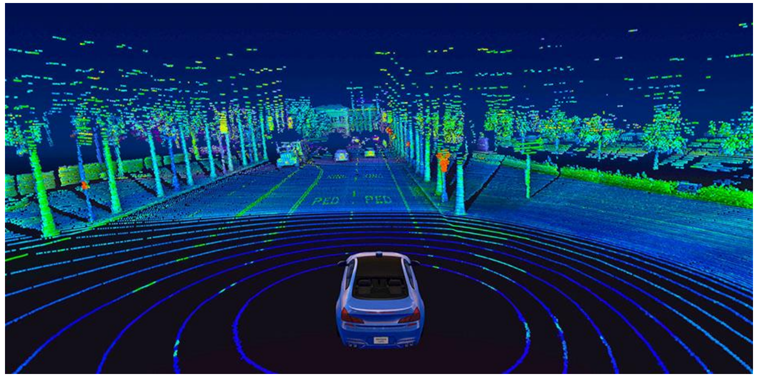
3D Perception & Learning Based Data Fusion

3D Semantic Segmentation with LiDAR scans

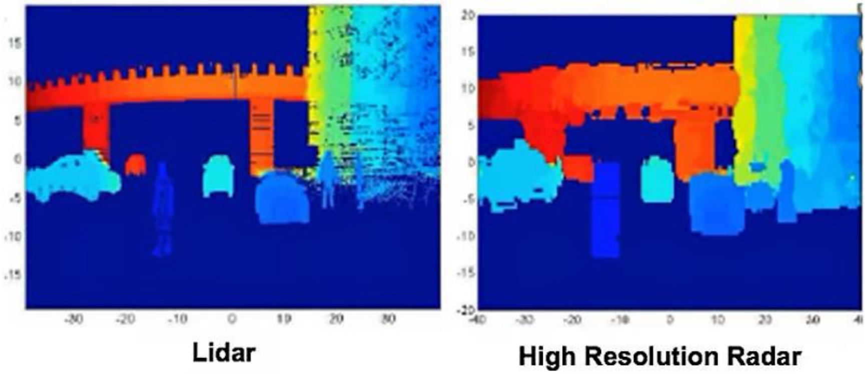
Giuseppe Trimigno

SENSOR

- ***Lidar (LIght Detection And Ranging)***, or ***Laser Range Sensors***, are exteroceptive and active sensors used for range distance measurement.
- Uncertainty on the estimated range is inversely proportional to the square of the received signal amplitude.
- They are composed by a transmitter and a receiver.



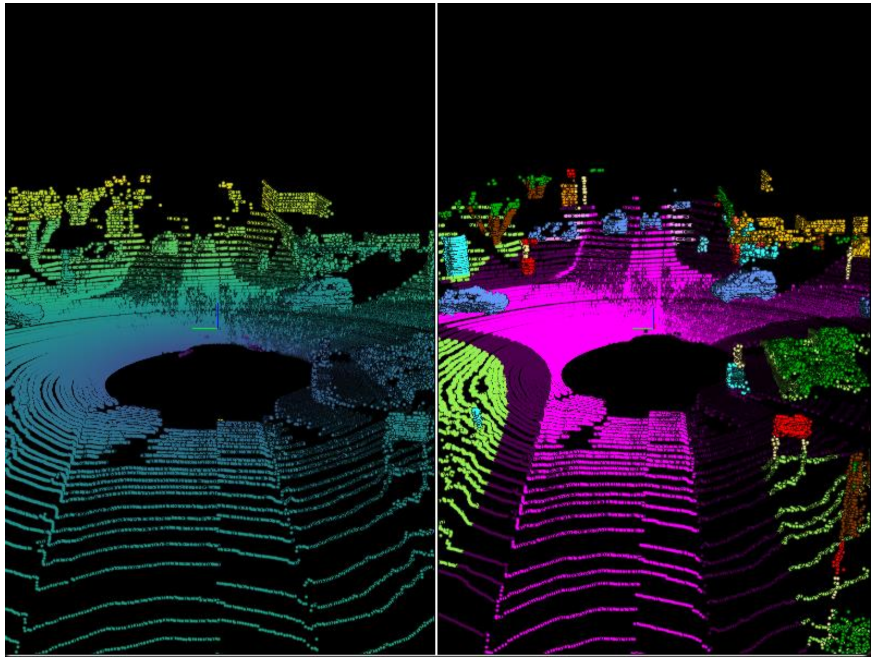
Operating principles:
- Pulsed laser (standard), which directly measures elapsed time.
- Phase shift measurement to produce range estimation.

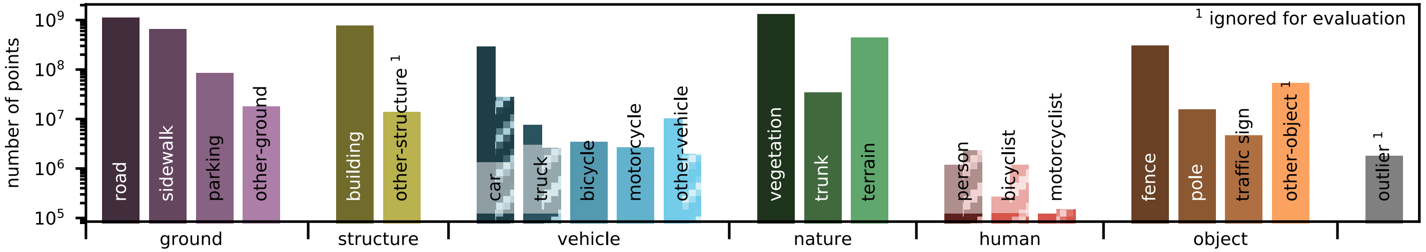


Do not confuse with radars!
- Both technologies are used for object detection and mapping, but...
- Lidar offer high-resolution data, but with a shorter range and an high cost.
- Radar have a longer range, are less expensive and more robust against adverse weather conditions.

DATASET & TASK

- **3D LiDAR Segmentation** is a task in which we want to infer the label of each three-dimensional point.
- **Single scan** evaluation task, in which we don't distinguish between moving and non-moving objects.
- **Multiple scan** evaluation task, where we distinguish between moving and non-moving.
- The most common metric for the evaluation of this task is mean **Jaccard**, defined as $\frac{1}{C}\sum_{k=1}^{C}\frac{TP_k}{TP_k+FP_k+FN_k}$.

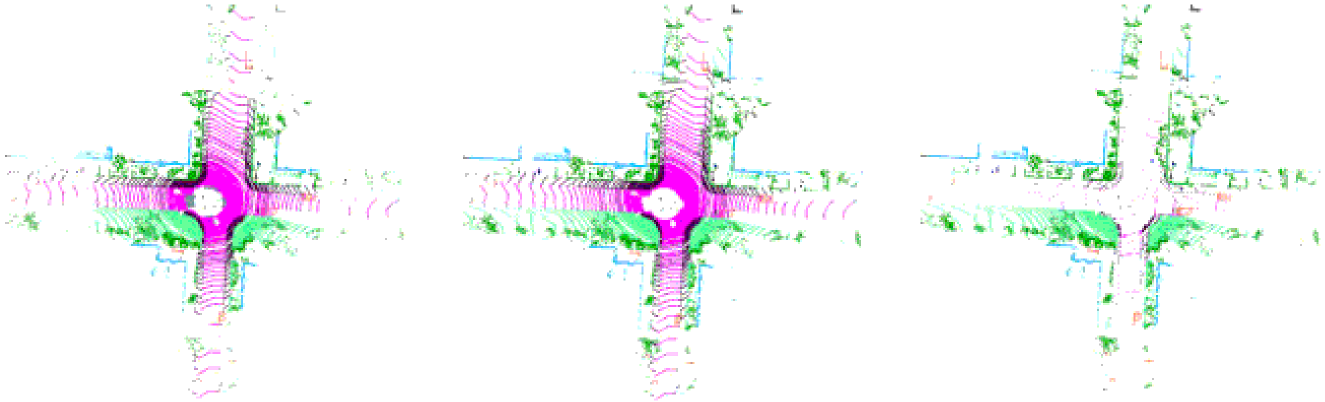




- **SemanticKITTI** is a large-scale dataset containing dense annotations for lidar scans in 22 scenes, totalling 28 different classes.

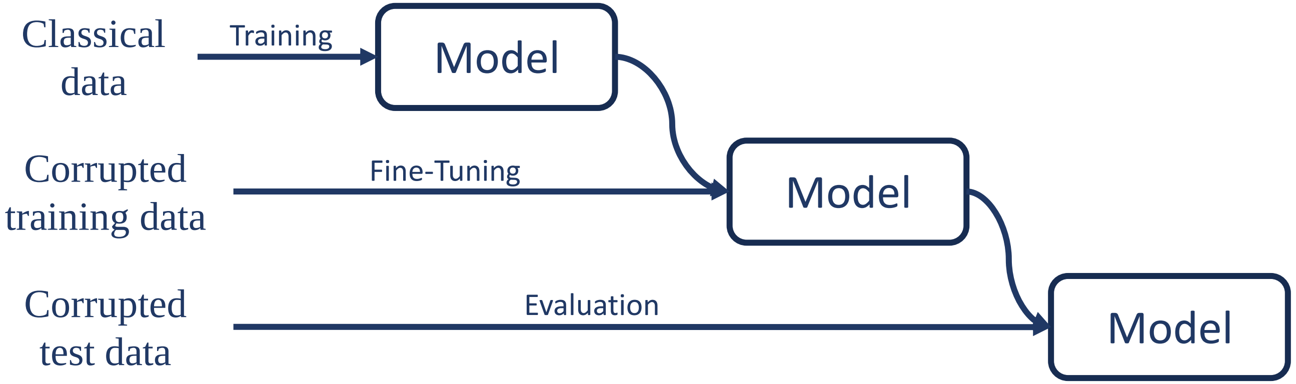
CORRUPTED DATASET

- In order to test the robustness of the models, three under out-of-distribution (**OoD**) scenarios, generated starting from classical data, have been considered.
- Adverse weather conditions (**fog** and **wet ground**) and external disturbances (**motion blur**).

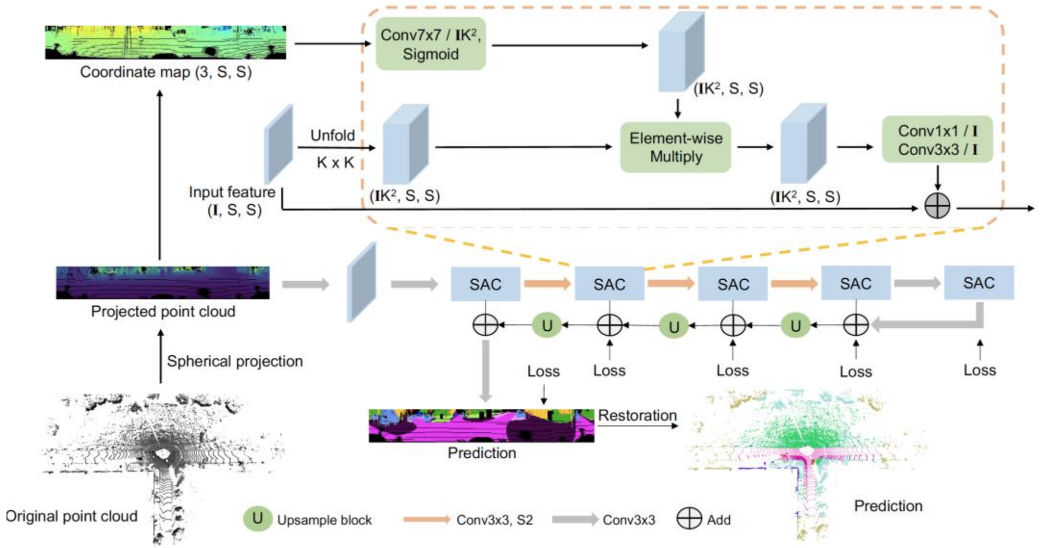


(a) Fog (b) Motion blur (c) Wet grond

- Evaluation of models, trained with classical data, on corrupted data, to observe if there is - and how much is - the performance drop.
- Fine-tuning of models, trained with classical data, using training corrupted data, to see if performance increase.



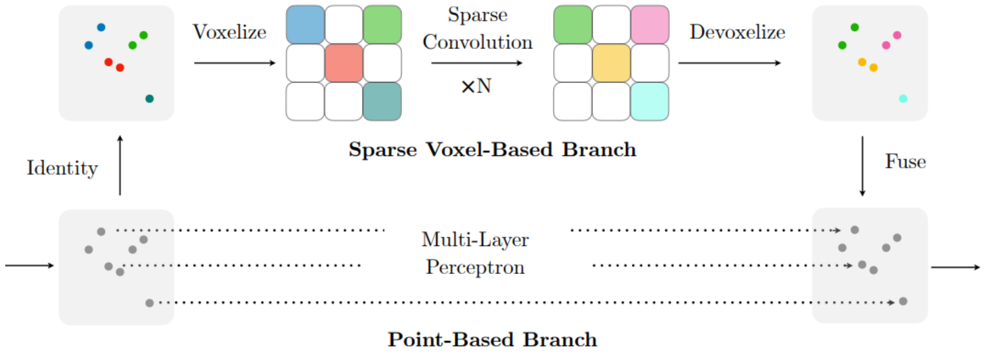
MODELS



(a) SqueezeSegV3's architecture

SqueezeSegV3

- The problem is that projecting a 3D point cloud to get a 2D LiDAR image and using convolution to process it ends in a drastic performance decrease.
- The solution is **SAC**, to deal with spatial variance in projected point clouds, in order to adopt different filters for different locations according to the input image since, for LiDAR images, the feature distribution across the image are no longer identical.



(b) Sparse Point-Voxel Convolution

SPVCNN (Sparse Point-Voxel Convolution)

- In PVConv, given a large outdoor scene, each voxel grid corresponds to a large area → hard to recognize small instances.
- Due to the limited computational resources, SparseConv cannot be very deep → the network has to downsample very aggressively.
- **SPVConv** overcome these limitations working with two different branches: Point-Based and Sparse Voxel-Based.

RESULTS

Model	Data	mIoU	Accuracy
SqueezeSegV3	Classical	0.527	**0.891**
SqueezeSegV3	Fog	0.444	0.825
SqueezeSegV3	Motion Blur	0.370	0.789
SqueezeSegV3	Wet Ground	0.496	0.868
SqueezeSegV3 Fine-tuned	Fog	**0.486**	**0.877**
SqueezeSegV3 Fine-tuned	Motion Blur	0.468	0.867
SqueezeSegV3 Fine-tuned	Wet Ground	0.502	**0.87**
SPVCNN	Classical	**0.604**	0.875
SPVCNN	Fog	-	-
SPVCNN	Motion Blur	0.552	0.836
SPVCNN	Wet Ground	0.575	0.865
SPVCNN Fine-tuned	Fog	-	-
SPVCNN Fine-tuned	Motion Blur	**0.630**	**0.875**
SPVCNN Fine-tuned	Wet Ground	**0.627**	0.863

Testing results (**mean accuracy** and **mean IoU**) for both **SqueezeSegV3** and **SPVCNN** models (fine-tuned and not), evaluated over **classical** and **corrupted** (fog, wet ground ans motion blur) data.

RESULTS

Model	Data	Car	Bicycle	Motorcycle	Truck	Other-veichle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-sign
SqueezeSegV3	Classical	0.861	0.309	0.478	0.507	0.424	0.520	0.521	0.000	0.945	0.473	0.816	0.003	0.802	0.472	0.825	0.524	0.720	0.423	0.381
SqueezeSegV3	Fog	0.853	0.245	0.417	0.489	0.325	0.421	0.428	0.000	0.815	0.231	0.662	0.005	0.754	0.279	0.760	0.440	0.609	0.384	0.319
SqueezeSegV3	Motion Blur	0.797	0.217	0.244	0.120	0.201	0.249	0.318	0.000	0.756	0.205	0.544	0.002	0.709	0.308	0.747	0.407	0.545	0.332	0.337
SqueezeSegV3	Wet Ground	0.870	0.296	0.424	0.476	0.389	0.484	0.500	0.000	0.843	0.328	0.679	0.005	0.798	0.455	0.827	0.523	0.718	0.425	0.388
SqueezeSegV3 Fine-tuned	Fog	0.851	0.241	0.453	0.395	0.324	0.471	0.505	0.000	0.935	0.423	0.797	0.005	0.788	0.442	0.806	0.509	0.692	0.327	0.270
SqueezeSegV3 Fine-tuned	Motion Blur	0.838	0.237	0.422	0.402	0.223	0.414	0.478	0.000	0.920	0.391	0.782	0.003	0.786	0.424	0.793	0.472	0.695	0.321	0.28
SqueezeSegV3 Fine-tuned	Wet Ground	0.865	0.283	0.478	0.389	0.330	0.491	0.563	0.000	0.903	0.436	0.738	0.003	0.794	0.460	0.819	0.520	0.724	0.390	0.355
SPVCNN	Classical	0.921	0.462	0.655	0.704	0.481	0.614	0.758	0.003	0.849	0.295	0.743	0.001	0.907	0.649	0.890	0.678	0.777	0.578	0.506
SPVCNN	Fog	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPVCNN	Motion Blur	0.898	0.461	0.645	0.477	0.405	0.509	0.765	0.010	0.770	0.204	0.606	0.000	0.854	0.603	0.868	0.662	0.683	0.560	0.497
SPVCNN	Wet Ground	0.928	0.499	0.657	0.458	0.476	0.594	0.757	0.001	0.701	0.222	0.620	0.001	0.904	0.654	0.892	0.675	0.792	0.580	0.506
SPVCNN Fine-tuned	Fog	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPVCNN Fine-tuned	Motion Blur	0.949	0.477	0.673	0.849	0.481	0.714	0.878	0.000	0.909	0.366	0.765	0.010	0.897	0.636	0.864	0.697	0.693	0.611	0.504
SPVCNN Fine-tuned	Wet Ground	0.957	0.508	0.671	0.896	0.439	0.739	0.881	0.000	0.857	0.322	0.681	0.034	0.899	0.614	0.864	0.699	0.714	0.616	0.520

Single-class testing **mean IoU** for both **SqueezeSegV3** and **SPVCNN** models (fine-tuned and not), evaluated over **classical** and **corrupted** (fog, wet ground ans motion blur) data.

QUESTIONS

Thanks for the attention!