RTFNet: RGB-Thermal Fusion Network for Semantic Segmentation of Urban Scenes

Yuxiang Sun, Weixun Zuo and Ming Liu, *Senior Member, IEEE*

***Abstract*—Semantic segmentation is a fundamental capability for autonomous vehicles. With the advancements of deep learning technologies, many effective semantic segmentation networks have been proposed in recent years. However, most of them are designed using RGB images from visible cameras. The quality of RGB images is prone to be degraded with unsatisfied lighting conditions, such as darkness and glares of oncoming headlights, which imposes critical challenges for the networks that use only RGB images. Different from visible cameras, thermal imaging cameras generate images using thermal radiations. They are able to see under various lighting conditions. In order to enable robust and accurate semantic segmentation for autonomous vehicles, we take the advantage of thermal images and fuse both the RGB and thermal information in a novel deep neural network. The main innovation of this letter is the architecture of the proposed network. We adopt the Encoder-Decoder design concept. ResNet is employed for feature extraction and a new decoder is developed to restore the feature map resolution. The experimental results prove that our network outperforms the state of the arts.**

***Index Terms*—Semantic Segmentation, Urban Scenes, Deep Neural Network, Thermal Images, Information Fusion.**

I. Introduction

SEMANTIC image segmentation of urban scenes is of critical significance to autonomous vehicle systems. For instance, it is a fundamental component for scene understanding, which ensures reliable operations of autonomous vehicles in real-world urban environments. Many other applications, such as path planing [1] and environment modelling [2]–[5], could also benefit from semantic segmentation.

Recent advancements of deep learning technologies have attracted great attentions from both the academia and industry. Many effective semantic segmentation algorithms based on deep neural networks have been proposed in recent years [6]–[12]. Among most of the state of the arts, Convolutional Neural Network (CNN) has been widely used and achieved great success [13]. However, the mainstream semantic segmentation networks are designed to work with 3-channel RGB images captured by visible cameras. The segmentation performance is prone to be degraded when the lighting conditions are not satisfied. For instance, most algorithms would fail to correctly segment objects in almost total darkness.

Thermal imaging cameras are able to see under various lighting conditions [14]. Different from visible cameras that work in visible light spectrum ranging from 0.4μm to 0.7μm, they form images using thermal radiations emitted by all matters with temperatures above absolute zero [15]. Almost anything that creates heat can be seen with thermal. Thermal imaging cameras are initially invented for military uses, but the price has dropped down in recent years so that the cameras could be increasingly used in civilian applications, such as remote sensing [16], autonomous surveillance [17] and Advanced Driver Assistance Systems (ADAS) [18]. It should be noted that thermal imaging cameras are different from Near-Infra-Red (NIR) cameras. The former detects wavelength up to 14μm, while the latter merely works in the NIR spectrum that ranges from 0.7μm to 1.4μm [14]. The NIR spectrum is not visible to human eyes, but can be sensed by silicon image sensors. Consequently, NIR cameras can be simply manufactured by adding a NIR filter in the front of the CMOS or CCD sensors of visible cameras, whereas the manufacturing of thermal imaging cameras are much more complicated.

In this letter, we exploit the benefits of images captured in thermal radiations, and fuse the RGB and thermal images to get robust and accurate semantic segmentation in urban scenes. The main novelty of our work is the proposed data fusion network architecture. Specifically, we adopt the Encoder-Decoder design concept [8]. ResNet [19] is employed in two encoders for feature extraction. A new decoder is developed to restore the feature-map resolution. The data fusion is performed in the encoding stage by the element-wise summation with feature maps. The contributions of this letter are summarized as follows:

1) We develop a novel deep neural network that fuses both the RGB and thermal information for semantic segmentation in urban scenes.
2) We prove that using thermal information is able to improve the semantic segmentation performance.
3) We compare our network with the state of the arts on a public dataset and achieve the superior performance.

The remainder of this letter is structured as follows. In section II, related work has been reviewed. In section III, we describe our network in detail. In section IV, experimental results and discussions are presented. Conclusions and future work are drawn in the last section.

Manuscript received November 7, 2018; Revised January 26, 2019; Accepted February 25, 2019. This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor Youngjin Choi and Reviewers' comments. This work was supported by the Hong Kong Research Grant Council (RGC) project 11210017 and 21202816, the Hong Kong University of Science and Technology Project IGN16EG12, the National Natural Science Foundation of China project U1713211. *(Corresponding author: Ming Liu.)*

Yuxiang Sun, Weixun Zuo and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China (email: eeyxsun@ust.hk, sun.yuxiang@outlook.com; wzuo@connect.ust.hk; eelium@ust.hk).

Digital Object Identifier (DOI): see top of this page.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

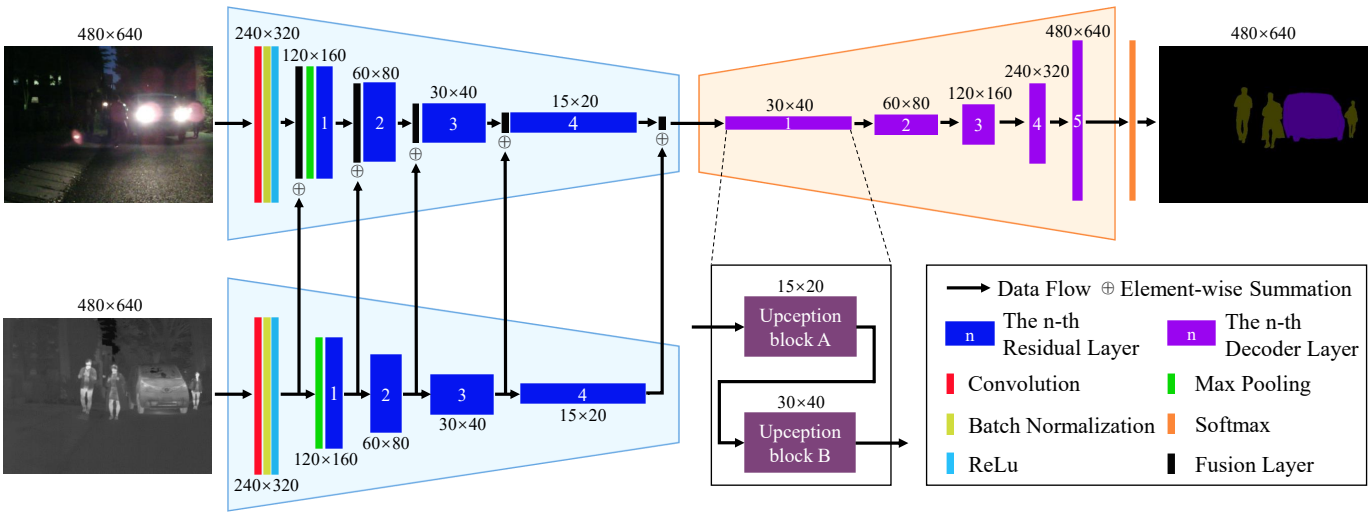


Fig. 1: The overall architecture of the proposed RTFNet. It consists of an RGB encoder, a thermal encoder and a decoder. The blue and yellow trapezoids indicate the encoders (left) and the decoder (right), respectively. We employ ResNet [19] with the average pooling and the fully connected layers removed as the feature extractor. The thermal feature maps are fused into the RGB encoder through the element-wise summation. There are 5 layers in the decoder, in which each layer sequentially consists of the Upception blocks A and B. The output resolutions of layers and blocks are indicated in the figure given for an example input of 480×640. The figure is best viewed in color.

II. Related Work

Semantic segmentation refers to associating each pixel of an image with a class label. We review selected semantic segmentation networks in this section.

A forerunner of deep neural network-based semantic segmentation algorithm is the Fully Convolutional Networks (FCN) proposed by Shelhamer *et al.* [7]. They adapted the existing image classification networks, such as VGG-16 [20] and GoogLeNet [21], into fully convolutional networks by replacing the fully connected layers with the convolutional ones. The downsampled feature maps are upsampled to the desired resolution through deconvolutional networks [6].

SegNet introduced the Encoder-Decoder concept for semantic segmentation [8]. It adopted the VGG network [20] as the encoder, and the mirrored version as the decoder. The locations of the maximum element during pooling were employed for the unpooling operations in the decoder.

UNet was initially proposed for biomedical image segmentation by Ronneberger *et al.* [9], but it generalized well to other domains. It consisted of a contracting path and an expansive path. Similar as the Encoder-Decoder architecture, the contracting path extracts feature maps from input images, while the expansive path restores the feature maps to the desired resolution. The feature maps from the encoder were passed to the decoder through short-cut connections.

PSPNet was designed by Zhao *et al.* [10] for semantic segmentation in complex scenes. They observed that incorporating contextual information is able to improve the segmentation performance. A 4-level pyramid pooling structure was developed in PSPNet to extract the contextual information from images, which covered the whole, half of and small portions of images.

Wang *et al.* [11] developed the DUC-HDC network with the proposed Dense Upsampling Convolution (DUC) and the Hybrid Dilated Convolution (HDC) techniques. The DUC was designed to get pixel-wise prediction maps in a learnable way. It firstly transformed the feature map at the end of the encoder to the one with more channels, and then shuffled the feature map to the desired shape. The HDC was designed to alleviate the gridding issue caused by the standard dilated convolution operation [22]. To this end, it employed different dilation rates in the different layers of the encoder.

ERFNet was proposed to efficiently parse urban scenes in real time by Romera *et al.* [12]. They proposed a non-bottleneck residual module using 1-D convolution filters. Specifically, the sizes of the convolutional kernels are 3×1 and 1×3, while the sizes of commonly used filters are of 2-D, such as 3×3. The proposed residual module using the 1-D filters were demonstrated to be able to reduce parameters, which was favour of real-time tasks. ERFNet was formed by integrating the proposed residual module with the Encoder-Decoder structure.

The aforementioned networks merely work with RGB images captured by visible cameras. Hazirbas *et al.* [23] proposed FuseNet using both the RGB and depth images provided by RGB-D cameras [24]–[26]. FuseNet adopted the Encoder-Decoder architecture. The encoder consisted of two parallel feature extraction modules that take as inputs the registered RGB and depth images, respectively. The depth feature maps were fused into the RGB branch through an element-wise summation. In the encoders of FuseNet, VGG-16 was employed as the backbone.

MFNet was proposed by Ha *et al.* [27] for semantic segmentation of urban scenes using both the visible and thermal cameras. Similar as FuseNet, MFNet also adopted the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

Encoder-Decoder architecture. Two identical feature extractors were designed for RGB and thermal images, respectively. A mini-inception block with dilated convolution was proposed in the encoders. To fuse the feature maps from the RGB and thermal encoders, a short-cut block was designed to concatenate the two feature maps from the two encoders. The concatenated feature map was then added to the output of the corresponding last layer of the decoder.

III. The Proposed Network

The motivation of this work is to enable robust and accurate semantic segmentation of urban scenes for autonomous vehicles. The key idea is to take the advantage of thermal cameras, and fuse both the RGB and thermal information to achieve the superior performance.

A. The Overall Architecture

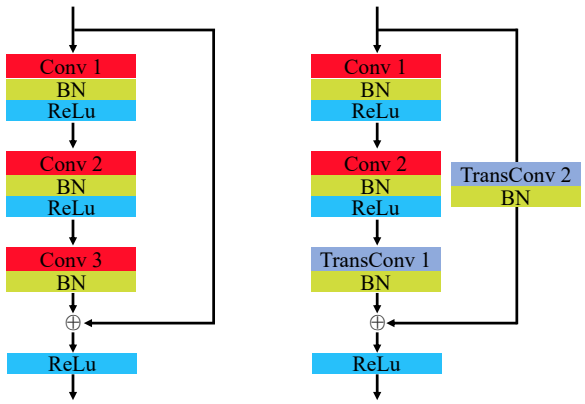
We present a novel deep neural network termed RTFNet for semantic segmentation of urban scenes. Fig. 1 displays the overall architecture of RTFNet. As the Encoder-Decoder structure has been confirmed as an effective architecture in many semantic segmentation networks [13], RTFNet also adopts this structure. As shown in Fig. 1, RTFNet consists of three modules: an RGB encoder and a thermal encoder that are employed to extract features from RGB and thermal images, respectively; a decoder that is used to restore the resolution of feature maps. Different from the related work, such as SegNet [8], the decoder module in RTFNet is not a mirrored version of the encoder module. The encoder and decoder are asymmetrically designed. We have two large encoders and one small decoder. At the end of RTFNet, we use a softmax layer to get the probability map for the semantic segmentation results.

B. The Encoders

We design two encoders to extract features from the RGB and thermal images, respectively. The structures of the two encoders are identical to each other except the number of input channels in the first layer. Different from FuseNet [23] and MFNet [27], we employ ResNet [19] as the feature extractor. In order to avoid the excessive loss of spatial information of feature maps, we delete the average pooling and the fully connected layers of ResNet. This also helps to reduce the model size.

ResNet starts with an initial block that sequentially includes a convolutional layer, a batch normalization layer and a ReLu activation layer. As ResNet is designed using 3-channel RGB images, we modify the number of input channels of the convolutional layer in the initial block of the thermal encoder to 1. Following the initial block, a max pooling layer and four residual layers are sequentially employed to gradually reduce the resolution and increase the number of channels of the feature maps. We refer readers to the ResNet paper [19] to get the details of the residual layers.

We fuse the RGB and the thermal information in the RGB encoder by element-wisely adding the corresponding RGB and thermal feature maps. Note that the feature-map shape is not changed after the fusion operation. As shown in Fig. 1, we place the fusion layers behind the initial block and each residual layer of ResNet. The output of the last fusion layer is taken as input for the decoder.



(a) The Upception block A. (b) The Upception block B.

Fig. 2: The architecture of the proposed Upception block. *Conv*, *TransConv* and *BN* refer to the convolutional layers, transposed convolutional layers and the batch normalization layers, respectively. The detailed configurations for the convolutional and transposed convolutional layers are listed in Tab. I. The figure is best viewed in color.

C. The Decoder

The decoder is mainly designed to get dense predictions. Through the decoder, the feature map resolution is gradually restored to that of the input images. We propose a network block termed Upception in this section. It consists of two sub-blocks: The Upception block A and B. The block A keeps the resolution and the number of feature map channels unchanged; The block B increases the resolution and decreases the channels of the feature maps.

Fig. 2 illustrates the architecture of the Upception block. In block A, there are 3 convolutional layers, through which the resolution and the number of feature channels are not changed. We introduce a short cut from the input to the output of the third batch normalization layer. The input and the feature map are element-wisely added up. In block B, the first convolutional layer (Conv 1) keeps the resolution unchanged and decreases the number of feature channels by a factor of 2. The second convolutional layer (Conv 2) keeps both the resolution and the number of feature channels unchanged. Similar as block A, the input is transferred through a short-cut and added with the output of the third batch normalization layer. Since the first transposed convolutional layer (TransConv 1) keeps the number of channels unchanged and increases the resolution by a factor of 2, the second transposed convolutional layer (TransConv 2) is required to increase the resolution and decrease the number of feature channels. Otherwise, the shapes of the feature maps will not match so that the summation operation cannot be performed. Detailed configurations for the neural network layers in the Upception blocks are displayed in Tab. I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

TABLE I: The detailed configurations for the convolutional (*Conv*) and transposed convolutional (*TransConv*) layers in the Upception blocks. *c*, *h*, *w*, *n* refer to the number of channels, the height and the width of the feature map, the number of classes of the semantic segmentation. (1-4) represents that the Upception blocks are from the 1-st to the 4-th decoder layers. (5) represents that the Upception blocks are from the 5-th (the last) decoder layer. The dash symbol denotes that the size is equal to that in (1-4).

	Name	Kernel Size	Stride	Padding	Input Size (1-4)	Output Size (1-4)	Input Size (5)	Output Size (5)
Upception Block A	Conv 1	1×1	1	0	c×h×w	c×h×w	-	-
	Conv 2	3×3	1	1	c×h×w	c×h×w	-	-
	Conv 3	3×3	1	1	c×h×w	c×h×w	-	-
Upception Block B	Conv 1	1×1	1	0	c×h×w	c/2×h×w	-	n×h×w
	Conv 2	3×3	1	1	c/2×h×w	c/2×h×w	n×h×w	n×h×w
	TransConv 1	2×2	2	0	c/2×h×w	c/2×2h×2w	n×h×w	n×2h×2w
	TransConv 2	2×2	2	0	c×h×w	c/2×2h×2w	-	n×2h×2w

TABLE II: The number (*c*) of the input channels of the first convolutional layer of Upception block A with different ResNet variants.

	RTFNet-18	RTFNet-34	RTFNet-50	RTFNet-101	RTFNet-152
c	512	512	512	2048	2048

As shown in Fig. 1, there are 5 layers in the decoder. Each decoder layer sequentially consists of an Upception block A and an Upception block B. The Upception block B enables each decoder layer to increase the feature-map resolution and decrease the number of feature channels by a factor of 2. Note that the number of the output channels of the Upception block B in the last decoder layer is set to the number of the semantic classes. As we use the dataset provided in [27], the number is set to 9 in this letter.

According to [19], the ResNet variants are named by the number of layers as ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152, respectively. Our RTFNet variants are named according to the used ResNet variants. Since the number of output channels of the last residual layer of ResNet differs between the variants, the number of the input channels of the first convolutional layer of Upception block A varies accordingly. Detailed numbers are displayed in Tab. II.

IV. Experimental Results and Discussions

In this section, we evaluate our proposed RTFNet and compare it with the state-of-the-art networks through extensive experiments on a public dataset.

A. The Dataset

We use the public dataset released in [27]. It records urban scenes using an InfReC R500 camera [28] that can stream RGB and thermal images simultaneously. The dataset contains 1569 pairs of RGB and thermal images, among which 820 are taken at daytime and 749 are taken at nighttime. There are 9 hand-labelled semantic classes including the unlabelled background class in the ground truth. The image resolution in the dataset is 480×640.

We follow the dataset splitting scheme proposed in [27]. The training set consists of 50% of the daytime images and 50% of the nighttime images. The validation set consists of 25% of the daytime images and 25% of the nighttime images. The other images are used for testing.

B. Training Details

We implement our proposed RTFNet using the PyTorch 0.4.1 with the CUDA 8.0 and cuDNN 7.0 libraries. Our RTFNet is trained on a PC with an Intel 3.6GHz i7 CPU and a single NVIDIA 1080 Ti graphics card. As the graphics card memories are limited to 11 GB, we accordingly adjust the batch sizes for different networks.

We train RTFNet with the pre-trained weights of ResNet provided by PyTorch except the first convolutional layer of ResNet in the thermal encoder, because we have 1-channel input data while ResNet is designed for 3. The first convolutional layer of the encoder as well as the convolutional and transposed convolutional layers in the decoder are initialized using the Xavier scheme [29].

We use the Stochastic Gradient Descent (SGD) optimization solver [30] for training. The momentum and weight decay are set to 0.9 and 0.0005, respectively. The initial learning rate is set to 0.01. We adopt the exponential decay scheme to gradually decrease the learning rate. The input training data are randomly shuffled before each epoch. Moreover, the training data are augmented using the flip technique. We train the network until convergence, at which no further decrease in the loss is observed.

C. Evaluation Metrics

We adopt two metrics for the quantitative evaluations of the semantic segmentation performance. The first is the Accuracy (Acc) for each class, which is also known as *recall*. The second is the Intersection over Union (IoU) for each class. The average values across all the classes for the two metrics are denoted as mAcc and mIoU, respectively. They are calculated in the formulas:

$$\mathrm{mAcc} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{1}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

TABLE III: The comparative results (%) on the test dataset from [27]. The results for the unlabelled class are not displayed. The value 0.0 represents that there is no true positives. In other words, the class cannot be detected totally. *3c* and *4c* represent that the networks are tested with the 3-channel RGB data and the 4-channel RGB-Thermal data, respectively. The bold font highlights the best result in each column. The superiority of our proposed RTFNet is clearly proven in this table.

Methods	Car		Person		Bike		Curve		Car Stop		Guardrail		Color Cone		Bump		mAcc	mIoU
	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU		
ERFNet (4c)	78.8	67.1	62.9	56.2	41.6	34.3	39.4	30.6	12.6	9.4	0.0	0.0	0.1	0.1	33.0	30.5	40.8	36.1
ERFNet (3c)	75.8	64.8	47.5	36.5	54.4	42.4	26.5	20.5	15.7	10.0	0.0	0.0	0.0	0.0	39.2	28.8	39.8	33.2
UNet (4c)	71.1	66.2	67.2	60.5	51.4	46.2	47.9	41.6	20.7	17.9	6.7	1.8	34.7	30.6	46.5	44.2	49.5	45.1
UNet (3c)	73.6	65.2	51.6	42.6	54.1	47.8	34.3	27.8	23.7	20.8	0.0	0.0	42.1	**35.8**	36.1	31.0	46.1	40.8
PSPNet (4c)	81.0	74.8	69.2	61.3	63.8	50.2	44.7	38.4	18.1	15.8	0.0	0.0	36.4	33.2	49.0	44.4	51.3	46.1
PSPNet (3c)	82.3	69.0	48.6	39.9	55.3	46.7	33.4	26.9	11.7	11.1	0.0	0.0	38.4	34.1	35.6	26.7	44.9	39.0
SegNet (4c)	67.5	65.3	60.3	55.7	61.0	51.1	46.3	38.4	10.4	10.0	0.0	0.0	41.9	12.0	55.3	51.5	49.1	42.3
SegNet (3c)	58.4	57.3	29.8	27.1	69.4	49.9	18.7	16.8	0.0	0.0	0.0	0.0	0.0	0.0	42.9	37.7	35.4	31.7
DUC-HDC (4c)	91.5	84.8	76.4	68.8	66.7	54.6	54.7	41.9	30.9	19.2	12.3	4.4	40.2	34.3	61.5	45.1	59.3	50.1
DUC-HDC (3c)	91.0	82.6	68.0	58.2	68.0	55.5	40.4	31.9	31.8	21.9	**39.5**	**6.8**	34.7	31.6	58.0	43.6	58.9	47.7
MFNet	77.2	65.9	67.0	58.9	53.9	42.9	36.2	29.9	12.5	9.9	0.1	0.0	30.3	25.2	30.0	27.7	45.1	39.7
FuseNet	81.0	75.6	75.2	66.3	64.5	51.9	51.0	37.8	17.4	15.0	0.0	0.0	31.1	21.4	51.9	45.0	52.4	45.6
RTFNet-50 (ours)	91.3	86.3	78.2	67.8	71.5	58.2	59.8	43.7	32.1	24.3	13.4	3.6	40.4	26.0	73.5	**57.2**	62.2	51.7
RTFNet-152 (ours)	**93.0**	**87.4**	**79.3**	**70.3**	**76.8**	**62.7**	**60.7**	**45.3**	**38.5**	**29.8**	0.0	0.0	**45.5**	29.1	**74.7**	55.7	**63.1**	**53.2**

$$\mathrm{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \tag{2}$$

where N is the number of classes. In this work, $N = 9$ including the unlabelled class. $\mathrm{TP}_i = \sum_{k=1}^{K} \mathrm{P}_{ii}^{k}$, $\mathrm{FP}_i = \sum_{k=1}^{K}\sum_{j=1,j\neq i}^{N} \mathrm{P}_{ji}^{k}$ and $\mathrm{FN}_i = \sum_{k=1}^{K}\sum_{j=1,j\neq i}^{N} \mathrm{P}_{ij}^{k}$ are the true positives, false positives and false negatives for each class i, where K is the number of tested frames, P_{ii}^{k} is the number of pixels for class i that is correctly classified as class i in the frame k, P_{ji}^{k} is the number of pixels for class j that is incorrectly classified as class i in the frame k, P_{ij}^{k} is the number of pixels for class i that is incorrectly classified as class j in the frame k.

D. Ablation Study

In the ablation study, we test RTFNet by removing the thermal encoder to see the benefits brought by using the thermal information. We term the variant as NTE because there is No Thermal Encoder (NTE). Similarly, we remove the RGB encoder from our RTFNet to see how the network performs when only given the thermal information. The variant has No RGB Encoder (NRE), which is accordingly termed as NRE. Moreover, to evaluate the benefit of our proposed Upception block, we replace each decoder layer of RTFNet with a simple decoder layer, which consists of a transposed convolutional layer followed by a batch normalization layer and a ReLu activation layer sequentially. This variant is termed as NUB, which represents that there is No Upception Block (NUB) in our RTFNet. Similar as our original design, there are 5 decoder layers in NUB to gradually restore the feature map to the desired resolution. In order to evaluate the performance of different encoders, all the variants including our RTFNet are tested with different ResNet configurations.

Fig. 3 illustrates the ablation study results. In general, the segmentation performance becomes better when using ResNet with more layers as the encoder. Particularly, the performance improves greatly from ResNet-34 to ResNet-50. However, more layers than ResNet-50 could not contribute too much for the performance improvement. Thus, ResNet-50 could be considered as a nice trade-off between the accuracy and speed. By comparing the results of NRE and NTE, we find that NRE generally gives better performance, but they are both inferior to our RTFNet. This proves that the data fusion is an effective approach to increase the performance, and the thermal information contributes remarkably in the data fusion. By comparing NUB with the others, we could find that our proposed Upception block plays a significant role in RTFNet because the performance decreases notably without it. This proves the effectiveness of our proposed Upception block.

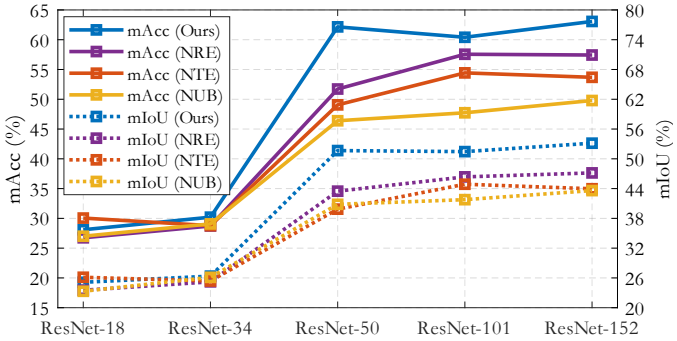


Fig. 3: The ablation study results. The horizontal axis of the plot shows different ResNet used as the encoder. NRE, NTE and NUB are three variants of our RTFNet. NRE and NTE have no RGB and thermal encoder, respectively. NUB has no Upception block. The figure clearly demonstrates the superiority of using ResNet with more layers as the encoder. In addition, the benefits brought by using the thermal information and our proposed Upception block are also proven by the comparison. The figure is best viewed in color.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

TABLE IV: The comparative results of mAcc (%) on the daytime and nighttime scenarios for the NRE and NTE variants. The bold font highlights the better results in each scenario.

Variants	Daytime		Nighttime	
	NRE	NTE	NRE	NTE
ResNet-50	48.16	**52.04**	**50.84**	43.51
ResNet-152	50.62	**54.50**	**56.84**	49.28

Tab. IV displays the testing results of NRE and NTE on the daytime and nighttime scenarios, respectively. We could find that only using the RGB information in the daytime scenario gives better results, while only using the thermal information in the nighttime scenario gives the better results. This is expected because RGB images are more informative in daytime and thermal images are more informative in nighttime.

TABLE V: The comparative results of mAcc and mIoU (%) for different fusion strategies on the test dataset. The bold font highlights the best results.

Variants	Ours		OLF		FCI	
	mAcc	mIou	mAcc	mIou	mAcc	mIou
ResNet-50	**62.2**	**51.7**	52.2	46.2	54.9	45.5
ResNet-152	**63.1**	**53.2**	60.5	51.4	56.3	48.8

In order to demonstrate the effectiveness of our fusion strategy. We delete the fusion layers except the last one from our RTFNet. We term this variant as OLF (Only Last Fusion). We also drop the thermal encoder and modify the RGB encoder to take as input the 4-channel RGB-Thermal data. The 4-channel data are obtained by simply concatenating the 3-channel RGB data with the 1-channel Thermal data. This variant is termed as FCI (Four-Channel Inputs). Tab. V displays the comparative results. We can see that our RTFNet achieves the best results in both mAcc and mIoU.

E. Comparative Results

We compare our RTFNet with ERFNet [12], UNet [9], PSPNet [10], SegNet [8], DUC-HDC [11], MFNet [27] and FuseNet [23] in this section. Because the networks except FuseNet and MFNet are all designed using the 3-channel RGB data, to ensure fair comparisons we train the networks with the 3-channel RGB data and the 4-channel RGB-Thermal data, respectively. We modify the input layers of these networks to accommodate the 4-channel RGB-Thermal data. All the networks are trained until the loss converges with the same augmented training images as those used for our RTFNet.

1) The Overall Results: Tab. III displays the quantitative comparative results for the networks. As most of the pixels are unlabelled in the dataset [27], the evaluation results for the unlabelled class are similar across different networks (around $96\% \sim 99\%$). They are less informative and not displayed in the table. In general, we can see from Tab. III that our RTFNet achieves the best results in terms of both the mAcc and mIoU metrics across all the networks. This proves the superiority of our proposed network. By comparing the RTFNet variants, we can see that RTFNet-152 is slightly better than RTFNet-50. In some cases, RTFNet-50 is even better than RTFNet-152. We conjecture the major reason is that more layers would be prone to causing over fitting, especially for our case that the number of training images in the dataset is limited.

We find that there are some 0.0 Acc and IoU results in the table, especially in the Guardrail class. As indicated in the dataset paper [27], the classes in the dataset are extremely unbalanced. The Guardrail class occupies the fewest portion of the pixels, so there are very few training data for the Guardrail class. We believe that the models are not well trained on this class due to the insufficient training data, so that the 0.0 results appear during the test. In addition, there are 393 images in the

TABLE VI: The comparative results (%) on the daytime and nighttime scenarios. *3c* and *4c* represent that the networks are tested with the 3-channel RGB inputs and the 4-channel RGB-Thermal inputs, respectively. The bold font highlights the best result in each column. Our proposed RTFNet outperforms the others with both the daytime and nighttime images.

Methods	Daytime		Nighttime	
	mAcc	mIoU	mAcc	mIoU
ERFNet (4c)	37.5	32.5	39.3	34.5
ERFNet (3c)	39.5	32.7	34.6	29.6
UNet (4c)	42.3	38.0	47.7	44.0
UNet (3c)	43.7	37.5	41.0	37.0
PSPNet (4c)	42.6	37.8	49.7	45.2
PSPNet (3c)	42.3	36.7	41.1	36.0
SegNet (4c)	39.9	34.6	47.4	41.7
SegNet (3c)	34.2	29.5	28.9	27.4
DUC-HDC (4c)	56.7	44.3	55.0	49.4
DUC-HDC (3c)	55.2	44.3	53.3	44.3
MFNet	42.6	36.1	41.4	36.8
FuseNet	49.5	41.0	48.9	43.9
RTFNet-50 (ours)	57.3	44.4	59.4	52.0
RTFNet-152 (ours)	**60.0**	**45.8**	**60.7**	**54.8**

TABLE VII: The inference speed for each network. *4c* represent that the networks are tested with the 4-channel RGB-Thermal inputs. *ms* and *FPS* represent the time cost in millisecond and the speed in Frame-Per-Second, respectively.

Methods	GTX 1080 Ti		Jetson TX2	
	ms	FPS	ms	FPS
ERFNet (4c)	5.81	172.06	38.89	25.71
UNet (4c)	2.59	386.63	15.59	64.13
PSPNet (4c)	9.09	110.01	49.23	20.31
SegNet (4c)	3.31	302.30	19.06	52.47
DUC-HDC (4c)	12.21	81.93	67.40	14.84
MFNet	4.35	229.86	25.74	38.85
FuseNet	3.92	255.27	22.25	44.94
RTFNet-50	11.25	88.87	59.56	16.79
RTFNet-152	29.35	34.07	508.71	1.97

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2904733, IEEE Robotics and Automation Letters

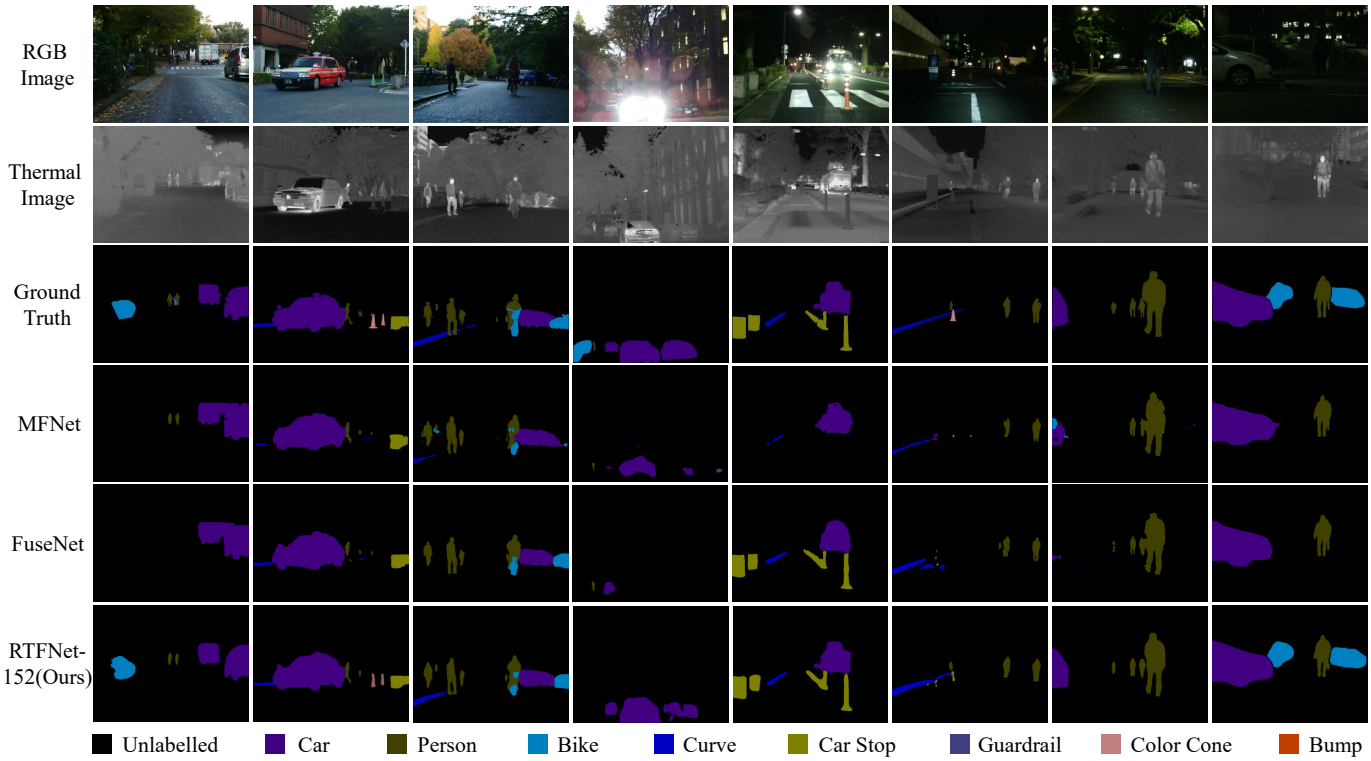


Fig. 4: The sample qualitative results for the data-fusion networks. The left 4 and right 4 columns present the results in typical lighting conditions in the daytime and nighttime scenarios, respectively. The columns 1-3 show high-dynamic range lighting conditions. The column 4 shows the glares of oncoming headlights. The columns 6-8 show almost total darkness lighting conditions. The comparative results prove the robustness and superiority of our approach. The figure is best viewed in color.

test dataset, but only 4 images containing the Guardrail class. Thus, we think that the extreme few pixels on the Guardrail class in the test dataset is another reason for the 0.0 results.

According to Tab. III, the second best network is DUC-HDC and the third would be FuseNet. The DUC-HDC outperforms ours in some classes, such as the guardrail and color cone, although it is not specially designed for the 4-channel RGB-Thermal data. This proves the generalization capability of DUC-HDC. For the other networks, the performance is similar to each other. By comparing the 3-channel and 4-channel results of ERFNet, UNet, PSPNet, SegNet and DUC-HDC, we find that all the 4-channel results are better than those of the 3-channel results. This proves that incorporating the thermal information could improve the segmentation performance.

2) The Daytime and Nighttime Results: We also evaluate the networks using the daytime and nighttime images from the test dataset. Tab. VI displays the comparative results. As we can see, our proposed network achieves the best results in both the two scenarios. In the daytime, we find that the 4-channel results of ERFNet and UNet are slightly inferior to the corresponding 3-channel results. We believe the reason is that the RGB and thermal images are slightly misaligned spatially and temporally [27]. The spatial and temporal misalignments would be caused by the camera calibration errors and the synchronization errors, respectively. In the daytime, both the RGB and thermal images are informative and visually clear. So the slight misalignments could confuse the network and hence degrade the performance. For the other networks, we think that the large receptive fields in their structures alleviate this problem. In the nighttime, the RGB images are little informative because they are not clear (almost black). The thermal information would dominate the segmentation, so the slight misalignments could not greatly influence the accuracy. Thus, we can see in Tab. VI that the 4-channel results are much better than the corresponding 3-channel results.

3) The Inference Speed: We measure the inference speed of the networks with an NVIDIA GeForce GTX 1080 Ti graphics card and an NVIDIA Jetson TX2 (Tegra X2) embedded platform. Tab. VII displays the average time cost on the test dataset given the input resolution of 480×640. As we can see, our RTFNet-50 exhibits a real-time inference speed on GTX 1080 Ti and an acceptable speed on Jetson TX2. Compared with RTFNet-152, we think that it would be better to use RTFNet-50 in practical applications, in which the inference speed is normally a critical concern. We find that the other networks are really fast on GTX 1080 Ti and most of them could run real-timely on Jetson TX2. However, they are unable to provide satisfactory accuracy as ours.

4) The Qualitative Demonstrations: Fig. 4 displays sample qualitative results for the data-fusion networks in typical daytime and nighttime scenarios. In general, we can see that our RTFNet is able to robustly and accurately segment the objects under various lighting conditions that are not well satisfied or even challenging. By comparing our results with the ground truth, we could find that the object boundaries segmented by RTFNet are not so sharp. This would be caused

by the heavy downsampling in the encoders, which greatly reduces the spatial information on the details. The columns 1-3 show the lighting conditions with high-dynamic ranges. The pedestrians could not be clearly seen in the RGB images even it is in daytime. The column 4 shows that the cars are hidden by the glares of the oncoming headlights. In the columns 6-8, the objects are almost invisible because there is almost no lighting in these scenarios. We can see that our RTFNet outperforms the other networks even they are all designed with the RGB and thermal data fusion. The comparison demonstrates the superiority of our network.

V. Conclusion

We proposed here a novel RGB and thermal data fusion-based network for the semantic segmentation of urban scenes. The experimental results demonstrate the superiority of our network in various scenarios, even in challenging lighting conditions. However, there are still several issues to be addressed in the future. Firstly, the inference speed is less competitive especially on embedded platforms. So we would like to speed up our network with an emphasis on the optimization for embedded platforms. Secondly, the object boundaries segmented by our network are not so sharp. To produce sharp boundaries and keep more detailed information, we will use short-cuts to introduce low-level feature maps to high-level feature maps. Lastly, the RGB or thermal image may be more informative than the other in some cases. For example, thermal images would be less informative for objects sharing similar temperature, which would be a negative side of thermal cameras. Giving lower weights to the less contributive information or discarding it completely would benefit the segmentation. In the future, we would like to develop discriminative mechanisms to find data that are more informative.

Acknowledgment

The authors would like to thank Peng Yun for the discussions on the training skills of deep neural networks.

References

[1] C. Wang, W. Chi, Y. Sun, and M. Q.-H. Meng, "Autonomous Robotic Exploration by Incremental Road Map Construction," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2019.

[2] J. Cheng, Y. Sun, and M. Q.-H. Meng, "A dense semantic mapping system based on CRF-RNN network," in *2017 18th International Conference on Advanced Robotics (ICAR)*, July 2017, pp. 589–594.

[3] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110 – 122, 2017.

[4] X. Sun, H. Ma, Y. Sun, and M. Liu, "A Novel Point Cloud Compression Algorithm Based On Clustering," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2132–2139, April 2019.

[5] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robotics and Autonomous Systems*, vol. 108, pp. 115 – 128, 2018.

[6] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1520–1528.

[7] E. Shelhamer and J. Long and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[8] V. Badrinarayanan and A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241.

[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6230–6239.

[11] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding Convolution for Semantic Segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018, pp. 1451–1460.

[12] E. Romera and J. M. Álvarez and L. M. Bergasa and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, Jan 2018.

[13] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *CoRR*, vol. abs/1704.06857, 2017.

[14] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine Vision and Applications*, vol. 25, no. 1, pp. 245–262, Jan 2014.

[15] M. Vollmer, M. Klaus-Peter *et al.*, *Infrared thermal imaging: fundamentals, research and applications*. John Wiley & Sons, 2017.

[16] T. Omar and M. L. Nehdi, "Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography," *Automation in Construction*, vol. 83, pp. 360 – 371, 2017.

[17] A. Carrio, Y. Lin, S. Saripalli, and P. Campoy, "Obstacle Detection System for Small UAVs using ADS-B and Thermal Imaging," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2, pp. 583–595, Dec 2017.

[18] J. S. Yoon, K. Park, S. Hwang, N. Kim, Y. Choi, F. Rameau, and I. s. Kweon, "Thermal-infrared based drivable region detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 978–985.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[22] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2018.

[23] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture," in *Computer Vision – ACCV 2016*. Cham: Springer International Publishing, 2017, pp. 213–228.

[24] Y. Sun, M. Liu, and M. Q.-H. Meng, "Active Perception for Foreground Segmentation: An RGB-D Data-Based Background Modeling Method," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2019.

[25] T. Yan, Y. Sun, T. Liu, C.-H. Cheung, and M. Q.-H. Meng, "A Locomotion Recognition System Using Depth Images," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6766–6772.

[26] Y. Sun, M. Liu, and M. Q.-H. Meng, "Invisibility: A moving-object removal approach for dynamic scene modelling using RGB-D camera," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2017, pp. 50–55.

[27] 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 *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 5108–5115.

[28] InfReC R500 Website. http://www.infrared.avio.co.jp/en/products/ir-thermo/lineup/r500/, accessed Mar. 1, 2019.

[29] Glorot, Xavier and Bengio, Yoshua, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[30] G. Montavon, G. Orr, and K.-R. Mller, *Neural Networks: Tricks of the Trade*, 2nd ed. Springer Publishing Company, Incorporated, 2012.