Joint SPLATNet3D is shown in Fig 24. The predictions for semantic segmentation in Fig 24 c shows that many regions of the background are misclassified as billboard. It is also interesting to note that the network misclassifies the regions of background which have similar color features as that of the billboard shown in Fig 24 a. The prediction for instance segmentation is shown in Fig 24 e. The results of billboard category is only considered owing to the nature of training the discriminative loss function. The network tends to cluster both the samples of billboard into a single instance. It is also observed that multiple small clusters are formed at random which can be seen as white and black patches in Fig 24 e.

V. Discussion

Semantic segmentation: The experiments with SPLATNet3D show that softmax loss function is more suitable than weighted softmax loss function for semantic segmentation given that random sampling was used. This observation is made on dataset with smaller background and it might not be the same for dataset with full background. Experiments on MT-PNet with dataset V2 and softmax loss function show small gains in accuracy of billboard category when compared to the background. The network does not show any gains on dataset with full background. This provides further scope to improve the accuracy for billboard category with other training configurations. This has been reflected in the experiments of joint semantic-instance segmentation with MT-PNet.

Joint Semantic-Instance segmentation: The experiments on semantic segmentation with Joint SPLATNet3D and MT-PNet show that, dataset with smaller background gives better results with softmax loss function. This observation is consistent with dataset containing single billboard and multiple billboards both having small background. The results of both the networks show further improvement with equal sampling on dataset with smaller background. This gain with equal sampling comes with the downside that many regions of the background are misclassified as billboard. Both two networks behave distinctly with experiments on equal sampling and weighted loss function. Joint SPLATNet3D with equal sampling shows similar results for softmax and weighted softmax loss functions. MT-PNet with equal sampling gives better results for weighted softmax loss function when compared to softmax loss function. However this observation is consistent only for datasets with smaller background.

The experiments on instance segmentation show that training the discriminative loss function without background samples has proved to be useful. This is attributed to the nature of labelling adopted for dataset generation. The datasets are labelled in a way such that the background is considered as a single instance whereas billboard contains multiple instances. The experiments on tuning the bandwidth(δ_d) for

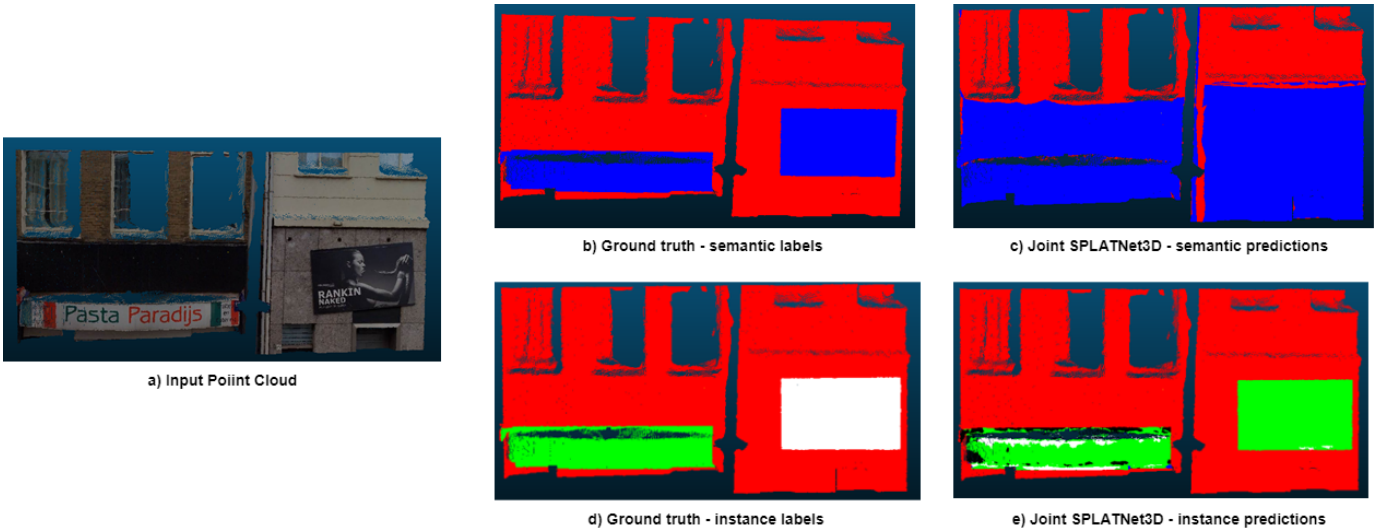

Fig. 24: Illustration of input point cloud, ground truth - semantic labels, semantic prediction, ground truth - instance labels, instance prediction in alphabetical order. The prediction results are with respect to Joint SPLATNet3D.

Joint SPLATNet3D show that the discriminative loss aggressively decays the feature vectors to very small values. The discriminative loss is expected to stop further decays in the feature vectors when the pull and push forces between the clusters are balanced. The decay shows that the loss function couldn't balance the forces and this could be attributed to the nature of feature vectors generated by Joint SPLATNet3D. Therefore the features generated by Joint SPLATNet3D are not very suitable for instance segmentation of billboards with discriminative loss function. Similar experiments on tuning bandwidth for MT-PNet does not does not show aggressive decay in feature vectors. But owing to its poor performance in semantic segmentation it is possible that the network does not effectively represent the features of billboards.

VI. Conclusion and Future scope

Automatic detection of billboards in 3D is a crucial asset for urban planning. This paper proposes Joint SPLATNet3D, a novel approach for semantic-instance segmentation of billboards. The proposed approach is based on SPLATNet3D and multi-task loss function. The network predicts a class label and an instance embedding for every 3D point. This paper describes the process of dataset generation, feasibility study of semantic segmentation and final experiments on joint semantic-instance segmentation for billboards. Preliminary experiments on semantic segmentation show that SPLATNet3D gives an average IoU of 75% in comparison to MT-PNet which gives 46%. Final experiments on joint training show that Joint SPLATNet3D gives an average IoU of 68% in comparison to MT-PNet which gives 48% for semantic segmentation. Training Joint SPLATNet3D with equal sampling improves the IoU for billboard class but reduces the IoU of background class. Experiments on instance segmentation show that training discriminative loss with samples of billboard category alone is useful. However the features generated by Joint SPLATNet3D are not very suitable for instance segmentation of billboards with discriminative loss function. Overall experiments show that Joint SPLATNet3D is more suitable than MT-PNet for semantic segmentation of billboards. The experiments in this paper are limited to one category of billboard with respect to background. This provides scope for further experimentation of semantic segmentation with all the categories of billboards. The observations on instance segmentation with Joint SPLATNet3D is limited to billboards dataset. Therefore Joint SPLATNet3D could be bench-marked with S3DIS dataset[42] which is considered as a standard dataset for instance segmentation.

Acknowledgement

This research project is done with Cyclomedia Technology in association with University of Twente, Netherlands.

References

[1] Li Yin, Qimin cheng, Zhenxin Wang, and Zhenfeng Shao. 'big data' for pedestrian volume: Exploring the use of google street view images for pedestrian counts. *Applied Geography*, 63:337–345, 07 2015.

[2] Degraen D Schoning J Runge N, Samsonov P. No more autobahn: Scenic route generation using googles street view. *In Proceedings of the International Conference on Intelligent User Interfaces*, 2016.

[3] Vladimir A. Krylov, Eamonn Kenny, and Rozenn Dahyot. Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing*, 10, 08 2017.

[4] Charles Ruizhongtai Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas Guibas. Frustum pointnets for 3d object detection from rgb-d data. pages 918–927, 06 2018.

[5] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection, 03 2019.

[6] R. Charles, Hao Su, Mo Kaichun, and Leonidas Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. pages 77–85, 07 2017.

[7] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. 12 2017.

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. pages 6526–6534, 07 2017.

[9] Rudolf Gerrit Verwaal Franciscus Antonius Van Den Heuvel, Bart Johannes Beers. Method and system for producing an image from a vehicle, 06 2011.

[10] Cyclomedia Technology. Billboard detection dataset. https://www.cyclomedia.com/nl/.

[11] D. Erhan C. Szegedy S. Reed C.-Y. Fu W. Liu, D. Anguelov and A. C. Berg. Ssd: Single shot multibox detector. *ECCV*, 2016.

[12] R. B. Girshick. Fast r-cnn. *ICCV*, 2015.

[13] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. pages 2530–2539, 06 2018.

[14] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. pages 8819–8828, 06 2019.

[15] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. *Deep Continuous Fusion for Multi-sensor 3D Object Detection: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, pages 663–678. 09 2018.

[16] Lilei Sun, Junqian Wang, Zhijun Hu, Yong xu, and Zhongwei Cui. Multi-view convolutional neural networks for mammographic image classification. *IEEE Access*, PP:1–1, 09 2019.

[17] Ahmed Nassar, Sébastien Lefèvre, and Jan Wegner. Simultaneous multi-view instance detection with learned geometric soft-constraints. pages 6558–6567, 10 2019.

[18] Yuxing Xie, Jiaojiao Tian, and Xiao Zhu. A review of point cloud semantic segmentation, 01 2020.

[19] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. 06 2016.

[20] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. pages 1513–1518, 09 2017.

[21] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds, 02 2019.

[22] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. 11 2017.

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. 06 2017.

[24] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. pages 10288–10297, 06 2019.

[25] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay Sarma, Michael Bronstein, and Justin Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38, 01 2018.

[26] Gusi Te, Wei Hu, Zongming Guo, and Amin Zheng. Rgcnn: Regularized graph cnn for point cloud segmentation, 06 2018.

[27] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. 06 2018.

[28] Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Real-time progressive 3d semantic segmentation for indoor scenes. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1089–1098, 2018.

[29] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. pages 2569–2578, 06 2018.

[30] Gregory Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. 03 2019.

[31] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *ArXiv*, abs/1708.02551, 2017.

[32] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. 2018.

[33] Varun Jampani, Martin Kiefel, and Peter Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. 06 2016.

[34] J. Baek A. Adams and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, pages 753–762, 2010.

[35] R. Girshick B. Hariharan, P. Arbelaez and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *CVPR*, pages 447–456, 2015.

[36] Padraig Cunningham and Sarah Delany. k-nearest neighbour classifiers. *Mult Classif Syst*, 04 2007.

[37] XiaoBin Li and Weiqiang Wang. Learning discriminative features via weights-biased softmax loss. *Pattern Recognition*, page 107405, 05 2020.

[38] J. Donahue S. Karayev J. Long R. Girshick S. Guadarrama Y. Jia, E. Shelhamer and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *ACM Multimedia*, 2014.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. 12 2019.

[40] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. June 2019.

[41] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis Machine Intelligence (PAMI)*, pages 603–619, 06 2002.

[42] Amir R Zamir Helen Jiang Ioannis Brilakis Martin Fischer Iro Armeni, Ozan Sener and Silvio Savarese. 3d

semantic parsing of large-scale indoor spaces. *CVPR*, pages 1534–1543, 2016.