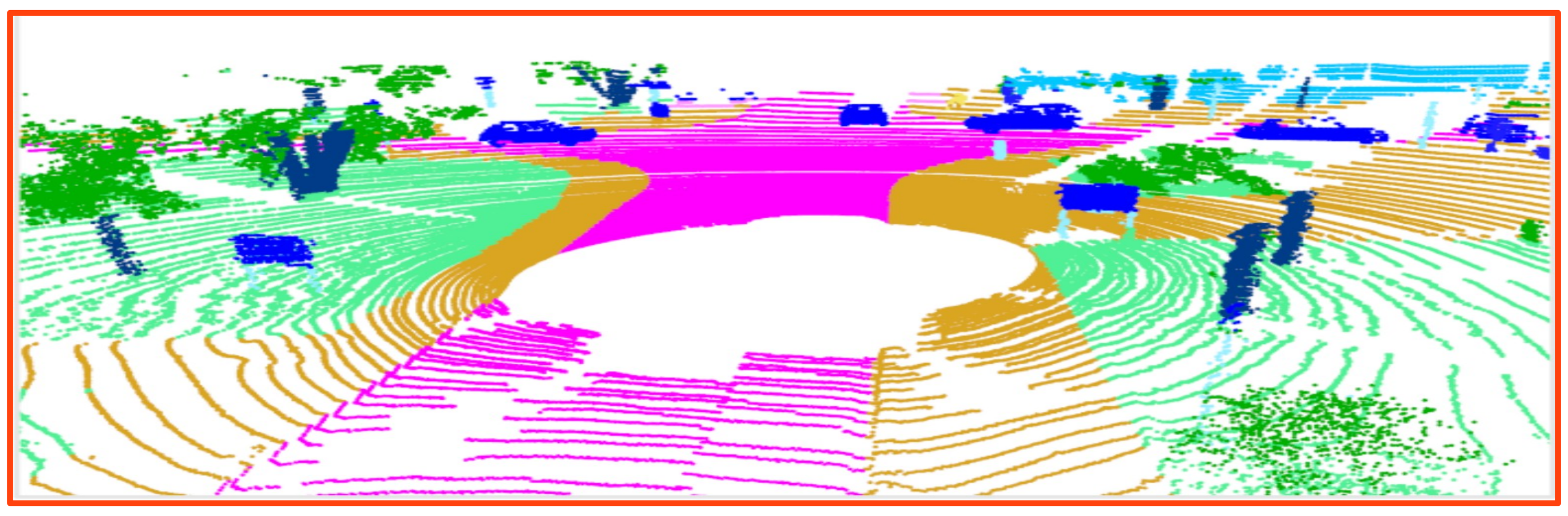
1st Open Day and Workshop
14th July 2022

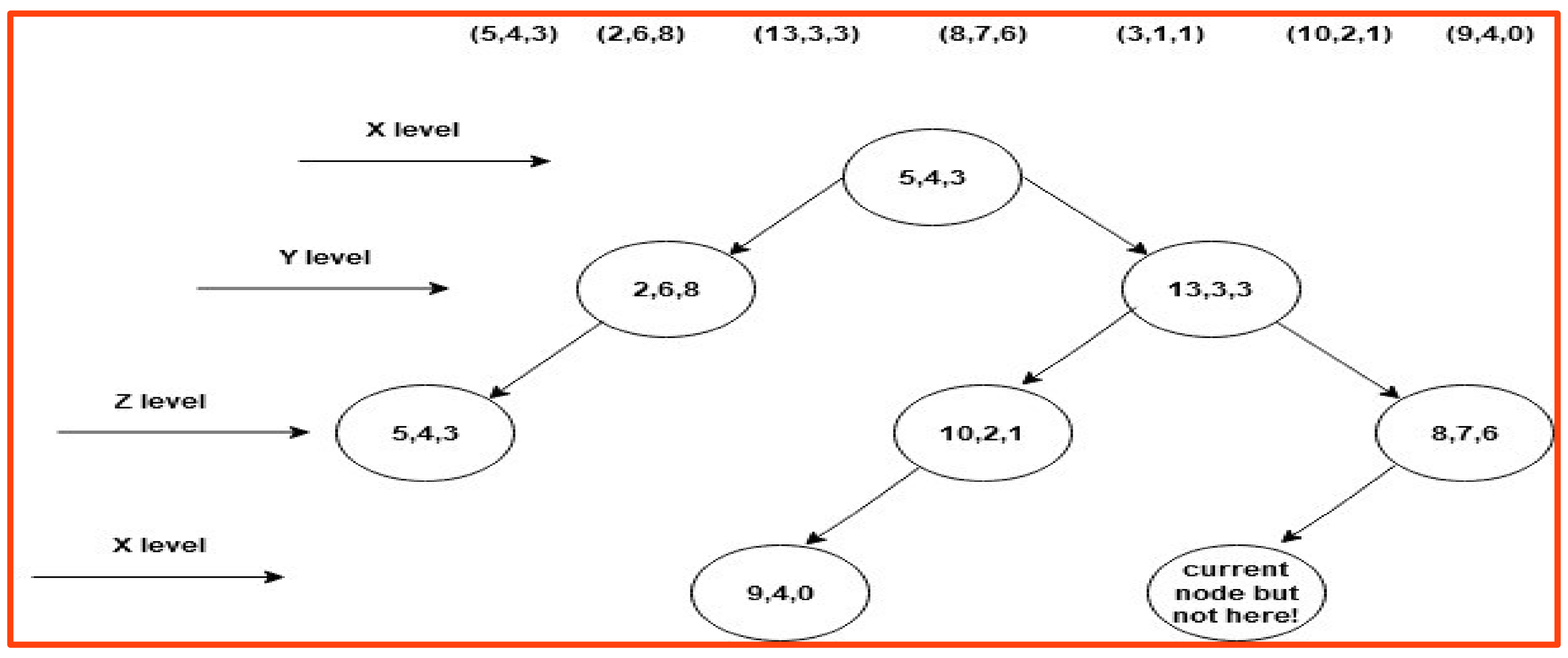
INSTITUTO DE SISTEMAS E ROBÓTICA
UNIVERSIDADE DE COIMBRA

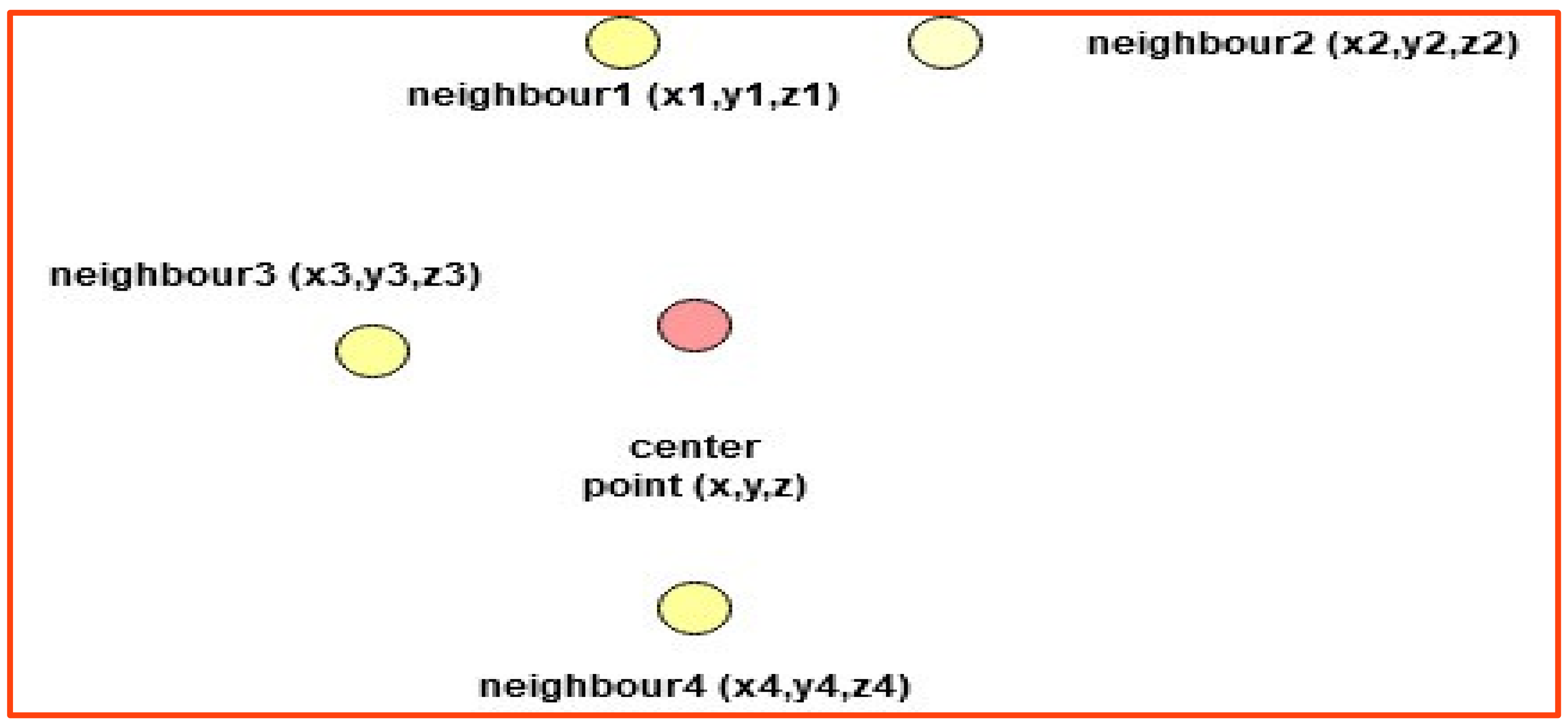
Semantic Segmentation of Large-Scale LiDAR Point Clouds for Forest Application

Habibu Mukhandi

- The research addresses the problem of applying deep learning algorithms directly on raw 3D-LiDAR data points captured in the forestry application.
- Permutation invariant neural network for raw point clouds data are typically unstructured, with no regular grid.
- Systematic sampling for outstandingly memory and computation efficient, the systematic sampling of points samples only a small subset of points while preserving spatial and geometrical features.
- Minimizing false positive rate especially in similar classes such as different species of trees in the forest.
- Measuring the impact of augmenting data set with synthetic data on improving robot navigation task.







kd tree on 3 dimension data. During insertion of the 3d points, the first node becomes the root node at level 0 and the next node goes to the left if its x coordinate is smaller than the root's x coordinate and goes to the right if its x is larger than root's x. At level 1 we compare y coordinates if smaller go left if larger go right. At level 2 we compare z coordinates and at level 3 we compared x coordinates as in level 0 and the process repeats itself. This process of finding nearest neighbours run in O(NlogN) time [6], where N stand for number of nodes. After creating a graph, we used K-Nearest Neighbours to find 2000 nearest neighbours for each point (to save computation we use square distance instead of square root to find Euclidean distance). We select 2000 neighbours because we want for every ≈ 100k points to get a number of points in order of magnitude of 100 when other authors do random sampling. Therefore, each node is set to have degree 2000 in our implementation. We take only one point from the 2000 points neighbourhood and discard the rest. The selected points are the systematically sampled points to be used as inputs to MLP. Our algorithm makes sure that selected points are not adjacent to each other which in essence can be compared to farthest point selection algorithm except our method is faster than farthest point selection.

- The Euclidean and Manhattan distances of the selected point relative to each of the 16 nearest neighbours are encoded to the feature to complete the feature map.

Relative point position encoding:

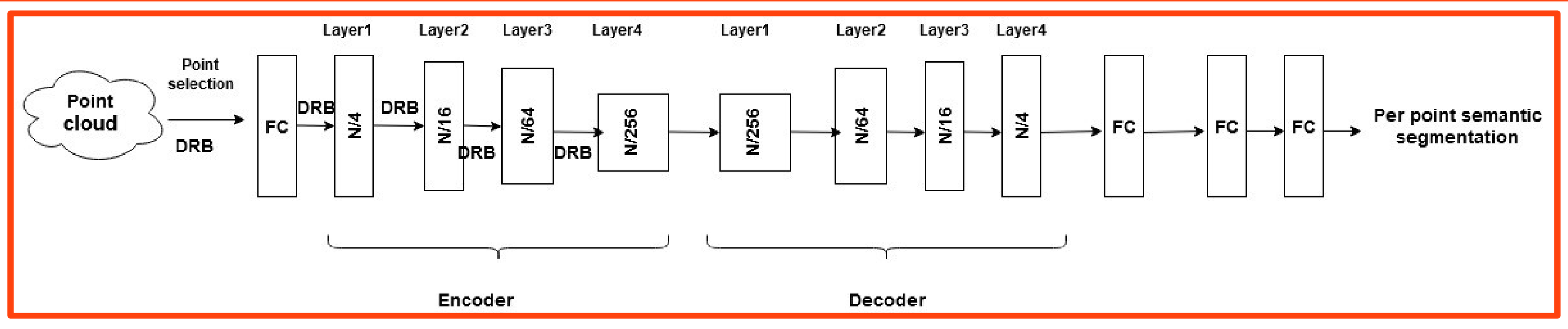
(4.1) $$r_i^k = MLP(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus \|p_i - p_i^k\|)$$

18

(5.1) $$\sigma_B^2 = W_a W_b (\mu_a - \mu_b)^2$$

σ_B^2 = inter-class (between class) variance; W_a, W_b = Number of points of the two similar classes respectively; μ_a, μ_b = mean of the norm of the vectors (we use $v^T v$ to save computation).

- where ⊕ is concatenate symbol, ‖.‖ is Euclidean distance, (pi − pki) is difference between centrepoint and neighbouring point k, pi is the selected point, pk i is each of the k = 16 neighbours. MLP() = input to MultiLayer Percepton and rki is relative point position encoding. Then the rki is concatenated with the other features such as the colors of each point RGB.



- Encoder and decoder for downsampling and upsampling the input points features for semantic segmentation task.
- FC stands for fully connected layer. DRB stands for Dilated residual block.
- N is the number of 3d points

	mIOU	accuracy	Man made	natural	High veg	Low veg.	buildngs	Hard scape	cars
State of the art	69.31	0.895	94.09	81.87	85.39	37.09	82.18	25.71	86.80
ours	72.23	0.913	94.74	85.72	89.52	39.49	85.77	28.85	90.77