Project Presentation
3D Object Detection(Waymo)

Jay Patel
Rutvik Patel

Project Progress

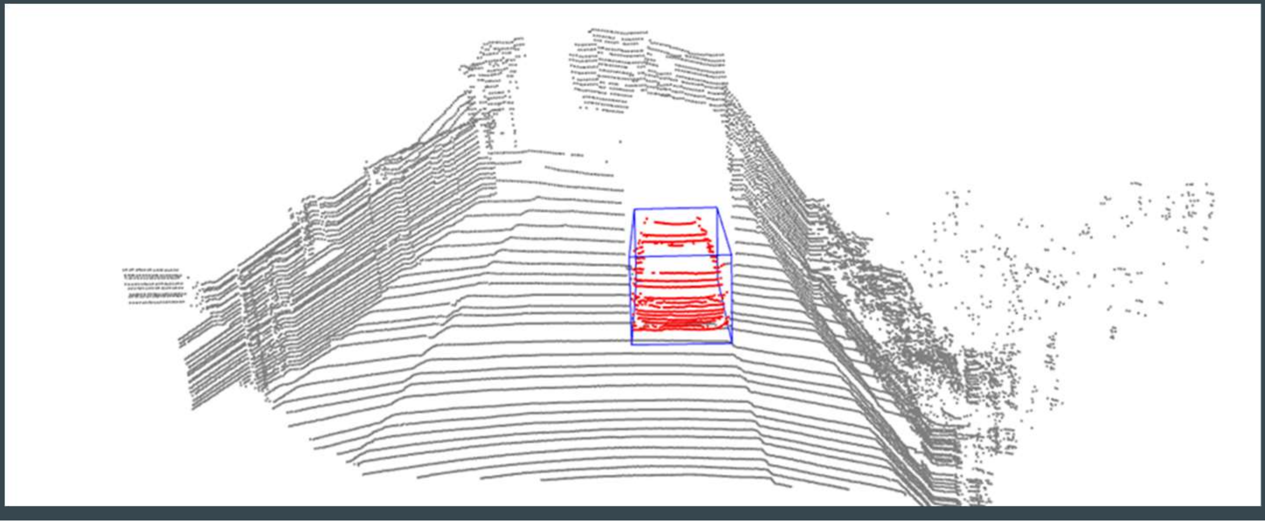
- Trained point pillars(MMDetection3D) on kitti dataset with secfpn backend and 25 epochs.
- Results:

```
Overall AP11@easy, moderate, hard:
bbox AP11:75.7596, 70.9100, 65.3331
bev  AP11:72.2993, 65.6330, 61.2926
3d   AP11:67.1383, 58.3646, 53.9808
aos  AP11:54.19, 50.79, 46.40
```

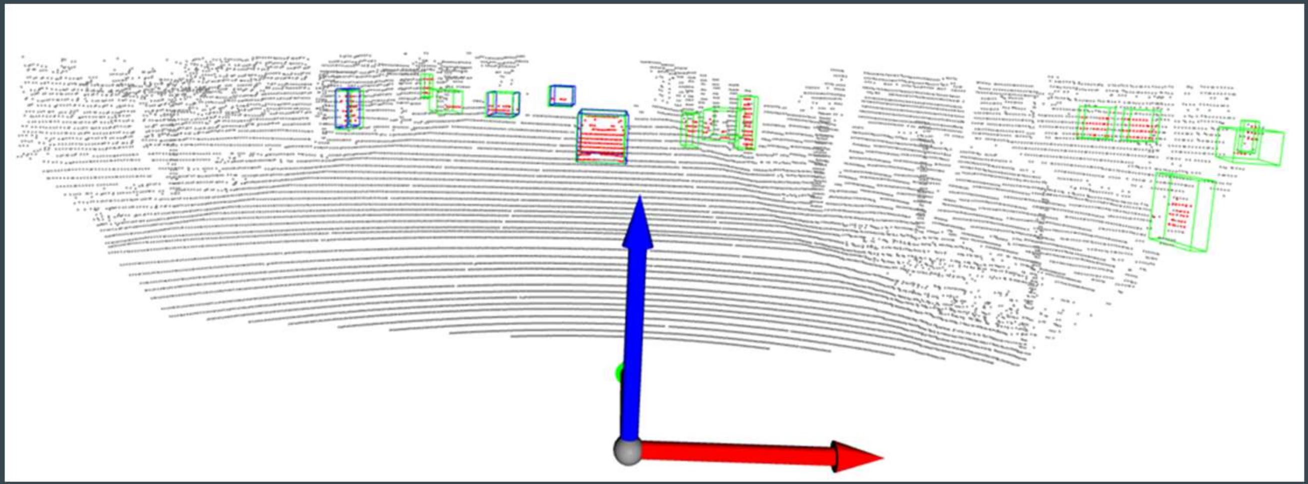
```
Overall AP40@easy, moderate, hard:
bbox AP40:77.5056, 71.1530, 66.6518
bev  AP40:73.2140, 65.6435, 61.1554
3d   AP40:66.8896, 57.7339, 52.6973
aos  AP40:54.69, 49.76, 46.22
```

- Made an attempt to train the model on Waymokitti dataset.
- Getting data folder velodyne_reduced not found error.

Point pillar training on kitti dataset



PointPillar testing on kitti dataset



3D Object Detection

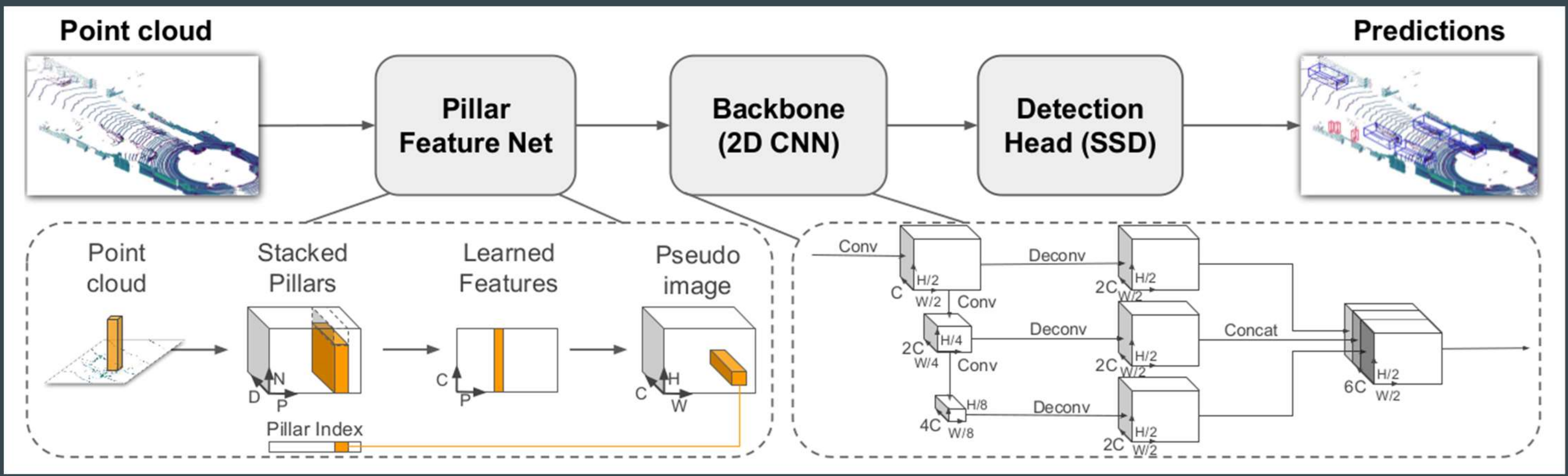
There are two main detection directions for object detection in Lidar information.

1. Project a 3D point cloud to a 2D image (fast speed, high reference resources,low precision)
2. Feature extraction directly using 3D point clouds (high accuracy, low inference speed)

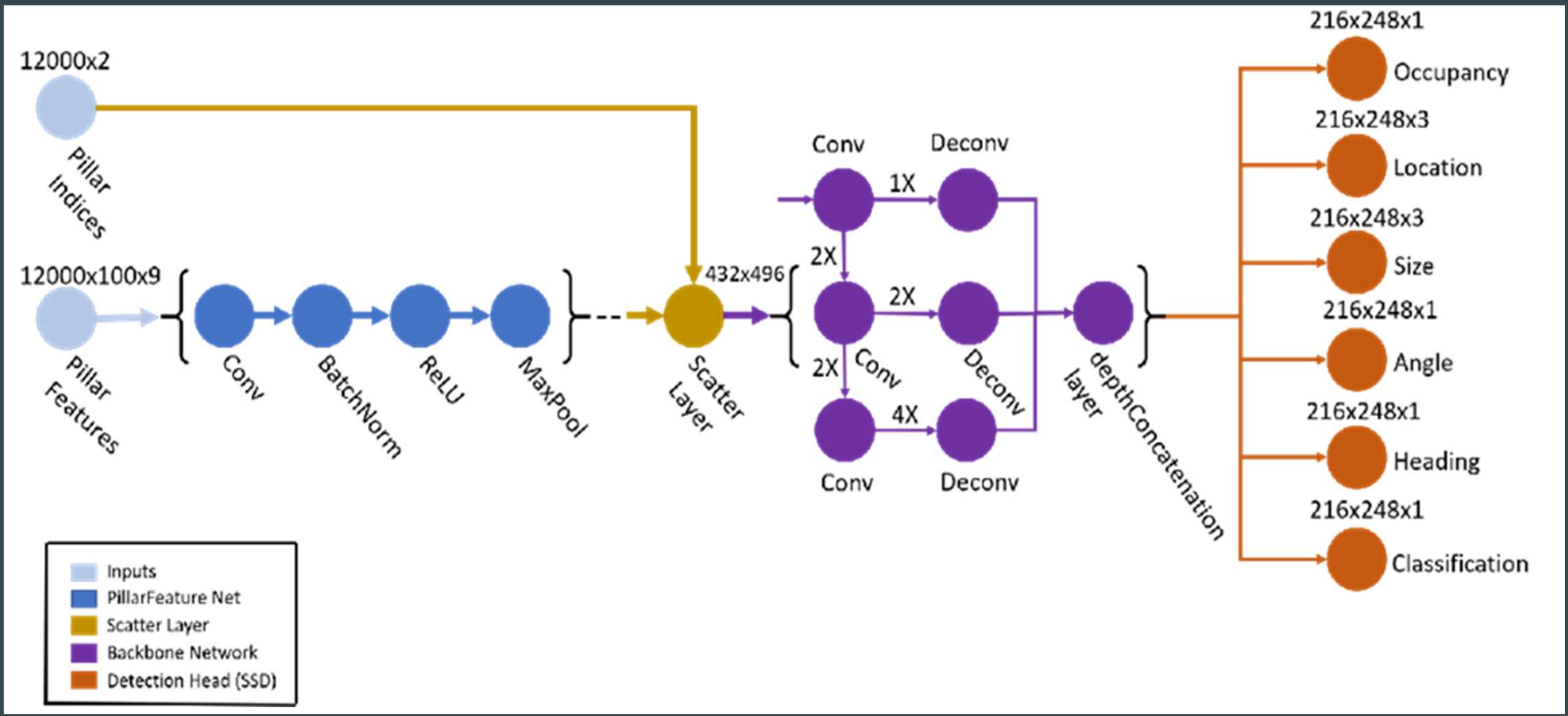
3D Object Detection

- The PointPillars method (abbreviated as PP) is quite excellent.
- The accuracy of the classification of the vehicle category is roughly equivalent to CONTFUSE, but the inference speed can reach 60hz far higher than other networks.
- PointPillars run at 62 fps which is orders of magnitude faster than the previous works in this area.
- Near-real-time object detection of point cloud data is almost possible.

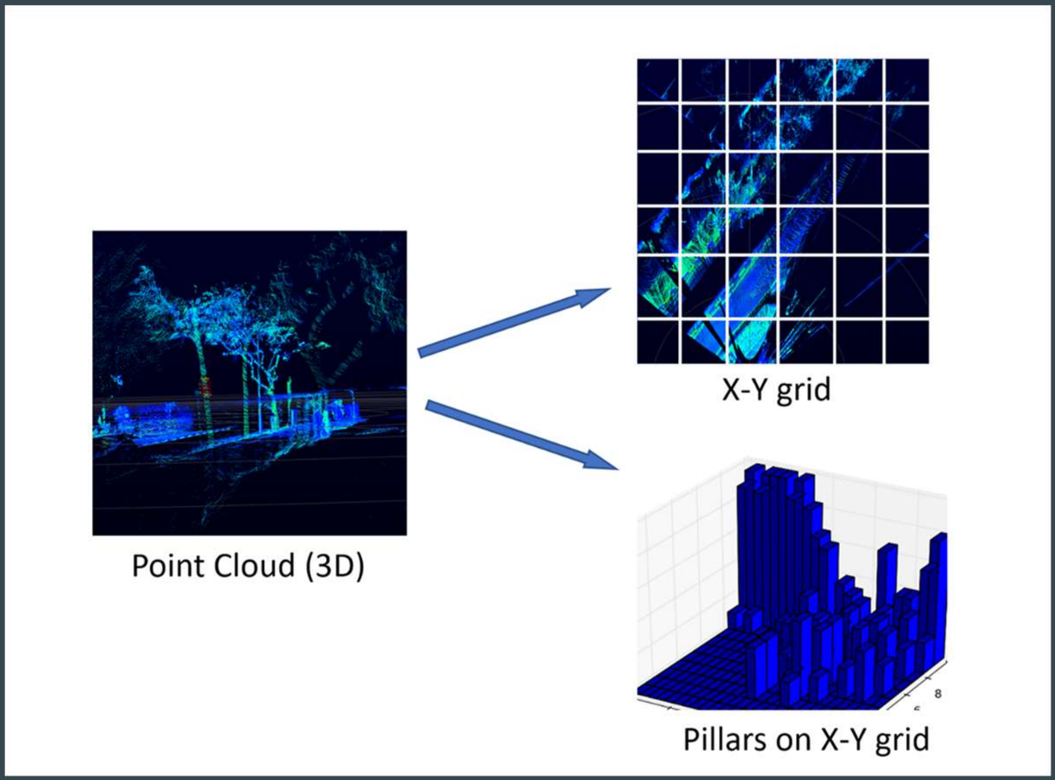
PointPillar Architecture



Point Pillar : layers



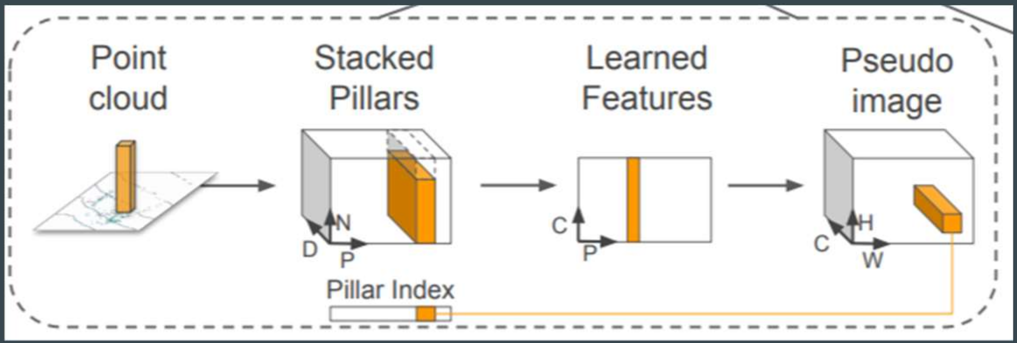
Feature Encoder (Pillar feature net)



- Converts the point cloud into a sparse pseudo image. First, the point cloud is divided into grids in the x-y coordinates, creating a set of pillars. Each point in the cloud, which is a 4-dimensional vector (x,y,z, reflectance), is converted to a 9-dimensional vector containing the additional information explained as follows:

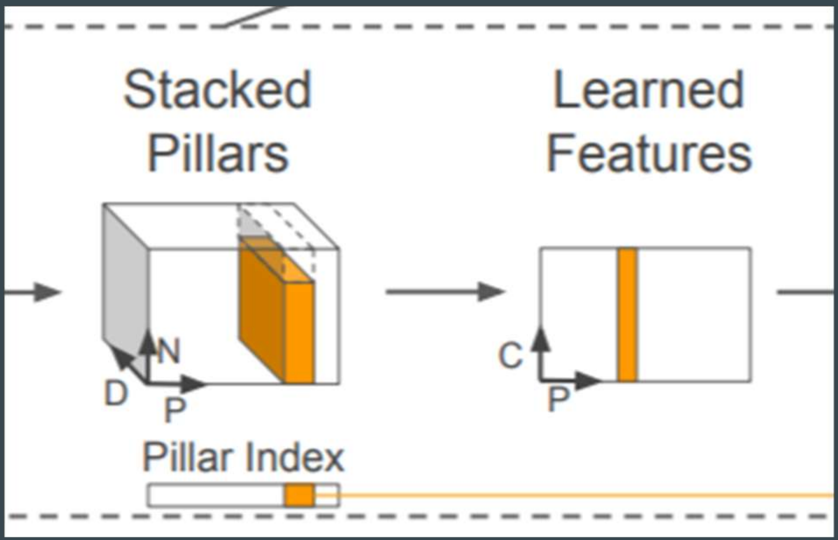
- Xc, Yc, Zc = Distance from the arithmetic mean of the pillar c the point belongs to in each dimension.
- Xp, Yp = Distance of the point from the center of the pillar in the x-y coordinate system.
- Hence, a point now contains the information D = [x,y,z,r,Xc,Yc,Zc,Xp,Yp].

From pillars to a dense tensor(stacked pillars)



- The set of pillars will be mostly empty due to sparsity of the point cloud, and the non-empty pillars will in general have few points in them. This sparsity is exploited by imposing a limit both on the number of non-empty pillars per sample (P) and on the number of points per pillar (N) to create a dense tensor of size (D, P, N). If a sample or pillar holds too much data to fit in this tensor the data is randomly sampled. Conversely, if a sample or pillar has too little data to populate the tensor, zero padding is applied.
- Note that D = [x,y,z,r,Xc,Yc,Zc,Xp,Yp] as explained in the previous section.

From Stacked Pillars to Learned Features

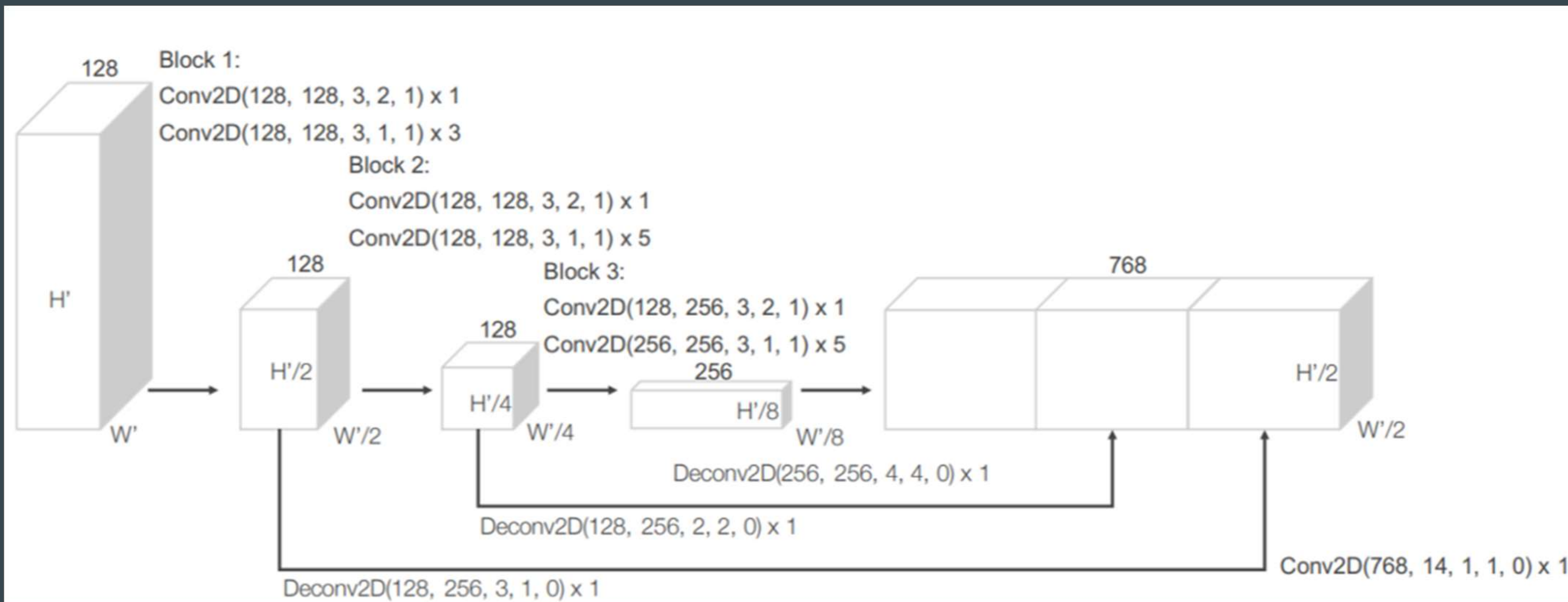


- We use PointNet to extract features from a point cloud type of data.
- PointNet basically applies to each point, a linear layer followed by BatchNorm and ReLU to generate high-level features, which in this case is of dimension (C,P,N). This is followed by a max pool operation which converts this (C,P,N) dimensional tensor to a (C,P) dimensional tensor.

Generating the Pseudo Image from Learned features

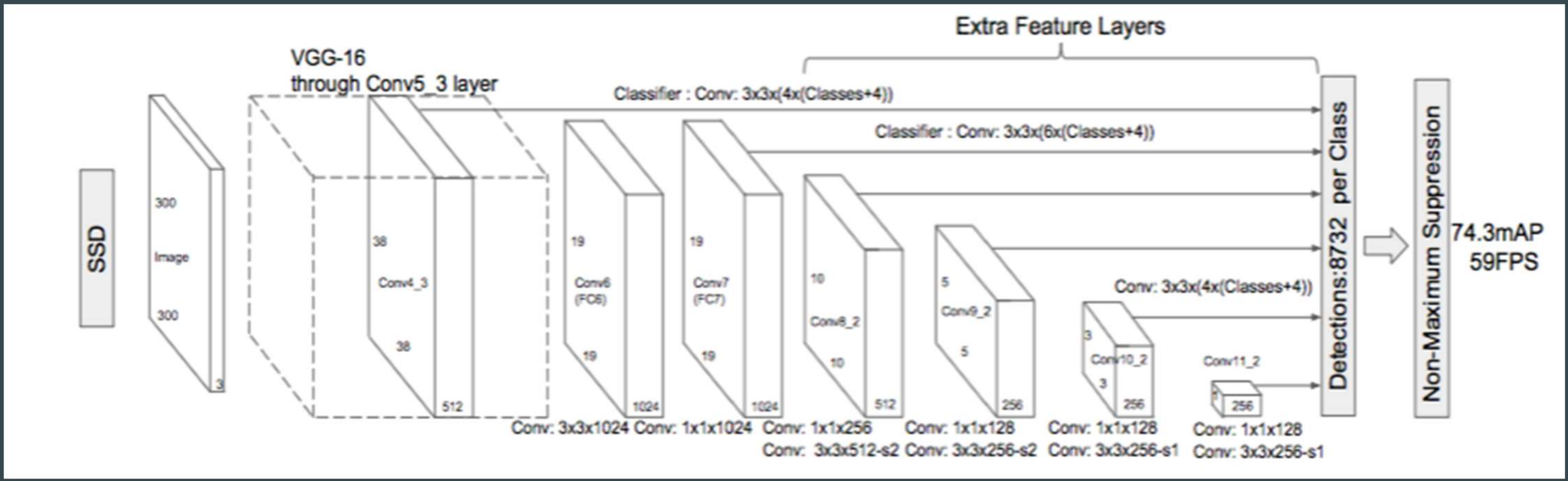
- The generated (C, P) tensor is transformed back to its original pillar using the Pillar index for each point.
- So originally, where the point was converted to a D dimensional vector, now it contains a C dimensional vector, which are the features obtained from a PointNet.

Backbone



An example of a backbone (RPN) Region Proposal Network used in Point Pillars.

Detection Head (SSD)



The objective of the SSD network is to generate 2D bounding boxes on the features generated from the backbone layer of the Point Pillars network.

Detection Head(SSD) contd.

Several important reasons for choosing SSD as a one-shot bounding box detection algorithm are:

- Fast inference.
- Uses features from well-studied networks like VGG.
- Great Accuracy.

References

1. Lang, Alex H., et al. "Pointpillars: Fast Encoders for Object Detection from Point Clouds." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, https://doi.org/10.1109/cvpr.2019.01298.