**Depth Map Generation Using Frontal Images and LiDAR Point Clouds**

**Introduction**

In this project, we aimed to combine the technologies and knowledge acquired in the previous two projects.

The main objective was to build a system capable of generating depth maps by fusing data from monocular cameras and LiDAR point clouds.

Ultimately, the system should generalize a small portion of the LiDAR projection onto the 2D image space to recreate the depth map for the entire camera image.

The aim was to compare two popular data fusion methods to determine if there are differences in their application to this specific task.

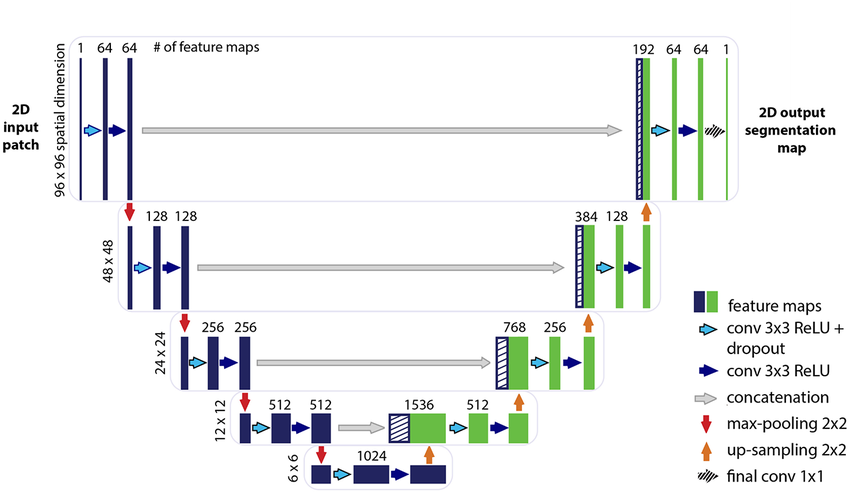
The first approach, **Data-level fusion**, involves combining the data from the two sensors before performing regression.

The second approach is based on the concept of **Prediction-level fusion**, which consists of merging the predictions of the two models trained on raw data.

**Tools and data**

For this particular project, a U-Net neural network was implemented in Python using the PyTorch library.

U-Net is a type of convolutional neural network that compresses an image to learn feature representations and then expands it to generate a different image. Thanks to this capability, U-Net is primarily used in segmentation and generative modeling tasks.



A portion of the Carla Sensor Fusion V8 dataset was used for both training and validation.

The dataset consists of 10,000 samples, including camera images and LiDAR point cloud projections as inputs for the network, and grayscale depth maps as the ground truth.

The dataset was split into 5,000 samples for training and 5,000 for testing. However, in the case of the late fusion approach, the test samples and their respective predictions were used to train the stacking meta-model, which then performed the final prediction on a sub-sample of 1,000 images.

**Implementation**

Several implementations of the model were developed to compare results following both data-level and prediction-level fusion approaches.

In the first case, a model was implemented that takes as input three-channel RGB images and 2D projections of LiDAR data, outputting grayscale images of the generated depth maps.

In the second case, multiple models were used to predict depth maps separately from the LiDAR and RGB images. Subsequently, a third model was employed to perform the stacking procedure and merge the predictions.

**Results and conclusions**

The results were found to be quite similar between the two tested approaches. In particular, by the end of the training set, the model achieved a loss of 0.01 ± 0.005 in both cases. However, predictions appeared to be of higher visual quality when using an early fusion technique.

Based on these observations, the fusion methodology did not prove to be crucial for this specific application.

Conversely, the accuracy of the dataset played an extremely important role. Assuming access to better LiDAR data, the model's accuracy could improve significantly.

**References**

Dataset - <https://www.kaggle.com/datasets/skylam/carla-sensor-fusion-v8>

Simple U-NET implementation - <https://github.com/TFonta/DeepLearningAndGenerativeModelsCourse/blob/main/ex11.ipynb>

Multimodality Data Fusion - <https://medium.com/haileleol-tibebu/data-fusion-78e68e65b2d1#:~:text=Data%20level%20fusion%20is%20a,the%20correlation%20between%20two%20sensors>