Warning: Do not delete slides.

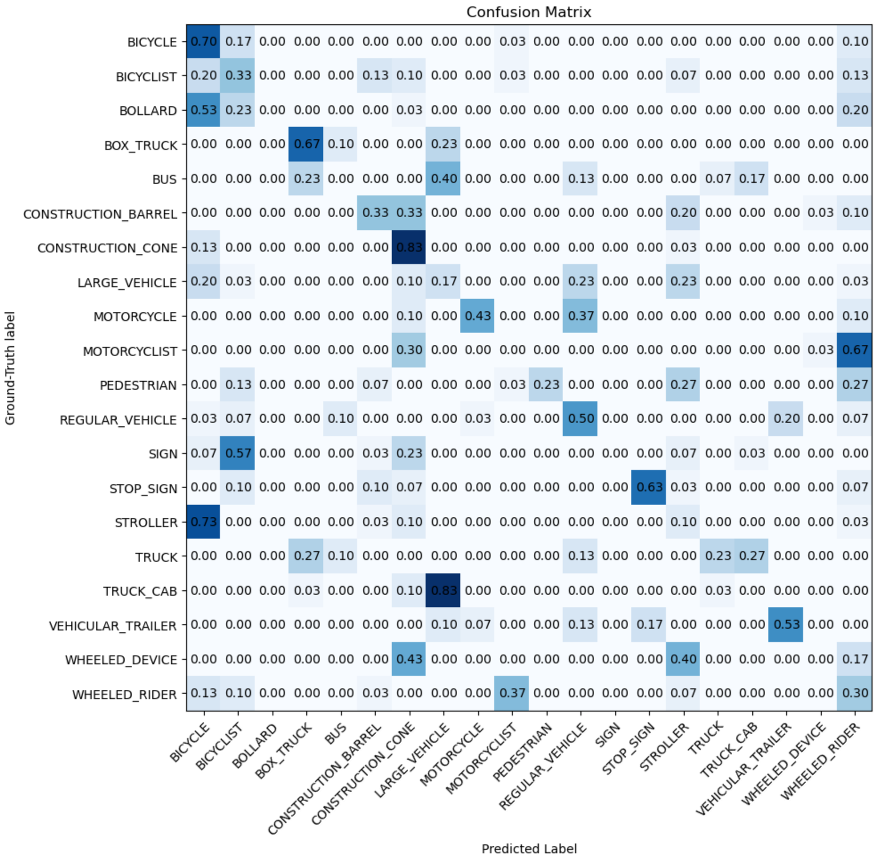
This includes extra credit slides and any problems you do not complete. All problems, including extra credit, must be assigned to a slide on Gradescope.

Failure to follow this will result in a penalty

CS 6476 Project 5

Auryn Yamamura
ayamamura6@gatech.edu
ayamamura6
904154249

Part 2: Voxel Baseline Model



Part 2: Voxel Baseline Model

[What are the top confusion pairs in the matrix and why might this be the case?]

’Truck Cab’ labeled as ‘Large Vehicle’ and ‘Stroller’ labeled as ‘Bicycle’. VoxelNet is essentially just downsampling and then discretizing our point clouds, so the loss of information can cause classes that are already similar to be classified as each other.

[To which classes would adding finer geometry help the most and why?]

It will help most for classes that are small: for instances of those classes, our points could possibly reside in all in one bin – so if we have more than one ”small” class this poses a major issue. Thus the need for finer geometry. Additionally, finer geometry will help in distinguishing between classes that are similar (for instance, human-based classes, like ‘Pedestrian’, ‘Motorcyclist’, ‘Bicyclist’, etc).

Part 2: Voxel Baseline Model

[Which classes does this baseline perform well for? Why?]

'Construction Cone' and 'Bicycle': these two have very distinct shapes so the pattern of cells they occupy are also probably quite distinct compared to patterns from other classes.

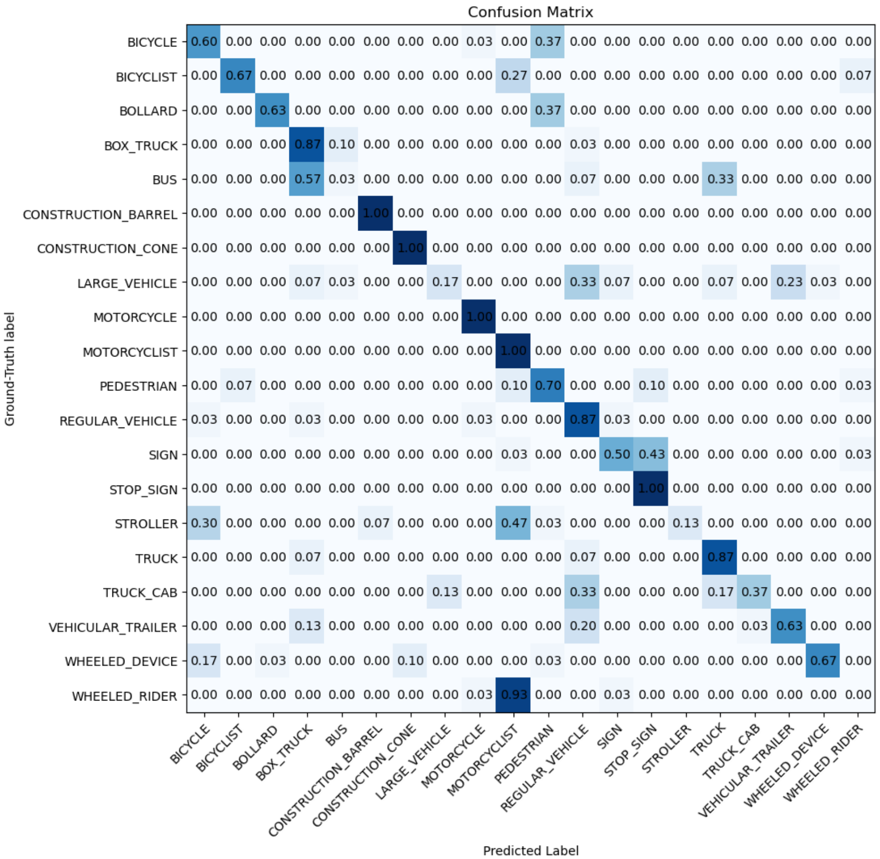
[What are some ways we can improve this voxel based model?]

We could also try classifying using various numbers of cells and have our final decision be an aggregate of all those class outputs. (Multi-Scale)

[Which mode (count or occupancy) achieves higher accuracy? Why do you think this is the case?]

Occupancy surprisingly achieves the higher accuracy. For all classes, the number of points per point cloud tend to vary, making count an unreliable indictor of class.

Part 3: Simplified PointNet



Part 3: Simplified PointNet

[Which classes does this model perform well for? Why?]

This model performs extremely well for ‘Construction Cone’, ‘Construction Barrel’, ‘Motorcycle’, and ‘Stop Sign’. I suspect this is because these classes tend to have very distinct shape or ’geometry’, so the model is more easily able to identify what they are since there’s not as much variation.

[Which classes are most misclassified? What were they classified as and why do you think this is the case?]

The most misclassified are ‘Wheeled Rider’ and ‘Bus’ which were mostly misclassified as ‘Motorcyclist’ and ‘Box Truck’ respectively. You’ll notice that these pairs tend to be quite similar in geometry (both buses and box trucks are ”box-like” while “Wheeled Rider” and “Motorcyclist” are both instances of humans).

It’s also possible that the ’Motorcyclist’/‘Box Truck’ had more consistent shapes/patterns throughout their samples whereas ‘Wheeled Rider’/’Bus’ might have varied more, causing the network to be less confident in choosing ‘Wheeled Rider’/’Bus’ over ‘Motorcyclist’/’Box Truck.’

Part 3: Simplified PointNet

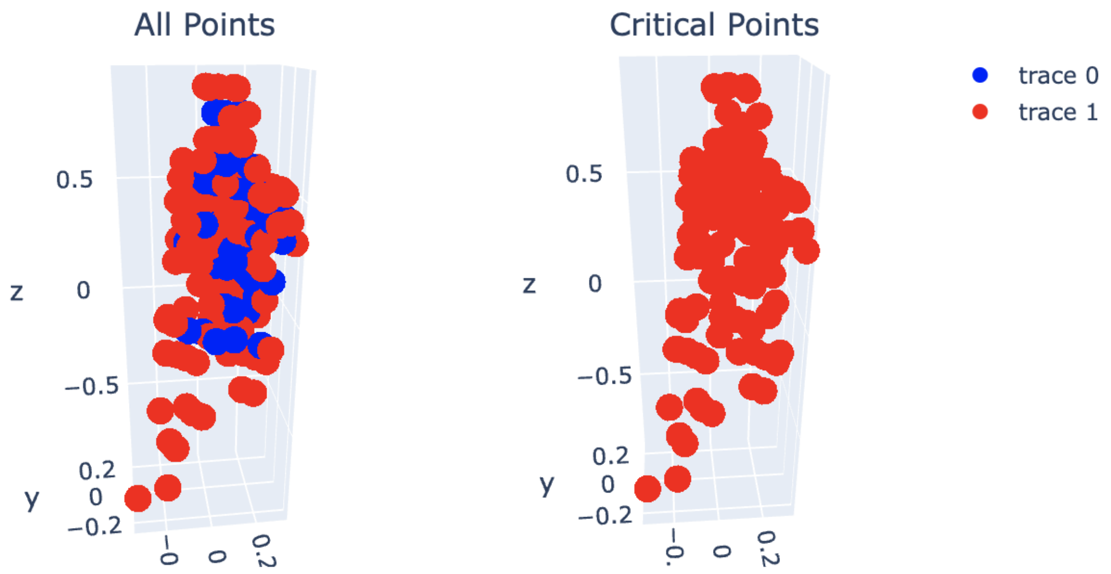
[What changes in the architecture or training process can be done to improve performance?]

We could try adding data augmentations to our training data (i.e. random points removed/added) to get the network to learn more general patterns. We could also try to implement a more hierarchical architecture like in PointNet++: we would partition our point clouds into neighborhoods (similar to VoxelNet) and get embeddings for each neighborhood using our naïve PointNet. We would then repeat this at multiple scales and aggregate all our features into one global feature via pooling.

[Did the top confusion pair in the matrix remain the same as in Part 2? Why or why not?]

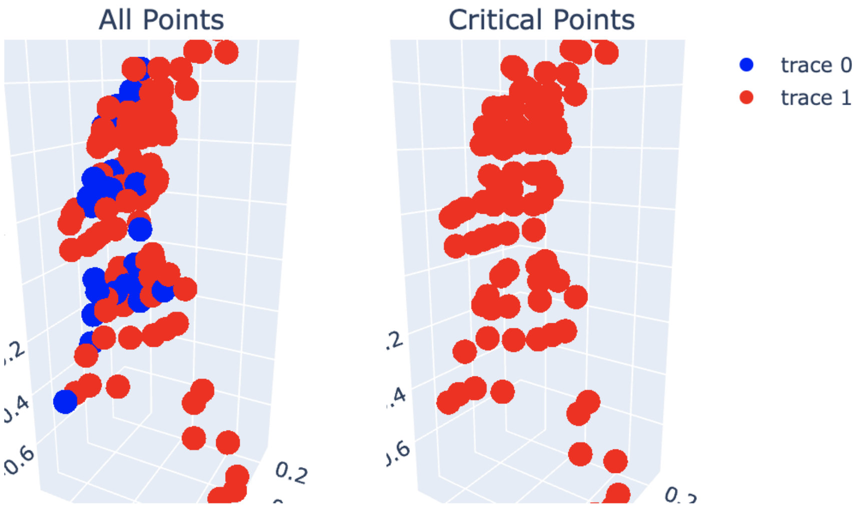
The top confusion pair changed from 'Truck Cab' as 'Large Vehicle' to 'Wheeled Rider' as 'Motorcyclist'. This change likely occurred as our PointNet learns about the finer geometric details of each class, so it probably used information about those details to distinguish instances of those classes apart.

Part 4: Analysis



‘PEDESTRIAN/11.txt’

Part 4: Analysis



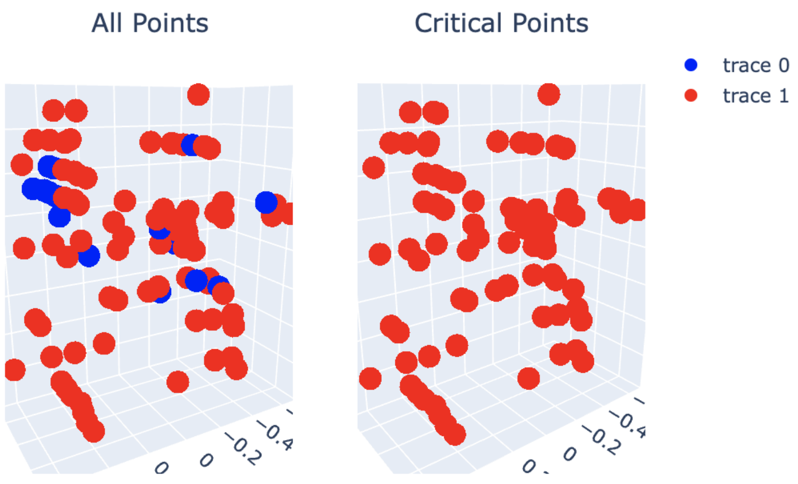
‘PEDESTRIAN/55.txt’

[Why would it make sense for these to be selected as critical points?]

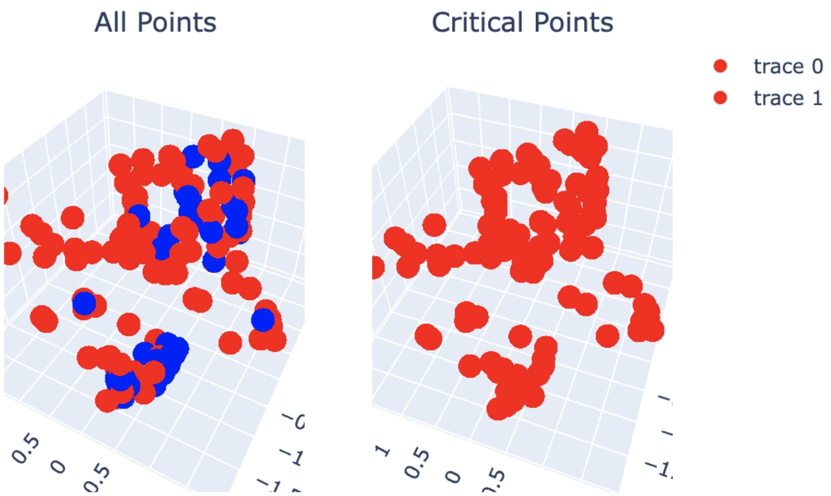
The points selected as critical points are the ones that contribute information to the global feature. From observation, it also appears that these points also contribute information about the shape/form of the object. One could argue that critical points here serve to “trace” out the human form, while the non-critical only seem to fill out that outline.

We could also argue that the non-critical points provide the same kind of information about the object as the critical points, but to lesser degree.

Part 4: Analysis

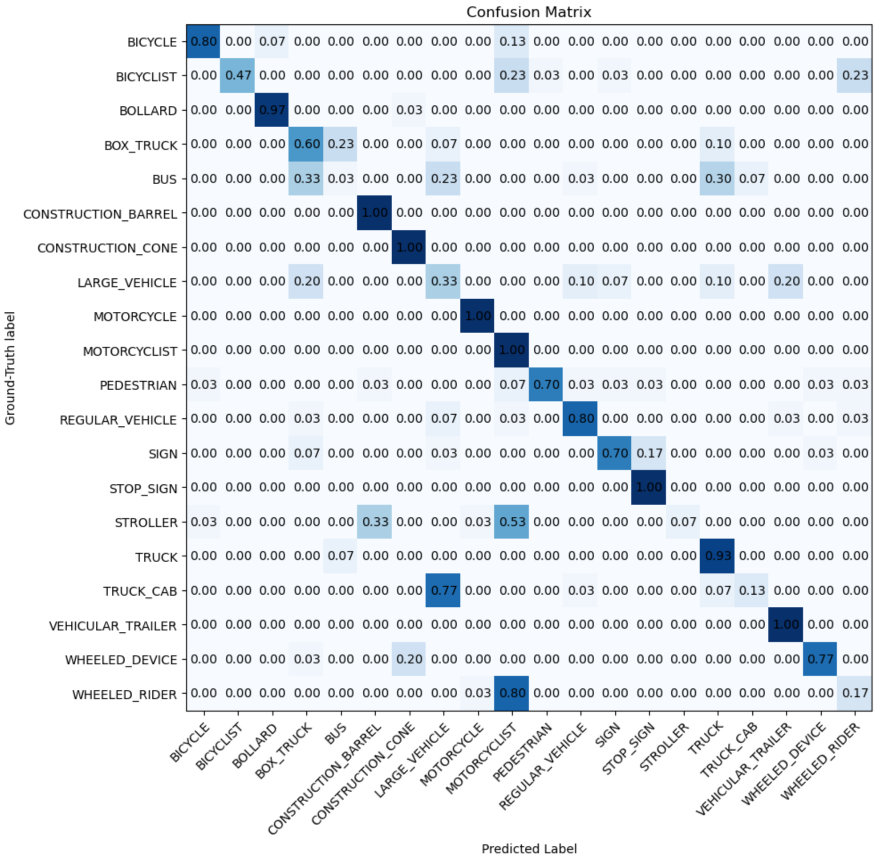


WHEELED_RIDER/175.txt



BUS/185.txt

Part 5: PointNet + T-Net (Extra Credit)



Part 5: PointNet + T-Net (Extra Credit)

[What is the motivation behind using T-Net in PointNet?]

We want the global feature representing our point cloud to be invariant under scalings/rotations/translations, which is why we implement a T-Net to learn a “canonical pose” (orientation and positioning) and a transform matrix to transform our point clouds to said “canonical pose.”

[Which classes saw the most improvement from including T-Net? Why do you think this is the case?]

‘Vehicular Trailer’ (+37%) and ‘Bollard’ saw the most improvement (+34%). Arguably classifying these objects heavily depends on its point cloud layout in the bounding volume. Without having a canonical “pose”, then it is understandable how our network conflated instances of these classes with other classes that are similarly shaped (i.e. ‘Bollard’/’Pedestrian’ and ‘Vehicular Trailer’/’Regular Vehicle’).

Additional Extra Credit (Bells & Whistles)

[describe any additional extra credit you implemented]