MAFNet: Segmentation of Road Potholes With Multimodal Attention Fusion Network for Autonomous Vehicles

Zhen Feng, *Graduate Student Member, IEEE*, Yanning Guo, Qing Liang, *Student Member, IEEE*, M. Usman Maqbool Bhutta, Hengli Wang, *Member, IEEE*, Ming Liu, *Senior Member, IEEE*, and Yuxiang Sun, *Member, IEEE*

***Abstract*— Road potholes can cause discomforts to passengers and even traffic accidents to vehicles. Accurate segmentation of road potholes is an important capability for autonomous vehicles to ensure safe driving. Some methods on road-pothole segmentation use single-modal data (i.e., RGB images). The main challenge faced by these methods is that the visual appearance of road potholes is often close to road areas, making these networks difficult to distinguish them. Recent methods resort to fusing RGB images with depth/disparity images for pothole segmentation. However, their performance is still not satisfactory in real-world applications. To achieve superior results, this article proposes a novel data fusion network for road-pothole segmentation, where a channel attention fusion module and a dual attention fusion (DAF) module are designed to hierarchically fuse the RGB and disparity data. We evaluate our proposed network using a public dataset, and the experimental results demonstrate the superiority over the state-of-the-art networks.**

***Index Terms*— Attention mechanism, autonomous vehicles, RGB-disparity fusion, road potholes, semantic segmentation.**

I. Introduction

ROAD pothole is a kind of negative obstacles that lie below the road surface. It usually appears after long-term use of roads without timely maintenance. The existence of road potholes is a potential threat to road users. It can cause bumps and discomforts to passengers. When vehicle speed is fast, it can even cause severe accidents, such as rollover to vehicles [1]. To alleviate the negative effects caused by road potholes, vehicles are often equipped with damping systems to reduce the vibrations [2]. However, adding damping systems is just a kind of remedial measures. To get alerts for road potholes in advance, it is necessary to detect or segment the potholes using vehicle-mounted sensors. Currently, many effective methods have been proposed using various sensors, such as visual camera [3], thermal camera [4], RGB-D camera [5], and Lidar [6].

In the field of using visual sensors, RGB images, disparity images, and depth images have been adopted in existing road-pothole segmentation methods. Each modal of data has its own advantages and disadvantages [7]. For example, RGB images contain rich visual texture information, but not robust to illumination conditions [8]. Methods using only RGB images could be degraded in darkness [9]. Disparity and depth images encode distance information, but cannot provide visual information. Fusing multimodal data could take advantages of each modal of data and has been proven to produce better performance in previous work [10].

However, the existing multimodal methods still cannot provide satisfactory performance in real-world applications. Segmenting potholes by fusing RGB images and disparity images has been proven to achieve satisfactory performance [11]. However, there are currently few studies on pothole segmentation by fusing RGB and disparity images. Existing algorithms perform unsatisfactorily, especially at the edges of potholes. We guess that this is mainly caused by the limitation of convolutional feature extraction and inappropriate fusion of the two modals of data. To achieve superior results, we propose a novel deep neural network that fuses RGB images and disparity images for road-pothole segmentation in this work. We replace the final stages of both encoders with a transformer, extracting more edge information through the multihead self-attention model. We design two kinds of fusion modules in the encoders based on channel attention and dual attention to fuse the two modals of data. Attention modules can focus the network on useful information and weaken useless information. Channel attention is used to adjust the weights between feature maps. Dual attention includes channel attention and spatial attention, where spatial attention is used

Manuscript received 7 July 2022; accepted 30 July 2022. Date of publication 22 August 2022; date of current version 26 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61973100, Grant 61876050, Grant 12150008, and Grant 62003286; and in part by the Startup Fund of The Hong Kong Polytechnic University under Grant P0034801. The Associate Editor coordinating the review process was Bruno Ando. *(Corresponding author: Yanning Guo.)*

Zhen Feng is with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China, and also with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: zfeng94@outlook.com).

Yanning Guo is with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: guoyn@hit.edu.cn).

Qing Liang, Hengli Wang, and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: qliangah@connect.ust.hk; hwangdf@connect.ust.hk; eelium@ust.hk).

M. Usman Maqbool Bhutta is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: mumbhutta@cuhk.edu.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: yx.sun@polyu.edu.hk).

Digital Object Identifier 10.1109/TIM.2022.3200100

to adjust the weights between different data in feature maps. We evaluate and compare our network with the state of the arts on a public dataset `Pothole-600` [11]. The contributions of this work are summarized as follows.

1) We propose a novel road-pothole segmentation network[1] by fusing RGB and disparity images, which integrates convolutional layers and transformer modules.
2) We adopt two data fusion modules in encoders based on channel attention and dual attention, and verify the effectiveness through ablation studies.

The remainder of this article is structured as follows. Section II presents a review for related work. Section III describes the details of our network. Section IV discusses the experimental results. The last section concludes this article and presents several promising research directions for future work.

II. Related Work

This section reviews related works on road-pothole segmentation and detection. We classify them into single-modal methods that use only RGB images, and multimodal methods that fuse RGB and other modality of data. Since many related works detect road cracks that are also negative obstacles, they are also reviewed here.

A. Single-Modal Methods

Zhang *et al.* [14] proposed a crack detection method based on convolutional neural network (CNN). This method divides an image into subimage blocks and classifies each sub-image block into cracks and noncrack regions. Pereira *et al.* [15] designed a classification-based CNN architecture to achieve pothole detection. They used four convolutions and pooling layers, as well as a fully-connected layer. The model can detect potholes under dry, humid, and dark conditions with various sizes and shapes. Mei and Gul [16] designed ConnCrack combining conditional Wasserstein generative adversarial network (cWGAN) and connectivity maps. The cWGAN is used for training, consisting of two separate neural networks called generator and discriminator, and the connectivity maps are used to resolve the scattered output caused by the deconvolution layers. Anand *et al.* [17] designed an autonomous crack and pothole detection system based on texture features. They removed the last convolutional layer of SqueezeNet [18] and added an encoding layer.

Mandal *et al.* [19] used YOLOv2 [20] to detect road cracks. In order to speed up training and improve the performance, they used transfer learning and froze the weights of ResNet-101 and RPN pre-trained on the COCO dataset. Suong and Kwon [21] also proposed a network based on YOLOv2 to detect potholes. They designed two architectures. The first one is based on the Darknet architecture of YOLOv2, which contains 31 layers, and another is based on the first one, attempting to reduce the computational costs and model size.

Dhiman and Klette [22] proposed two deep learning-based pothole detection methods, using stereo vision to estimate the 3-D point cloud information of the environment, and further analyzing the road environments in front of the vehicle. Masihullah *et al.* [23] adopted few-shot learning approach and introduced the channel attention module to DeepLabv3+ [24] to realize the segmentation of potholes in RGB images.

B. Multimodal Methods

In addition to RGB images, other data, such as thermal images and depth images, are also used for pothole and crack detection. Bhatia *et al.* [25] designed a CNN based on a residual network, which takes as input thermal images to detect potholes. Beckman *et al.* [26] proposed a method based on a faster region-based CNN (Faster R-CNN) to detect concrete spalling damages with RGB and depth data. Multimodal information fusion brings better segmentation results, so Pan *et al.* [27] fused multispectral images obtained by unmanned aerial vehicle to detect potholes and cracks on asphalt roads. Fan *et al.* [11] proposed AA-RTFNet based on RTFNet to fuse the RGB images and disparity images to segment the pothole. They introduced an attention module in the skip-connection between the encoder and the decoder. They also released the `Pothole-600` dataset that contains 600 pairs of RGB and disparity images.

C. Difference From Existing Work

Our work lies in the category of multimodal methods. A major issue of existing methods, such as AA-RTFNet [11], is that they do not perform well on the edges of potholes. We conjecture that the reasons are using only the convolutional layers to extract features, as well as using the simple elementwise addition for fusion. Our MAFNet builds on RTFNet, but has the following differences. First, we believe that the multihead self-attention model in transformer could better preserve edge information. So, we replace the last stage of the RTFNet encoder with a transformer module. Second, we add a channel attention and a spatial attention for feature fusion instead of directly elementwise adding the two feature maps from the two modals of data. Although AA-RTFNet and our MAFNet both build on RTFNet, our network differs from AA-RTFNet that we introduce the attention module into encoders instead of between encoders and decoder.

III. Proposed Network

A. Overall Architecture

We propose a new network called multimodal attention fusion network (MAFNet) for road-pothole segmentation. Fig. 1 shows the overall architecture.

We develop our network based on RTFNet [13]. So, MAFNet also follows the two-encoders-one-decoder paradigm, where both the two encoders have the same initial block and four stages. We define the input set for encoders as $\{(R_i, D_i)|R_i, D_i \in \mathbb{R}^{H\times W\times 3}, i = 1, \ldots, M\}$, where M represents the number of images. We employ ResNet-34 [12] as the backbone of the encoders, and replace the fourth stage of the backbone with a transformer module borrowed from [28]. We call the initial block, 1st, 2nd, and 3rd stages of ResNet-34 as the initial block, 1st, 2nd, and 3rd stages

[1]The code is available at https://github.com/lab-sun/MAFNet

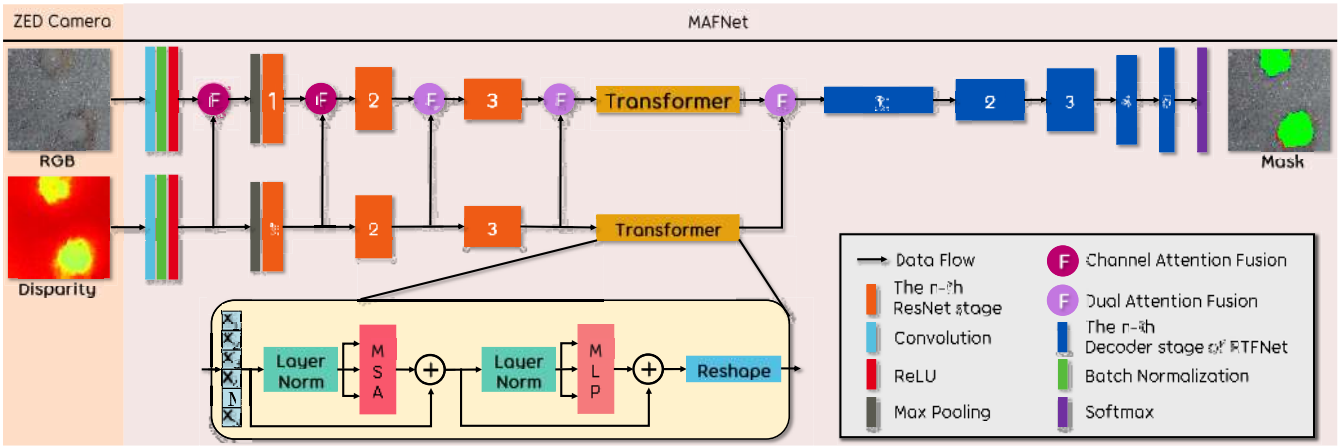

Fig. 1. Overall architecture of our proposed MAFNet. There are two five-stage encoders and one five-stage decoder, which are the RGB encoder, disparity encoder, and decoder. We change the last stage of ResNet [12] to the transformer module. The disparity feature maps extracted from the initial block and the first stage of the encoder are fused into the RGB encoder through the CAF module. The disparity feature maps extracted from the last three stages are fused into the RGB encoder through the DAF module. For the decoder, we directly borrow the decoder from RTFNet [13] as our decoder. The images in the dataset are captured by a ZED stereo camera. The figure is best viewed in color.

of our encoder respectively, and the final transformer module as the 4th stage of the encoder. The encoders are designed to extract feature maps from the input RGB and disparity images. We denote the nth stage of the RGB encoder and the disparity encoder as $g_R^{(n)}(R_i)$ and $g_D^{(n)}(D_i)$, respectively. In the fusion part, first, the RGB feature maps and the disparity feature maps are fused by elementwise addition, and the preliminary fusion result is obtained. This preliminary fusion result has the same dimension as the feature map. Then, the attention model is used to adjust the weight between different feature information for this preliminary fusion result, and the final fusion result is obtained. The final fusion result has the same dimension as the original feature map. The first two stages of fusion use the channel attention model, and the last three stages of fusion use the dual-attention model. The feature maps extracted by the initial blocks, and the first stages of the encoders are respectively fused using our proposed channel attention-based fusion module. We denote the nth fusion module as $f_{\text{fuse}}^{(n)}(g_R^{(n)}(R_i), g_D^{(n)}(D_i))$. The feature maps extracted by the last three stages of the encoders are fused using our proposed dual attention-based fusion module. The final fused output of the two encoders is fed into the decoder. The encoder can be denoted as (1), shown at the bottom of the page. The decoder is used to restore the feature map resolution and generate the segmentation map. We directly borrow the RTFNet decoder [13] as our decoder.

B. Fusion Modules

To fuse the RGB and disparity data, we design two types of fusion modules respectively based on channel attention [29] and dual attention [30]. So, they are named as channel attention fusion (CAF) module and dual attention fusion (DAF) module. We denote them as f_c and f_d. For each module, we try two kinds of architectures to choose the better one.

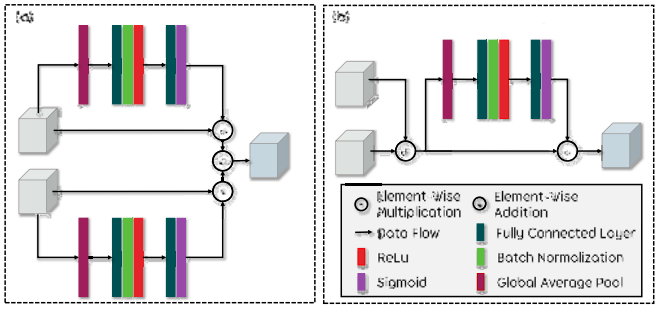

Fig. 2. Structures of the two kinds of CAF module. (a) Structures of CAF-1A2F, which adopted from [11]. (b) Structures of CAF-1F2A. The figure is best viewed in color.

1) CAF Module: Fig. 2 shows the two kinds of architectures of our proposed CAF module, which are named as CAF-1st-Attention-2nd-Fusion (CAF-1A2F, denoted as f_c^{AF}) and CAF-1st-Fusion-2nd-Attention (CAF-1F2A, denoted as f_c^{FA}). In CAF-1A2F, the feature maps from the two modals are first weighted by channel attention networks, and then the weighted feature maps are fused via elementwise addition. On the contrary, in CAF-1F2A, the feature maps are first fused via elementwise addition, then weighted by a channel attention network. We adopt the channel attention model in the encoder to fuse two modalities and discuss the positional relationship between the attention model and the fusion of elementwise summation through two different structures (CAF-1A2F and CAF-1F2A).

The channel attention networks are shown in Fig. 2. In CAF-1A2F, we first employ a global average pool to calculate the average value of the feature maps from the two channels. Then, two fully connected layers are used to

$$f(R_i, D_i) = f_{\text{fuse}}^{(5)}\Big(\ldots f_{\text{fuse}}^{(2)}\Big(g_R^{(2)}\Big(f_{\text{fuse}}^{(1)}\big(g_R^{(1)}(R_i), g_D^{(1)}(D_i)\big)\Big), g_D^{(2)}(D_i)\Big), \ldots, g_D^{(5)}(D_i)\Big) \tag{1}$$

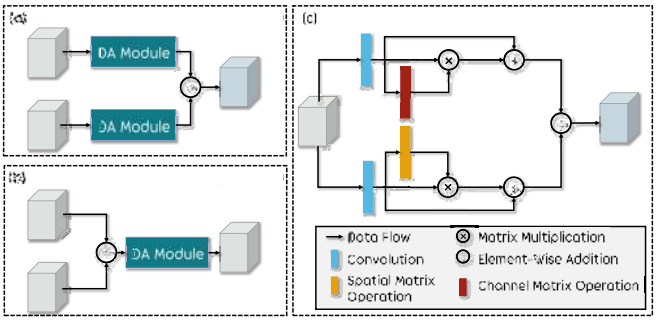

Fig. 3. Structures of the two kinds of DAF modules. (a) Structures of DAF-1A2F, which adopted from [11]. (b) Structures of DAF-1F2A. (c) Structure of the DA module. The figure is best viewed in color.

get the weighting factor from the average value. There is a batch normalization layer and a ReLU activation layer after the first fully connected layer. The input channels of the first fully connected layer is 64 and the output channels of the first fully connected layer is 32. There is a Sigmoid function after the second fully connected layer, which ensures the output weighting value ranging from 0 to 1. The second fully connected layer restores the number of channels from 32 to 64. We denote this process as $w(X_i)$. The two input feature maps are respectively weighted by the two weighting factors via elementwise multiplication. Finally, the weighted feature maps are fused by elementwise addition. CAF-1F2A adopts the same attention network as CAF-1A2F, but CAF-1F2A first fuses the two input feature maps, and then feed the fused result to the attention network. f_c^{AF} and f_c^{FA} are calculated as follows:

$$f_c^{\mathrm{AF}} = W(R_i, w(R_i)) \oplus W(D_i, w(D_i)) \tag{2}$$

$$f_c^{\mathrm{FA}} = W(R_i \oplus D_i, w(R_i \oplus D_i)) \tag{3}$$

where the $\oplus$ means the elementwise summation, $C_{n\times m} = A_{jk} + B_{jk}, j \in 1, 2, \ldots, n, k \in 1, 2, \ldots, m$. The $W(A, b)$ means the elementwise multiplication, $W(A_{n\times m}, b) = A_{jk} \times b, j \in 1, 2, \ldots, n, k \in 1, 2, \ldots, m$.

We take the CAF-1F2A structure as an example to introduce the process of CAF in detail through the pseudocode of this module. The pseudocode is shown in Algorithm 1.

2) DAF Module: Similar as the CAF module, we also have DAF-1st-Attention-2nd-Fusion (DAF-1A2F, denoted as f_d^{AF}) and DAF-1st-Fusion-2nd-Attention (DAF-1F2A, denoted as f_d^{FA}), which are illustrated in Fig. 3. We adopt the dual attention model in the encoder to fuse the two modalities and discuss the positional relationship between the attention model and the fusion of elementwise summation through two different structures (DAF-1A2F and DAF-1F2A). In the DAF module, we adopt a dual attention (DA) module [30], which includes a spatial matrix operation (denoted as f_s) and a channel matrix operation (denoted as f_c). The spatial matrix operation resizes the shape of feature maps from $C \times H \times W$ to $(H \times W) \times (H \times W)$, and the channel matrix operation resizes the shape of feature maps from $C \times H \times W$ to $C \times C$, where C represents the number of channels of the feature map, H represents the height of the feature map, and W represents the width of the feature map.

Algorithm 1 Process of CAF-1F2A (f_c^{FA})

Data: RGB feature maps F_R, disparity feature maps F_D
Result: Fusion result Q

$F_{fuse} \leftarrow F_R \oplus F_D$;
$w_{FA} = w(F_{fuse})$;
$F_{new} = W(F_{fuse}, w_{FA})$;
$Q \leftarrow F_{new}$;

Algorithm 2 Process of DAF-1F2A (f_d^{FA})

Data: RGB feature maps F_R, disparity feature maps F_D
Result: Fusion result Q

$F_{fuse} \leftarrow F_R \oplus F_D$;
$F_c = f_c^{FA}(F_{fuse})$ `// channel attention;`
$F_s = f_s^{FA}(F_{fuse})$ `// spatial attention;`
$F_{new} = F_c \oplus F_s$;
$Q \leftarrow F_{new}$;

Fig. 3(c) shows the structure of the DA module. There are mainly two branches in the DA module. Each branch consists of three sub-branches. In the first main branch, the input features are first fed into a 3×3 convolution layer that keeps the number of channels unchanged, and then the processed feature map is fed into three sub-branches. In the bottom sub-branch of the first main branch, a new feature map is obtained through the channel matrix operation. Then the new feature map is fused with the feature map from the middle sub-branch by matrix multiplication. The fused feature map is then elementwise added with the feature map from the top sub-branch to produce the final output. The second main branch is similar to the first main branch, except that there is the spatial matrix operation instead of the channel matrix operation. Finally, the outputs from the two main branches are fused via elementwise addition to produce the final output. The process can be denoted as $f_d = f_c \oplus f_s$. We refer readers to this article [30] for more details of the DA module. f_d^{AF} and f_d^{FA} are calculated as follows:

$$f_d^{\mathrm{AF}} = f_d(R_i) \oplus f_d(D_i) \tag{4}$$

$$f_d^{\mathrm{FA}} = f_d(R_i \oplus D_i). \tag{5}$$

We take the DAF-1F2A structure as an example to introduce the process of DAF in detail through the pseudocode of this module. The pseudocode is shown in Algorithm 2.

C. Encoders

The RGB and disparity encoders share the same architecture. As aforementioned, we adopt ResNet-34 [12] as the encoder backbone, and replace the 4th stage of ResNet with the transformer module. There is a 7×7 three-channel convolutional layer, a batch normalization layer, and a ReLU activation layer in the initial block. The initial block reduces the resolution by half and increases the number of channels from 3 to 64. The first stage of the encoder contains a max pooling layer and the first residual block, keeping the number of channels unchanged and reducing the resolution by half.

TABLE I

DETAILED CONFIGURATIONS OF THE ENCODER AND THE DECODER. C AND S IN BRACKETS REPRESENT CHANNEL AND SIZE, RESPECTIVELY. THE INITIAL BLOCK OF THE ENCODER CONTAINS THE CONVOLUTION, BATCH NORMALIZATION, AND RELU LAYERS. THE 1ST STAGE OF THE ENCODER CONTAINS THE MAX POOLING AND THE 1ST STAGE OF THE RESNET

	Encoder					Decoder				
	initial block	1st stage	2nd stage	3rd stage	4th stage	1st stage	2nd stage	3rd stage	4th stage	5th stage
input(C)	3	64	64	128	256	512	256	128	64	32
output(C)	64	64	128	256	512	256	128	64	32	2
input(S)	512×512	256×256	128×128	64×64	32×32	16×16	32×32	64×64	128×128	256×256
output(S)	256×256	128×128	64×64	32×32	16×16	32×32	64×64	128×128	256×256	512×512

Each subsequent stage reduces the resolution by half and doubles the number of channels. We adopt the transformer module refereed from [28] as the last stage of the encoders. In the transformer module, we design a three-layer 16-head self-attention module. Each layer includes a layer norm layer, a multihead self-attention (MSA) layer, a layer norm layer, and a multilayer perceptron (MLP) layer in sequence. We refer readers to [31] for more details about the transformer. The resolution of the final output of the encoder is 16×16, and the number of channels is 512. The detailed configurations of the encoder are displayed in Table I. The encoders can be denoted as (1), where the $f_{\text{fuse}}^{(1,2)} = f_c$ and $f_{\text{fuse}}^{(3,4,5)} = f_d$.

D. Decoder

The decoder is designed to restore the feature map resolution and produce the segmentation map. As aforementioned, we directly use the RTFNet decoder as our decoder. There are five stages in the decoder. Each decoder doubles the feature map resolution. The final resolution is the same as the input. A softmax layer is placed at the end to compute the probabilities of each pixel belonging to each class. The configurations are also displayed in Table I. This module is denoted as (6), where the L means the output of MAFNet

$$L = D_{\text{coder}}(f(R_i, D_i)). \tag{6}$$

We detail the pipeline of MAFNet in pseudocode and the pseudocode is shown in Algorithm 3.

Algorithm 3 Process of MAFNet

Data: RGB images R, disparity image D
Result: Semantic Segmentation Results L

$F_R \leftarrow R$;
$F_D \leftarrow D$;
for $i \leftarrow 1$ **to** 5 **do** // Encoder
  $F_R \leftarrow g_R^{(i)}(F_R)$;
  $F_D \leftarrow g_D^{(i)}(F_D)$;
  if $i \leq 2$ **then**
    $f_{fuse} = f_c^{FA}$ // CAF Module;
  else
    $f_{fuse} = f_d^{FA}$ // DAF Module;
  end
  $F_R \leftarrow f_{fuse}(F_R, F_D)$;
end
$L \leftarrow Decoder(F_R)$ // Decoder;

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Dataset

We use the public `Pothole-600` dataset [11] for the experiments. This dataset was recorded using a ZED stereo camera. Disparity images are provided by applying the PT-SRP algorithm [32] on the captured stereo images. There are in total 600 pairs of RGB and disparity images with the resolution of 400×400. Hand-labeled ground-truth segmentation masks are provided along with the images. In our experiments, we re-scale the image size to 512×512.

We follow the image splitting scheme of [11]. The training set contains 240 pairs of RGB and disparity images. The validation and testing sets contain 180 pairs of images, respectively. We generate an augmented training set by flipping, rotating, and cropping the images of the original training set. The augmented training set contains 960 pairs of images. The details of the augmentation process is that we first flip the original training images along the x-axis to obtain a new set of 240 image pairs (Set A), and then flip the images of Set A along the y-axis to get Set B. We randomly rotate the 240 original training images from 0° to 10° to generate the Set C. Finally, we randomly crop (within 50 pixels) the images of Set C to generate the Set D. The augmented training set is the union of Set A, Set B, Set C, and Set D. So the total number of image pairs of the augmented training set is $4\times\ 240 = 960$. The final training set is the union of the original training set and the augmented training set. So there are totally $240 + 960 = 1200$ pairs of images for training.

B. Training Details

We implement our proposed MAFNet by using PyTorch. Our MAFNet is trained on a PC with an Intel 3.6 GHz i7 CPU and two NVIDIA RTX 3060 (12 GB GPU RAM) graphics cards, but we only use one card to train the network. We test our network inference speed on three different PCs. The first one is the aforementioned NVIDIA RTX 3060 PC. The second one is with an AMD R7 2.9 GHz CPU and a single NVIDIA RTX 2060 (6 GB GPU RAM) graphics card. The third one is with an Intel 3.6 GHz i7 CPU and two NVIDIA RTX 3090 (24GB GPU RAM) graphics cards.

We first create a baseline by replacing the last encoder of RTFNet with a four-head transformer module, and replacing the elementwise addition fusion of RTFNet with the CAF-1F2A module. We train this baseline with the pretrained weight of ResNet provided by PyTorch. Then, we train our

TABLE II

ABLATION STUDY RESULTS (%) OF THE VARIANTS THAT ADOPT THE CAF-1A2F AND DAF-1A2F ON THE POTHOLE-600 TESTING SET. *AF* PREFIX REPRESENTS THE CAF-1A2F AND THE DAF-1A2F ARE ADOPTED IN THE MODULES. *NF* PREFIX REPRESENTS THAT THERE IS ON FUSION MODULE IN THE NETWORK. *1T4H* REPRESENTS THAT THE TRANSFORMER MODULE HAS ONE LAYER AND FOUR ATTENTION HEADS IN MSA. WE USE BOLD FONT TO HIGHLIGHT THE BEST RESULTS FOR EACH CLASS

Variants	Background				Pothole				mPre	mRec	mFsc	mIoU
	Pre	Rec	Fsc	IoU	Pre	Rec	Fsc	IoU				
NF-1T4H	97.99	**99.31**	**98.65**	**97.33**	**92.04**	79.71	85.43	74.57	**95.02**	89.51	92.04	85.95
AF-1T4H	98.65	98.48	98.56	97.17	85.06	86.57	85.81	75.15	91.86	92.52	92.19	86.16
AF-1T8H	98.36	98.41	98.39	96.82	84.10	83.62	83.86	72.20	91.23	91.02	91.12	84.51
AF-1T16H	98.44	98.83	98.63	97.31	87.84	84.38	**86.08**	**75.56**	93.14	91.61	**92.36**	**86.43**
AF-1T32H	98.56	98.52	98.54	97.12	85.32	85.64	85.48	74.64	91.94	92.08	92.01	85.88
AF-2T4H	98.65	98.02	98.33	96.72	81.46	86.59	83.94	72.33	90.05	92.31	91.14	84.53
AF-2T8H	98.75	98.23	98.49	97.03	83.24	87.64	85.38	74.49	91.00	92.93	91.94	85.76
AF-2T16H	98.73	98.34	98.53	97.11	84.05	87.33	85.66	74.92	91.39	92.84	92.10	86.01
AF-2T32H	98.38	98.84	98.61	97.25	87.82	83.73	85.73	75.02	93.10	91.28	92.17	86.13
AF-3T4H	98.75	98.33	98.54	97.12	84.03	87.62	85.78	75.11	91.39	92.97	92.16	86.12
AF-3T8H	98.42	98.63	98.53	97.09	86.06	84.18	85.11	74.08	92.24	91.41	91.82	85.59
AF-3T16H	98.53	98.48	98.50	97.05	84.89	85.33	85.11	74.08	91.71	91.90	91.81	85.56
AF-3T32H	98.54	98.41	98.48	97.00	84.37	85.48	84.93	73.80	91.46	91.95	91.70	85.40
AF-4T4H	98.44	98.32	98.38	96.81	83.44	84.49	83.96	72.36	90.94	91.41	91.17	84.59
AF-4T8H	**98.97**	98.01	98.49	97.02	81.92	**89.82**	85.69	74.96	90.45	**93.92**	92.09	85.99
AF-4T16H	98.85	97.89	98.37	96.79	80.83	88.60	84.54	73.22	89.84	93.25	91.45	85.00
AF-4T32H	98.44	98.55	98.50	97.04	85.41	84.40	84.90	73.76	91.92	91.48	91.70	85.40
AF-5T4H	98.58	98.46	98.52	97.08	84.82	85.81	85.31	74.38	91.70	92.13	91.91	85.73
AF-5T8H	98.69	97.63	98.16	96.38	78.65	87.09	82.65	70.44	88.67	92.36	90.41	83.41
AF-5T16H	98.46	98.24	98.35	96.75	82.84	84.65	83.73	72.02	90.65	91.45	91.04	84.39
AF-5T32H	98.82	97.96	98.39	96.83	81.31	88.36	84.69	73.44	90.07	93.16	91.54	85.14

MAFNet with the pretrained weight of this baseline. Specifically, we reuse the weight of the initial block, the first three stages of the encoders and all stages of the decoder of the baseline. We use the stochastic gradient descent (SGD) optimization function with the initial learning rate of 0.1 and the momentum of 0.9. The learning rate is decreased using the exponential strategy with the decay rate of 0.95. The training is stopped when the validation loss converges.

C. Evaluation Metrics

We employ four quantitative metrics that are used in to evaluate the semantic segmentation performance, the F-score (Fsc), the recall (Rec), the precision (Pre), and the intersection over union (IoU). They are calculated as follows:

$$\text{Fsc} = \frac{2 \times \text{Rec} \times \text{Pre}}{\text{Rec} + \text{Pre}} \tag{7}$$

$$\text{Rec} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

$$\text{Pre} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{9}$$

$$\text{IoU} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives} + \text{False Positives}}. \tag{10}$$

There are two classes (i.e., pothole and unlabelled background) in the ground-truth images of the dataset. We calculate the Fsc, Rec, Pre, and IoU values for the two classes. The average values (denoted as mFsc, mRec, mPre, and mIoU) are the mean values over the two classes.

D. Ablation Study

1) Ablation on Data Fusion: In the ablation study, we first test MAFNet by removing the fusion module. We name this variant as NF-1T4H because it has No the proposed fusion (NF) module. The fusion module is replaced with elementwise addition. 1T4H means that in the transformer module there is one transformer layer and four self-attention heads (1T4H).

As aforementioned, we design two structures for the fusion modules CAF and DAF. We name the variants with AF prefix when the CAF-1A2F and the DAF-1A2F are adopted in the modules. Similarly, we name the variants with FA prefix when the CAF-1F2A and the DAF-1F2A are adopted in the modules.

In MAFNet, the first two fusion modules are CAFs, and the last three fusion modules are DAFs. So all the AF variants include two CAF-1A2Fs and three DAF-1A2Fs, and all the FA variants include two CAF-1F2As and three DAF-1F2As.

2) Ablation on the Transformer Module: For the transformer module, we try different numbers of transformer layers and attention heads. As the number of layers increases, the transformer module is expected to be more powerful, but it also increases the number of module parameters, thereby increases the time cost. So, the number of transformer layers is not the more the better. To tradeoff the performance and speed, the number of transformer layers is chosen from 1 to 5. The more attention heads of MSA, the more information can be extracted, but at the same time, it will make the network more redundant, make the network larger, and reduce the inference speed of the network. To tradeoff the performance and the speed, the number of attention heads of MSA is not the more the better. We design the number of attention heads of MSA to be 4, 8, 16, and 32. For the naming, we take AF-3T8H as an example, the name means that the transformer module of the variant has three transformer layers and eight attention heads.

3) Quantitative Results: Table II displays the results of NF-1T4H and all the AF variants we designed on the testing

TABLE III

ABLATION STUDY RESULTS (%) OF THE VARIANTS THAT ADOPT THE CAF-1F2A AND DAF-1F2A ON THE POTHOLE-600 TESTING SET. *FA* PREFIX REPRESENTS THE CAF-1F2A AND THE DAF-1F2A ARE ADOPTED IN THE MODULES. *NF* PREFIX REPRESENTS THAT THERE IS ON FUSION MODULE IN THE NETWORK. *1T4H* REPRESENTS THAT THE TRANSFORMER MODULE HAS ONE LAYER AND FOUR ATTENTION HEADS IN MSA. WE USE BOLD FONT TO HIGHLIGHT THE BEST RESULTS FOR EACH CLASS

Variants	Background				Pothole				mPre	mRec	mFsc	mIoU
	Pre	Rec	Fsc	IoU	Pre	Rec	Fsc	IoU				
NF-1T4H	97.99	**99.31**	98.65	97.33	**92.04**	79.71	85.43	74.57	**95.02**	89.51	92.04	85.95
FA-1T4H	98.33	98.75	98.54	97.13	87.02	83.31	85.13	74.10	92.68	91.03	91.83	85.62
FA-1T8H	98.54	98.33	98.44	96.92	83.69	85.50	84.58	73.29	91.12	91.92	91.51	85.10
FA-1T16H	**98.92**	97.43	98.17	96.40	77.68	**89.34**	83.11	71.09	88.30	93.39	90.64	83.75
FA-1T32H	98.65	98.29	98.47	96.99	83.54	86.59	85.04	73.97	91.09	92.44	91.75	85.48
FA-2T4H	98.82	98.14	98.48	97.00	82.63	88.32	85.38	74.49	90.72	93.23	91.93	85.75
FA-2T8H	98.26	98.97	98.61	97.26	88.89	82.52	85.59	74.81	93.57	90.74	92.10	86.03
FA-2T16H	98.56	98.02	98.29	96.64	81.29	85.74	83.45	71.61	89.93	91.88	90.87	84.12
FA-2T32H	98.40	98.63	98.51	97.07	85.98	83.99	84.97	73.87	92.19	91.31	91.74	85.47
FA-3T4H	98.65	97.90	98.28	96.61	80.57	86.65	83.50	71.67	89.61	92.28	90.89	84.14
FA-3T8H	98.13	98.90	98.51	97.06	88.08	81.16	84.48	73.12	93.10	90.03	91.49	85.09
FA-3T16H	98.83	98.54	**98.69**	**97.41**	85.88	88.39	**87.11**	**77.17**	92.35	**93.47**	**92.90**	**87.29**
FA-3T32H	98.49	98.25	98.37	96.80	82.99	85.00	83.98	72.39	90.74	91.63	91.18	84.59
FA-4T4H	98.66	98.38	98.52	97.08	84.26	86.70	85.46	74.62	91.46	92.54	91.99	85.85
FA-4T8H	98.40	98.87	98.63	97.30	88.13	83.92	85.97	75.40	93.26	91.39	92.30	86.35
FA-4T16H	98.73	98.52	98.63	97.29	85.58	87.41	86.48	76.18	92.16	92.97	92.56	86.74
FA-4T32H	98.48	98.35	98.41	96.88	83.77	84.82	84.29	72.85	91.12	91.58	91.35	84.86
FA-5T4H	98.60	98.58	98.59	97.22	85.85	86.05	85.95	75.36	92.23	92.31	92.27	86.20
FA-5T8H	98.53	98.70	98.61	97.26	86.79	85.30	86.04	75.50	92.66	92.00	92.33	86.38
FA-5T16H	98.33	98.96	98.65	97.33	88.96	83.28	86.03	75.48	93.65	91.12	92.34	86.41
FA-5T32H	98.70	98.34	98.52	97.08	84.01	87.07	85.51	74.69	91.35	92.70	92.01	85.88

set of the `Pothole-600` dataset. We can see that the proposed fusion modules could generally improve the performance. The values for the background are close to 100% and similar between all the variants we designed. We conjecture the reason could be that background occupy most of the pixels in the images so that the networks learn well for the background.

Table III displays the results for all of AF variants in Table III on the `Pothole-600` testing set which contains 180 pairs of images. From Table II, we can see that compared with the AF series variants in Table II, the best results of the FA series variants in Table II are concentrated on the FA-3T16H variant except the Pre of all classes and the Rec of the background and the pothole. The best Pre for background is 98.92%, for pothole is 92.04%, and for mPre is 95.02%. The best Rec for background is 99.31%, for pothole is 89.34%, and for mRec is 93.47%. The best Fsc for background is 98.69%, for pothole is 87.11%, and for mFsc is 92.90%. The best IoU for background is 97.41%, for pothole is 77.17%, and for mIoU is 87.29%.

Comparing AF-4T8H with FA-3T16H, although the Rec result of FA-3T16H is slightly lower than the result of AF-4T8H, the result of IoU is much larger than that of AF-4T8H. Comparing the segmentation results of the two structures of FA-3T16H and AF-1T16H, the result of AF-1T16H is lower than the result of FA-3T16H in all evaluation metrics except the Rec of background and the Pre of background and mPre. The above results show that when the feature maps are fused by the CAF-1F2A and the DAF-1F2A, the final segmentation result is better and more stable.

Tables II and III show that FA-3T16H generally has the best performance. So we use this variant in the following comparative study. This result is also in line with our intuitive understanding. Although the increase in the number of transformer layers can improve the learning ability of the entire network, the amount of data we use is relatively small, which may cause the network over-fitting during the learning process. According to the results in Tables II and III, we can find that the results do not increase monotonously with the increase of the number of transformer layers. In MSA, more attention heads can focus on more information. However, no matter how many attention heads there are, the data length of all information is fixed. In our network, the data length of all attention heads is 1024. More attention heads means that the length of the data representing a piece of information is shorter, which makes them unable to fully express the information. In other words, the less attention head means that too long data is used to represent a feature information, which is redundant and allows the network to extract less feature information. In short, the appropriate number of transformer layers and the number of attention heads are very important for a good semantic segmentation result.

From all the experimental results, it can be seen that among the variants with the same number of transformer layers and self-attention heads, the results of the AF variants are inferior to those of the FA variants. For example, the mIoU value of FA-4T8H is higher than that of AF-4T8H. It shows that the fusion schemes have a significant impact on the network performance. We choose the best fusion scheme (i.e., the FA structure) to design our MAFNet.

4) Inference Speed: We use the same testing set to test the inference speed of all variants on the above three graphics cards with the input resolution of 512 $\times$ 512. The inference speed are displayed in Fig. 4 (using RTX 2060 graphics card), Fig. 5 (using RTX 3060 graphics card), and Fig. 6 (using RTX 3090 graphics card), respectively. In these figures, *1T*

TABLE IV

COMPARATIVE RESULTS (%) ON THE POTHOLE-600 TESTING SET. *Methods* REPRESENTS THE NAME OF THE NETWORKS. *DLV3+* REPRESENTS DEEPLABV3+. *r*, *d* AND *6c* REPRESENT THAT THE NETWORKS ARE TRAINED AND TEST WITH ONLY RGB IMAGES, ONLY DISPARITY IMAGES, AND BOTH RGB IMAGES AND DISPARITY IMAGES, RESPECTIVELY. WE USE BOLD FONT TO HIGHLIGHT THE BEST RESULTS FOR EACH CLASS. THE DATA IN THE TABLE DIRECTLY PROVES THE SUPERIORITY OF OUR PROPOSED MAFNET

Variants	Background				Pothole				mPre	mRec	mFsc	mIoU
	Pre	Rec	Fsc	IoU	Pre	Rec	Fsc	IoU				
UNet++(r)	97.51	96.35	96.93	94.04	67.48	75.48	71.26	55.35	82.50	85.92	84.09	74.69
UNet++(d)	98.32	98.73	98.53	97.09	86.77	83.22	84.96	73.85	92.55	90.97	91.74	85.47
UNet++(6c)	98.36	98.73	98.54	97.13	86.84	83.54	85.16	74.15	92.60	91.13	91.85	85.64
PSPNet(r)	96.79	95.04	95.90	92.13	58.06	68.53	62.87	45.84	77.43	81.78	79.38	68.99
PSPNet(d)	98.49	98.08	98.28	96.63	81.59	95.04	83.28	71.35	90.04	91.56	90.78	83.99
PSPNet(6c)	98.63	98.34	98.48	97.01	83.87	86.35	85.09	74.05	91.25	92.34	91.79	85.53
DLV3+(r)	96.51	96.33	96.42	93.08	64.04	65.21	64.62	47.73	80.27	80.77	80.52	70.41
DLV3+(d)	98.49	98.47	98.48	97.01	84.81	84.91	84.86	73.7	91.65	91.69	91.67	85.35
DLV3+(6c)	98.71	98.39	98.55	97.14	84.43	87.18	85.78	75.11	91.57	92.78	92.17	86.12
MA-Net(r)	96.75	94.10	95.41	91.22	53.79	68.50	60.26	43.12	75.27	81.30	77.83	67.17
MA-Net(d)	98.32	98.31	98.32	96.69	83.16	83.26	83.21	71.25	90.74	90.79	90.76	83.97
MA-Net(6c)	98.24	98.43	98.33	96.72	84.03	82.37	83.19	71.22	91.13	90.40	90.76	83.97
TransUNet(r)	96.63	97.69	97.16	94.47	74.13	66.00	69.83	53.65	85.38	81.85	83.49	74.06
TransUNet(d)	98.13	98.81	98.47	96.98	87.28	81.18	84.12	72.59	92.70	90.00	91.29	84.79
TransUNet(6c)	98.43	98.39	98.41	96.87	83.99	84.35	84.17	72.67	91.21	91.37	91.29	84.77
RTFNet	98.01	98.98	98.49	97.03	88.70	79.95	84.10	72.56	93.35	89.46	91.29	84.79
FuseNet	98.19	98.76	98.47	96.99	86.86	81.82	84.26	72.81	92.53	90.29	91.37	84.90
SegNet	96.47	**99.55**	97.99	96.05	93.40	63.67	75.72	60.93	94.94	81.61	86.85	78.49
MFNet	98.22	98.73	98.48	97.00	86.68	82.19	84.37	72.97	92.45	90.46	91.43	84.99
AA-RTFNet	97.56	99.43	98.49	97.02	**93.02**	75.15	83.14	71.14	**95.29**	87.29	90.81	84.08
MAFNet(Ours)	**98.83**	98.54	**98.69**	**97.41**	85.88	**88.39**	**87.11**	**77.17**	92.35	**93.47**	**92.90**	**87.29**

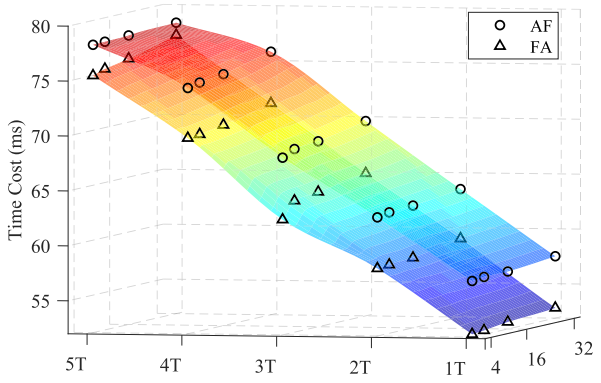

Fig. 4. Inference speed for each variant evaluated on RTX 2060. The figure clearly demonstrates that increasing the number of attention heads can only slightly increases the inference time. The figure is best viewed in color.

represents that there is one transformer layer in the transformer module. The number 4, 16, and 32 represent that there are 4, 16, and 32 attention heads in MSA, respectively. *AF* represents that the CAF-1A2F and the DAF-1A2F are adopted in the network. *FA* represents that the CAF-1F2A and the DAF-1F2A are adopted in the network. Based on the measured discrete results, we fit the distribution of the inference time in 3-D space, which could be helpful to analyze the underlying patterns between different variants. In these figures, the discrete points are the true measured values, and the surface between the discrete points is interpolated. We use different colors to represent different inference time, and the time cost increases from blue to red.

As we can see, all the variants exhibit a real-time inference speed on RTX 3060 (the fast speed is 21.58 ms from FA-1T4H and the slowest is 34.03 ms from AF-5T32H) and RTX 3090 (the fast speed is 14.91 ms from FA-1T4H and the slowest is 23.79 ms from AF-5T32H), and an acceptable speed on RTX 2060 (the fast speed is 52.56 ms from FA-1T4H and the slowest is 78.48 ms from AF-5T32H).

These figures clearly demonstrate that increasing the number of attention heads for MSA in transformer only slightly increases the inference time for one image. The line between the measurement points of the variants with the same transformer is approximately parallel to the coordinate plane of the attention head and the number of network layers, indicating that the inference time of these variants is almost the same. However, the increase in the number of layer of transformer greatly increase the inference time. The color of the line between the measurement points of the variants with the same attention head changes sharply, indicating that the reasoning time gap of these variants is huge. The inference speed increases by the same amount when the transformer module increases a layer. Compared with the CAF-1F2A and the DAF-1F2A, the CAF-1A2F, and the DAF-1A2F require more time cost to infer an image.

5) Number of Network Parameters: The number of a network parameter could measure the size of a network. According to our observation, the number of parameters for the variants satisfies the following:

$$P = B_i + n(L + h \times H) \tag{11}$$

where the P represents the number of parameters of a network. B_i represents the type of network, which is 37 409 156 for AF variants, and is 36 969 444 for FA variants. n represents the number of transformer layer. L represents the basic parameters

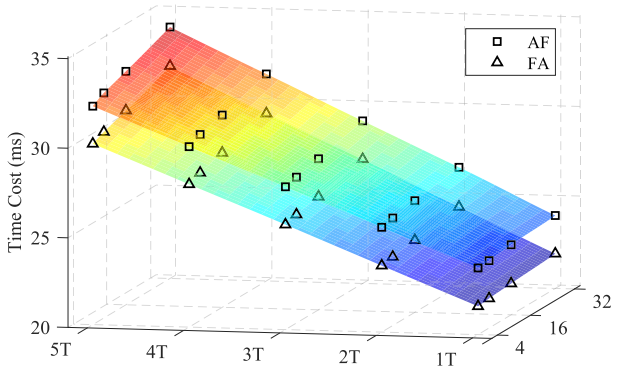

Fig. 5. Inference speed for each variant evaluated on RTX 3060. The figure clearly demonstrates that increasing the number of attention heads can only slightly increases the inference time. The figure is best viewed in color.

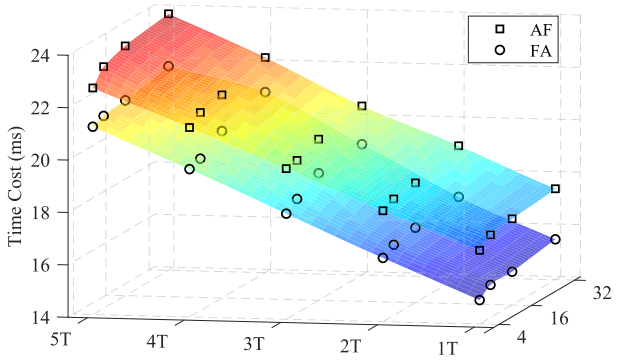

Fig. 6. Inference speed for each variant evaluated on RTX 3090. The figure clearly demonstrates that increasing the number of attention heads can only slightly increases the inference time. The figure is best viewed in color.

of each transformer layer, which is 25 192 448. h represents the number of attention heads. H represents the increase of the parameter when an attention head is added, which is 8. We can see that the number of network parameters increases with the number of transformer layers and attention heads.

E. Comparative Study

We compare our proposed MAFNet with UNet++ [33], MA-Net [34], PSPNet [35], TransUNet [28], DeepLabv3+ [24], FuseNet [36], SegNet [37], RTFNet [13], MFNet [38], and AA-RTFNet [11] in this section. The comparison networks can be divided into two categories: single-modal networks and multimodal networks. To ensure fair comparisons, we train the single-modal networks with the three-channel RGB images, the three-channel disparity images and six-channel RGB-disparity images, respectively. The input layers of single-modal networks are modified to accommodate the six-channel RGB-disparity images. We use the implementations from the library[2] for the single-modal networks except TransUNet, and use the implementation from the library[3] for TransUNet. All parameters of the function adopt default values in the libraries. We also modify the one-channel encoder in FuseNet, RTFNet, and AA-RTFNet to accept the three-channel disparity images. All the networks are trained until the validation loss converges.

[2]https://github.com/qubvel/segmentation_models.pytorch
[3]https://github.com/The-AI-Summer/self-attention-cv

TABLE V

INFERENCE SPEED FOR EACH NETWORK. THE NETWORKS ARE TESTED WITH THE SAME TESTING SET ON RTX 2060, RTX 3060, AND RTX 3090, RESPECTIVELY. *DLV3+* REPRESENTS DEEPLABV3+. ms REPRESENTS MILLISECOND AND *FPS* REPRESENTS THE FRAME-PER-SECOND. WE USE SIX-CHANNEL DATA TO TEST THE SINGLE-MODE NETWORKS

Methods	RTX 2060		RTX 3060		RTX 3090	
	ms	FPS	ms	FPS	ms	FPS
UNet++	54.64	18.30	24.47	40.87	11.53	86.69
PSPNet	10.01	99.85	4.64	215.42	3.09	323.19
DLV3+	31.64	31.61	11.53	86.70	7.94	125.89
MA-Net	33.78	29.61	13.99	71.49	9.50	105.31
TransUNet	111.55	8.96	60.38	16.56	28.56	35.01
SegNet	21.34	46.86	10.57	94.58	7.66	130.61
MFNet	15.38	65.04	6.72	148.90	6.26	159.65
FuseNet	97.53	10.25	60.64	16.49	31.88	31.36
AA-RTFNet	39.21	25.50	24.15	41.41	19.99	50.03
RTFNet	45.33	22.06	18.92	52.87	12.15	82.31
MAFNet(Ours)	64.27	15.56	26.31	38.01	18.85	53.06

1) Overall Results: We display the quantitative comparative results for all the networks in Table IV. As we can see, our proposed MAFNet achieves the best results in terms of all the metrics across all the networks, except the Rec values of the background. The superiority of our MAFNet is demonstrated by the comparative results.

From Table IV, we can see that the results for all the single-modal networks trained with disparity images are better than those trained with RGB images. This indicates that disparity images are beneficial to the pothole segmentation. The second and third best networks are single-modal DeepLabv3+ (DLV3+) and single-modal UNet++, both of which are trained with six-channel RGB-disparity images. Their performance is not only better than multimodal networks but also better than themselves trained with three-channel data. For the other networks, the results for the single-modal networks trained with six-channel RGB-disparity images are the best, compared with those trained alone with RGB images or disparity images. It proves that using multimodal information is effective to improve the performance. From the results, we can find that our MAFNet significantly outperforms RTFNet (higher by 2.5%) and AA-RTFNet (higher by 3.21%). It shows that the transformer, CAF, and DAF in the encoder have a significant impact on the performance of the network.

It should be denoted that AA-RTFNet is designed based on RTFNet, and has achieved better results on the augmented `Pothole-600` dataset [11] than RTFNet. However, the results for AA-RTFNet in our experiments are worse than those of RTFNet. We think the reason could lie in the different augmentation methods adopted in the two works.

2) Inference Speed: The inference running time is a crucial evaluation metrics besides accuracy. We test these networks with the same testing set on RTX 2060, RTX 3060, and RTX 3090, respectively. To ensure fair comparison, we test the single-modal networks with six-channel data. Table V displays the average running time on the testing set. The input images resolution is 512 × 512. According to Table V, our MAFNet (FA-3T16H) exhibits a real-time inference speed on RTX 3060 and RTX 3090, and an acceptable speed on RTX

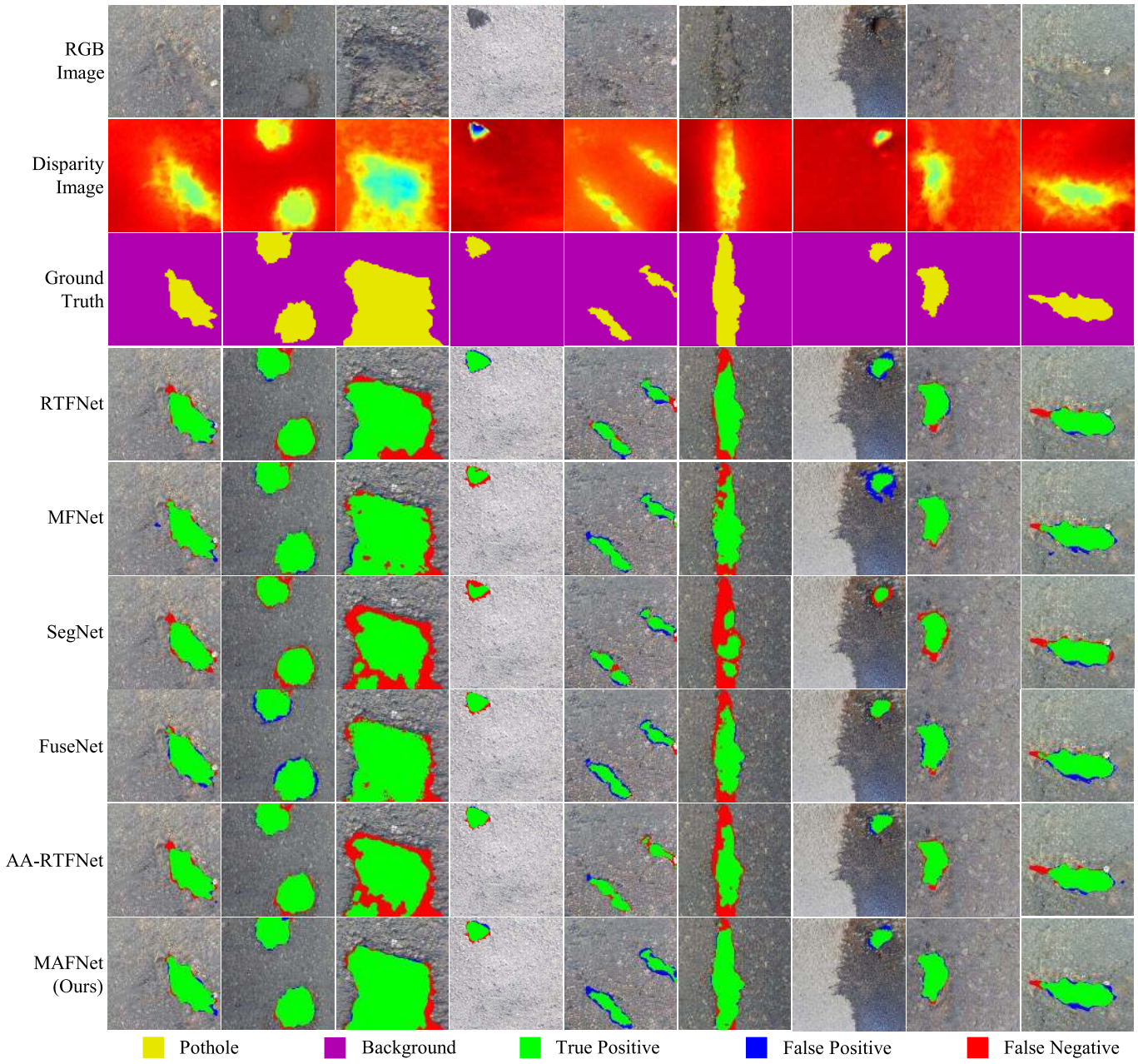

Fig. 7. Sample qualitative demonstrations for the networks. Every column shows the results for different networks testing with the same RGB and disparity images. The top three rows display the RGB images, disparity images, and the ground truth, respectively. Rows from the 3rd to the 9th show the experimental results for RTFNet, MFNet, SegNet, FuseNet, AA-RTFNet, and our MAFNet on the Pothole-600 testing set. The comparative results shows that our proposed network MAFNet generally exhibits better performance. The figure is best viewed in color.

2060. Although our inference speed is the slowest among all multimodal fusion networks, our segmentation performance is better than others. In addition, our inference speed is faster than the TransUNet that also uses the convolution and transformer structure.

3) Qualitative Demonstrations: The qualitative demonstrations are displayed in Fig. 7. In general, our proposed MAFNet exhibits more accurate and robust segmentation performance than the others. In particular, our network significantly outperforms the other networks on pothole edges. But there are still some false detected pixels on edges compared to the ground truth. We think the reason could be the multiple resolution reduction from 512×512 to 16×16 in the encoders, which leads to inappropriate edge spatial information for decoding. This reason could be validated from that the edge segmentation results of larger potholes are better than those of the small potholes. As we can see from the columns 4, 5, and 7, the smaller the potholes area, the less data are used for encoding in the last layer of encoders, so that it is more difficult to decode the edge information.

V. Conclusion

We proposed here a novel RGB-disparity fusion network for road-pothole segmentation, in which two data-fusion modules based on channel attention and dual attention were proposed. To find better data-fusion module structures, the appropriate

numbers of transformer layers, and attention heads, we tried a number of variants and evaluated their segmentation performance as well as the inference speed. We also compared our network with state-of-the-art single-modal and multimodal semantic segmentation networks. The experimental results demonstrate the superiority of our network.

However, there still exist some limitations. First, our network can only run at a real-time inference speed on RTX 3060 or better cards, which might be not suitable to be used on resource-constrained vehicles. This could be alleviated by using lightweight technologies, such as knowledge distillation or model compression. Second, the edge information might be ignored due to excessive downsampling in the encoder. We will use skip connections to introduce the feature maps from the encoder to the same level of the decoder to reduce the information loss. At the same time, edge features will also be learned by introducing a new loss function.

REFERENCES

[1] T. Verster and E. Fourie, "The good, the bad and the ugly of south African fatal road accidents," *South Afr. J. Sci.*, vol. 114, no. 7/8, pp. 63–69, Jul. 2018.

[2] L. Luo, M. Feng, J. Wu, and R. Leung, "Autonomous pothole detection using deep region-based convolutional neural network with cloud computing," *Smart Struct. Syst.*, vol. 24, pp. 745–757, Dec. 2019.

[3] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4906–4911, Nov. 2020.

[4] S. Gupta, P. Sharma, D. Sharma, V. Gupta, and N. Sambyal, "Detection and localization of potholes in thermal images using deep neural networks," *Multimedia Tools Appl.*, vol. 79, nos. 35–36, pp. 26265–26284, Sep. 2020.

[5] Y. L. Chen *et al.*, "Inexpensive multimodal sensor fusion system for autonomous data acquisition of road surface conditions," *IEEE Sensors J.*, vol. 16, no. 21, pp. 7731–7743, Nov. 2016.

[6] R. Ravi, H. Ayman, and D. Bullock, "Pothole mapping and patching quantity estimates using LiDAR-based mobile mapping systems," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, Jul. 2020, Art. no. 036119812092700.

[7] Y. Yang, S. Cao, S. Huang, and W. Wan, "Multimodal medical image fusion based on weighted local energy matching measurement and improved spatial frequency," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–16, 2021.

[8] Y.-B. Liu, M. Zeng, and Q.-H. Meng, "Unstructured road vanishing point detection using convolutional neural networks and heatmap regression," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.

[9] J. Ma, L. Tang, M. Xu, H. Zhang, and G. Xiao, "STDFusionNet: An infrared and visible image fusion network based on salient target detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[10] R. Fan, U. Ozgunalp, Y. Wang, M. Liu, and I. Pitas, "Rethinking road surface 3-D reconstruction and pothole detection: From perspective transformation to disparity map segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5799–5808, Jul. 2022.

[11] R. Fan, H. Wang, M. J. Bocus, and M. Liu, "We learn better road pothole detection: From attention aggregation to adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 285–300.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 770–778.

[13] Y. Sun, W. Zuo, and M. Liu, "RTFNet: RGB-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2576–2583, Jul. 2019.

[14] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.

[15] V. Pereira, S. Tamura, S. Hayamizu, and H. Fukai, "A deep learning-based approach for road pothole detection in Timor Leste," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat. (SOLI)*, Jul. 2018, pp. 279–284.

[16] Q. Mei and M. Gül, "A cost effective solution for pavement crack inspection using cameras and deep neural networks," *Construct. Building Mater.*, vol. 256, Sep. 2020, Art. no. 119397.

[17] S. Anand, S. Gupta, V. Darbari, and S. Kohli, "Crack-pot: Autonomous road crack and pothole detection," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2018, pp. 1–6.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*.

[19] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated road crack detection using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 5212–5215.

[20] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[21] L. K. Suong and K. Jangwoo, "Detection of potholes using a deep convolutional neural network," *Universal Comput. Sci.*, vol. 24, no. 9, pp. 1244–1257, 2018.

[22] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3536–3550, Aug. 2020.

[23] S. Masihullah, R. Garg, P. Mukherjee, and A. Ray, "Attention based coupled framework for road and pothole segmentation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 5812–5819.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision—ECCV 2018*. Cham, Switzerland: Springer, 2018, pp. 833–851.

[25] Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks based potholes detection using thermal imaging," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 3, pp. 578–588, Mar. 2022.

[26] G. H. Beckman, D. Polyzois, and Y.-J. Cha, "Deep learning-based automatic volumetric damage quantification using depth camera," *Autom. Construct.*, vol. 99, pp. 114–124, Mar. 2019.

[27] Y. Pan, X. Zhang, G. Cervone, and L. Yang, "Detection of asphalt pavement potholes and cracks based on the unmanned aerial vehicle multispectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 10, pp. 3701–3712, Oct. 2018.

[28] J. Chen *et al.*, "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[30] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[31] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[32] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3D reconstruction based on dense subpixel disparity map estimation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3025–3035, Jun. 2018.

[33] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 3–11.

[34] T. Fan, G. Wang, Y. Li, and H. Wang, "MA-Net: A multi-scale attention network for liver and tumor segmentation," *IEEE Access*, vol. 8, pp. 179656–179665, 2020.

[35] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jul. 2017, pp. 6230–6239.

[36] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. Asian Conf. Comput. Vision*, 2016, pp. 213–228.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Oct. 2016.

[38] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5108–5115.

Zhen Feng (Graduate Student Member, IEEE) received the B.S. degree in automation and the M.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2017 and 2019, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. He is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering, Harbin Institute of Technology.

His current research interests include semantic segmentation, computer vision, autonomous driving, and deep learning.

Yanning Guo received the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2008 and 2012, respectively.

He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology, where he currently teaches and performs research in the fields of deep space exploration, satellite attitude control, and nonlinear control.

Qing Liang (Student Member, IEEE) received the B.A. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2013, and the master's degree in instrument science and technology from Beihang University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

His current research interests include sensor fusion, low-cost localization, and mobile robots.

M. Usman Maqbool Bhutta received the B.Sc. degree in mathematics and physics from Bahauddin Zakariya University, Multan, Pakistan, in 2007, the M.Phil. degree in image processing and communication and the M.Sc. degree in electronics from the Department of Electronics, Quaid-i-Azam University, Islamabad, Pakistan, in 2012 and 2009, respectively, and the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2021.

He is currently a Post-Doctoral Fellow with The Chinese University of Hong Kong, Hong Kong. His research interests include long-term place recognition for autonomous systems and mixed-reality applications.

Dr. Bhutta won the Best Student Paper Finalist Award at IEEE-CYBER in 2018.

Hengli Wang (Member, IEEE) received the B.E. degree from Zhejiang University, Hangzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

His research interests include computer vision, robot navigation, and deep learning.

Ming Liu (Senior Member, IEEE) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zürich, Zürich, Switzerland, in 2013.

During his master's degree with Tongji University, he stayed one year at Erlangen-Nünberg University, Erlangen, Germany, and the Fraunhofer Institute IISB, Erlangen, as a Master Visiting Scholar. He is currently an Associate Professor with the Department of Electronic and Computer Engineering, the Department of Computer Science and Engineering, and the Cheng Kar-Shun Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong. His research interests include dynamic environment modeling, deep learning for robotics, 3-D mapping, machine learning, and visual control.

Yuxiang Sun (Member, IEEE) received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017.

He was a Research Associate with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. He is currently a Research Assistant Professor with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. His current research interests include autonomous driving, deep learning, robotics, and autonomous systems, and semantic scene understanding.

Dr. Sun serves as an Associate Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS, the IEEE International Conference on Robotics and Automation, and the IEEE/RSJ International Conference on Intelligent Robots and Systems.