UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Neural Networks and Deep Learning

A.Y. 2022/2023

Final Project: 3D Object Classification Using Graph Convolutional Network

Candon Matteo 2020353[1], Nicoletti Gianpietro 2053042[2], Rizzetto Nicola 2052417[3]

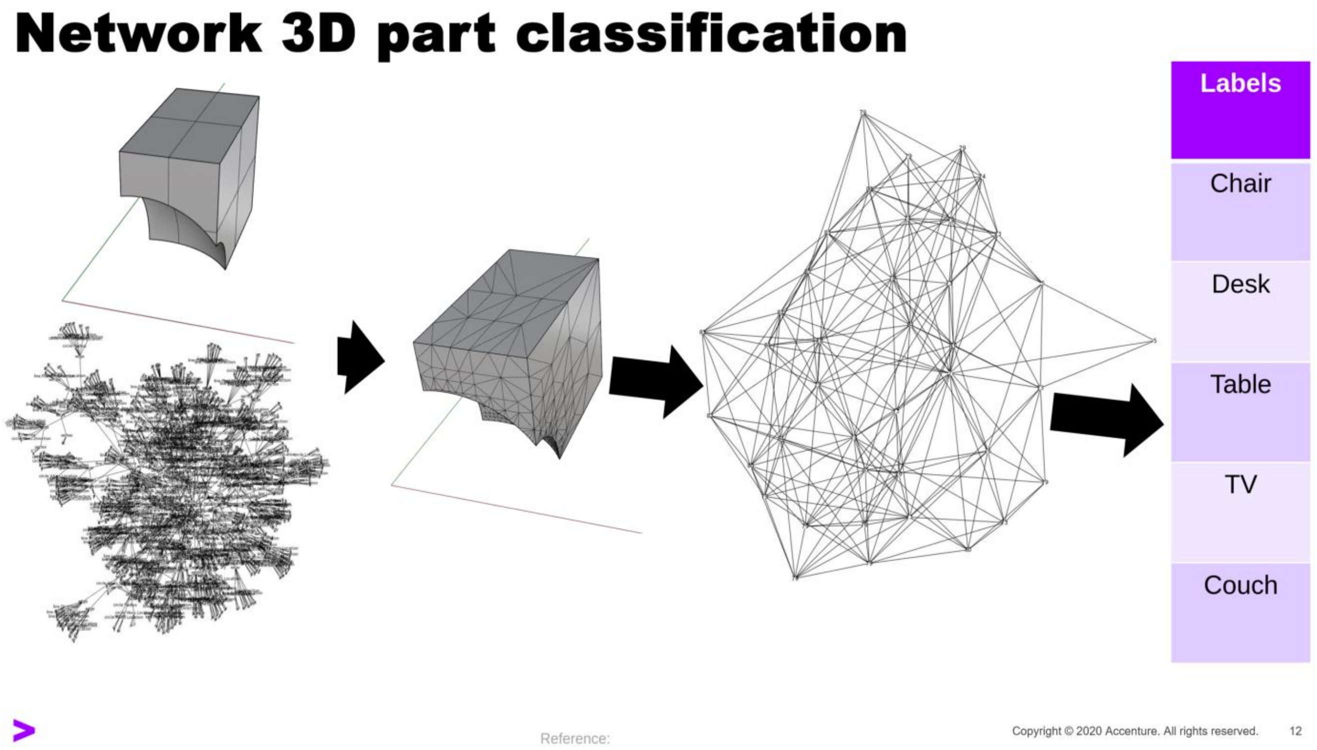
[1]*Department of Information Engineering, ICT for Internet and Multimedia, University of Padua*

[2]*Department of Information Engineering, ICT for Internet and Multimedia, University of Padua*

[3]*Department of Information Engineering, Computer Engineering, University of Padua*

Introduction

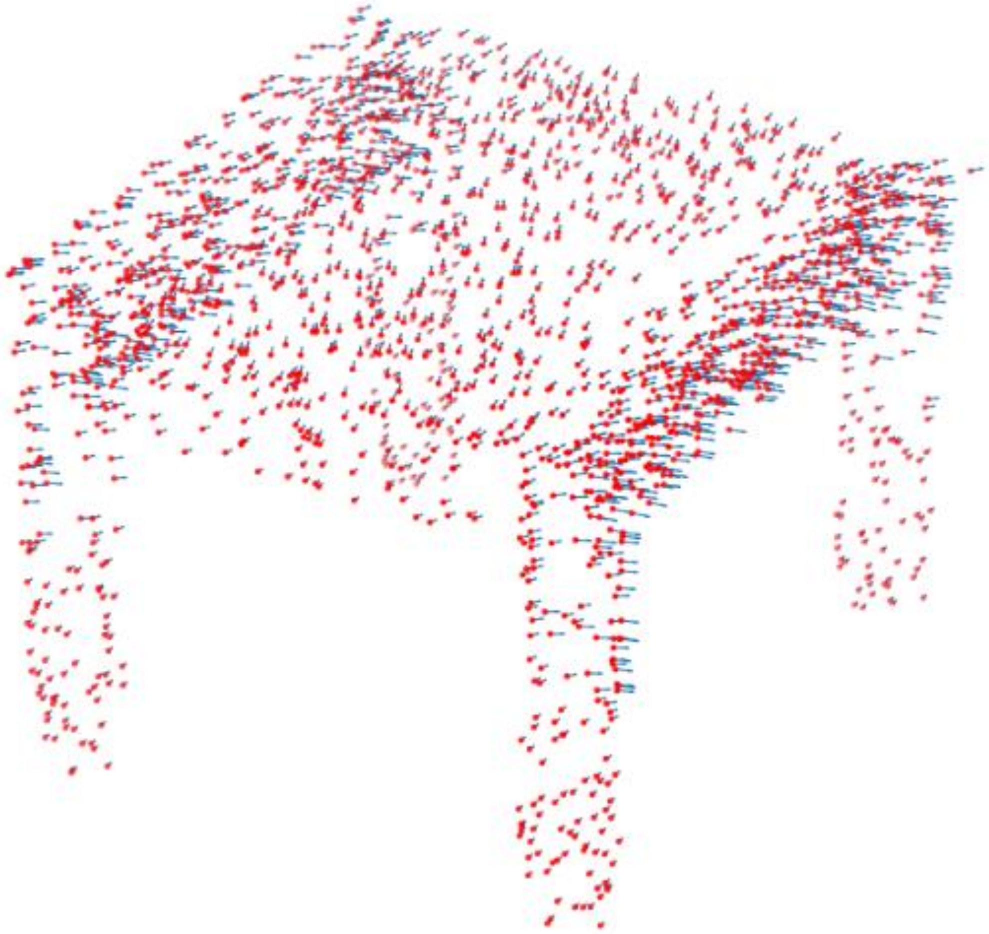
- The project aims to **classify 3D Objects**, with the structure of point clouds. Instead of using voxel grids and CNNs, the idea was to **interpret a point cloud as a graph and use Graph Convolutional filters**. The final model would then be compared to the PointNet model.
- The dataset used was **ModelNet10**, with 3991 train samples and 908 test samples, divided into 10 object classes.



Graph building and data augmentation

- Since a point cloud is a **structureless** set of points, we decided to build a graph with the points as nodes and building edges between a node and its 50 closest neighbors. The resulting graphs for each point cloud contained 2048 nodes and 102400 edges.
- For **data augmentation** we first estimated the **normal vectors** for each point, in order to give an idea of the surface of the object.
- To have more samples, we took the original dataset and added **Gaussian noise**, one version with mean 0 and variance 5 and the other with mean 0 and variance 10, for a total of 11973 training samples.

Estimation of normal vectors



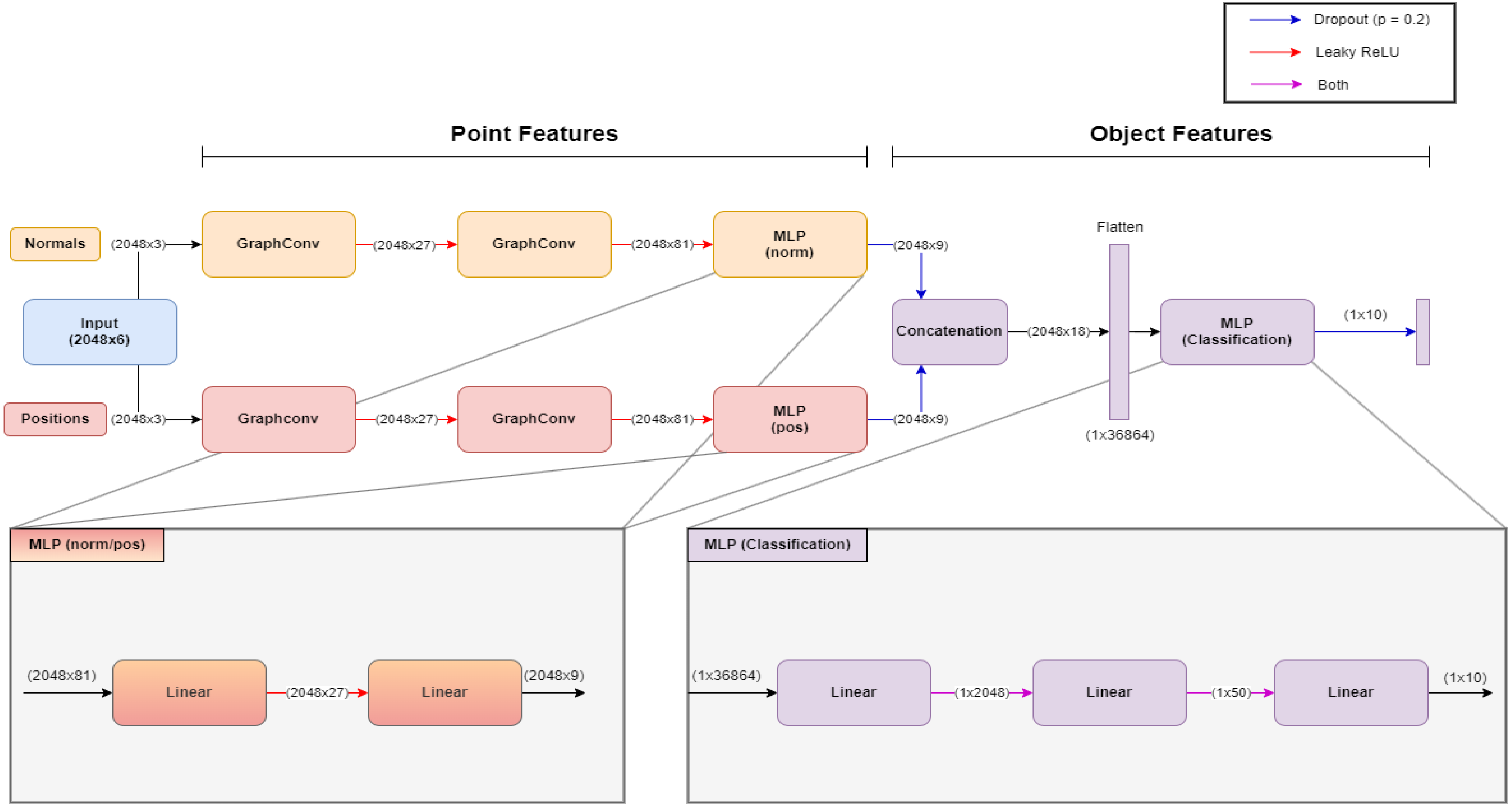
Graph Convolutional Networks

- GCN are a type of Neural Networks that apply convolution to a graph structure
- A layer of a GCN applies the following function to its input:

$$\sigma\left(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}FW\right)$$

- D = degree of the nodes
- Â = A + I = adjacency matrix with self loop added
- F = feature matrix of the nodes
- W = weight matrix

GraphConv Model



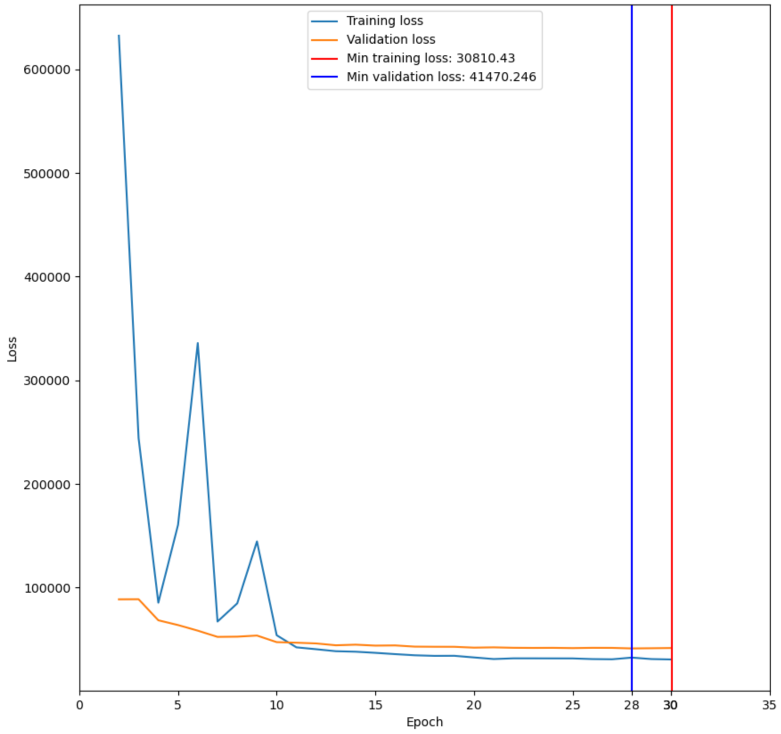
Training procedure

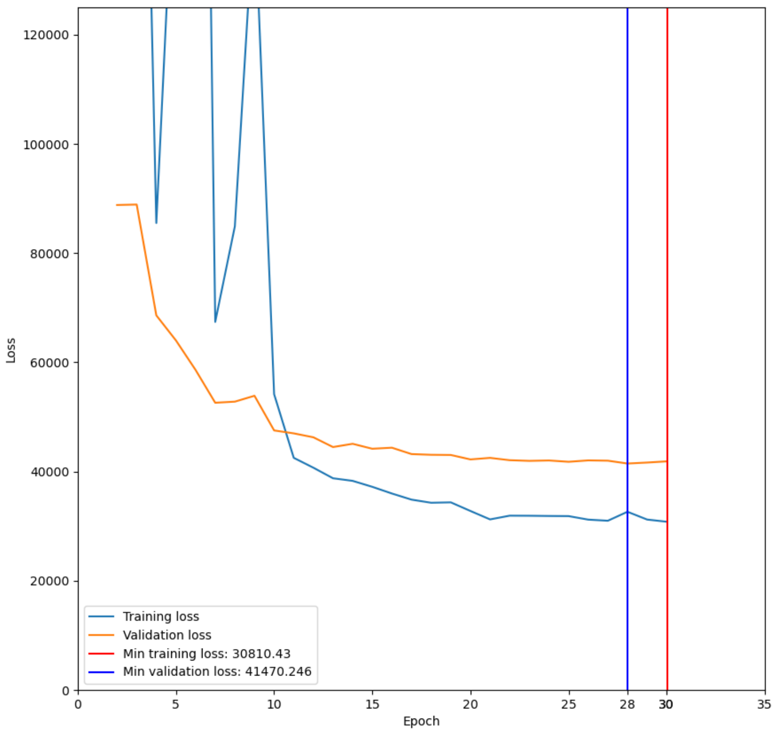
- The labels were redefined as a one-hot-encoding, with the difference of having a **weight** = 1000 instead of 1 in the corresponding index. This was done to force the model to assume the probability of being in a specific class. Backpropagation was regulated using **RMSProp** using a **weights decay of 10^{-2}** as optimizer and using the **mean gradient of 20 samples at once**. Also, learning rate decays every 10 epochs of a factor 10.

Trained Model

Given the unusual one-hot-encoding, we found more appropriate to use the MSE loss, **reason why the losses are so high**.

	Best train loss	Best validation loss
Training set	13177.574	12747.697
Validation set	41867.780	41470.242

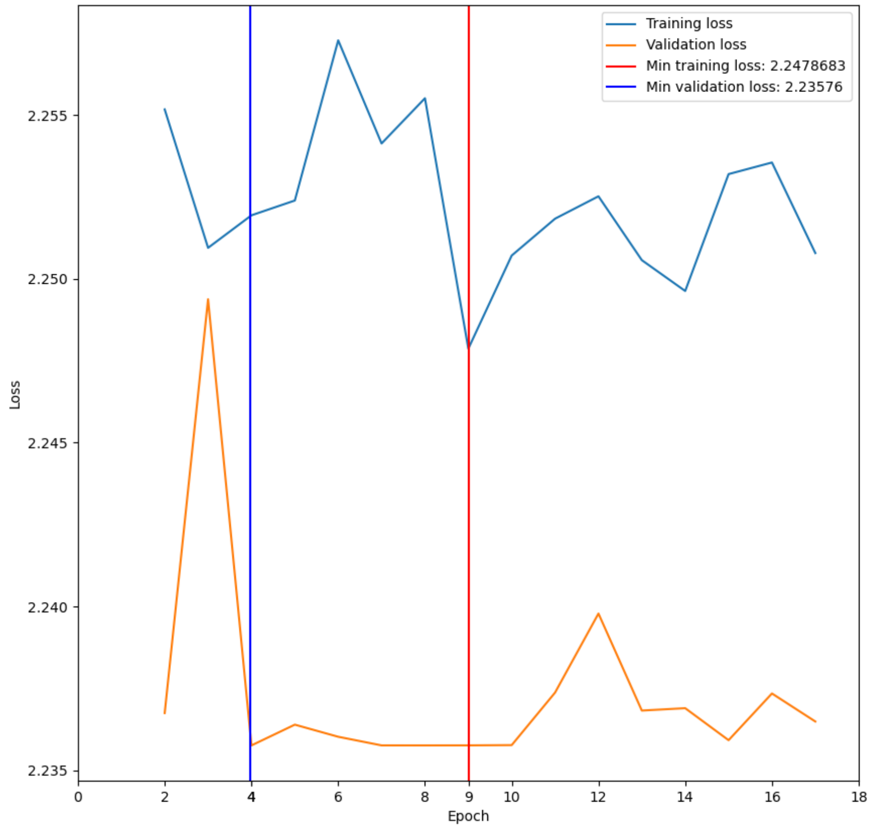




Design Choices

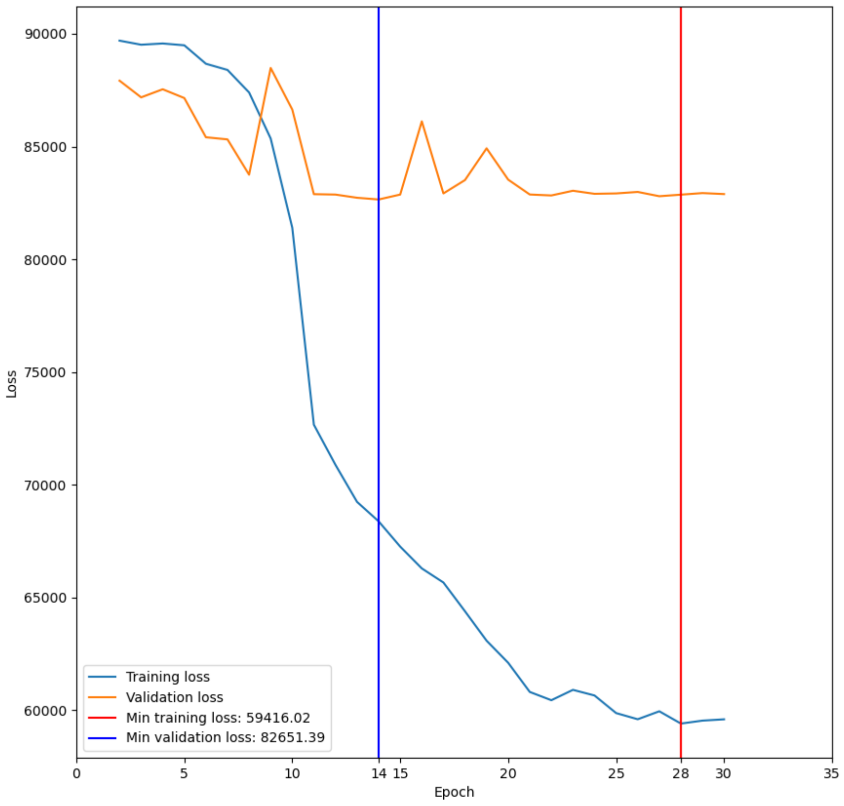
Model with normalized input, Softmax as output function and Cross Entropy as loss function:

Train: 22.28%, **Test:** 11.01%



Model with normalized input and MSE as loss function:

Train: 23.72%, **Test:** 32.93%

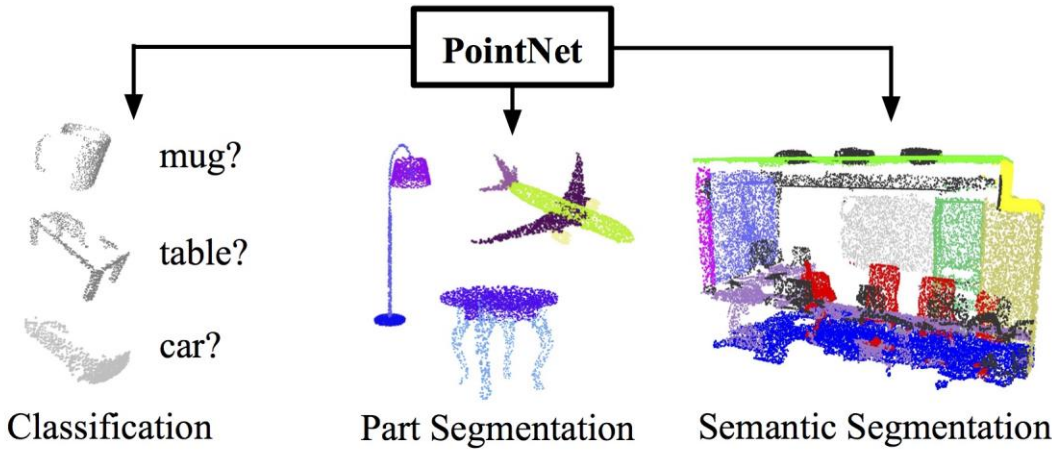


PointNet model

The PointNet model is designed to process unordered set of 3D point clouds.

The model is developed to deal with Model Classification, Part Segmentation and Semantic Segmentation.

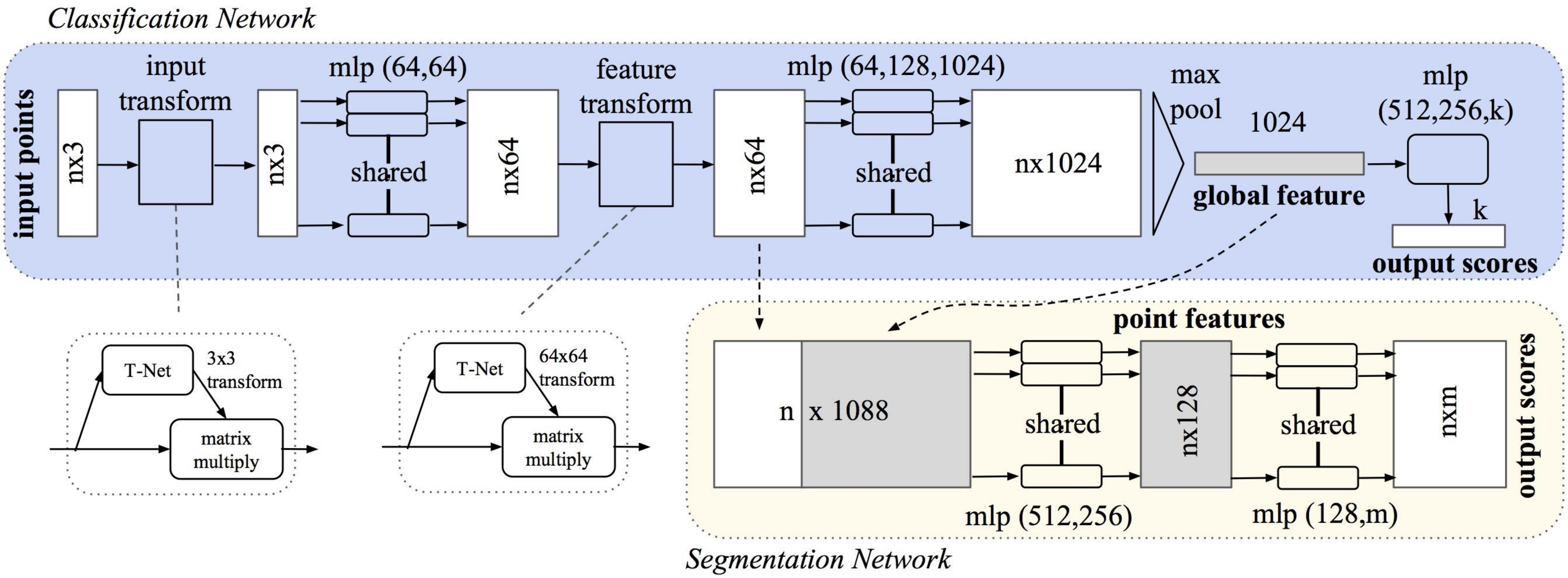
Since the input points are unstructured, the network needs to be invariant to translations and rotations.



Main Blocks

- **T-Net:** transforms the input first in a canonical representation and then applies an affine transformation for alignment
- **Symmetry block:** the input gets compressed by a MLP and then a max pooling layer is applied
- **Information aggregation block:** taken the input, its global features are concatenated with the features of each point, obtaining both global and local information

PointNet Structure



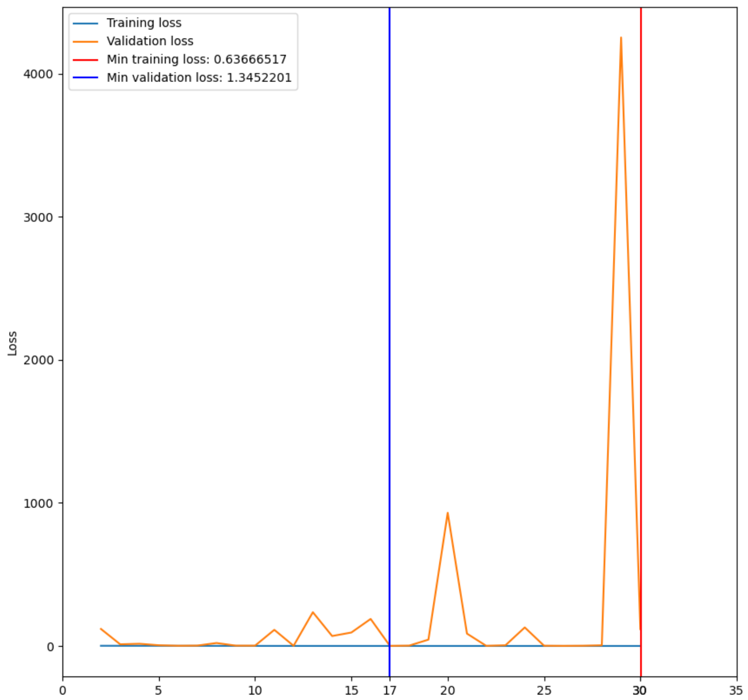
PointNet and training

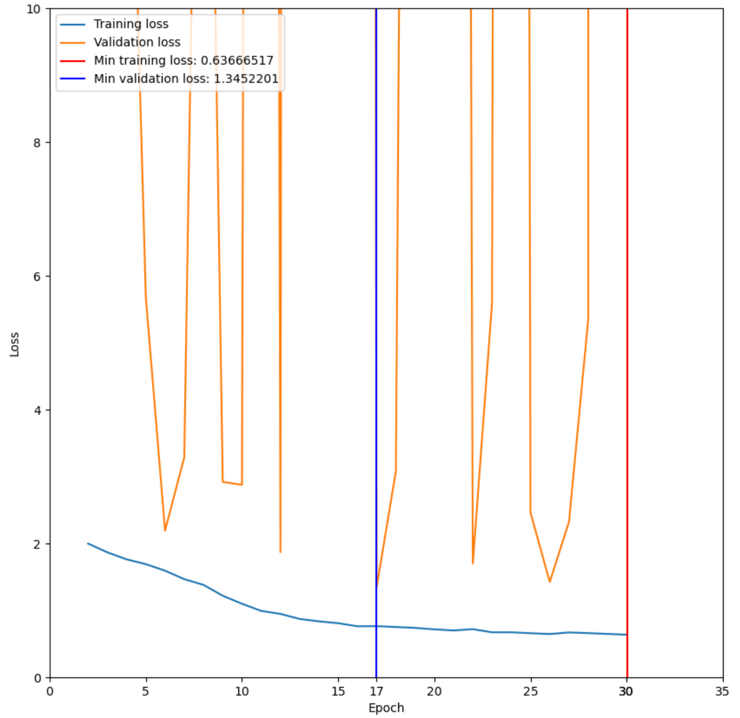
- The training done on the PointNet was different than the one presented in the paper, since for a true comparison the dataset fed to the network was the one implemented during the project, with some slight modifications to the original one (data augmentation, points per cloud). Also the training process was modified.
- The data augmentation performed on the dataset is the same as for the GCN Model, for a true comparison.
- For this model we used RMSProp and Learning Rate Decay, that is reduced of a factor 10 every 10 epochs

Trained PointNet model

The resulting graph is not very meaningful, given the unusual value for the validation loss (reported also in other applications of the PointNet(*)), and we can observe how the model overfits.

	Best train loss	Best validation loss
Training set	34.69	1.35
Validation set	118.35	1.34





```
model.compile(
    loss="sparse_categorical_crossentropy",
    optimizer=keras.optimizers.Adam(learning_rate=0.001),
    metrics=["sparse_categorical_accuracy"],
)

model.fit(train_dataset, epochs=20, validation_data=test_dataset)
```

```
ccuracy: 0.2724 - val_loss: 5804697916006203392.0000 - val_sparse_categorical_accuracy: 0.3073
ccuracy: 0.3443 - val_loss: 836343949164544.0000 - val_sparse_categorical_accuracy: 0.3425
ccuracy: 0.4260 - val_loss: 15107376738729984.0000 - val_sparse_categorical_accuracy: 0.3084
ccuracy: 0.4939 - val_loss: 6823221.0000 - val_sparse_categorical_accuracy: 0.3304
ccuracy: 0.5560 - val_loss: 675110905872323182592.0000 - val_sparse_categorical_accuracy: 0.4493
ccuracy: 0.6081 - val_loss: 600389124096.0000 - val_sparse_categorical_accuracy: 0.5749
ccuracy: 0.6394 - val_loss: 680423464582760103936.0000 - val_sparse_categorical_accuracy: 0.4912
ccuracy: 0.6575 - val_loss: 44108689408.0000 - val_sparse_categorical_accuracy: 0.6410
ccuracy: 0.6725 - val_loss: 873314112.0000 - val_sparse_categorical_accuracy: 0.6112
ccuracy: 0.7018 - val_loss: 13168980992.0000 - val_sparse_categorical_accuracy: 0.6784
ccuracy: 0.7056 - val_loss: 36888236785664.0000 - val_sparse_categorical_accuracy: 0.6586
ccuracy: 0.7166 - val_loss: 85375.9844 - val_sparse_categorical_accuracy: 0.7026
ccuracy: 0.7447 - val_loss: 7.7962 - val_sparse_categorical_accuracy: 0.5441
ccuracy: 0.7444 - val_loss: 66469.9062 - val_sparse_categorical_accuracy: 0.6134
ccuracy: 0.7695 - val_loss: 519227186348032.0000 - val_sparse_categorical_accuracy: 0.6949
ccuracy: 0.7702 - val_loss: 5263462156149188460544.0000 - val_sparse_categorical_accuracy: 0.6520
ccuracy: 0.7903 - val_loss: 142240048.0000 - val_sparse_categorical_accuracy: 0.7941
ccuracy: 0.7855 - val_loss: 2.6049 - val_sparse_categorical_accuracy: 0.5022
ccuracy: 0.8003 - val_loss: 1152819181305987072.0000 - val_sparse_categorical_accuracy: 0.7753
ccuracy: 0.8176 - val_loss: 12854714433536.0000 - val_sparse_categorical_accuracy: 0.7390
```

(*) **Taken from:**
https://keras.io/examples/vision/pointnet/

Results

- As a final analysis we ran the best models (best training and best validation losses) on the original train and test sets, so with no noisy samples.
- For the developed model, the best model was the Best TL, with a test accuracy of 82.158 %, slightly better than the 81.938 % of the Best VL.

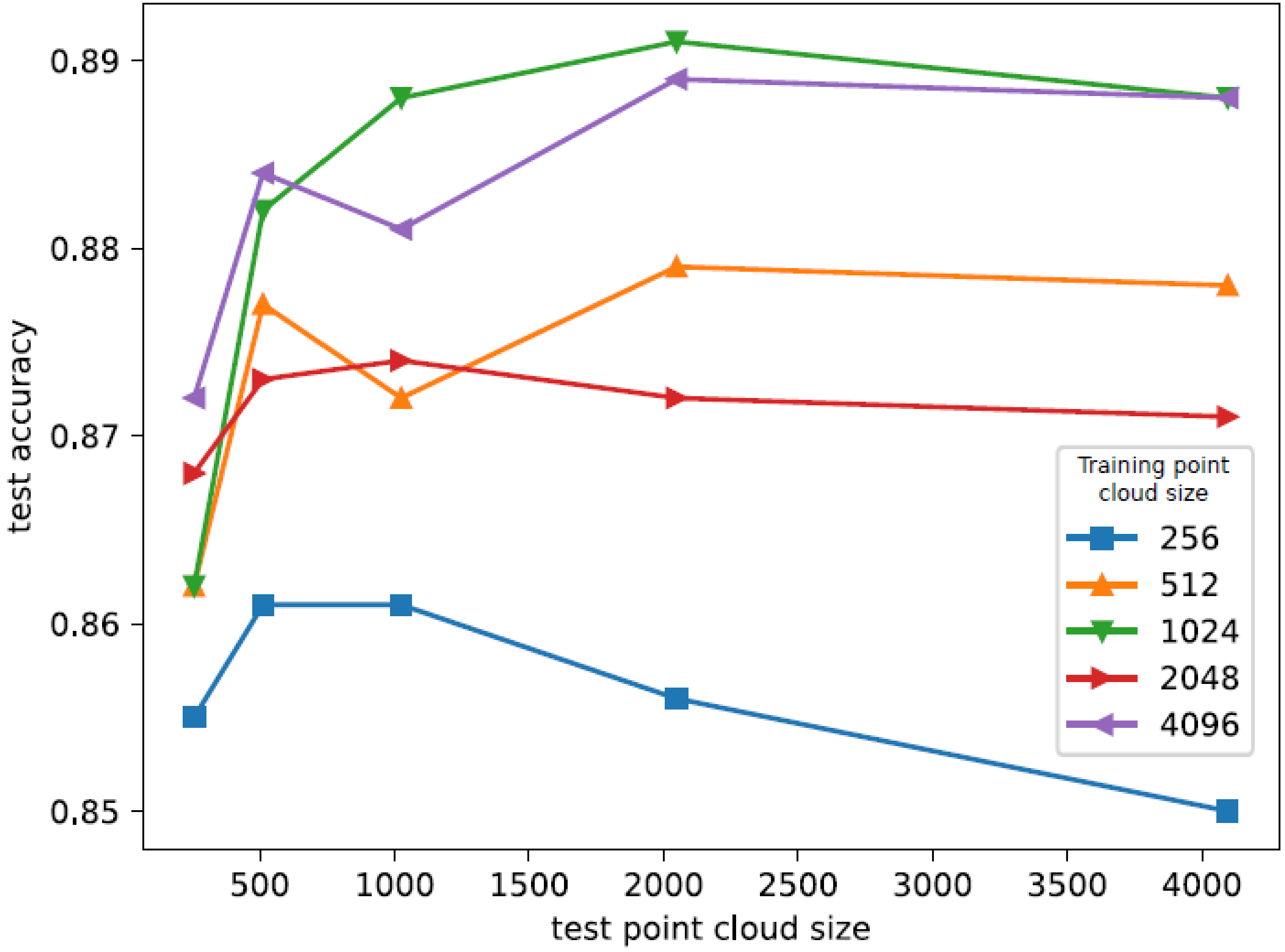
- Before considering the PointNet model, it must be noted that it performs best with 1024 or 4096 points per point cloud, so we expected a lower performance.(**)
- In the end, for the PointNet model, the best model was indeed the Best TL, with a test accuracy of 74.44 %. The accuracy is far lower than the results from the paper, but since the dataset is very different this was expected.

Graph Convolutional Network

		Best TL	**Best VL**
Training	**Loss**	18755.615	18696.443
	Accuracy	94.838%	94.688%
Test	**Loss**	34897.816	34794.234
	Accuracy	82.158%	81.938%

PointNet Model

		Best TL	**Best VL**
Training	**Loss**	44.85	1.23
	Accuracy	79.48%	63.21%
Test	**Loss**	0.75	1.53
	Accuracy	74.44%	49.66%



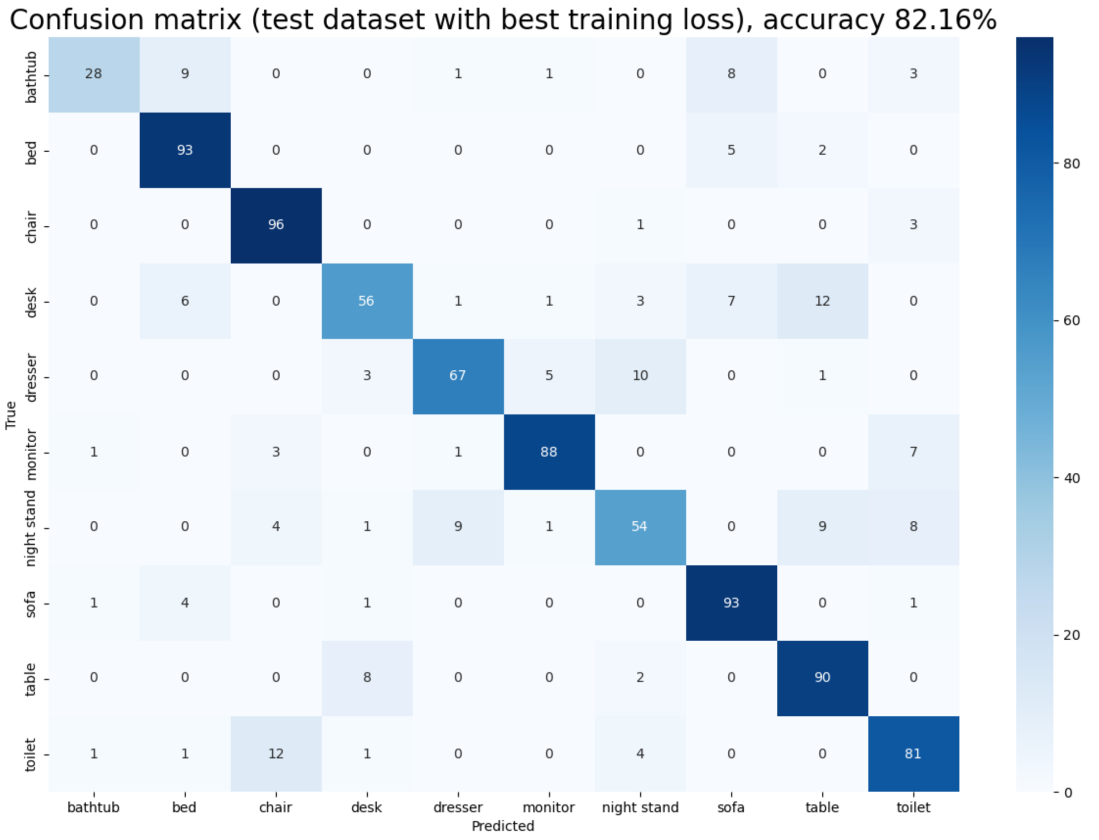
(**) Taken from:
D. G. Shayan Hoshyari, Zicong Fan, "Point cloud classification with pointnet"

Università degli Studi di Padova

Final considerations

Dipartimento di Ingegneria dell'Informazione

- The developed model is overall pretty solid, since it has a better performance of PointNet using this dataset and this preprocessing. Future work may include improving its structure to process heavier datasets or adapt it to more complex graph for Graph Classification.



1222 · 2022
800 ANNI

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Thank you for your attention