**Siamese Networks for LiDAR-Camera Fusion in Autonomous Driving**

The sources provide a comprehensive explanation of the **Siamese network** as a core component of the proposed LiDAR-Camera Fusion algorithm for real-time object detection in autonomous driving.

Here's a discussion of what the sources say about the Siamese network:

• **Definition and Purpose**

    ◦ A Siamese neural network is described as being "composed of a couple of neural networks that can process two different inputs at one time".

    ◦ In the context of this paper, it is constructed specifically "as the feature encoder based on Yolo-v5 for object detection". Its primary role is to process the two distinct, yet similar, input modalities: **RGB camera images and 2D depth images converted from LiDAR point clouds**.

    ◦ The RGB image and the corresponding depth image are considered "similar" because the depth image is projected from the LiDAR point cloud according to the RGB image captured at the same scene, and they share common feature information about objects (e.g., cars, people) while having the same size.

• **Rationale for its Use in Fusion**

    ◦ The authors chose the Siamese network because it "performs excellently in processing two similar inputs" and has been successfully applied in fields like face recognition and object tracking.

    ◦ It helps to **fuse accurate position information from LiDAR with dense texture information from the camera at the feature layer**. This feature-layer fusion strategy is preferred over data-layer or decision-layer fusion because it involves a smaller data volume of feature tensors, reducing processing time, and builds connections at multiple convolutional depths to strengthen the correlation between modalities.

    ◦ By maximizing different feature representations through comparing the similarity of the two inputs, the Siamese network is able to "capture more features".

    ◦ This construction allows the CNN to "learn more abstract feature information from multi-modality data," improving the "possibility and accuracy of the object being detected". The two branches of the network are jointly trained and then tested after convergence.

• **Architecture and Key Components** The Siamese network structure proposed in this paper includes three key components:

    1. **Siamese Framework Construction**: This involves **two parallel identical branches** to process the RGB image and its corresponding depth image simultaneously. The "backbone" of the network consists of an "RGB branch and the depth branch".

    2. **Cross Feature Fusion Block Design**: This block is crucial for integrating the multi-modality data.

        ▪ It is composed of "four Cross Stage Partial (CSP) blocks, three addlayers, four concatenation layers, two upsample layers, and six CBLs".

        ▪ **Three addlayers** are strategically placed after the 2nd, 3rd, and 4th CSP blocks of the Siamese network. These layers extract feature maps from both branches and perform an **additive operation** to fuse the multi-modality data. This additive operation does not change the dimension of feature maps, thus avoiding additional computational burden.

        ▪ Following the fusion, the generated feature maps undergo further processing, including up-sampling and concatenation, to deepen the integration level through "multi-time and multi-size fusions". This block "adjusts the weights of the two branches according to data from the other branch, which strengthens the correlation of multi-modality data".

    3. **CSPDarknet as the Backbone**: **CSPDarknet** is employed as the backbone for both branches of the Siamese network.

        ▪ CSPDarknet is recognized for its successful use in object detection and strong performance.

        ▪ It typically comprises "five convolutional layers and four CSP blocks". The CSP blocks incorporate short-cut connections (similar to Resnet) to mitigate the vanishing gradient problem in deeper networks and reuse feature maps to reduce weight parameters.

    ◦ **Detection Neck and Head**:

        ▪ A **Feature Pyramid Network (FPN)** is used as the "detection neck" to establish connections between multi-scale feature maps, combining low-level and high-level semantic information to enrich the feature maps.

        ▪ A **one-stage detection architecture** forms the "detection head," which simultaneously predicts object classes and locations. The output has three predictions, each with 3(K+5) channels, where K is the number of classes, 4 channels are for bounding box localization, and 1 for objectness prediction score.

• **Contribution to Performance**

    ◦ The use of the Siamese network is a key reason for the algorithm's "superior performance and real-time efficiency".

    ◦ It enables the model to "better learn depth and texture information based on multi-modality data".

    ◦ The experimental results show that the fusion-based algorithm, utilizing the Siamese network, exhibits the "best overall performance" compared to single-RGB-based and single-LiDAR-based algorithms. This improved performance is particularly noted in the "object detection and position regression of small targets, targets at the border of the image, and occluded targets".

    ◦ Its ability to process data efficiently is also highlighted, contributing to the algorithm's "real-time detection" capability, with a running time of "0.03 s per frame".

| **Sl No** | **Neighbor** | **Border Length (km)** | **Forested Regions Along Border** |
| --- | --- | --- | --- |
| 01 | Bangladesh | 4,096.7 | Tripura, Mizoram, Meghalaya, Assam, West Bengal — all have high forest cover |
| 02 | China | 3,488 | Arunachal Pradesh, Uttarakhand, Himachal Pradesh — dense Himalayan forests |
| 03 | Pakistan | 3,323 | Jammu & Kashmir, Punjab, Rajasthan — sparse forests except J&K |
| 04 | Nepal | 1,751 | Uttarakhand, Bihar, West Bengal |