MIMF: Mutual Information-Driven Multimodal Fusion

Zhenhong Zou[1,2], Linhao Zhao[1,2], Xinyu Zhang[1,2(✉)], Zhiwei Li[1,2], Dafeng Jin[1,2], and Tao Luo[3]

[1] State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China
xyzhang@tsinghua.edu.cn
[2] School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China
[3] China North Vehicle Research Institute, Beijing 100072, China

Abstract. In this paper, we propose a novel adaptive multimodal fusion network MIMF that is driven by the mutual information between the input data and the target recognition pattern. Due to the variant weather and road conditions, the real scenes can be far more complicated than those in the training dataset. That constructs a non-ignorable challenge for multimodal fusion models that obey fixed fusion modes, especially for autonomous driving. To address the problem, we leverage mutual information for adaptive modal selection in fusion, which measures the relation between the input and target output. We therefore design a weight-fusion module based on MI, and integrate it into our feature fusion lane line segmentation network. We evaluate it with the KITTI and A2D2 datasets, in which we simulate the extreme malfunction of sensors like modality loss problem. The result demonstrates the benefit of our method in practical application, and informs the future research into development of multimodal fusion as well.

Keywords: Multimodal fusion · Mutual information · Dynamic algorithm · Autonomous driving

1 Introduction

Autonomous driving requires robust models to sense the environment with multiple sensors and generate the perception accordingly, however, though existing multimodal methods can perform well in most scenes, their fusion strategies may fail severely in some abnormal scenarios [1, 2]. For instance, bad weather like rainy and foggy days can put obstacles in the way of camera's work [3]. The sensors themselves also contain potential perception deviations such as the noise in the LiDAR point clouds intensity [4]. In addition to these external and internal problems, there is another common but disturbing trouble in practice, data streams from different sensors do not always match in time due to the hardware limitation [5]. As the result, these problems lead to the uncertainty in data, hence widen the model performance gap between the datasets and real conditions and prevent the application of multimodal fusion methods.

© Springer Nature Singapore Pte Ltd. 2021
F. Sun et al. (Eds.): ICCSIP 2020, CCIS 1397, pp. 142–150, 2021.
https://doi.org/10.1007/978-981-16-2336-3_13

In order to overcome the obstacles, researchers have proposed several approaches to enhance the robustness of models. Some research proposed to select a main modality like images, to guide the fusion detection [5, 6] depending on the prior knowledge of sensors under different conditions, but did not solve the problem yet. Others' work was about specific problems such as foggy [3] and illumination changes [7], which may not be universal for other cases. These methods either focused on a specific issue, or were not real robust models. Instead, Caltagirone et al. proposed to learn an adaptive fusion weight in the LiDAR-camera network [2]. Mario et al. [3] and Yang et al. [8] applied dropout to build adaptive models, while Kim et al. [9] used gate to decide which data to fuse. These solution focus on the balance in multimodal fusion. That is, how to select the proper sensor or feature dynamically, rather than using mixing them in a fixed way?

Inspired by the mutual information (MI) [10, 11], that measures the relation between two variables, people refer to the amount of information in models [11–15]. A network is supposed to reach its best during information acquisition. Therefore, the information maximization equals to the fusion efficiency maximization to some extent. To address efficient usage of MI, some research contributes to the MI estimation in neural network [12, 13]. Based on the previous work Deep InfoMax (DIM) [13], we proposed a novel MI-based data fusion that figures the weight for feature fusion dynamically. The key idea of our work is real-time calculation on the MI value of multimodal features and recognition targets, which further generalize the fusion tendentiousness on them. We build an end-to-end model and examine it on LiDAR-camera fusion lane line segmentation task on the KITTI and A2D2 datasets [16–18].

The rest of the article is organized as follows. In the Sect. 2, we provide the definition the adaptive multimodal fusion problem. In the Sect. 3, we first present the backbone of our LiDAR-camera fusion network, then illustrate the integration of DIM. In the Sect. 4, we present the experiment procedure, the results and discussion.

2 Problem Statement

Let $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]^{\mathrm{T}}$ denotes the data of different modalities, $\boldsymbol{W}$ presents the weight matrix in neural networks, and $\boldsymbol{Y}$ denotes the target to be recognized. In deep learning, we train the model with the optimization goal

$$\widehat{\boldsymbol{W}} := \operatorname{argmin}_{\boldsymbol{W}} \|\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{X}\| + \|\boldsymbol{W}\| \tag{1}$$

where the multiplication contains normal matrix multiplication and Hadamard product. Suppose $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ are coherent information source for recognition, for example, the LiDAR point clouds $\boldsymbol{X}_1$ and camera images $\boldsymbol{X}_2$ provide relative measurements of the same objects, although they are in different domains. For the common and basic fusion, weighted-sum as feature fusion, it can be formulated as,

$$Z(\boldsymbol{X}) = A\boldsymbol{W}_0[\boldsymbol{X}_1, \boldsymbol{X}_2]^{\mathrm{T}} = \boldsymbol{W}_0[\alpha_1\boldsymbol{X}_1, \alpha_2\boldsymbol{X}_2]^{\mathrm{T}} \tag{2}$$

where $\boldsymbol{A} = [\alpha_1\boldsymbol{I}, \alpha_2\boldsymbol{I}]^{\mathrm{T}}$ and $\boldsymbol{W} = \boldsymbol{W}_1\boldsymbol{A}\boldsymbol{W}_0$. Notice that $\boldsymbol{W}_0 = [\boldsymbol{W}_{01}, \boldsymbol{W}_{02}]^{\mathrm{T}}$ is individual for different modalities, that means for $\boldsymbol{W}_0\boldsymbol{X}$ we use Hadamard product, but for $\boldsymbol{W}_{0_i}\boldsymbol{X}_i$ we use both matrix product and Hadamard product.

Then, Eq. 1 is written as,

$$\hat{\boldsymbol{W}} := \text{argmin}_{\boldsymbol{W}} \boldsymbol{Y} - \boldsymbol{W}_1 \boldsymbol{Z}(\boldsymbol{X}) + \boldsymbol{W} = \text{argmin}_{\boldsymbol{W}} \boldsymbol{Y} - \boldsymbol{W}_1 \boldsymbol{A} \boldsymbol{W}_0 \boldsymbol{X} + \boldsymbol{W} \tag{3}$$

Now we consider the computation of $\boldsymbol{A}$. Usually, $\boldsymbol{A}$ is an empirical preset coefficient matrix or is learned from the training data. However, in practical usage, the real-time collected $\hat{\boldsymbol{X}}$ is different from the training set, that indicates the domain-gap between $\boldsymbol{A}\boldsymbol{W}_0\boldsymbol{X}$ and $\boldsymbol{A}\boldsymbol{W}_0\hat{\boldsymbol{X}}$, and constructs a severe bias in fusion. Therefore, the key is to figure out a dynamic adjustment algorithm for $\boldsymbol{A}$. In the following section, we will present how to apply the mutual information to obtain it by $\boldsymbol{A} \sim \text{MI} := \text{I}(\boldsymbol{X}; \boldsymbol{Y})$.

3 MIMF Network

3.1 Multimodal Feature Fusion Network

In this paper, we select the common middle feature fusion (MF) as backbone network, which presents robustness in general tests and is regarded as the balance among early fusion, middle fusion, and late fusion [1]. We use an encoder-decoder architecture, in this way, the network can be easy to modify and compare the performance change. The network comprises two pipelines in the encoder for point clouds and images, with 3 convolutional blocks in both branches. To process more complex features in the images, we replace the convolutional blocks with ResNet-34 blocks except the first one. We fuse the features of two modalities by concatenation when two pipelines merge as shown in the Eq. 2. The information will be mix up in the following convolutional layers. Each convolutional block includes a convolution layer, a batch normalization layer, and a ReLU activation layer. The blocks in decoder distinguish the ones in the encoder for they use transposed convolution to recover the feature maps. In order to better utilize the raw information, we add skip-connection between the encoder layers and decoder layers. In MF, we do not assign a fusion weight, instead, the network learns the adaptive weight. But in MIMF, we embed the DIM module to provide a prior weight that can not only work as regularization, but also avoid influence of bad observations.

3.2 DIM Module for MI Estimation

DIM was proposed by Hjelm et al. [13]. Based on MINE [12], which is regarded as an efficient estimator for mutual information of two feature maps in neural networks. in this paper, we modify the DIM to fit our fusion network. For two variables $\boldsymbol{X}$, $\boldsymbol{Y}$, their mutual information $I(\boldsymbol{X}; \boldsymbol{Y})$ is,

$$I(\boldsymbol{X}; \boldsymbol{Y}) = \Sigma_{x \in \boldsymbol{X}} \Sigma_{y \in \boldsymbol{Y}} p(x, y) \log \frac{p(x,y)}{p(x)p(y)} \tag{4}$$

$$I(\boldsymbol{X}; \boldsymbol{Y}) = D_{KL}(P_{\boldsymbol{XY}} || P_{\boldsymbol{X}} \otimes P_{\boldsymbol{Y}}) \tag{5}$$

where D_{KL} is the KL-divergence. It is defined as:

$$D_{KL}(P \| Q) := E_P \left[\log \frac{\mathrm{d}P}{\mathrm{d}Q} \right] \tag{6}$$

We mark P_{XY} as J, and $P_X \otimes P_Y$ as M. By using the DV-distribution form and nature of KL-divergence, we obtain the lower bound $\hat{I}$ of $I(\boldsymbol{X}; \boldsymbol{Y})$:

$$I(\boldsymbol{X}; \boldsymbol{Y}) \geq \hat{I}(\boldsymbol{X}; \boldsymbol{Y}) = E_J\left[T_\omega(x, y)\right] - \log \mathrm{E}_{\mathrm{M}}\left[\mathrm{e}^{\mathrm{T}_\omega(\mathrm{x},\mathrm{y})}\right] \tag{7}$$

where $T_\omega : \mathrm{x} \times \mathrm{y} \rightarrow \mathbb{R}$ is a function parameterized by ω that can be used in the Eq. 7 to approximate $I(\boldsymbol{X}; \boldsymbol{Y})$. We simply present a sample of T_ω and consider it enough to the expected function [13]. Provided that the multimodal features $\boldsymbol{X}_1, \boldsymbol{X}_2$ have the same dimension, which can be achieved by feature alignment, and the size is (C, H, M, N). C is the number of channels in convolutional layers, and (H, M, N) is the size of a channel. We note the map in each channel as $\boldsymbol{X}_{i_n}$, $n \in [1, \mathrm{C}]$. Therefore, we rewrite the Eq. 7 as:

$$I\left(\boldsymbol{X}_{i_n}; Y\right) \doteq \hat{I}\left(\boldsymbol{X}_{i_n}; \boldsymbol{Y}\right) = \log\left(\Sigma S e^{u-u_{max}}\right) + u_{max} - \log(\Sigma S) - \frac{\Sigma\left(u_{avg} \cdot S\right)}{\Sigma S} \tag{8}$$

where $S = 1 - \overline{S}$, and $\overline{S} = H \times H$ is a diagonal matrix. Besides, $U = X_n \times Y$, and u_{max} is the maximum value in matrix U, while u_{avg} is the average value. Therefore, the mutual information is represented as below, $\mathrm{C} \in \{1, 2\}$ in our model:

$$I(\boldsymbol{X}_i; Y) = \frac{1}{C} \sum_{n=1}^{C} I\left(\boldsymbol{X}_{i_n}; \boldsymbol{Y}\right) \tag{9}$$

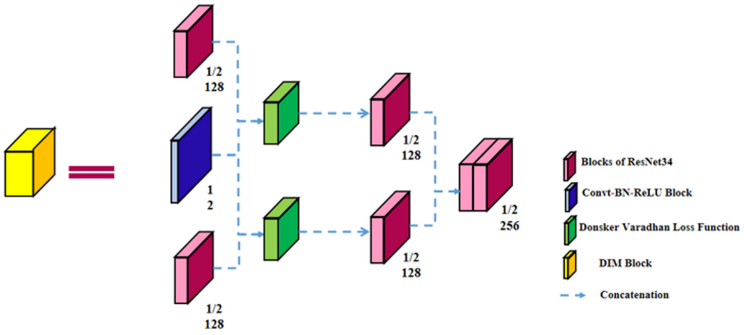


Fig. 1. The structure of DIM block in our fusion network. It adopts features from two modalities, the images and point clouds respectively, as $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, while taking the features from the last convolution layer as $\hat{\boldsymbol{Y}}$. Then, DIM block figures out the mutual information $I\left(\boldsymbol{X}_i, \hat{\boldsymbol{Y}}\right)$.

3.3 Mutual Information-Driven Multimodal Fusion

Recurrent Training Process. In the Sect. 3.2, we present how to compute the MI. To apply it in the weighted-fusion model, we integrate it into the network and training-testing procedure as well. As shown in Fig. 1, we take two branches in the DIM block, which are for two data. DIM block computes the mutual information between them and the expected feature $\boldsymbol{Y}$ respectively. However, we cannot obtain $\boldsymbol{Y}$ ahead of the network

computation. Instead, we make the time-continuity assumption: for each i, $\boldsymbol{X}_i$ is a given stable time sequence, that means $\boldsymbol{X}_i^t \approx \boldsymbol{X}_i^{t-1}$ and $\boldsymbol{Y}^t \approx \boldsymbol{Y}^{t-1}$. In autonomous driving, that indicates two frames of a sensor observations are similar because of the continuity of scenarios and events. With this assumption, when we acquire a well-trained model in test, we can treat the recognition of last frame as an approximation of the target at current time, especially in a sequence model. Obviously, the fault rate of the last recognition will be enlarged. But when DIM is integrated into a robust backbone, we can ignore it in most time, and use a reset strategy to reduce the cumulative error. However, we have only implemented a single-frame recognition model and lack enough time-series data, thus we simply use compute the current data cyclically. Specifically, we compute on it for the first time to simulate the 'last frame result', and use it in the second computation. Therefore, we finish a DIM process approximately in testing. The Fig. 2 presents the overall structure of MIMF. The yellow block is the DIM module, and the rest is the MF baseline. The RGB images and point clouds are processed in two separated pipelines in the encoder, and get fused in the DIM module. The sizes of feature maps are not changed in DIM. That means DIM is flexible for most models. When DIM outputs the $I\left(\boldsymbol{X}_i, \hat{\boldsymbol{Y}}\right)$ as above, we normalize them by

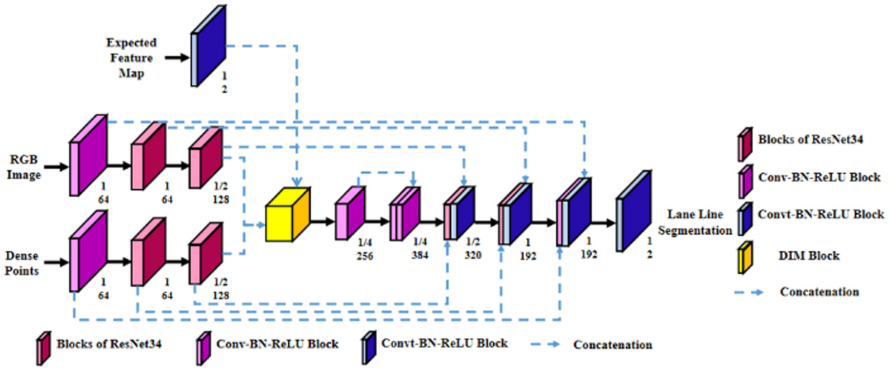


Fig. 2. The overview of the architecture of MIMF. It comprises a standard feature fusion in the middle of an encoder-decoder network, and the DIM block during fusion. Before fusion, MIMF has two individual pipelines to process different modal data.

$$\alpha_i = \frac{I\left(\boldsymbol{X}_i, \hat{\boldsymbol{Y}}\right)}{\Sigma I\left(\boldsymbol{X}_i, \hat{\boldsymbol{Y}}\right)} \tag{10}$$

MI as Fusion and Regularization. With Eq. 10 we obtain the fusion weight $\boldsymbol{A}$ in Eq. 2. As a prior knowledge of the target of the tasks, MIMF pre-fetches data with a bias. It further forces the fusion models to focus on more relevant information in testing. The bias makes it unwilling to get affected by the fault measurements or information loss data in complex scenes in practice. In addition, we observe that MIMF performs better on the normal data. We explain the result with the random regularization effect which is similar with the dropout. As MI is independent to the network or data, instead, it is

determined by both the data and target simultaneously, it will be treated as a random process under a distribution different from those of the noised data. Therefore, by learning the data-independent input, the network avoids over-fitting the data. Notice our method can only operate the case when as least one modality has good observation. Otherwise, the dominated data will lead to serious problem.

4 Experiment

4.1 Dataset and Metrics

To evaluate our models, we select pictures by ignoring the roads with intersections or without forward lines. Finally, we pick up around 400 data pairs from the KITTI road detection track [17], and around 1000 pairs from the A2D2 dataset [18]. We use 60% of the data as the training set, 10% for validation and rest for testing. The image resolution is 1242 × 375 in KITTI, and 1920 × 1208 in A2D2. KITTI uses a 64-line Velodyne to generate point clouds, but A2D2 combines one 8-line and two 16-line LiDARs. The difference in LiDARs causes the gap in performance, but it would not matter in the evaluation of the adaptive fusion. Because KITTI dataset has no lane line labels, we add pixel-level annotation to it by hand. Labeled lines are supposed to be not only parallel to the driving direction but also on the driving area. To reduce noise in the annotation, we do not estimate any markings, behind obstacles like vehicles and poles on the roadside. Different from KITTI, A2D2 provides similar lane line labels but they ignore the intervals in dash lines.

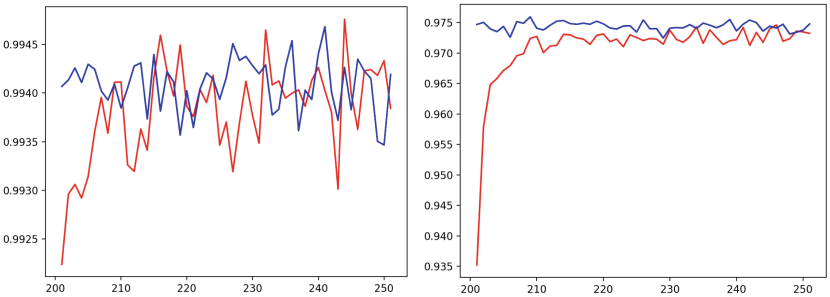
We focus more on the recall of lane line and compute it as the lane accuracy. We also consider the F2-score to balance in case the network over fits any class, and count the mean recall on both class as the *mAcc*.

Implementation and Training. To integrate LiDAR point clouds and RGB images in the same network, projection and value normalization are essential in preprocess. To project the point clouds onto the image plane, given a point $P_v = \left(\mathrm{x}_v;\ \mathrm{y}_v;\ \mathrm{z}_v\right)^{\mathrm{T}}$, we calculate:

$$Pv = K_v[R_v; T_v]P_v \tag{11}$$

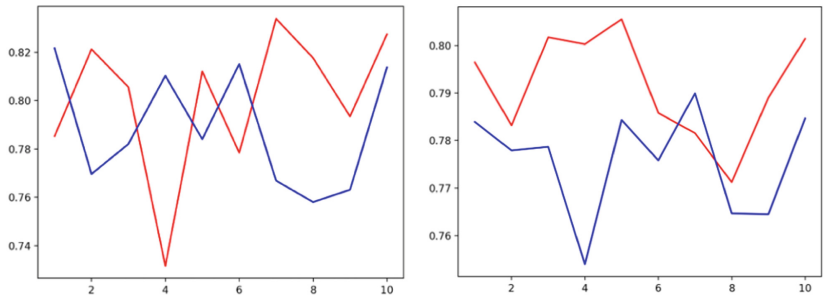
where Kv; Rv; Tv refer to the camera calibration matrix, rotation matrix and translation matrix respectively. Then the projected front-view point cloud reflectance map will be cropped to the same size of RGB images at 128 × 256. After that, the value of both reflectance map and the RGB images will be normalize to [0, 1] interval. After data pre-processing, we train our model for 250 epochs on two datasets respectively. As shown in the Sect. 3, we generate the simulated ground truth of target features in the first round of training, in which we also get the result of original MF model. Then in the second round, we train the MIMF with the features. In testing, we use the pre-trained MF, just as the procedure in training, to get target features, and test MF and MIMF.

Result and Analysis. We present the training record in the Fig. 3, and the result of testing in the Fig. 4. Note that we only put the training record of the last 50 epochs in the figures, in which we can see the MIMF performs worse than MF at first, but they converge together at last, that indicates the random disturbance from the independent mutual information. However, in testing, we observe that MIMF performs better than MF by 1–2%, which indicates the potential regularization function of MI-driven models on the small training data. Note that due to the unknown unstable calculation in MIMF, we have different output in testing, for which we process it tenth and count the average value. Though the result is not deterministic, the DIM in MIMF output the stable fusion weight, which is $1.25 : 0.75$ for images and point clouds fusion in normal data in KITTI. For A2D2, the ratio is $1.35 : 0.65$, and that meats the prior in dataset when we declare that the LiDARs in A2D2 is not so suitable for segmentation tasks. We further complete the modality loss on the KITTI dataset. With a prior knowledge of the MI of each sensor, MIMF can keep the performance elimination in an acceptable range, while MF only recall 50.29% of the lane pixels, far less than the result on normal data.



(a) (b)

Fig. 3. The comparison of training process between the baseline and MIMF on the KITTI and A2D2 datasets. The blue lines are the MF baseline and the red are the MIMF. The lines present the accuracy during training, which finally converge together. The X-axis indicates the epochs. (Color figure online)



(a) (b)

Fig. 4. The comparison of testing process between the baseline and MIMF on the testing datasets. The blue lines are the MF baseline and the red are the MIMF. The X-axis indicates the epochs. (Color figure online)

5 Conclusion

In this paper, we propose a novel adaptive multimodal fusion network named MIMF. It is driven by the mutual information between the input data and the target recognition patterns. By leveraging the mutual information in fusion, our weight-fusion module is able to perform adaptively based on the variant data. We further observe the regularization effect of our MI-driven method. The evaluation result on the KITTI and A2D2 datasets demonstrates the benefit of our method in practical application. In the following research, we will complete the experiments on more complex segmentation tasks and integrate a more flexible MI-estimator will better real-time processing procedure.

Acknowledgements. This work was supported by the National High Technology Research and Development Program of China under Grant No. 2018YFE0204300, the Beijing Science and Technology Plan Project No. Z191100007419008, the Guoqiang Research Institute Project No. 2019GQG1010, and the National Natural Science Foundation of China under Grant No. U1964203.

References

1. Feng, D., et al.: Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges. IEEE Trans. Intell. Transport. Syst (2019)
2. Caltagirone, L., Bellone, M., Svensson, L., Wahde, M.: LIDAR-camera fusion for road detection using fully convolutional neural networks. Robot. Autonom. Syst. **111**, 125–131 (2019)
3. Mario, B., et al.: Seeing through fog without seeing fog: deep multimodal sensor fusion in unseen adverse weather. In: CVPR (2020)
4. Carballo, A. et al.: LIBRE: The multiple 3D LiDAR dataset. ArXiv, abs/2003.06129
5. Vora, S., Lang, A., Helou, B., Beijbom, O.: PointPainting: sequential fusion for 3D object detection. In: CVPR (2020)
6. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.: Frustum PointNets for 3D object detection from RGB-D data. In: CVPR (2018)
7. Su, Y., Gao, Y., Zhang, Y., Álvarez, J.M., Yang, J., Kong, H.: An illumination-invariant nonparametric model for urban road detection. IEEE Trans. Intell. Vehicles **4**, 14–23 (2019)
8. Yang, B., Liang, M., Urtasun, R.: HDNET: exploiting HD maps for 3D object detection. In: CoRL (2018)
9. Kim, J., Koh, J., Kim, Y., Choi, J., Hwang, Y., Choi, J.W.: Robust deep multi-modal learning based on gated information fusion network. In: ACCV (2018)
10. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
11. Gabrié, M., et al.: Entropy and mutual information in models of deep neural networks. In: NeurIPS (2018)
12. Belghazi, M.I., et al.: Mutual information neural estimation. In: ICML (2018)
13. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization (2019)
14. Bramon, R., et al.: Multimodal data fusion based on mutual information. IEEE Trans. Visual. Comput. Graph. **18**, 1574–1587 (2012)
15. Yousef, A., Iftekharuddin, K.: Shoreline extraction from the fusion of LiDAR DEM data and aerial images using mutual information and genetic algorithms. In: IJCNN (2014)

16. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: spatial CNN for traffic scene understanding. In: AAAI (2018)
17. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR (2012)
18. Geyer, J., et al.: A2D2: Audi autonomous driving dataset. ArXiv, abs/2004.06320