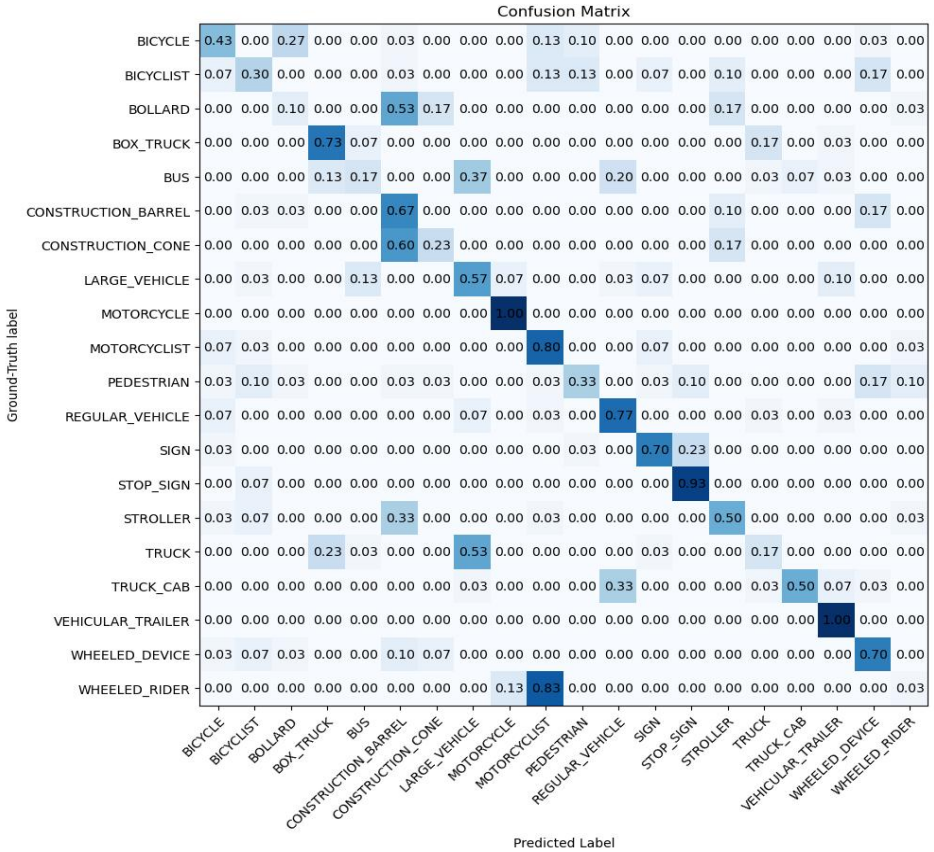
Warning: Do not delete slides.
This includes extra credit slides and any problems you do not complete. All problems, including extra credit, must be assigned to a slide on Gradescope.
Failure to follow this will result in a penalty

CS 6476 Project 5

Chuyu Xu
cxu433@gatech.edu
cxu433
904019969

Part 2: Voxel Baseline Model



et]

Part 2: Voxel Baseline Model

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

WHEELED_RIDER(True) -> MOTORCYCLIST (Predicted) (0.83)

TRUCK(True) -> LARGE_VEHICLE (Predicted) (0.53)

The model is very confused between these two. This is likely because their geometric shapes are almost identical. The labels are also semantically very similar.

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

Adding finer geometry would help the most for classes that are confused due to simple shapes looking alike. The best are the classes confused with BOLLARD, such as: STROLLER, CONSTRUCTION_CONE

Currently, the model likely confuses these because they all look like simple "upright poles" in a low-resolution point cloud. With finer geometry, the model could see the crucial details that make them different.

Part 2: Voxel Baseline Model

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

VEHICULAR_TRAILER(1.00), MOTORCYCLIST (0.87)

These classes have distinct, large, and consistent 3D shapes that are well-captured by a coarse voxel grid.

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

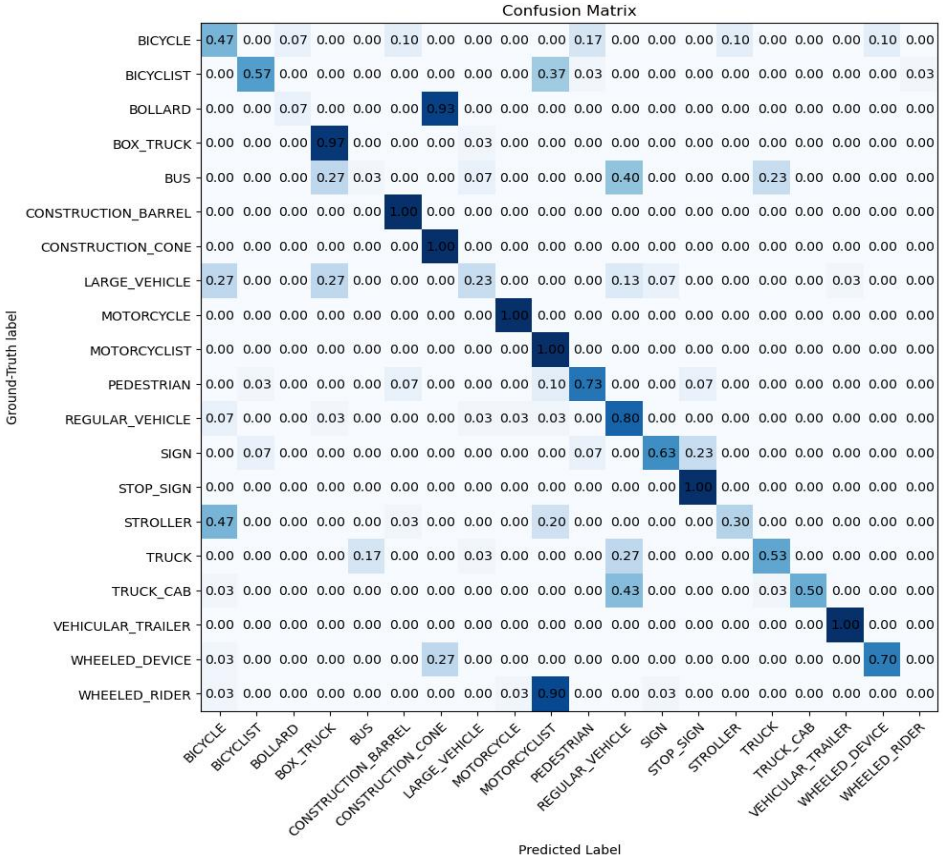
Improving the manual features, like using higher resolution, or multi-resolution. Then, use a deeper classifier or VFE to learn the features.

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

Occupancy mode likely achieves higher accuracy.

The "count" mode is highly sensitive to the density of the point cloud. A dense object and a sparse object will have vastly different feature values, even if their shape is similar. The "occupancy" mode ignores density and acts as a normalizer. It only captures the pure 3D shape. This binary shape is often a more robust and generalizable feature for the classifier to learn from.

Part 3: Simplified PointNet



Part 3: Simplified PointNet

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

MOTORCYCLE(1.00), MOTORCYCLIST(1.00)
STOP_SIGN(1.00), VEHICULAR_TRAILER(1.00)

This is because these objects have very unique and distinct geometric shapes, like a cone, a cylinder, a person on a bike. PointNet is very good at learning these unambiguous 3D signatures.

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

BUS, WHEELED_RIDER, BOLLARD

From a sparse lidar scanner, a bus, a truck, and a regular vehicle are all just very large, tall, "boxy" shapes.

The model is not advanced enough, especially without a T-Net to fix rotation, to tell the subtle differences between them. A bus seen from the side looks almost identical to a large truck from the side.

Part 3: Simplified PointNet

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

Implement the T-Net. It would help the model correctly classify objects that are rotated, which is likely the cause of much of the confusion.

Add Data Augmentation. Randomly rotating, scaling, and adding jitter to the training point clouds would force the model to learn the true shape of an object, not just the view of it.

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

Yes.

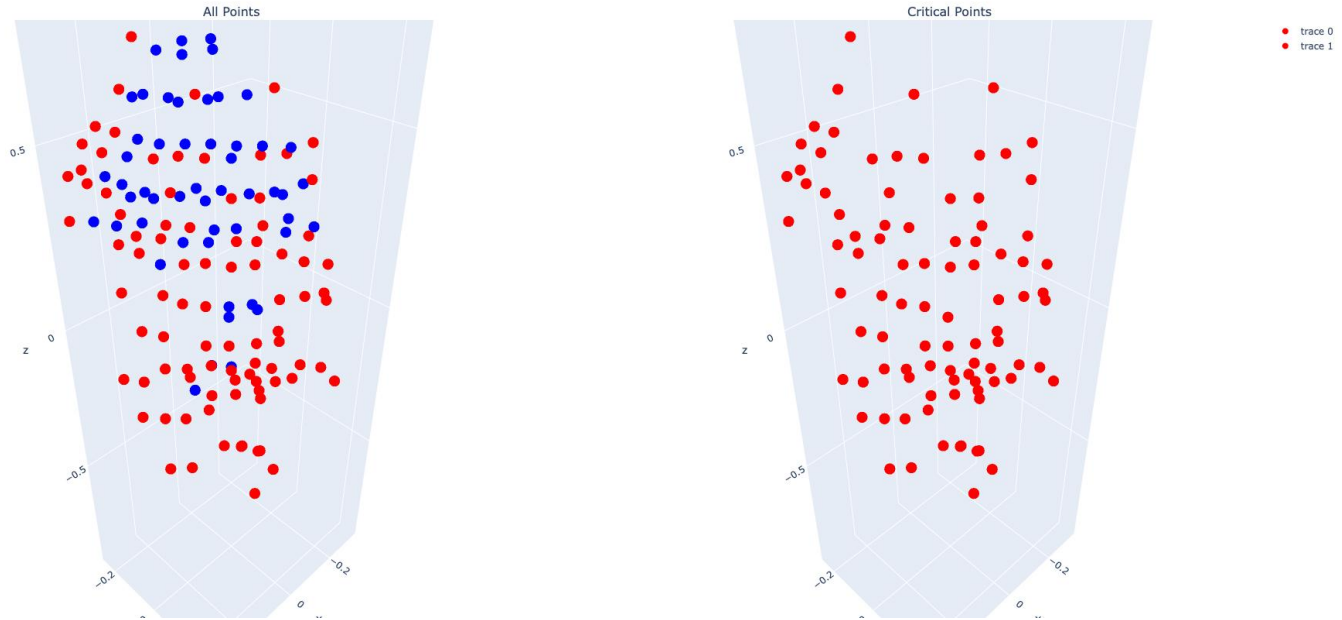
In the Voxel Baseline, the worst confusion was WHEELED_RIDER-> MOTORCYCLIST (0.83).

In this PointNet, the worst confusion is still WHEELED_RIDER-> MOTORCYCLIST(0.90).

Because both classes share an almost identical geometric shape: a person on a wheeled machine. PointNet's core design relies on Global Max Pooling, which compresses all points into a single feature vector that only describes the overall 3D shape. Because the overall shapes of a MOTORCYCLIST and a WHEELED_RIDER are so similar, their global feature vectors are also nearly identical. The model learns the general concept of "person on wheels" but fails to capture the subtle geometric details.

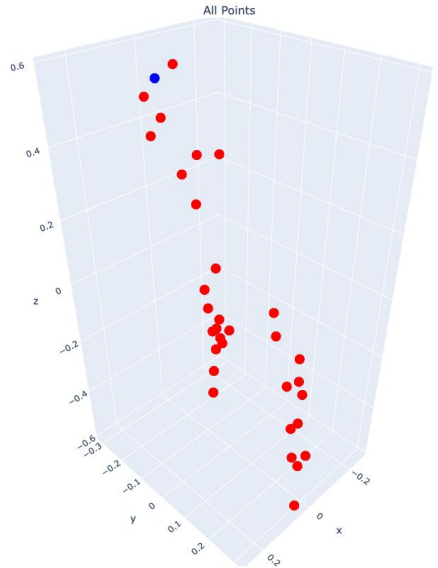
Part 4: Analysis

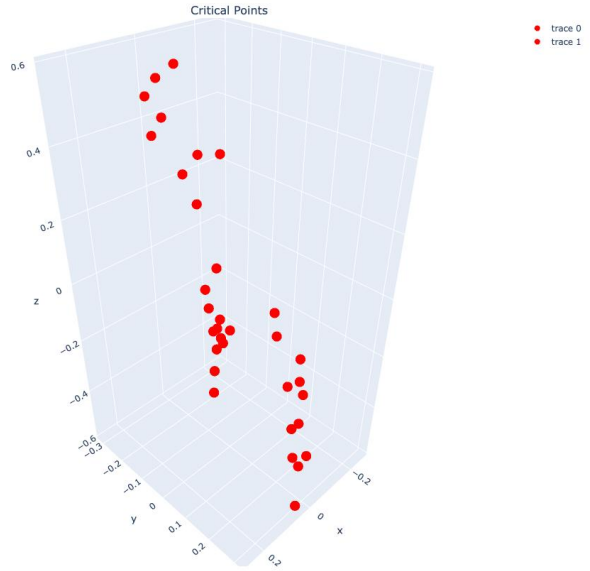
[insert visualization of critical points on the PEDESTRIAN/11.txt point cloud]



Part 4: Analysis

[insert visualization critical points on one other point cloud, why would it make sense for these to be selected as critical points?]





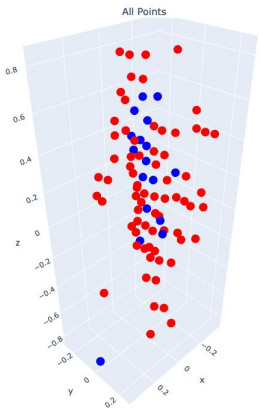
The model selects these points because they are the most geometrically important points that best summarize the object's unique 3D shape.

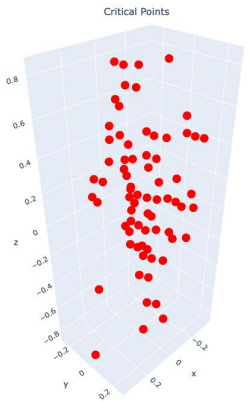
Therefore, it makes sense that the critical points are the tips, corners, and edges that lie on the oudside and form a skeleton of the pedestrain's unique 3D structure.

Part 4: Analysis

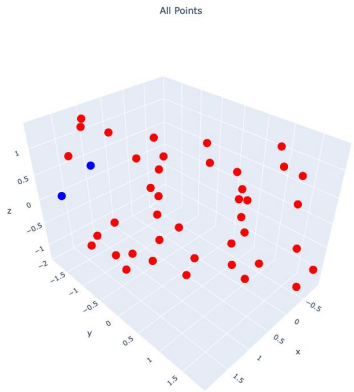
[Plot critical points for two more classes which are misclassified]

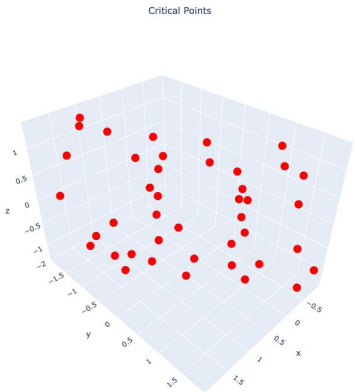
WHEELED_RIDER:



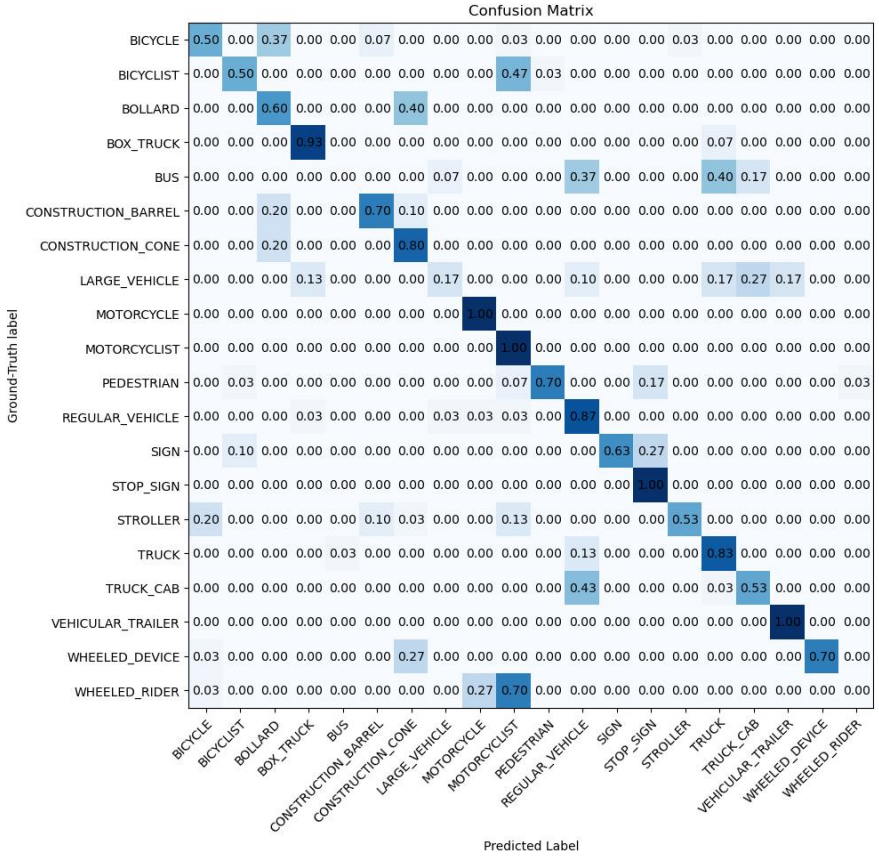


BUS:





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



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

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

The motivation is to make the PointNet model robust to transformations, especially rotations.

The basic PointNet model is sensitive if the input point cloud is rotated. The T-Net is a small, separate network that predicts the input's transformation. It calculates a 3 x 3 matrix that is then used to rotate the point cloud back to a standard orientation before it gets classified. This helps the main PointNet recognize the object, no matter how it is rotated.

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

BOLLARD (0.07->0.60)
A BOLLARD (a cylinder) and a CONSTRUCTION_CONE (a cone) are both upright and symmetrical shapes. The base model couldn't tell their profiles apart, especially if they were slightly tilted. The T-Net learns to straighten up all these simple objects into a canonical pose. Once aligned, the difference between a cylinder and a cone became perfectly clear to the classifier, and the confusion dropped to 0%.

TRUCK(0.53->0.83)
A truck from the side looks almost identical to a bus. The model was confused by these different viewing angles. The T-Net learned to rotate all these large, boxy vehicles into a standard orientation, e.g. all facing forward. Once they were all aligned, the main PointNet could easily see the subtle geometric differences and stopped confusing them.

Additional Extra Credit (Bells & Whistles)

[describe any additional extra credit you implemented]