AlgoSurg Phase 2 report

This report contains overall model and the trained results on ModelNet10 network.

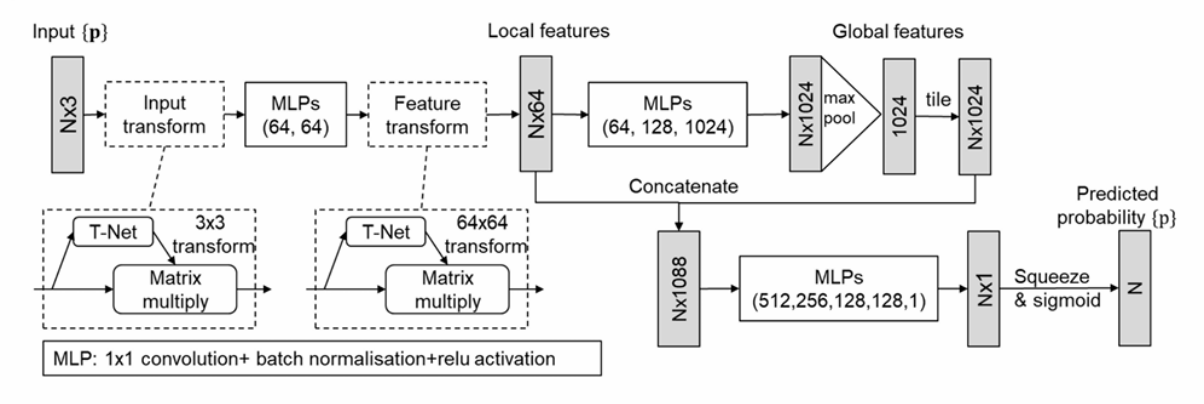
About the DATASET

ModelNet 10 data is used for training and testing the model . It contain 10 classes as shown in figure

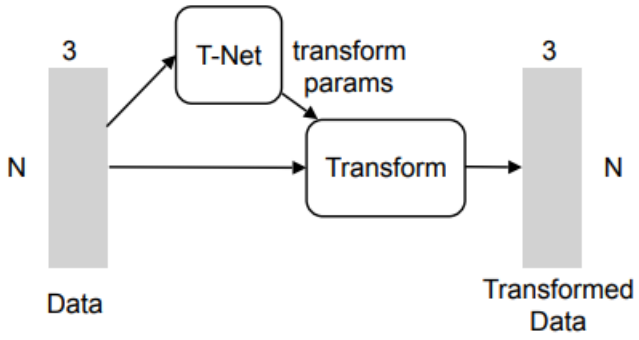
```
processing class: chair
processing class: monitor
processing class: dresser
processing class: sofa
processing class: table
processing class: bathtub
processing class: bed
processing class: toilet
processing class: desk
processing class: night_stand
```

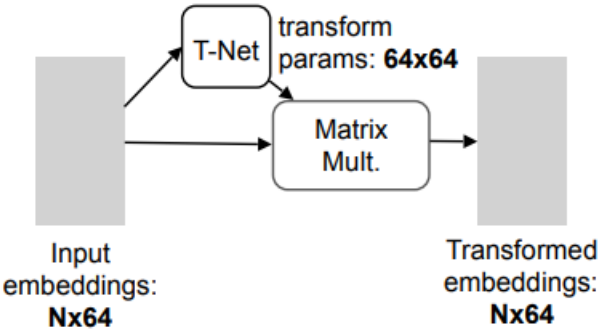
About the model

Model is based on PointNet Architecture which is required for Object Classification, Object Part Segmentation and Semantic Scene Parsing . We are going to do the Point Cloud Segmentation. Below are some images about the model and it's working .



Idea: Data dependent transformation for automatic alignment





Regularization:

Transform matrix A 64x64 close to orthogonal:

$$L_{reg} = \|I - AA^T\|_F^2$$

```
Model: "pointnet"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=================================================================================================
 input_2 (InputLayer)        [(None, 2048, 3)]         0         []

 conv1d_11 (Conv1D)          (None, 2048, 32)          128       ['input_2[0][0]']

 batch_normalization_17 (Ba  (None, 2048, 32)          128       ['conv1d_11[0][0]']
 tchNormalization)

 activation_17 (Activation)  (None, 2048, 32)          0         ['batch_normalization_17[0][0]
                                                                 ']

 conv1d_12 (Conv1D)          (None, 2048, 64)          2112      ['activation_17[0][0]']

 batch_normalization_18 (Ba  (None, 2048, 64)          256       ['conv1d_12[0][0]']
 tchNormalization)

 activation_18 (Activation)  (None, 2048, 64)          0         ['batch_normalization_18[0][0]
                                                                 ']

 conv1d_13 (Conv1D)          (None, 2048, 512)         33280     ['activation_18[0][0]']

 batch_normalization_19 (Ba  (None, 2048, 512)         2048      ['conv1d_13[0][0]']
 tchNormalization)

 activation_19 (Activation)  (None, 2048, 512)         0         ['batch_normalization_19[0][0]
                                                                 ']
```

conv1d_13 (Conv1D)	(None, 2048, 512)	33280	['activation_18[0][0]']
batch_normalization_19 (BatchNormalization)	(None, 2048, 512)	2048	['conv1d_13[0][0]']
activation_19 (Activation)	(None, 2048, 512)	0	['batch_normalization_19[0][0]']
global_max_pooling1d_3 (GlobalMaxPooling1D)	(None, 512)	0	['activation_19[0][0]']
dense_9 (Dense)	(None, 256)	131328	['global_max_pooling1d_3[0][0]']
batch_normalization_20 (BatchNormalization)	(None, 256)	1024	['dense_9[0][0]']
activation_20 (Activation)	(None, 256)	0	['batch_normalization_20[0][0]']
dense_10 (Dense)	(None, 128)	32896	['activation_20[0][0]']
batch_normalization_21 (BatchNormalization)	(None, 128)	512	['dense_10[0][0]']
activation_21 (Activation)	(None, 128)	0	['batch_normalization_21[0][0]']
dense_11 (Dense)	(None, 9)	1161	['activation_21[0][0]']

Accuracy and Result Prediction on Train & Test Data

```
Epoch 1/20
125/125 [==============================] - 32s 156ms/step - loss: 2.8899 - sparse_categorical_accuracy: 0.4089 - val_loss: 237185139736576.0000 - val_sparse_categorical_accuracy: 0.46
Epoch 2/20
125/125 [==============================] - 19s 151ms/step - loss: 2.6810 - sparse_categorical_accuracy: 0.5031 - val_loss: 40986005322334208.0000 - val_sparse_categorical_accuracy: 0.4
Epoch 3/20
125/125 [==============================] - 19s 153ms/step - loss: 2.4626 - sparse_categorical_accuracy: 0.5783 - val_loss: 14728410693632.0000 - val_sparse_categorical_accuracy: 0.561
Epoch 4/20
125/125 [==============================] - 19s 156ms/step - loss: 2.3104 - sparse_categorical_accuracy: 0.6232 - val_loss: 327843807232.0000 - val_sparse_categorical_accuracy: 0.5430
Epoch 5/20
125/125 [==============================] - 19s 153ms/step - loss: 2.1744 - sparse_categorical_accuracy: 0.6700 - val_loss: 18371270656.0000 - val_sparse_categorical_accuracy: 0.6927
Epoch 6/20
125/125 [==============================] - 19s 154ms/step - loss: 2.1695 - sparse_categorical_accuracy: 0.6625 - val_loss: 27843062923264.0000 - val_sparse_categorical_accuracy: 0.699
Epoch 7/20
125/125 [==============================] - 19s 155ms/step - loss: 1.9785 - sparse_categorical_accuracy: 0.7156 - val_loss: 426117932515328.0000 - val_sparse_categorical_accuracy: 0.66
Epoch 8/20
125/125 [==============================] - 19s 154ms/step - loss: 1.9320 - sparse_categorical_accuracy: 0.7382 - val_loss: 1324498534902687344.0000 - val_sparse_categorical_accuracy:
Epoch 9/20
125/125 [==============================] - 19s 154ms/step - loss: 1.9178 - sparse_categorical_accuracy: 0.7477 - val_loss: 304670944.0000 - val_sparse_categorical_accuracy: 0.7104
Epoch 10/20
125/125 [==============================] - 19s 155ms/step - loss: 1.8966 - sparse_categorical_accuracy: 0.7479 - val_loss: 11885.8848 - val_sparse_categorical_accuracy: 0.7104
Epoch 11/20
125/125 [==============================] - 20s 158ms/step - loss: 1.7712 - sparse_categorical_accuracy: 0.7798 - val_loss: 21955338093723648.0000 - val_sparse_categorical_accuracy: 0.4
Epoch 12/20
125/125 [==============================] - 19s 155ms/step - loss: 1.7563 - sparse_categorical_accuracy: 0.7900 - val_loss: 39405637746032640.0000 - val_sparse_categorical_accuracy: 0.7
Epoch 13/20
125/125 [==============================] - 19s 155ms/step - loss: 1.7298 - sparse_categorical_accuracy: 0.7860 - val_loss: 28019282411520.0000 - val_sparse_categorical_accuracy: 0.809
Epoch 14/20
```

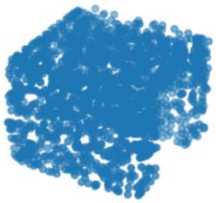
```
125/125 [==============================] - 19s 154ms/step - loss: 1.7036 - sparse_categorical_accuracy: 0.8066 - val_loss: 4126198528.0000 - val_sparse_categorical_accuracy: 0.8348
Epoch 15/20
125/125 [==============================] - 19s 155ms/step - loss: 1.7609 - sparse_categorical_accuracy: 0.7963 - val_loss: 1453408387072.0000 - val_sparse_categorical_accuracy: 0.7731
Epoch 16/20
125/125 [==============================] - 19s 154ms/step - loss: 1.6526 - sparse_categorical_accuracy: 0.8146 - val_loss: 2897902752882701107 20.0000 - val_sparse_categorical_accuracy:
Epoch 17/20
125/125 [==============================] - 19s 155ms/step - loss: 1.6390 - sparse_categorical_accuracy: 0.8266 - val_loss: 216930677948416.0000 - val_sparse_categorical_accuracy: 0.80
Epoch 18/20
125/125 [==============================] - 19s 154ms/step - loss: 1.6376 - sparse_categorical_accuracy: 0.8151 - val_loss: 73319408.0000 - val_sparse_categorical_accuracy: 0.8117
Epoch 19/20
125/125 [==============================] - 19s 154ms/step - loss: 1.6045 - sparse_categorical_accuracy: 0.8311 - val_loss: 33479497728.0000 - val_sparse_categorical_accuracy: 0.7808
Epoch 20/20
125/125 [==============================] - 20s 156ms/step - loss: 1.5756 - sparse_categorical_accuracy: 0.8411 - val_loss: 121.2330 - val_sparse_categorical_accuracy: 0.7445
<keras.src.callbacks.History at 0x78bdb8dad5d0>
```

pred: sofa, label: sofa

pred: bed, label: bed

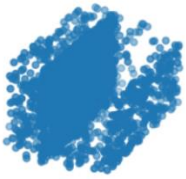
pred: chair, label: chair

pred: table, label: desk



pred: chair, label: chair

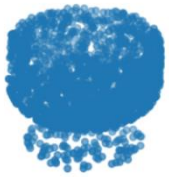
pred: monitor, label: monitor



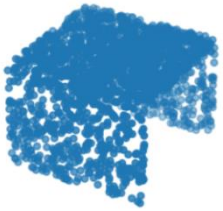
pred: sofa, label: sofa

pred: monitor, label: monitor

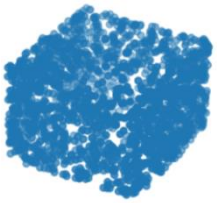
pred: chair, label: chair



pred: chair, label: table



pred: chair, label: dresser



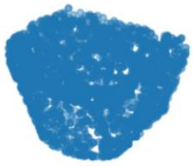
pred: table, label: monitor

pred: chair, label: night_stand

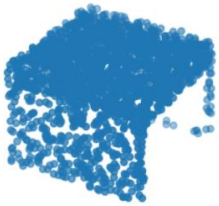
pred: chair, label: chair

pred: chair, label: chair

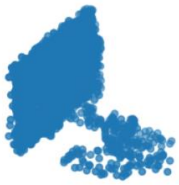
pred: table, label: toilet



pred: chair, label: desk



pred: chair, label: monitor



pred: chair, label: desk

pred: chair, label: table