Optimizing 3D Object Classification with Deep Learning on Geometry Representations

Ruiqi Wang
rqwang@stanford.edu

Introduction

Accurately classifying 3D data poses a significant challenge due to the high dimensionality and complexity of 3D data, which makes it difficult to extract meaningful features and identify discriminative patterns.

This project aims to investigate the effectiveness of different 3D data representation schemes and machine learning methods for classifying 3D objects. Specifically,

- Which classification method and 3D representation scheme combination delivers the best classification performance?
- Which classification method and 3D representation scheme combination is the most efficient method given the performance-cost trade-off?

Data & Geometry Representations

We use the 3D MNIST dataset [1] for both training and testing. The dataset contains 39,206 meshes for the 10 digits. We use a train/validation/test split of 60/20/20.

To test the robustness of the models, we generate two additional datasets, one with the meshes uniformly randomly rotating around the upright axis, the other with the meshes rotating with uniformly random 3D rotation matrices.

Single-View Images We use single-view images as the baseline geometry representation. For each 3D mesh, we generate a 56x56 grayscale image with the camera facing towards the -y axis.

Multi-View Images We generate 56x56 grayscale images with the camera facing towards $\pm$ x, $\pm$ y, and $\pm$ z.

Voxel Grids We generate voxels of size 0.02 from a mesh and pad the resulting voxel grid to 36x36x36.

Point Clouds We sample 2500 points uniformly on the surface of a mesh, and then generate 500 Poisson disk[2] samples as the point cloud based on the uniformly sampled points.

Figure 1: Geometry Representations (from left to right: mesh, point cloud, voxel grid, multi-view images)

Baselines

K-Nearest Neighbors (KNN)

For image and voxel representations, we first reduce the dimension of the input vectors to 9 using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), then fit a KNN model (K=3) on the reduced feature space. For point clouds, because the data structure is unordered, we fit a KNN model with the Hausdorff distance.

Support-Vector Machine (SVM)

For voxel grids with random rotations, low-dimensional features fail to capture the distinctions between various classes. Therefore, we employ the support vector machine (SVM) with the radial basis function (RBF) kernel as an alternative baseline. We train the SVM on PCA features with 100 dimensions.

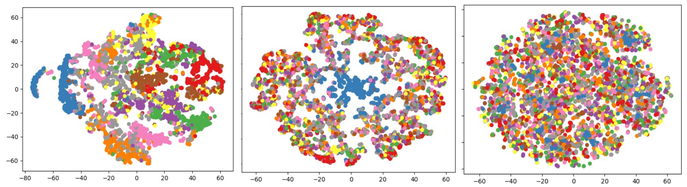


Figure 2: T-SNE plot of PCA features of voxel grids (from left to right: without rotation, rotation on 1 axis, free 3D rotation)

Network Architectures

For single-view images, the neural network has one 2D convolutional layer (with kernel size 3), followed by two fully-connected (FC) layers.

For multi-view images, we pass each view through a 2D CNN, then max-pool across different views and pass the result to FC layers.

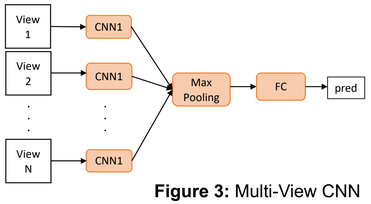


Figure 3: Multi-View CNN

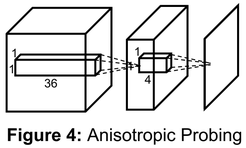


Figure 4: Anisotropic Probing

For voxel grids, we use two different architectures. One is similar to the single-view network, with the 2D convolutional kernels replaced by 3D kernels. The other is similar to the multi-view network, with the 2D kernels replaced by Anisotropic Probing[3] kernels.

For point clouds, we apply the input and feature transform design in PointNet [4].

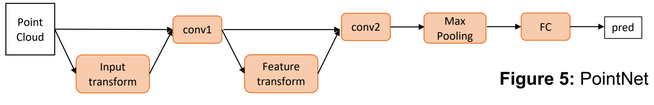


Figure 5: PointNet

Results

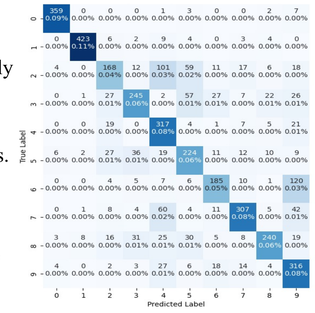
The table below shows the classification accuracy achieved by the models on the test sets. The three accuracy columns correspond to the original dataset, the dataset with random rotation around a single axis, and the dataset with random 3D rotations.

Model	Data	Accuracy%$^{(0)}$	Accuracy%$^{(1)}$	Accuracy%$^{(2)}$	# params
KNN + PCA	Single-View	88.12	68.74	42.30	
KNN + LDA		88.45	71.21	34.78	
CNN		**97.50**	88.55	63.46	75,306
MVCNN	Multi-View (4)	96.40	89.37	83.25	103,018
MVCNN	Multi-View (6)	96.58	93.88	**84.52**	103,018
KNN + PCA	Voxel Grid	80.29	31.16		
SVM + PCA		95.39	81.46	55.94	
VoxelCNN		**97.40**	91.66	65.17	132,938
VoxelCNN (aniso prob)		96.25	91.36	55.41	84,535
VoxelCNN (aniso, multi)		96.86	92.48	65.25	85,360
KNN (Hausdorff)	Point Cloud	93.42			
Point Net (vanilla)		94.62	90.18	64.23	83,978
Point Net (wo feature trans)		96.20	92.68	70.35	167,827
Point Net (w feature trans)		**97.12**	**95.13**	73.92	383,539

On the original dataset, all geometry representations yield satisfactory results. However, the multi-view model has a slightly lower accuracy. This can be attributed to the fact that the front views already capture 2D digit information, while the other views introduce similarities, such as the left and right views all resembling "1" and introducing noise in the classification process.

On the dataset with rotations around a single axis, the models demonstrate consistently high accuracy, and we see performance gains from multi-view images compared to single-view images.

For the dataset with 3D rotations, multi-view images yields the most favorable results. As shown in the confusion matrix, the Point Net model struggles to differentiate between 2 and 5, 4 and 7, 6 and 9, as they look similarly after rotation. Notably, the models on voxel grids exhibit the lowest accuracy. One plausible explanation is that other representations provide only surface information, while it is challenging for the models to directly extract surface-related details from voxel grids.



Future Works

In our future work, we intend to enhance the performance of our neural networks by conducting experiments involving varying model architectures (e.g. numbers of layers, layer sizes, and normalization methods) and the resolutions of the geometry representations. In addition, we plan to investigate whether the disparities in performance stem from architectural designs or the mere augmentation of parameter count. Moreover, we seek to extend our experiments to larger datasets, such as the ShapeNet dataset, to assess the generalizability of our conclusions across diverse object classes and more complex 3D data.

References

1. Hsueh-Ti D. Liu. Mesh MNIST. https://www.dgp.toronto.edu/~hsuehtil/.
2. Cem Yuksel. 2015. Sample Elimination for Generating Poisson Disk Sample Sets. Computer Graphics Forum 34, 2 (May 2015), 25–32.
3. Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. 2016. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 5648–5656.
4. Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE 1, 2 (2017), 4.