Accurate Object Detection in Smart Transportation Using Multiple Cameras

Zhinan Qiao[1], Andrew Sansom[1], Mara McGuire[2], Andrew Kalaani[3], Xu Ma[1], Qing Yang[1] and Song Fu[1]
[1] University of North Texas, Denton, TX 76203
[2] Texas A&M University - Corpus Christi, Corpus Christi, TX 78412
[3] Georgia Southern University, Statesboro, GA 30458

***Abstract*—Recently, more and more attention has been paid to the connected object detection for better performance. One of the most interesting fields is learning from multiple resources in a connected fashion. In this paper, we present a connected object detection method using multiple cameras for the smart transportation system. The proposed architecture consists of three parts: an alignment framework, a deep multi-view fusion network and an object detection network. Experiments are conducted to illustrate the performance of our proposed architecture.**

***Keywords*-multi-view, alignment, fusion network, object detection**

I. Introduction

With the rising amount of computation resources, great progress has been made in the area of computer vision with the help of convolution networks [1], [2]. This progress has driven the development of many aspects of the smart transportation system, such as autonomous vehicles and advanced driving. One of the fundamental problems of smart transportation is to build a quick and accurate object detection algorithm in the real world since an advanced driving vehicle must have the capacity to detect surrounding objects promptly and precisely to ensure its safety [3]. On another hand, the cameras with improved function but cheaper price emerged, which enables the assumption to equip multiple cameras on a single vehicle to assist autonomous driving in a relatively economical way.

How to handle occlusion is one of the toughest issues in object detection tasks. Some researches, especially those focused on the methods for 3D object detection address this problem by employing bird view [4], [5]. This type of data representation, however, may obscure the intrinsic nature of the 3D object. Furthermore, these bird view based algorithms require the use of LiDAR equipment, which is much expensive than cameras. On the other hand, LiDAR data always tends to be sparse and not adequately accurate. Another issue of object detection method is that the single front view will lose the information of the real-world 3D structure. For instance, if we look at a cone in the front view, what we observe will be only a circle. These issues point out a crucial problem that we may have different detection results of the same scene at different perspectives using the same algorithm. In order to gain more information from a certain scene for accurate object detection, we propose to adapt the images captured by different cameras from different viewpoints. Base on the previous researches, deep learning has shown its mighty force to extract deep features which are informative to represent the original images [6], we proposed a joint framework grounded on state-of-the-art deep neural networks.

Multi-view image detection always involves image alignment, re-scale, and fusion. Conventionally, image alignment algorithms are aimed at finding the correspondence of images with distinct ratios of overlapping [7], and the differences between views can not be too large or too small. As we proposed to combine and maintain more information via combining multiple images through fusion, the first challenge of our research is finding the appropriate method to align the images. In this paper, we proposed three means addressing the image alignment issue and demonstrate the feasibility of every method. The second challenge of our research is the design of an effective fusion network. Inspired by [8], we proposed a dense fusion network that follows a cascade structure to avoid losing information in the fusion process. We adopt the idea of the dense network and employed joint operation to fuse different views as it proved its ability to excavate the deep information among different sources without losing the original image data presentation. The final ingredient of our framework is an object detection network to ultimately testing the accuracy of our method. Instead of using the popular region-based detector, we employ an end-to-end detector to fit the real-time requirement for autonomous driving vehicles. As shown in Fig. 1, we constructed a joint framework to demonstrate our assumption. The result of the alignment framework will feed to the fusion network and the outcome of fusion will further be tested by the object detection network.

The main contributions of our method are: 1) we propose three approaches for aligning images taken from different perspectives and conduct the experiments to exam the validity of every method; 2) we build a trainable fusion network based on DenseNet to extract features from input images and perform the fusion operation; 3) we induct an evaluation of the fused features by applying state-of-the-art object detection network.

II. Related Work

We proposed a joint approach for the task that adopting multiple cameras for accurate object detection. The holonomic framework follows a cascade structure: first align the corresponding regions (bounding boxes), then fuse the calibrated images to gain more information, and apply the object detector at last to test the improvement.

978-1-7281-6059-7/20/$31.00 ©2020 IEEE
DOI 10.1109/MetroCAD48866.2020.00011

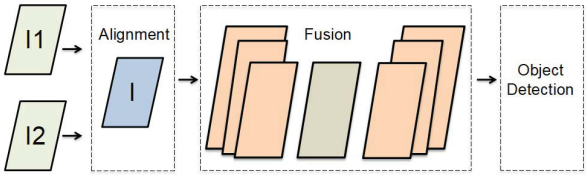

Fig. 1: An overview of the proposed network architecture. Our approach aligns two input images, send the alignment result to a fusion network, and evaluate the result through an object detection network.

A. Alignment

The task of image alignment can be conducted through several aspects. Calibration is one of the conventional and classical approaches by exploiting the setting of cameras. Zhang et al. [9] proposed a flexible calibration technique by observing a planar pattern presented by cameras at different angles, which is demonstrated effective.

Feature-based alignment is another key method to match images base on the correspondence of features. Li et al. [10] designed their automatic feature-based localization and comparison method in a two-step manner: firstly implement the general localization; then perform accurate localization through point to point matching.

Alternatively, some researchers conduct alignment by finding the matching between pixels. Pixel-based alignment is also called direct alignment, which is based on searching for the alignment where most pixels agree [11]. Mohammadzade et al. [12] successfully conduct their experiments for the task of face recognition by finding the correspondence between pixels of faces and demonstrate that their method outperforms state-of-the-art eye-alignment methods.

B. Fusion

Fusion data from different sources in different formats is not a brand new idea. Most existing works fuse the bird view of LiDAR data and the front view of image data, then apply convolutional network to make prediction. Chen et al. [13] proposed their framework to exploit multiple views from different sensors based on a region fusion architecture.

However, it is worth noticing that the previous works employing LiDAR and cameras trends to be expensive not only because of the cost of various equipment but also the rising cost of computing caused by different data formats. Based on this consideration, one scheme is to estimate object pose and shape through multiple cameras and this achieves a state-of-art performance [14]. Johns et al. [15] proposed a multi-view recognition method by firstly decomposing the image sequence into image pairs, then assign the weight of the images pairs according to their contribution to train the classifier.

To our knowledge, this research is the first proposal to fuse the images from different perspectives using the neural networks. However, fusion images with different properties have been explored by multiple researchers. Liu et al. [16] trained their neural network by both original and blurred images patches. Their network jointly adopt the activity measurement and fusion strategy and obtained state-of-the-art performance. [17] combined the most widely used fusion strategies: multi-scale transform and sparse representation to implement their fusion network.

Inspired by the aforementioned fusion strategies, we adopt the idea of DenseNet [8] to implement our fusion network. DenseNet directly connects all the layers under the premise of ensuring the maximum information transmission between the layers in the network. In a conventional convolutional neural network, assuming it has L layers, there will be L connections, but in DenseNet, there will be L(L+1)/2 connections, which means the input of each layer comes from the output of all the previous layers. [18] adopted dense block in their fusion network to refine the performance of multi-scale fusion and proved the dense block can effectively extract the information from the inputs.

C. Object Detection

Various methods for object detection can be found in the literature. Considering the efficiency requirement of smart transportation, especially autonomous driving vehicles, we consider three end-to-end state-of-the-art object detection networks as our experimental tools.

MobileNet [19] was firstly proposed by Howard et al. MobileNet's main contribution is to replace the previous standard convolutions with depth-wise separable convolutions to improve the computational efficiency and reduce the parameter amount of convolutional networks. It defaults to the assumption that the conventional convolution kernel has a decomposition characteristic, which is similar to a linear combination in the channel dimension mapping of feature maps. The Google team demonstrated the effectiveness of MobileNet as an efficient infrastructure network through a variety of experiments.

Yolo V3 [20] is the latest algorithm in the Yolo series, it contains both reservations and improvements to the previous algorithms. YOLO V3 uses multiple-scale fusion methods to make predictions; a new network is leveraged to implement feature extraction and each bounding box uses a multi-label classification strategy to predict which classes the bounding box might contain. Faster detection speed and higher detection accuracy have been achieved by adapting these improvements.

Qiong et at [21] proposed their single-stage object detector based on recurrent rolling convolution (RRC) architecture. RRC automatically find suitable contextual information for SSD to improve target detection performance. Comparing to the R-CNN's two stages structure: extraction and classification, RRC only requires a single process to get the detection result and achieved state-of-the-art detection accuracy.

III. Proposed Framework

Our proposed framework contains three key components: the alignment framework; the fusion network and the object detection network.

Fig. 2: An example of corresponding images in data 1 and data 2. The differences between them are observable but not too palpable.

A. Data

The data set we use is provided by KITTI Vision Benchmark Suite, which is a real-world computer vision benchmark designed for the researches of autonomous driving. [22]. Specifically, we make use of one of the binocular data set in which the images are captured by two-color cameras (camera 2 and camera 3) equipped on the top of a vehicle. The height above the ground of the cameras are identical and the distance between them is 0.54 meter. The data set can perfectly fit our requirements: the differences between views exist but not too large. And the identical height of the cameras simplified the alignment process by restricting the position change within the horizontal direction. Each data set contains 7481 training images and 7518 testing images. The label of the first data set is also provided. We are going to use ”data 1” and ”data 2” to infer the first and second data set respectively in the remaining part of this paper. In this data set, the label for the training data of data 1 is available but the label for data 2 is not provided. Fig 2 shows a pair of corresponding images in data 1 and data 2. We can observe from the images that the differences between them are perceptible but not too distinct.

B. Alignment

1) Calibration: In the aforementioned data set, the calibration matrix from a base camera to every other camera is given. The first attempt we assume to align different images is to calibrate two images through a calibration matrix. In state-of-the-art benchmark data suit [22], the mapping between 3D reference labels and 2-D image labels is a key component of image calibration, which forces the user to take the distortion of images into consideration [23].

2) Generate Label via Object Detector: we also proposed an alignment method by using state-of-the-art object detection networks. Concretely, given an image(image 1) with the ground-truth label, we assume the image(image 2) we try to align with image 1 contains the same amount of object that can be detected by art object detection networks. Thus we run the detector on image 2 and compare the label generated with the ground truth label. In the following step, we compare the position of all the labels of image 1 and image 2 and assume the bounding boxes with the minimal position difference represent the same object.

3) Generate Label via Pooling: The last alignment strategy we conceived is based on the assumption that the bounding boxes of the corresponding regions in image 1 and image 2 are identical. As we mentioned before, the height above the ground of the cameras and identical, so the absolute height of corresponding bounding boxes remain unchanged. Then we set the original position of the bounding box in image 2 (bounding box 2) using the location value of bounding box 1, and slide the window to the right by the stride of 1 pixel. While sliding, we implement average pooling of the window and compare the difference of the image cropped by the window in image 1 and image 2. The bounding boxes which have the minimal difference will be considered as the corresponding alignment regions.

C. Fusion

We construct our fusion network based on the insight of the splendid DenseNet [8]. DenseNet is a convolutional neural network with dense connections. In this network, there is a direct connection between any two layers. That is to say, the input of each layer of the network is the union of the output of all the previous layers, and the feature map learned by a layer is also directly transmitted to all layers after it. Due to the aforementioned network structure, the DenseNet naturally possesses the ability to extract and maintain the most informative features of the inputs. By fusion the feature map extracted by the DenseNet, we made the assumption that more features will be reserved in the fused image and further benefit the object detection process.

Inspired by [24], we build our end-to-end trainable fusion network by combining the feature extraction network, fusion network and image reconstruction network. The feature extraction network is utilized to extract deep features of the input images through the embedded dense blocks. Then we feed the deep features to the fusion network and then apply the image reconstruction network to reconstruct the images. Both the feature extraction network and image reconstruction network followed a three-layer structure to minimize information loss. We proposed to apply both addition and L1 norm fusion strategies in the fusion phase and adopt an object detector to demonstrate the effectiveness of the fusion process. Due to time constraint, we only implement the addition strategy in the experiments.

D. Object Detection

In the object detection stage, we adopted three states-of-the-art object detection networks [19], [21], [20] to evaluate our calibration and fusion network. By establishing the contrastive experiments, RRC performs more fitful for our data and proposed experiment. Thus all the object detection tasks will be implemented using RRC model in the rest part of this paper.

IV. Experiment

We conducted all the experiments on the Ubuntu 18.04 platform equipped with 2.80GHz Intel(R) Xeon(R) X3460

CPU. For frameworks (e.g. fusion and object detection) that process faster on GPU, we use a Tesla K40c (12GB RAM). We implement the proposed deep learning framework on the widely used open-source framework Caffe [25].

A. Alignment

1) Calibration: The data set provided by KITTI Vision Benchmark Suite [7] provided the corrected rotation matrix (R) and mapping matrix (P), which can be directly used in combination with the camera internal parameters and distortion parameters to generate a calibration parameter. As we mentioned before, let Ti and Ri denote the rotation and translation matrices from camera 0 to camera i, we can deviate the calibration matrices for camera 2 and 3 are shown below.

$$R = R_3^{(-1)} * R_3, \tag{1}$$

$$T = T_2 - T_3. \tag{2}$$

After applying the aforementioned calibration method, we found that the calibration parameter varies along with the distance from the camera to the object. Thus although we can calibrate individual pair of images, we are not able to find a fixed algorithm to calibrate all the images.

2) Generate Label via Object Detector: Under the situation that the data 1 has the ground truth label, we proposed our second alignment method by making use of the capability of state-of-the-art object detection network. In KITTI data set, the coordinate plane is defined by Xmin, Xmax, Ymin, and Ymax. We conducted the experiments to generate the labels of image3 data set through the following steps:

- Use Object Detection Network to generate the label of all the images in data 2.
- As the pair of corresponding images in data 1 and data 2 data set have minor differences. We assume that we can always find the matching relationship between the regions in those two data sets and the bounding boxes have an identical size.
- Camera 2 and camera 3 have the same height above the ground, we assume the corresponding bounding boxes have the same Ymax and Ymin value.
- As camera 3 is located on the right side of camera 2. We assume that all the corresponding regions in image 2 should be found to the right of the region position in image 1.
- Thus we compute the sum of the squared difference of the two corresponding bounding boxes and assign the label of the region in image 1 to the region with the smallest deviation value in image 2 (the position of the region in image 2 must locate to the right of the region in image 1).

We had conducted the experiments on the entire data set which contains 7481 images. The result shows the number of regions of the corresponding images cannot fit well. In our 7481 training examples, only 5215 of them get an identical number of regions comparing with image 1. Also, the algorithm that finding minimal deviation can not guarantee to obtain real mapping.

3) Generate Label via Pooling: We designed our finial assumption based on the two previous attempts in deference to the following steps.

- Given two corresponding images, we firstly define the bounding box of image 2 (bounding box 2) as the same size of the bounding box of image 1 (bounding box 1).
- As we mentioned before, we set the Ymax and Ymin value of the bounding box 2 as the same value of bounding box 1. Thus we only need to find the Xmax and Xmin value now.
- we set the location of the start point of bounding box 2 using the location value of bounding box 1, and slide the window to the right by the stride of 1 pixel.
- We implement average pooling of the window and compare the difference of the image cropped by the window in image 2 and image 3.
- In the last step, assign the label of bounding box 2 to the bounding box 3 which has a minimal difference.

The experimental result of this method shows great improvement. We are able to align 7476 of the images. Fig 3 shows the alignment result of a pair of images in data 1 and data 2. We can observe from the figure that even though the relative interval between the objects varies, we can still obtain a desirable result. The following experiments will be conducted base on this calibration method.

B. Fusion

As the result of pure calibration is not desirable, we proposed our fusion network based on the idea of region fusion. Our fusion network consists three major parts: an input network to extract features from multiple images; a fusion network to fuse the features, and an output network to reconstruct the image from the fused feature map. As shown in Fig 4, our region fusion task can be further divided into the following steps.

- After alignment, we obtain the corresponding labels in image 1 and image 2. Then we extract the image patches cropped by the bounding boxes.
- Implement region fusion based on our fusion network. In this process, we choose addition as the fusion strategy and adopt pixel and system structure loss to find the best parameter for our model.
- Stitch the fused region back to image 1 using the ground truth label of image 1. The output of the fuse model will be the input of the object detection model.

Within the fusion network, we follow the method proposed by [26] to use addition as the fusion strategy. Inspired by [24], In order to avoid information lost in the image reconstruction process, we minimize the loss function L to train our fusion framework. As shown in equation 3, lost function L is the combination of weighted structural similarity loss and pixel loss.

$$L = \lambda * L_s + L_p. \tag{3}$$

The equation of pixel loss is shown in equation 4 where R and F stand for the fused image and the output result.

Fig. 3: The result of applying Pooling method to data 2. The green rectangles illustrate the bounding box of detected object.

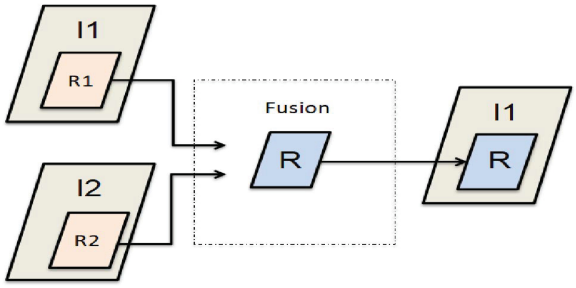

Fig. 4: Region based fusion framework. We fuse the correspond regions of a pair of images and put the fused patches back to image 1.

Essentially, it is the Euclidean distance between the two images.

$$L_p = ||R, F||_2. \tag{4}$$

The similarity loss is calculated by Equation 5, where SSIM is the function to generate structural similarity proposed by Wang et al. [27].

$$L_s = 1 - SSIM(R, F). \tag{5}$$

Fig 5 shows the result of our region fusion method. The outcome demonstrates our network successfully aligns the regions and strong features can be observed along the edges.

C. Object Detection

In the object detection stage, we leveraged the RRC object detection model and retrain the model using Caffe framework [25]. The RRC model was pre-trained on the KITTI binocular data set and achieve competitive accuracy on object detection tasks. We set the training batch size as 1 and learning rate as 1^{-8}. Following the setting of RRC model, we set the epoch as 60,000, but due to time constraint, we only provide the result after training for 1,000 epochs. Fig 6 depicted the object detection result of the fused image. We can observe from the result that the vehicles had been successfully detected.

TABLE I: Result of Object Detection

Original 1	**Easy**	**Moderate**	**Hard**
Accuracy	90.51 %	89.89 %	80.67 %
Original 2	**Easy**	**Moderate**	**Hard**
Accuracy	89.84 %	80.89 %	71.64 %
Fuse	**Easy**	**Moderate**	**Hard**
Accuracy	89.86 %	80.92 %	71.66 %

Table 1 displays the result of the object detection model. The "Original 1" row shows the result when we employ the original model and use un-fused images for testing. The "Original 2" row shows the result when we employ the original model but use the fused image for testing. We can observe a rational decrease in accuracy. The result fits our assumption because the features become different during the fusion process, so the model trained on original features can not achieve similar accuracy. The "Fusion" row displays the result when we train the RRC model using fused images for 1,000 epochs. Due to the computation limitation of GPU, we in fact only employ 13.4% of our training data for once. However, we can still observe slight accuracy improvement under such strict situation. At the same time, we can always detect the loss

Fig. 5: The result of region-based fusion. The first two images in the first row represent the corresponding regions in a pair of images. And the last image in the first row is the fusion result of the two regions. The image in the second row is the synthesized result of the original image and fused region.

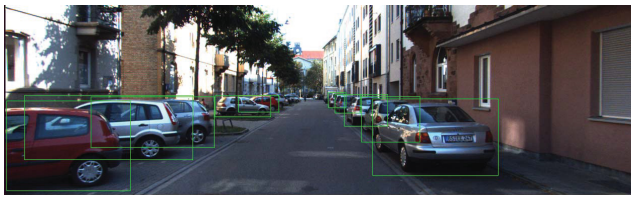

Fig. 6: The result of our model after training for 1,000 epochs.

is decreasing during the training process. Grounded on the aforementioned phenomenons, we have the confidence to make the assumption that we will achieve higher accuracy after completing 60,000 epochs.

V. Conclusion

In this paper, we proposed and conducted experiments to test our assumption that we can improve the object detection accuracy via fusion images taken from different perspectives and demonstrated the assumption by our joint framework. The result proves the deep learning network provides the feasibility of optimal feature extraction and fusion toward the object detection task and this unique capability can be further employed to other domains.

One limitation of our current implementation is that it requires the pair of input images contain a relatively large overlapping area and similarly target objects. We will further explore situations where input images show more variances.

ACKNOWLEDGMENT

The work is supported by National Science Foundation (NSF) grants NSF CNS-1761641 and CNS-1852134.

References

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] D. M. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 87–93 vol.1.

[4] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 641–656.

[5] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *arXiv preprint arXiv:1712.02294*, 2017.

[6] X. Zhang, H. Zhang, Y. Zhang, Y. Yang, M. Wang, H.-B. Luan, J. Li, and T.-S. Chua, "Deep fusion of multiple semantic cues for complex event recognition." *IEEE Transactions on Image Processing*, pp. 1033–1046, 2016.

[7] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, pp. 1–104, 2007.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, p. 3.

[9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, 2000.

[10] Y. Li and P. Gu, "Feature-based alignment and comparison between portion and whole of free-form surfaces," *CIRP Annals-Manufacturing Technology*, pp. 135–138, 2005.

[11] R. Szeliski *et al.*, "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, pp. 1–104, 2007.

[12] H. Mohammadzade, A. Sayyafan, and B. Ghojogh, "Pixel-level alignment of facial images for high accuracy recognition using ensemble of patches," *JOSA A*, pp. 1149–1159, 2018.

[13] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, p. 3.

[14] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and map-

ping (slam): Part ii," *IEEE Robotics & Automation Magazine*, pp. 108–117, 2006.
[15] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3813–3822.
[16] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Information Fusion*, pp. 191–207, 2017.
[17] Y. Liu, S. Liu, and Z. Wang, "A general framework for image fusion based on multi-scale transform and sparse representation," *Information Fusion*, pp. 147–164, 2015.
[18] C. Tian, C. Li, and J. Shi, "Dense fusion classmate network for land cover classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 192–196.
[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[20] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[21] J. QiongYan and Y. LiXu, "Accurate single stage detector using recurrent rolling convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[22] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems*, 2013.
[23] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *IEEE Computer Society Conference onComputer Vision and Pattern Recognition*, 1997, pp. 1106–1112.
[24] H. Li and X.-J. Wu, "Densefuse: A fusion approach to infrared and visible images," *arXiv preprint arXiv:1804.08361*, 2018.
[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.
[26] K. R. Prabhakar, V. S. Srikar, and R. V. Babu, "Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs," in *IEEE International Conference on Computer Vision*, 2017, pp. 4724–4732.
[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, pp. 600–612, 2004.