Enabling Efficient Deep Convolutional Neural Network-based Sensor Fusion for Autonomous Driving

Xiaoming Zeng*, Zhendong Wang*, {xiaoming.zeng, zhendong.wang}@utdallas.edu,
Electrical and Computer Engineering Department, The University of Texas at Dallas, Richardson, US
Yang Hu, yang.hu4@utdallas.edu, hy6356@gmail.com,
ECE Department, The University of Texas at Dallas, US, School of Integrated Circuits, Tsinghua University, China
[1] These two authors contributed equally.

***Abstract*—Autonomous driving demands accurate perception and safe decision-making. To achieve this, automated vehicles are typically equipped with multiple sensors (e.g., cameras, Lidar, etc.), enabling them to exploit complementary environmental contexts by fusing data from different sensing modalities. With the success of Deep Convolutional Neural Network (DCNN), the fusion between multiple DCNNs has been proved to be a promising strategy to achieve satisfactory perception accuracy. However, existing mainstream DCNN fusion strategies conduct fusion by simply element-wisely adding feature maps extracted from different modalities together at various stages, failing to consider whether the features being fused are matched or not. Therefore, we first propose a feature disparity metric to quantitatively measure the degree of feature disparity between the fusing feature maps. Then, we propose a Fusion-filter as the Feature-matching techniques to tackle the feature-mismatching issue. We also propose a Layer-sharing technique in the deep layer of the DCNN to achieve high accuracy. With the assistance of feature disparity working as an additional loss, our proposed technologies enable DCNN to learn corresponding feature maps with similar characteristics and complementary visual context from different modalities. Evaluations demonstrate that our proposed fusion techniques can achieve higher accuracy on KITTI dataset with less computation resources consumption.**

***Index Terms*—Sensor Fusion, DCNN, Feature-matching, Autonomous Driving**

I. Introduction

The era of driving automation is coming. The safety of an automated vehicle hinges crucially upon the accuracy of perception. Therefore, many studies [1]–[4], [7], [8] have employed multi-modal sensors such as cameras and LiDARs that can provide complementary sensing information to deliver better and robust perception performance. In this paper, we focus on the free road segmentation since it is a cornerstone module among all autonomous driving tasks. It is a typical application that benefits from such multi-modal sensing technology as shown in Fig.1. In this multi-modal sensor fusion setup, both RGB images(captured from cameras) and depth images(pre-processed from 3D point cloud collected by LiDAR) are employed, as depicted in Fig.1 (a) and Fig.1 (b) respectively. We can observe that the RGB and depth images are a pair of interpretations of the same scene at the same moment, providing an opportunity to exploit them for a better perception accuracy. In contrast, employing only one sensing modality often fails the task in some driving scenarios, for example, using only RGB camera under unfavorable lighting conditions such as dark night, overexposure, and etc.

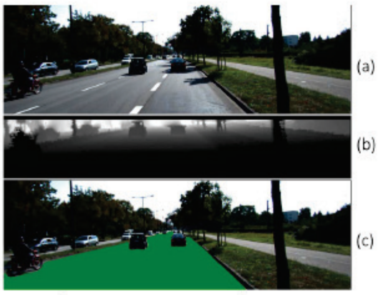


Fig. 1. The free road segmentation result by fusion. (a) RGB input image. (b) Depth input image. (c) segmentation result. The free (drivable) road is represented in green pixels.

Among these fusion architectures, though there is no conclusive evidence that one fusion method is absolutely better than others, the middle fusion method with element-wise summation [1]–[4] is the dominant method in recent works as we can notice in the KITTI benchmarks [6]. The reason is two-fold. First, the middle fusion architecture often adopts crossing connections between network branches in all fusion stages (i.e., from shallow to deep layer), and these crossing connections can learn characteristics from the training data, that where and to what degree the integration should be carried out [1], and thus providing better accuracy compared to the early [7] and late fusions [8] architectures. Second, fusion operations can be achieved by executing element-wise summation between corresponding intermediate feature maps, which is easy to be implemented.

Although it is a fact that middle fusion with element-wise summation provide the best accuracy to date, we frustratedly observe that existing solutions all fail to consider the *intrinsic relationship between two sets of feature maps to be fused.* Specifically, naively conducting the element-wise summation of intermediate feature maps from multi-modal sensing data can lead to such a scenario, where a sensor A's feature map A_1 that represents the feature X is fused with a sensor B's feature map B_1 that represents the feature Y, since A_1 and B_1 is product of convolution using different filters. the fusion

DAC '22, July 10–14, 2022, San Francisco, CA, USA

ACM ISBN 978-1-4503-9142-9/22/07…$15.00
https://doi.org/10.1145/3489517.3530444

of mismatched features can generate meaningless information that might harm the accuracy and eventually threat the driving safety.

In this work, we set out to explore the impacts of feature matching to the accuracy of sensor fusion. Specifically, we propose a ***feature disparity (FD)*** metric to quantitatively describe the differences between features from different sensing modals. Consequently, we argue that in middle fusion the feature maps to be fused should possess similar visual characteristic with complementary content, that being said, with low feature disparity. To achieve this goal, we first propose ***Fusion filter*** that learns the feature-matching relationship between the feature maps to be fused to guarantee the feature matching. Second, we propose ***Layer-sharing*** network architecture which allows the deep layers to share the same filters based on our observation that features processed in deep layers tend to be similar and the feature matching is preserved. Besides, we utilize the feature disparity metric as an additional sub-objective loss function(i.e., ***Feature Disparity Loss***) to further constrain two sub-network branches to learn similar features during the training stage. This will not be executed in the inference stage, hence does not affect the inference latency.

Finally, we implement comprehensive evaluations on models with our proposed techniques on the KITTI dataset [6], and demonstrate that the feature-matching techniques can effectively reduce the feature disparity between the feature maps to be fused and achieve better accuracy than that of the state-of-the-art RoadSeg [3] adopting the naive fusion. Meanwhile, the Layer-sharing technique can effectively reduce the computational overhead of the fused network with a comparable or better accuracy compared to the RoadSeg.

To summarize, this paper makes the following contributions:

- To the best of our knowledge, we are the first to identify the feature-mismatching issue while performing element-wise fusion in a DCNN-based middle fusion method. Accordingly, a ***feature disparity metric*** is proposed to quantitatively measure the degree of feature deviation between feature maps to be fused. Feature disparity is then adopted as a individual loss in addition to the baseline segmentation loss, enabling DCNN to learn similar features with complementary content extracted from different network branches.
- We propose the technique, ***Fuse-filter***, to address the feature-mismatching issue when fusing two independent DCNN feature maps at various stages. Furthermore, we propose the ***Layer-sharing*** network architecture which allows the deep layers to share the same filters in the fused network so that the feature matching property is preserved.
- Our evaluation results among models equipped with different proposed fusion schemes on KITTI road dataset reveal that our proposed ***Fuse-filter*** can achieve better accuracy, while ***Layer-sharing*** can obtain comparable accuracy with less computational resources demand on KITTI dataset.

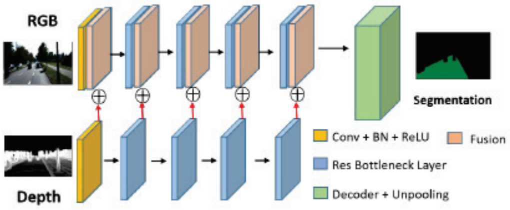


Fig. 2. Architecture of baseline RoadSeg.

II. Background and Challenges

A. DCNN-based Sensor Fusion

As a representative subset of DNN architectures, the Deep Convolutional Neural Network(DCNN) is widely applied in vision-related tasks [11]–[13]. Typically, DCNN is a stacked structure composed of multiple layers [5]. From shallow to deep, convolutional layers can hierarchically extract embedded visual features by following the sliding-window method from traditional image processing [9]. Specifically, each set of filters slides over the input image and performs convolution operation to produce the output feature map with a certain characteristic corresponding to the content inside the sliding filter that can perform edge extraction, image sharpening, etc.

As a mainstream sensor fusion method, the DCNN-based fusion architecture leverages multiple separate neural networks to process the data from the multiple independent sensors as the perception approaches using single-modality data (e.g., either the camera data or the LiDAR data) can fail the task. For instance, the LiDAR point cloud data is sparse and without the fine texture of the object being scanned upon, using only LiDAR data might lead to unfavorable perception accuracy.

In this work, We target the **free road segmentation** task, which is crucial to the driving automation system to distinguish the free drivable road from the surroundings. We adopt **RoadSeg** [3] as the baseline. It is the state-of-the-art open-source DCNN architecture with a middle fusion element-wise summation scheme between the separate RGB and Depth network branches, which ranks at the top of KITTI road segmentation task. RoadSeg also adopts the popular encoder and decoder architecture with ResNet [14] structure being the backbone network, as shown in Fig. 2. Specifically, an RGB encoder and a Depth encoder are employed as the two branches to extract the features from RGB and Depth channels, respectively. Meanwhile, at each fusion stage, the extracted RGB and depth features are fused via the element-wise summation operation. Then, the fused feature maps from the encoder are fed to the decoder to generate the final drivable road segmentation result.

B. Feature Disparity Assessment

Feature disparity assessment [15] is a fundamental process used in a variety of traditional computer vision tasks, and it mainly compares the feature disparity between two correspondent images. The most standard feature disparity measure is using the L2 metric, which naively compares the pixel-level value difference between two images.

TABLE I
FEATURE DISPARITY METRIC COMPARISON

Feature disparity metric	Spatial information	luminance disparity
MI, Cross-bin	✕	✕
SSIM	✓	✕
Feature Disparity	✓	✓

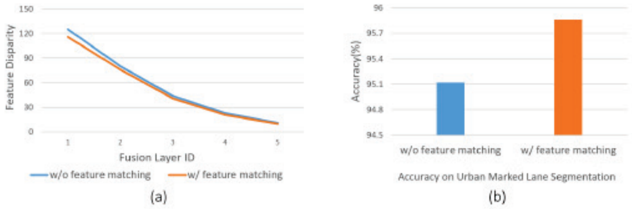


Fig. 3. (a)Feature disparity between two sets of feature maps to be fused at different fusion layers.(b) Corresponding accuracy performance evaluated with and without feature-matching technique.

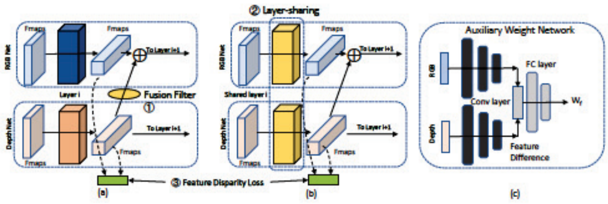


Fig. 4. (a) Feature-matching technique: Fusion-filter, which is denoted as a yellow oval box. (b) Layer-sharing method. The shared layer between two network branches is presented as a yellow square. (c) Auxiliary Weight Network.

Beyond the standard L2 metric, several other methods to measure disparity between features are proposed in traditional computer vision field [15]–[18] as shown in first two rows of Table I. The representative feature disparity metrics such as mutual information (MI) [17] and cross-bin [18] mainly focus on the statistical pixel-level mean and variation in luminance, lacking the spatial information embedded in the feature map. Addressing the lack of spatial information is deemed as the major feature in our task. Although the structural disparity measure (SSIM) [16] takes the structural information into account, it favors two images to be similar in terms of pixel-level intensity in luminance all across the image, which is not applicable to our case.

C. Problem Definition

As we focus on the DCNN-based middle fusion architecture with the element-wise summation technique, the fusion is carried out by directly element-wisely adding the intermediate feature maps from different modalities. Due to the fact that the feature maps to be fused are extracted by the two separate DCNN branches which use their own associated filters, there are great chances that the feature maps to be fused possess mismatched characteristics with each other. Accordingly, simply element-wisely summing the feature maps can cause the chaotic and unfavorable results.

In order to quantitatively measure the degree of feature deviation between the feature maps to be fused, our concern is two-fold. Firstly, the spatial features from the two sets of feature maps to be fused should be represented and compared. Secondly, the pixel-level difference in luminance of two feature maps is anticipated as the two feature maps are obtained from two different sensing modalities. For example, for the same driving scene, the RGB image obtained from camera may be darker during the night, while depth image converted from LiDAR point cloud will not be affected by the light condition. In this case, the overall pixel-level luminance differs in the two images, and the previous metrics would fail since they are sensitive to this pixel-level difference in luminance of the two images. Therefore, we choose the edge information as the representative feature of each intermediate feature map, and conduct comparison between the extracted edges. Because the edge sketches by nature can well preserve the spatial information and they can be identified as long as there exists pixel-level difference on different objects.

By borrowing the edge detection [19] idea from traditional computer vision field, we adopt opencv edge detection library [10] to extract the edge sketches of each feature map, then the Feature Disparity will be obtained by conducting comparison between the extracted edge from their corresponding feature maps, which is formally described in Eq.1.

$$\mathcal{D}_{fd} = \frac{1}{C}\sum_{i=0}^{C}[\mathcal{E}_i(f_{Rc}) - \mathcal{E}_i(f_{Dc})]^2 \tag{1}$$

Note that the channel-wise edge extraction process, denoted as $\mathcal{E}$, will be first performed on feature maps from both RGB branch(f_{Rc}) and Depth branch(f_{Dc}), then the pixel-level feature disparity will be obtained across all the corresponding channels. C represents the total number of channels.

Fig.3 (a) shows the result of our proposed feature disparity metric applied on the intermediate RGB and Depth feature maps at five different fusion stages over ten randomly selected RGB and Depth input-image pairs. We observe that the feature mismatch between corresponding RGB and Depth feature maps exists at all fusion layers, and it gets less significant as the fusion layer gets deeper. The blue line denotes the raw feature disparity in the baseline model, and after applying our proposed feature matching technique in next chapter, the feature disparity can be reduced as the orange line shows. accordingly, a better accuracy performance is gained as shown in Fig.3 (b). In addition, we learn that high-level feature maps tend to hold similar features, offering an opportunity to share the feature-extracting filters residing in deep layers between two network branches. By sharing these deep layers, the model's computational overhead in terms of total number of parameters can be reduced (see Sec. III).

III. DESIGN

To address the aforementioned feature-mismatching issue, we first present a feature-matching technique by leveraging **Fusion-filter** to guarantee that two feature maps to be fused at each fusion stage possess similar characteristics with complementary contents (① in Fig.4 (a)). Second, we observe that the high-level feature maps from two fusion branches often carry similar characteristics, a **Layer-sharing** method is proposed to reduce the computational overhead and network parameter volume (② in Fig.4 (b)). Third, we apply the **Feature disparity loss** in addition to the baseline segmentation loss function to further make the model learn filters that can extract analogous features from each modality (③ in Fig.4 (a) and (b)).

A. Feature-matching Utilizing Fusion-Filter

As we discussed in Sec. II-C, the best fusion paradigm would be the feature maps holding similar characteristics with complementary content. To achieve this goal, we introduce Fusion-filter technique to address the feature-mismatching issue, which target learning the feature-matching relationship between two sets of feature maps to be fused from the training data.

Without the loss of generality, we introduce the Fusion-filter on top of the baseline architecture. As shown in Fig. 4(a), the Fusion-filter, presented as a yellow circle between two separate neural networks, is employed before the stage where the Depth feature maps are element-wisely summed with RGB feature maps. The usage of Fusion-filter can be described as Eq. 2. At the fusion stage i, the Depth feature maps f_{Di} are firstly convoluted with the corresponding fusion filter W_f. This convolution process is denoted as F_f. Secondly, the resulting intermediate RGB feature maps $f_{Ri}^{'}$ is updated by the summation between original RGB feature maps f_{Ri} and the aforementioned convolution result.

$$f_{R_i}^{'} = f_{R_i} + F_f(f_{D_i}; W_f) \tag{2}$$

The Fusion-filter is designed to reconstruct the Depth feature maps by conducting a convolution with Fusion-filter W_f, which is capable to learn the matching relationship from Depth to RGB feature maps from the training data. Note that the kernel size of the fusion filter is 1×1, since it only aims at reorganizing the mapping relationship between those two sets of feature maps. On the other hand, the extra memory access for Fusion-filter parameters and corresponding convolution calculation will be introduced (see Sec. IV-B).

B. Leveraging Layer-sharing Method

As Fig. 3 (a) shows, with layer going deeper, the feature disparity between the two sets of feature maps significantly decreases, which offers us a great opportunity to propose a Layer-sharing method that allows the two fused networks to share filters in deep layers while leaving shallow layers stay the same. Fig. 4 (b) shows an example of sharing convolution layer (indicated by the yellow rectangle) between the two networks. That is, the Depth and RGB network branches share the same convolution filters but process their own data independently.

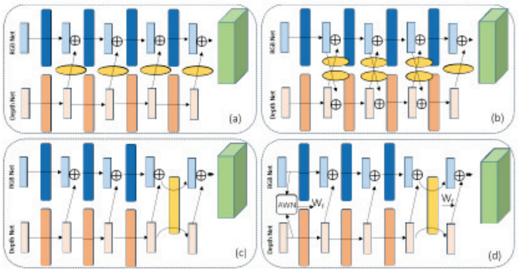


Fig. 5. Models with different fusion schemes: (a)AllFilter_U, (b)AllFilter_B, (c)BaseSharing, (d)WeightedSharing.

Furthermore, after analyzing the baseline architecture, we find that in non-shared architecture, there is an implicit weight embedded in each set of filters from two network branches when the fusion is carried out. Therefore, we propose to apply an **Auxiliary Weight Network (AWN)** (indicated by w_f in Fig. 4 (c)) on the resulting feature maps after the layering sharing, considering that the feature maps of two network branches are extracted from different sensing modalities and carry different weights.

Specifically, we first adopt the same convolution layers to extract the high level feature maps from the two input sensing modalities. Then, the difference of the two sets of extracted high level feature maps will be fed into a stacked full-connected layer to generate the auxiliary weight parameter W_f, which represents the weight of feature maps in Depth Network branch when it is fused with its counterpart RGB feature maps, as depicted in Fig.5 (d).

C. Feature Disparity in Objective Loss

Besides the objective loss function (i.e., segmentation loss), we further utilize the feature disparity metric as an additional sub-objective loss function (i.e., Feature Disparity Loss) to further constrain the model to gradually extract features with the same characteristics yet complementary contents from its own sensing modality. Therefore, two modalities can together reach a consensus on what they are perceiving, a better resulting accuracy performance can be reaped (see Sec.IV-B). With the Feature Disparity Loss included (indicated by the green box in Fig. 4 (a) and (b)), the overall objective loss function can be formally described as follows,

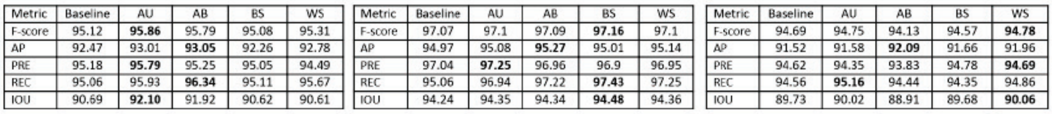
$$\mathcal{L}_{loss} = \mathcal{L}_{Segmentation} + \alpha \sum_{i} \mathcal{D}_{fd-i} \tag{3}$$

Apart from segmentation loss $\mathcal{L}_{Segmentation}$, we formulate the Feature Disparity Loss $\mathcal{D}_{fd-i}$ at fusion stage i by comparing the edge characteristic between the two sets of the feature maps instead of comparing feature maps directly. Moreover, a tuning knob α is assigned to the feature disparity loss to decide how much it weighs in the overall loss function. Note that the proposed loss function will only be applied during the training process, thus it does not increase the inference latency.

IV. Evaluation

A. Experimental Setup

1) Training Environment: The models proposed are trained and tested on a single NVIDIA's Quadro RTX 8000 GPU



Metric	Baseline	AU	AB	BS	WS
F-score	95.12	**95.86**	95.79	95.08	95.31
AP	92.47	93.01	**93.05**	92.26	92.78
PRE	95.18	**95.79**	95.25	95.05	94.49
REC	95.06	95.93	**96.34**	95.11	95.67
IOU	90.69	**92.10**	91.92	90.62	90.61

(a) UM

Metric	Baseline	AU	AB	BS	WS
F-score	97.07	97.1	97.09	**97.16**	97.1
AP	94.97	95.08	**95.27**	95.01	95.14
PRE	97.04	**97.25**	96.96	96.9	96.95
REC	95.06	96.94	97.22	**97.43**	97.25
IOU	94.24	94.35	94.34	**94.48**	94.36

(b) UMM

Metric	Baseline	AU	AB	BS	WS
F-score	94.69	94.75	94.13	94.57	**94.78**
AP	91.52	91.58	**92.09**	91.66	91.96
PRE	94.62	94.35	93.83	94.78	**94.69**
REC	94.56	**95.16**	94.44	94.35	94.86
IOU	89.73	90.02	88.91	89.68	**90.06**

(c) UU

Fig. 6. Accuracy performance of proposed models under different road scenes in tables. In order to adapt the paper width, we term the architecture AllFilter_U as AU, AllFilter_B as AB, BaseSharing as BS and WeightedSharing as WS respectively. The best model is highlighted under each metric.

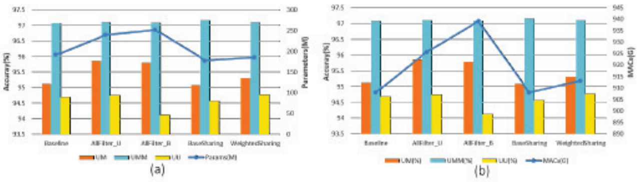


Fig. 7. Model performance in terms of accuracy, total number of MACs and parameters.

with CUDA 10.0 and PyTorch 1.1. As discussed in II-A, we adopt RoadSeg [3] as our baseline, and the final objective loss is comprised of two parts as shown in Eq. 3. From our experimental experience, we empirically set α to 0.3.

2) Dataset and Metrics: KITTI road dataset [6] is widely used for autonomous driving research as a benchmark. It contains 289 image pairs(RGB image and Depth image) for training and 290 image pairs for testing, where within each pair there both containing three different road scene categories including urban marked roads (UM), urban multiple marked lanes (UMM), and urban unmarked roads (UU). Note that the Depth images are generated from lidar point-cloud data by utilizing the pre-processing method proposed in the baseline [3]. Generally, there are four common metrics for performance evaluation: F-score, AP, PRE, REC, and IOU [3]. For fair evaluation, KITTI does not provide ground-truths for testing images, and the final segmentation results of testing images would be converted to a bird's eye view before submitting to KITTI evaluation server.

B. Performance of Our Proposed Models

With our proposed feature matching technique and layer-sharing method, various architectures can be derived from combining different fusion schemes with the baseline. The corresponding diagrammatic presentation can be found in Fig.5. In Fig.5 (a), we apply unidirectional Fusion-filter from Depth branch to RGB branch at each fusion stage and term this architecture as **AllFilter_U**. Similarly, we call architecture from the Fig.5 (b) as **AllFilter_B** for applying bidirectional Fusion-filter. Fig.5 (c) is termed as **BaseSharing** as the last convolutional stage been shared between two branches. **WeightedSharing** in figure (d) is named for the Auxiliary Weight Network(AWN) applied on top of BaseSharing. Note that the yellow oval box represents Fusion-filters and the yellow square box represents the shared layer, and the Auxiliary Weight Network is presented in a white box in Fig.5 (d). We experiment with these models first by training them on the GPU platform and then evaluate the trained model with testing images. Note that for the baseline model, we can achieve as best as 95.12% for UM, 97.07% for UMM, and 94.69% for UU in our experimental environment respectively, which is lower than reported in [3].

1) Accuracy and Corresponding Computational Overhead: As we can see in tables of Fig. 6, the overall accuracy performance of all metrics are presented under three evaluation road scenes. Generally, our proposed models perform better than baseline nearly in every metric under different road scenes. However, the accuracy improvement of our proposed models under UU is less significant, since UU is the most challenging one. Among all metrics, AllFilter_U performs better than others in UM, the same as BaseSharing in UMM, and WeightedSharing in UU category while they all top three metrics out of five as shown in tables in Fig. 6.

For models with the Fusion-filter technique, we have evaluated two types of models, **AllFilter_U** and **AllFilter_B**. By taking advantage of the unidirectional fusion-filter, AllFilter_U is able to carry out fusion with more matched feature maps, which is witnessed as the orange line shown in Fig.3 (a). It outperforms baseline in every category and tops three accuracy metrics out of five in UM category, one in UM and one in UU. With bidirectional Fusion-filter, AllFilter_B outperforms baseline in UM and UMM category in nearly every metric. The reason might be with the help of Fusion-filters across two branches, more balanced segmentation results can be obtained. However, with the introduction of Fusion-filters, higher computational overhead in terms of MACs and parameters is introduced as we can see in Fig. 7.

For the Layer-sharing method, both **BaseSharing** and **WeightedSharing** are evaluated. BaseSharing achieves three best accuracy performance out of five metrics in UMM road scene category with least computational cost as observed from Fig. 6 and Fig. 7. And after AWN is applied on top of BaseSharing, WeightedSharing outperforms Baseline in all three road scenarios and all metrics, and it even achieves the best performance under three metrics in the challenging UU scenario among all the proposed models as the introduced weight parameter can dynamically adjust the weight from one network branch to the other based on different input during fusion. Moreover, WeightedSharing still carries less model parameters than Baseline as shown in Fig.7.

2) Ablation Study for Feature Disparity Loss: We conduct ablation evaluation on our proposed models with Feature Disparity Loss, as shown in Fig. 8. Three typical architectures (Baseline, AllFilter_U and BaseSharing) are evaluated using the same KITTI dataset. Specifically, Baseline, AllFilter_U

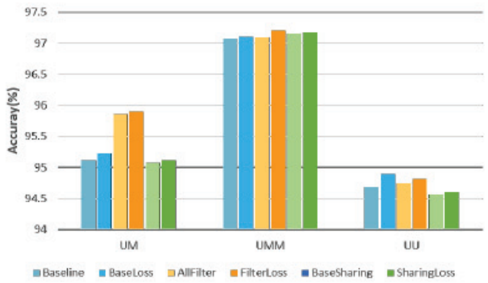


Fig. 8. Ablation study for Feature Disparity Loss. The "loss" indicates the Feature Disparity Loss plus segmentation loss.

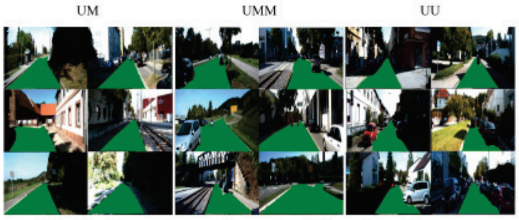


Fig. 9. Example results under different road scenes.

and BaseSharing are trained with only segmentation loss while BaseLoss, FilterLoss and SharingLoss are trained with both segmentation loss and the proposed Feature Disparity loss. As shown in the figure, architectures trained with Feature Disparity Loss outperforms the same architecture trained without it. This result proofs the efficacy of the concept of feature-matching. Especially, for Baseline trained with Feature Disparity Loss, BaseLoss even achieves the best accuracy in UU road scenario under F-score metric.

3) Qualitative Samples: Fig. 9 shows some qualitative results of model-AllFilter on the test set of the KITTI benchmark. Three road scenes(UM, UMM, UU) are presented from left to right. To prove the robustness of our model, we deliberately select the road scenes with over-exposure or shadows. From these samples, we observe our model is robust to these adversarial lighting conditions on road.

CONCLUSION

In order to achieve higher perception accuracy, we investigate the various DCNN fusion architectures, and we identify that feature-mismatching issue exists in directly element-wisely adding feature maps from different sensing modalities, which is commonly adopted by most DCNN middle fusion methods. To tackle this issue, we first propose a feature disparity metric that quantitatively measures the degree of feature disparity between feature maps to be fused. After identifying the feature-mismatching issue exists in conventional element-wise middle fusion architectures, we propose a Feature-matching technique, Fusion-filter, to address the issue. We further learn that feature maps in deeper layer appears to possess less different features, Layer-sharing method is thus proposed to reduce the model's computational overhead. Together with the Feature Disparity Loss, the proposed models can learn corresponding features from different modalities to achieve higher accuracy. Experimental results demonstrate that our proposed Feature matching techniques can achieve better accuracy than the baseline, and the Layer-sharing method achieves comparable accuracy with less computational overhead compared to the baseline.

ACKNOWLEDGMENT

This work is supported in part by NSF grants CCF 1822985, and CCF 1943490 (CAREER). Yang Hu is the corresponding author.

REFERENCES

[1] Caltagirone, L., Bellone, M., Svensson, L. and Wahde, M., 2019. LIDAR–camera fusion for road detection using fully convolutional neural networks. Robotics and Autonomous Systems, 111, pp.125-131.

[2] Chen, Z., Zhang, J. and Tao, D., 2019. Progressive lidar adaptation for road detection. IEEE/CAA Journal of Automatica Sinica, 6(3), pp.693-702.

[3] Fan, R., Wang, H., Cai, P. and Liu, M., 2020, August. Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. In European Conference on Computer Vision (pp. 340-356). Springer, Cham.

[4] Hazirbas, C., Ma, L., Domokos, C. and Cremers, D., 2016, November. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In Asian conference on computer vision (pp. 213-228). Springer, Cham.

[5] LeCun, Y., Kavukcuoglu, K. and Farabet, C., 2010, May. Convolutional networks and applications in vision. In Proceedings of 2010 IEEE international symposium on circuits and systems (pp. 253-256). IEEE.

[6] Geiger, A., Lenz, P. and Urtasun, R., 2012, June. Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354-3361). IEEE.

[7] Wulff, F., Schäufele, B., Sawade, O., Becker, D., Henke, B. and Radusch, I., 2018, June. Early fusion of camera and lidar for robust road detection based on U-Net FCN. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 1426-1431). IEEE.

[8] Du, X., Ang, M.H., Karaman, S. and Rus, D., 2018, May. A general pipeline for 3d detection of vehicles. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3194-3200). IEEE.

[9] Dumoulin, V. and Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

[10] Xie, G. and Lu, W., 2013. Image edge detection based on opencv. International Journal of Electronics and Electrical Engineering, 1(2), pp.104-106.

[11] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[12] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, pp.1097-1105.

[13] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[15] Haghighat, M.B.A., Aghagolzadeh, A. and Seyedarabi, H., 2011. A non-reference image fusion metric based on mutual information of image features. Computers Electrical Engineering, 37(5), pp.744-756.

[16] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp.600-612.

[17] Qu, G., Zhang, D. and Yan, P., 2002. Information measure for performance of image fusion. Electronics letters, 38(7), pp.313-315.

[18] Ling, H. and Okada, K., 2006, June. Diffusion distance for histogram comparison. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 1, pp. 246-253). IEEE.

[19] Basu, M., 2002. Gaussian-based edge-detection methods-a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(3), pp.252-260.