Depth Embedded Recurrent Predictive Parsing Network for Video Scenes

Lingli Zhou, Haofeng Zhang, Yang Long, Ling Shao, *Senior Member, IEEE*, and Jingyu Yang

***Abstract*—Semantic segmentation-based scene parsing plays an important role in automatic driving and autonomous navigation. However, most of the previous models only consider static images, and fail to parse sequential images because they do not take the spatial-temporal continuity between consecutive frames in a video into account. In this paper, we propose a depth embedded recurrent predictive parsing network (RPPNet), which analyzes preceding consecutive stereo pairs for parsing result. In this way, RPPNet effectively learns the dynamic information from historical stereo pairs, so as to correctly predict the representations of the next frame. The other contribution of this paper is to systematically study the video scene parsing (VSP) task, in which we use the RPPNet to facilitate conventional image paring features by adding spatial-temporal information. The experimental results show that our proposed method RPPNet can achieve fine predictive parsing results on cityscapes and the predictive features of RPPNet can significantly improve conventional image parsing networks in VSP task.**

***Index Terms*—Recurrent predictive parsing network (RPPNet), spatial-temporal continuity, video scene parsing, depth embedded, long short term memory (LSTM).**

I. Introduction

THE purpose of video scene parsing is to classify every pixel of all frames in scene videos, which is useful for many applications [1]. In recent years, convolutional neural networks have been well applied to image scene parsing tasks [2]–[6]. However, there are still some problems in applying these networks to the Video Scene Parsing (VSP) task directly. The most fundamental reason is that these networks can only parse the scene of videos frame by frame, thus the correlation and continuity between video sequences are neglected and will bring much noise to the final results. Besides, video annotation data is very scarce in present, since annotation is a labor-intensive and time-consuming work. Therefore, we aim to find a solution in the paper that can parse the video scene images in the circumstance where the video sequences are sufficient while the annotations are in short. In view of the above problems, Jin *et al.* [7] proposed a novel Predictive Parsing Network (PPNet) to predict the parsing map of the target frame given only its preceding frames, which is instructive for our task. Even if the parsing results of PPNet are of some referential, there is quite room for improvements.

Inspired by PPNet, we elaborately design a depth embedded Recurrent Predictive Parsing Network (RPPNet) that has the ability to predict the parsing map of the target frame effectively. In this method, we learn the previous frames of the target frame through a recurrent strategy, which can learn the dynamic trend between frames and can bring about more structural details for final predictive parsing results, but does not require the ground-truth maps of the entire sequences. There are two innovative components in our proposed network to meet the above capabilities. First, in most networks, such as PPNet, the input of them is a single image, while our network takes a stereo image pair containing both the left and right image of the same scene taken at the same time as input. These binocular images of a sequence implicitly contain depth information of the scene. In the video sequence, features with depth information can effectively provide more dynamic information of scene changes than that of single RGB features, which can enhance the continuity and consistency between frames. Therefore, compared with monocular images, binocular images can play a more important role in VSP task. Second, for predicting scene maps, PPNet simply concatenates several preceding frames, and then the network extracts features by doing convolution operations. In our network, we use the Long Short Term Memory (LSTM) network [8]–[10], a kind of recurrent neural networks (RNN), to predict the features of the target frame. The features extracted from the preceding frames are chronologically inputted into the LSTM network, and then the predictive parsing map can be obtained by convolving the predicted features of the target image.

As mentioned above, image scene parsing networks should not be directly applied to VSP task, which will bring noise and discontinuity to the results. Therefore, we further adaptively integrate the spatial-temporal features obtained from RPPNet with features from any conventional image scene paring model to learn more discriminative representations, and enhance VSP performance substantially in the presence of the target frames. For example, in the second row of Fig. 1, the areas marked by the red boxes are discontinuous with context. But these discontinuities in the fourth row of the figure

Manuscript received October 27, 2018; revised January 20, 2019; accepted March 31, 2019. This work was supported in part by the National Science Foundation of China under Grant 61872187, Grant 61773215, and Grant 61871444, in part by the National Defense Pre-Research Foundation under Grant 41412010302 and Grant 41412010101, and in part by the Medical Research Council (MRC) Innovation Fellowship under Grant MR/S003916/1. The Associate Editor for this paper was J. Sanchez-Medina. *(Corresponding author: Haofeng Zhang.)*

L. Zhou, H. Zhang, and J. Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zhoulingli@njust.edu.cn; zhanghf@njust.edu.cn; yangjy@njust.edu.cn).

Y. Long is with the Open Laboratory, School of Computing, University of Newcastle, Newcastle upon Tyne NE4 5TG, U.K. (e-mail: yang.long@ieee.org).

L. Shao is with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates (e-mail: ling.shao@ieee.org).

Digital Object Identifier 10.1109/TITS.2019.2909053

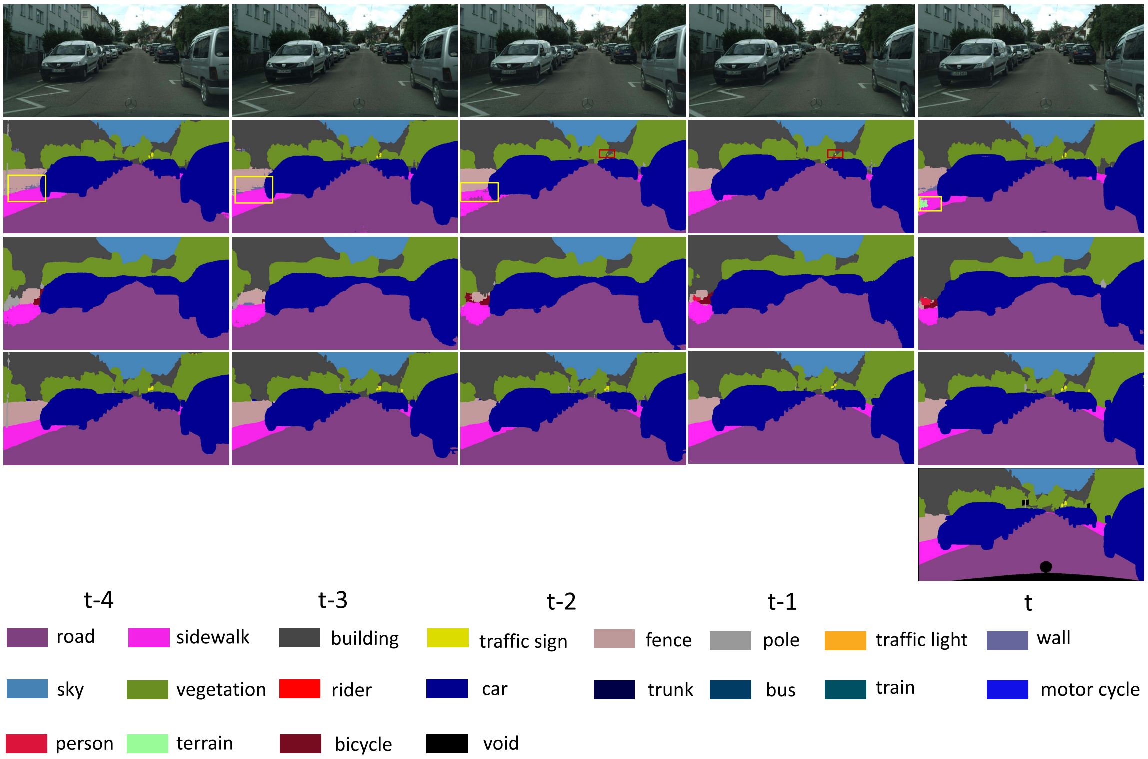


Fig. 1. Illustration on predictive parsing results of the Recurrent Predictive Parsing Network (RPPNet) and examples of improved results for the traditional image parsing network with the help of RPPNet. **Top**: five consecutive frames of a video from the Cityscapes dataset. **Second row**: the parsing results of the above frames produced by the common image parsing network (CIPNet) based on VGGNet. We can see much noise marked by the yellow boxes and discontinuities of objects marked by the red boxes. **Third row**: results from RPPNet, it can not only generate spatial-temporal parsing results but also help the traditional image parsing networks. **Fourth row**: parsing maps produced by the integrated network (ITNet), which takes advantage of CIPNet (the second row) and RPPNet (the third row). The noise of the yellow box are eliminated and the discontinuities marked by the red boxes are resolved. **Bottom**: ground-truth annotation of the last frame. Best viewed in color and zoomed pdf.

are eliminated because of the adoption of spatial-temporal information.

It is worthwhile to list the contributions of our work,

1) We propose a novel depth embedded recurrent predictive parsing network (RPPNet) for VSP task, which takes binocular images as input, so it can capture more dynamic information between frames to ensure the temporal and spatial consistency of features. In addition, the advantage of the predictive capability of LSTM network is applied to continuous sequences so as to obtain predictive features;
2) In order to generate more accurate video parsing maps, the predicted features and the features obtained from conventional image parsing models are fused to further improve the performance of conventional models on VSP task;
3) The experiments on the popular city scene dataset show that RPPNet produces instructive predictive parsing results and our method can significantly improve the performance of conventional parsing methods on video sequences, especially on the classes with small areas, such as pole and pedestrian.

The rest of the paper is organized as follows. We first review existing architectures of parsing tasks and the functions of LSTM in Section 2. Details of our RPPNet and approaches are introduced in Section 3. Experimental results are described in Section 4, and the last is a brief conclusion of the whole paper in Section 5.

II. Related Work

A. Scene Parsing Networks

Recently, convolutional neural architectures [11]–[14] have obtained remarkable results in image parsing tasks and played an important role in many applications, such as autonomous driving and navigation. Among them, fully convolutional network (FCN) [2] and Deeplab [13], [15], [16] are the most prominent. However, these networks mostly act on individual and static images, and lack the consideration of continuous video frames. In this article, our task is to parse video sequences. It is obviously not good enough to apply the above parsing methods directly to every frame of the video sequences because these methods ignore the temporal and spatial consistency between frames and lead to bad results. The coherence information between consecutive frames is especially important for capturing and predicting dynamic objects of videos in our predictive parsing task. In order to get consistency between frames, some methods [17], [18] try to utilize 3D data which contains more spatial motion information or seek help from optical flow [19]. With more and more datasets containing depth images, there is an increasing

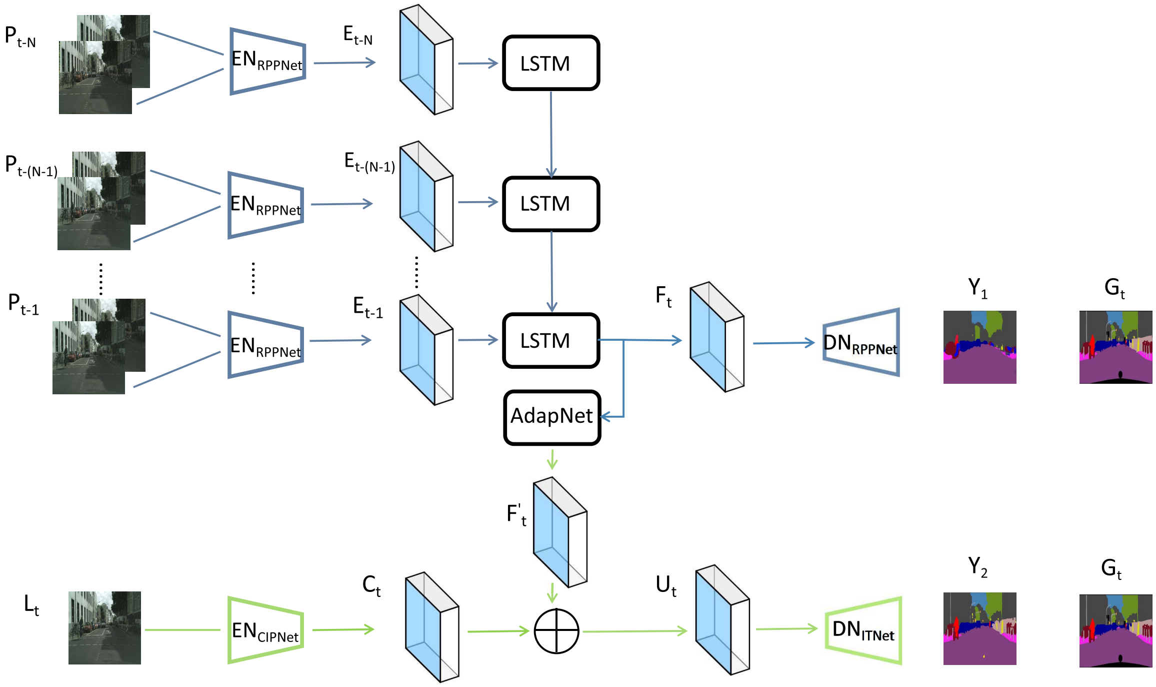


Fig. 2. (a) The upper part of the figure with blue flow line is the framework of the recurrent predictive parsing network (RPPNet). EN_{RPPNet} and DN_{RPPNet} represent the encoder and decoder of RPPNet respectively. RPPNet feeds the features extracted from the historical image pairs $\{P_{t-N}, ..., P_{t-1}\}$ to LSTM in chronological order and DN_{RPPNet} decodes the predictive features F_t of the target image generated by LSTM to obtain the predictive parsing result Y_1. (b) The bottom part of the figure with green flow lines is the framework of the integrated network (ITNet). Similarly, EN_{CIPNet} represents the encoder of the conventional image parsing network (CIPNet) and DN_{ITNet} represents the decoder of ITNet. First, ITNet combines the predictive features F_t processed by AdaptNet with features C_t of the target image encoded by CIPNet. Second, DN_{ITNet} decodes the fused features U_t to get the final parsing result Y_2.

number of methods using RGB-D images for scene parsing task [17], [20], [21]. We also consider using depth information to improve the accuracy of capturing dynamic transformations and obtaining structural details. However, different from [17], [18], [22], we take a simpler but more efficient method by taking advantage of 3D structure hiding in stereo pairs instead of using depth images directly. Jin *et al.* [7] firstly proposed a novel predictive feature learning method called PPNet to predict the parsing result of the target image using preceding consecutive sequences. This network can learn the features of spatial-temporal coherence between the target frame and its previous frames effectively. Inspired by [7], we also learn from anterior sequence frames to obtain features of the target frame, but the difference is the adoption of stereo image pairs and recurrent network in our method.

B. LSTM

The main purpose of recurrent neural networks (RNNs) [23]–[25] is to process and predict sequence data. RNN can not only mine sequential information in data but also make full use of the great power of expression of semantic information, so it has made a breakthrough in speech recognition [26], machine translation [27], [28] and time series analysis [29] *etc.* LSTM [8], a special kind of RNN, is designed to solve the problem of long-term dependence in RNN. For many tasks, recurrent neural networks using LSTM are better than standard recurrent neural networks. Different from prediction technique of [7], we draw support from sequence prediction ability of LSTM and classify the features predicted by LSTM to get final predictive parsing results.

III. Approach

In this paper, our approach solves two problems in different settings with two steps: first, how to get the predictive parsing results of the target frames in advance when the target images are not available; second, can we integrate temporal continuity obtained by the former with the target image features so as to further improve the performance of the parsing results in the circumstance that the target images are available.

Let $\{P_{w/o}/P_w, G_t\}$ denotes a video or an image sequence, where $P_{w/o} = \{P_{t-N}, \ldots, P_{t-1}\}$ stands for $N-1$ consecutive image pairs before the target frame, and $P_w = \{P_{t-N}, \ldots, P_{t-1}, P_t\}$ represents image pairs with the target frame, where, P_t is the target frame. Here each image pair P_i is consist of pair-wised stereo images, including a left image L_i and a right image R_i. Let G_t denote ground-truth annotation with C classes of the target left image L_t. The task of first setting can be considered as seeking a function F_1 that maps $P_{w/o}$ to the predictive parsing map of L_t, and which can be expressed as a formula $Y_1 = F_1(P_{w/o})$, where Y_1 stands for parsing result of L_t. Similarly, $Y_2 = F_2(P_w)$ is the formula related to the second setting, where Y_2 is the direct parsing result of L_t too. The biggest difference between the two tasks is that there are only preceding frames in the former setting, while the latter has the target frame P_t available in advance.

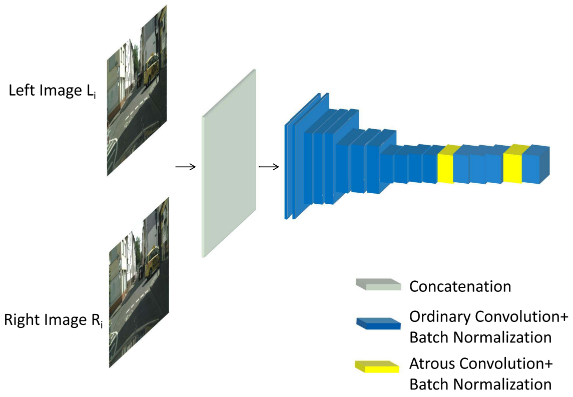

Left Image L_i
Right Image R_i
→
→
Concatenation
Ordinary Convolution+ Batch Normalization
Atrous Convolution+ Batch Normalization

Fig. 3. The encoder part of RPPNet (EN_{RPPNet}). It takes a left image L_i in size $H \times W \times 3$ and its corresponding right image R_i as input. After concatenation and several convolution operations, EN_{RPPNet} outputs the features in size $H/16 \times W/16 \times 1024$.

TABLE I

DETAILED PARAMETERS OF EN_{RPPNet}. WHERE, k IS THE KERNEL SIZE, c IS THE NUMBER OF CHANNELS, s REPRESENTS THE STRIDE, AND d STANDS FOR THE DILATION RATE. [·] MEANS NUMBER OF TIMES OF EXECUTIONS, AND THE '×' IN FIRST COLUMN HAS THE SAME MEANING

Name	Layer Setting	Output dimension
Input_left		$H \times W \times 3$
Input_right		$H \times W \times 3$
Concatenation		$H \times W \times 6$
Conv0_0+BN	$k = 3 \times 3, c = 32, s = 2$	$H/2 \times W/2 \times 32$
Conv0_1+BN	$k = 3 \times 3, c = 32, s = 1$	$H/2 \times W/2 \times 32$
Conv1_0+BN	$k = 3 \times 3, c = 64, s = 2$	$H/4 \times W/4 \times 64$
Conv1_×+BN	$[k = 3 \times 3, c = 64, s = 1] \times 2$	$H/4 \times W/4 \times 64$
Conv2_0+BN	$k = 3 \times 3, c = 128, s = 2$	$H/8 \times W/8 \times 128$
Conv2_×+BN	$[k = 3 \times 3, c = 128, s = 1] \times 2$	$H/8 \times W/8 \times 128$
Conv3_0+BN	$k = 3 \times 3, c = 256, s = 2$	$H/16 \times W/16 \times 256$
Conv3_×+BN	$[k = 3 \times 3, c = 256, s = 1] \times 3$	$H/16 \times W/16 \times 256$
Atrous_conv0+BN	$k = 3 \times 3, c = 512, d = 2$	$H/16 \times W/16 \times 512$
Conv4_×+BN	$[k = 3 \times 3, c = 512, s = 1] \times 3$	$H/16 \times W/16 \times 512$
Atrous_conv1+BN	$k = 3 \times 3, c = 1024, d = 2$	$H/16 \times W/16 \times 1024$
Conv5+BN	$k = 3 \times 3, c = 1024, s = 1$	$H/16 \times W/16 \times 1024$

In the following subsections, we will describe the proposed RPPNet in detail. Fist, two important components, including depth embedded convolution and recurrent network, and the general framework of RPPNet will be introduced in subsection A and subsection B respectively, and then how to enhance the parsing performance of the conventional image parsing networks by adopting predictive features will be explained in subsection C.

A. Depth Embedded and Recurrent Network

If the regularity of dynamic change between consecutive frames can be learned more accurately, the feature of next frame can be predicted more reliably, which is the key component to improve the performance of scene predictive parsing. The proposed RPPNet is inspired from this point, thus we put emphasize on tracking spatial-temporal dynamics in the network architecture design. There are two novel strategies to make full use of context information and capture dynamics in our scene predictive parsing network. One is the adoption of stereo images as input, the other is the long-short term memory (LSTM) is employed to predict features of the target frame.

1) Depth Embedded: Most scene parsing networks only use monocular images as input, but recently, the role of depth information in scene parsing tasks has attracted much attention, and many works try to bring it into full play in the field of deep learning tasks such as scene parsing [11], [12] and object detection [30], and we are also agree with the positive role of depth information in the scene parsing task. For one hand, depth information can provide more structural details of the scene, and for the other hand, it can supplement dynamic information in video sequences and thus beneficial to predict the changes of objects in the next frame. Therefore, we employ this important clue in our network. However, different from these methods [17], [18], we do not use RGB-D images directly, nor do we explicitly calculate the depth or disparity map of the input images [22]. Imitating the human eyes, RPPNet simultaneously processes a pair of images in the same scene, as shown in Fig. 3. We concatenate the left image L_i and the right image R_i as a whole, which is then fed into the encoder part of RPPNet. Therefore, by adopting the latent depth information in binocular images, the encoder of RPPNet can learn more precise structural information of objects comparing to traditional single image encoder. The parameters of the encoder part of RPPNet can be found in Tab. I.

2) Recurrent Network: Another key element of RPPNet is the usage of LSTM to learn predictive features from the images of preceding frames. Each frame in a video is related to its previous frames, because they are temporally and spatially continuous. Conventional neural networks has the problem that it cannot use the information of the previous frames, which can be well solved by recurrent neural networks (RNNs). The most prominent characteristic of RNN is some outputs of its neurons can be transferred as input again, thus RNN can take advantage of previous information, in other words, the network has memory. LSTM is a special version of RNN. The biggest difference between it and ordinary RNNs is that LSTM is able to remember information far from the current input without the phenomenon of gradient disappearance because of its specially designed gate operations. The application of LSTM to predict target features is illustrated in Fig. 2. Therefore, the encoder of RPPNet (denoted as EN_{RPPNet}) maps each image pair of the sequence $P_{w/o}$ to spatial-temporal representations, which are then fed to the followed LSTM network in chronological order to produce predictive representations F_t of the target frame.

B. Recurrent Predictive Parsing Network (RPPNet)

The framework of RPPNet is illustrated in the upper part of Fig. 2. The RPPNet consists of three components, i.e. the encoder part EN_{RPPNet}, which learns features $E_i = EN_{RPPNet}(P_i)$ from the preceding image pairs, the LSTM network, which generates predictive features of the target image $F_t = LSTM(E_{t-N}, \ldots, E_{t-1})$, and the up-sampling part (denoted as DN_{RPPNet}), which first assigns each point of the shrunken feature map F_t with a predefined label, and then generates the final predictive parsing result Y_1

by up-sampling the preliminary result and expanding its size to the same size as the original input image.

There are many encoding architectures, often represented by the two most common networks VGGNet [31] and ResNet [32], for EN_{RPPNet} to choose. But considering that the input is an image sequence, especially the sequence consists of multiple image pairs, we abandon the above two networks, instead we redesign a feature extraction network as the encoder for RPPNet elaborately, and we refer the reader to Fig. 3 for more details. For the design of this part, we follow two criterias. First, it is proposed in [31] that the use of stacked small-kernel convolution instead of large-kernel convolution not only reduces the number of parameters but also learns more features because of containing more linear transformations. So we only use small-kernel convolution for stacking in the network. Second, we adopt atrous convolution at the end of the network with reference to [13]. Atrous convolution can not only play the role of downsampling, but also avoid the resolution of feature map becoming too small. For an image pair P_i with the size of $H \times W \times 3$ at time i, it is first concatenated into size of $H \times W \times 6$ by EN_{RPPNet}, which is then followed by a series of convolution operations. Note that all images must be normalized before entering the network and each convolution layer is followed by a batch normalized (BN) layer, because BN can both accelerate network convergence and prevent over-fitting. In the multiple convolutional layers of the network, down-sampling is performed for feature maps by setting the stride to 2 in order to enlarge the receptive field. On the back of the EN_{RPPNet}, we add several atrous convolution layers to avoid feature maps being too small to lose more important structural information. At last, for N previous consequent image pairs $\{P_{i-N}, \ldots, P_{i-2}, P_{i-1}\}$, we can get N feature maps $\{E_{i-N}, \ldots, E_{i-2}, E_{i-1}\}$ with size of $H/16 \times W/16 \times 1024$, and then these feature maps are fed to LSTM network in chronological order to predict features F_t of the target image, as illustrated in Fig. 2. For the LSTM part, although there are many variants of LSTM, we adopt the basic structure without major changes since the comparative experiments of Greff *et al.* in 2015 [33] show that effects on LSTM variations are similar and they are just different in certain tasks.

Throughout the feature extraction phase, EN_{RPPNet} makes full use of the context information of the video data and produces predictive features for our first task. It adopts binocular images simply but efficiently, which can provide more structural information for predictive features implicitly and facilitate the network to capture more dynamic information. In addition, to generate the predictive feature map, the preceding information should be applied to improve the accuracy of prediction, thus the LSTM network is employed in our predictive parsing task to remember the previous information. For the first task of generating predictive result, since the absence of the target image, we use its preceding sequence in the video to learn its representations. DN_{RPPNet} classifies the predictive features by making a 1×1 convolutional operation on it, and gets the preliminary parsing result of size $H/16 \times W/16 \times C$, where C is the category number of each pixel. Then the preliminary parsing result are followed by four continuous deconvolution layers, which are all with the kernel size of 4×4 and the stride size of 2. These deconvolutional operations expand the size of the preliminary parsing result to $H \times W \times C$. The reason we choose deconvolutional layers for upsampling is that they can restore more structural details compared with simple linear interpolation. At last, a softmax is exploited to generate the final predictive parsing result. Given a sequence of m image pairs, we can use cross entropy to define the loss function for training RPPNet as,

$$\begin{aligned}&\mathcal{L}_1(Y_1|P_{w/o},\theta_{RPPNet})\\&\quad= -\frac{1}{2m}\sum_{t=0}^{m}\sum_{(p,q)\in L_t}\sum_{i\in C}(G_{t_{(p,q,i)}}logY_{1_{(p,q,i)}}\\&\qquad+(1-G_{t_{(p,q,i)}})log(1-Y_{1_{(p,q,i)}}))+\lambda\sum_{w\in\theta_{RPPNet}}w^2, \qquad (1)\end{aligned}$$

where, θ_{RPPNet} is the parameter of RPPNet. m is the batch size for training, $Y_{1_{(p,q,i)}}$ denotes the predictive parsing probability of class i at location (p, q) for the predictive parsing result Y_1, and $G_{t_{(p,q,i)}}$ has similar meaning but for ground-truth. λ is the balancing coefficient for the regularization term. Since the Eq. 1 is a differentiable function, we can minimize it to obtain optimal parameters by applying Stochastic Gradient Descent (SGD) during training.

C. Integrated Parsing Network (ITNet)

The conventional image parsing network (short for CIPNet) should not be directly applied to video sequences because it cannot capture the connections between the target images and the preceding frames. Video sequences are continuous and integrated, thus we should not handle them separately. However, the representations of the target images obtained from CIPNet are also very important, thus combining the target features and the predictive features together is an effective way to achieve complementary effect. In this section, we propose an integrated network, namely ITNet, to integrate RPPNet and CIPNet into a joint architecture to improve the parsing performance.

The ITNet takes advantage of spatial-temporal features of RPPNet and independent ones of CIPNet. As shown in the below part of Fig. 2, the predictive feature F_t', which is the output of the adaptive network, and the feature C_t of CIPNet are concatenated to generate the joint feature U_t. EN_{CIPNet} is the encoder part of CIPNet by removing the classification layers, and can generate a dense feature map for the input image. For CIPNet, various existing image parsing networks, such as FCN [2] and Deeplab [13], [15], [16] can be employed. In the paper, we pick two popular image parsing networks which are based on the two most widely-used feature extraction networks VGGNet and ResNet, and more details will be given in Section 4.1.

Note that the predictive features and target image features cannot be combined simply, because they have different scales and distributions. As mentioned in [34], naively concatenated features may cause that the "large" features overwrite the "smaller" ones, and lead to poor performance. This issue is also considered in [7], which adopts a similar method to [34]

TABLE II

PER-CLASS RESULTS OF THE CONTRAST EXPERIMENT BETWEEN PPNET AND RPPNET ON THE CITYSCAPES TEST SET

Model	**road(%)**	**swalk(%)**	**build.(%)**	**wall(%)**	**fence(%)**	**pole(%)**	**tlight(%)**	**sign(%)**	**veg.(%)**
PPNet	92.89	58.34	77.76	13.21	17.44	11.39	**24.71**	**30.91**	82.15
RPPNet (N=4)	**94.61**	**62.13**	**81.89**	**19.29**	**22.67**	**12.87**	21.90	26.04	**83.53**
terrain(%)	**sky(%)**	**person(%)**	**rider(%)**	**car(%)**	**truck(%)**	**bus(%)**	**train(%)**	**mbike(%)**	**bike(%)**
40.15	85.12	39.25	7.23	72.69	2.25	8.77	5.76	8.75	**36.35**
40.87	**86.86**	**39.40**	**7.93**	**80.07**	**13.32**	**34.39**	**15.25**	**9.07**	36.20

but powerful approach to solve the problem. In our method, we use an adaptive networks, namely AdapNet, which draws on their experience. Concretely, when the sizes of the two feature maps do not match, a deconvolutional operation is utilized on the smaller one. After that, a convolutional layer with a kernel of 1 × 1 is conducted to generate the predictive features F_t' to make consistent with C_t. Although this operation of AdapNet is just a simple linear transformation, it eases the parameter adjustment complexity in the deconvolution operation of the integrated network during training.

After that, we concatenate F_t' and C_t to generate the joint feature U_t. Same as RPPNet, ITNet also uses deconvolutional layers to enlarge the parsing map as same as RPPNet, and we denote these deconvolutional layers and softmax operation as DN_{ITNet}. Finally, the loss function of the integrated network can be defined same as the loss function of RPPNet,

$$\begin{aligned}\mathcal{L}_2&(Y_2|P_w,\theta_{ITNet})\\ &= -\frac{1}{2m}\sum_{t=0}^{m}\sum_{(p,q)\in L_t}\sum_{i\in C}(G_{t_{(p,q,i)}} logY_{2_{(p,q,i)}}\\ &\quad + (1-G_{t_{(p,q,i)}})log(1-Y_{2_{(p,q,i)}})) + \alpha\sum_{w\in\theta_{ITNet}} w^2, \qquad (2)\end{aligned}$$

where, α controls the balance of the two items. The Eq. 2 is also a differentiable function, thus we can optimize it by applying Stochastic Gradient Descent (SGD) during training too.

IV. EXPERIMENTS

A. Experimental Settings and Implementation Details

1) Dataset and Evaluational Metrics: In our experiment, we choose Cityscapes dataset [35] for our testing. There are two reasons for that, first, it is a big and rich dataset compared with the datasets like CamVid [36] and Leuven [37]. Cityscapes is tailored for urban scene understanding and the data is recorded from 50 different cities, ensuring to fully capture the polymorphism and complexity of real-world urban scenes. Second, the data type contained in the dataset is very suitable for our experiments. Cityscapes have 5000 sequences with fine pixel-wise annotations for the nineteenth frame per sequence. Among the 5000 sequences, there are 2975 for training, 500 for validation and 1525 for testing. Specially, Cityscapes provides the right image for every corresponding left image, which is the guarantee for taking image pairs as input. There are 34 visual classes for annotation, which are grouped into eight coarse categories. But some classes are too rare, and only 19 classes of them are included in this assessment, which can be seen Fig. 1.

TABLE III

COMPARISON BETWEEN PPNET AND RPPNET ON TWO METRICS OF PA AND MIOU ON THE CITYSCAPES TEST SET

Method	**Pixel Acc. (%)**	**mIoU (%)**
PPNet	87.35	37.63
RPPNet (N=4)	**89.02**	**41.49**

In accordance with conventional methods, we use Intersection-over-Union (IoU) [38] and Pixel Accuracy (PA) as evaluation metrics for Cityscapes. Given an image, the IoU metric stands for the similarity between the computed parsing region and the ground truth region in a selected class, and is defined as the size of the intersection divided by the union of the two regions [39]. The IoU metric can take into account the problems of class imbalance that generally exist in such problem settings. For example, if an algorithm predicts that each pixel of the image is the background, the IoU measurement can effectively penalize it because the intersection between the predicted region and the ground truth region will be zero, resulting in a zero IoU count. PA is defined as the ratio of all correctly classified pixels to all valid pixels. In addition, we also add mIoU as another indicator, which is defined as the average of all IoU values from 19 classes.

2) Implementation Details: Both of our two models including RPPNet and ITNet are implemented based on the public deep learning architecture TensorFlow [40]. Throughout the training, we crop the images randomly same as that done in the literature [14] and set the batch size to 1 due to the limited GPU memory. We take two strategies to prevent overfitting, one is that we add the dropout operation to some layers of the networks during training; the other is that we add L_2 regularization of parameters for the loss functions, as shown in Eq. 1 and Eq. 2. During training, λ and α are set to 0.2, and we choose Adam optimizer, which is computationally efficient and requires relatively less memory and leads to widely usage in many applications. The learning rate is reduced from 0.01 to 0.0001 by using the "exponential decay" policy.

RPPNet contains a total of approximately 140M trainable parameters, and we first train them from the scratch for about 60K iterations. Then the parameters of RPPNet are utilized as pre-trained parameters of ITNet. For the conventional image parsing part of ITNet, that is EN_{CIPNet}, when it is based on VGGNet, we randomly initialize its parameters; while it is based on ResNet, we adopt the pre-trained parameters from ImageNet [41]. The modified baselines based on the two architectures will be elaborated in the following part.

B. Comparison With Other Predictive Parsing Network

As we have known, there is only one deep predictive parsing work, Predictive Parsing Network (PPNet) [7], for

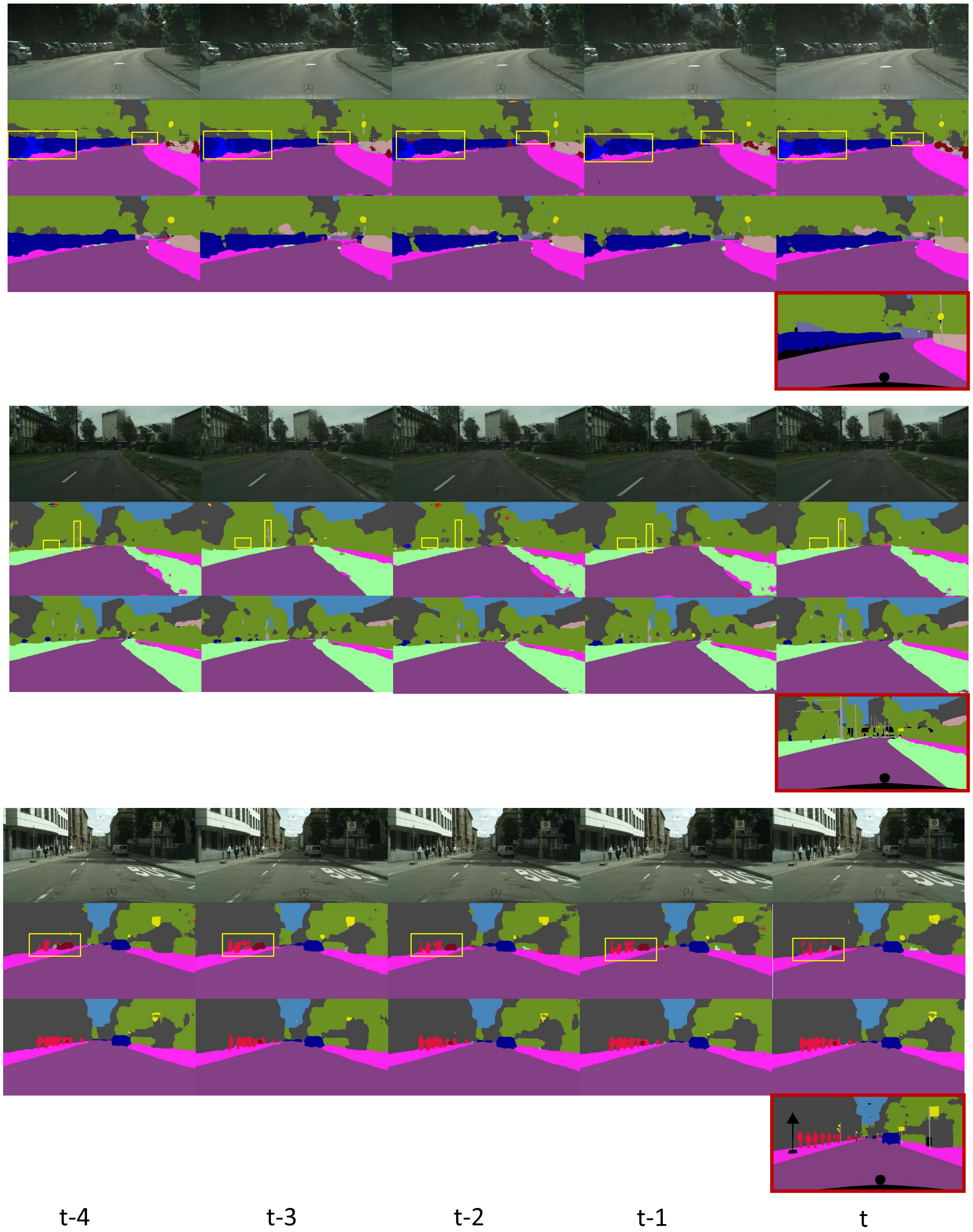


Fig. 4. Visual comparison of the parsing results of PPNet and RPPNet on the Cityscapes test set. **Top row**: Five consecutive left images of a sequence. **Second row**: Predictive parsing maps produced by PPNet, of which each map is predicted from 4 preceding frames of the current image. Unsatisfactory parts highlighted by yellow can be amended by RPPNet (the third row). **Third row**: Predictive parsing maps produced by RPPNet. **Bottom**: the ground-truth annotation (with red boundary) of the frame *t* provided by the dataset. Best viewed in color and zoomed pdf.

video scene predictive parsing. Although PPNet also predicts the result of the target image with the help of preceding frames and tries to capture spatial-temporal dynamic information, its structure is too simple to make full use of the preceding frames and cannot predict more structural details of the scene, while our method can make significant improvement in capturing dynamic information and predicting structural information.

To proof that our RPPNet is superior in the predictive scene parsing task, we compare our RPPNet with PPNet and recorded the results in Table III. According to [7], PPNet only uses 4 previous frames, thus for the sake of fairness, we also use the preceding 4 frames by setting N to 4. As shown in Table III, our method exceeds PPNet greatly by 1.67 and 3.86 and reaches 89.02 and 41.49 in terms of PA(%) and mIoU(%) respectively. More details of per-class values are also

TABLE IV

PER-CLASS VALUES OF IOU FOR DIFFERENT N ON THE CITYSCAPES VALIDATION SET

Model	**road(%)**	**swalk(%)**	**build.(%)**	**wall(%)**	**fence(%)**	**pole(%)**	**tlight(%)**	**sign(%)**	**veg.(%)**
RPPNet (N=3)	**95.44**	**60.63**	82.24	**25.17**	20.72	12.92	**22.38**	**26.10**	**84.61**
RPPNet (N=4)	95.42	59.78	82.11	20.62	**21.47**	**13.07**	21.44	26.04	84.55
RPPNet (N=5)	95.41	59.72	**82.25**	24.25	21.00	12.72	21.17	25.64	84.45
RPPNet (N=6)	93.56	52.30	77.17	22.99	17.49	1.82	9.67	10.98	78.38
terrain(%)	**sky(%)**	**person(%)**	**rider(%)**	**car(%)**	**truck(%)**	**bus(%)**	**train(%)**	**mbike(%)**	**bike(%)**
42.22	87.69	40.27	9.38	79.08	14.70	35.83	17.43	9.87	35.42
39.60	87.08	40.90	9.74	79.04	12.02	33.82	**19.89**	10.87	**36.03**
39.10	**88.01**	**41.14**	**10.20**	**79.53**	**14.78**	**38.72**	17.19	**13.05**	35.49
34.09	81.68	28.63	2.35	68.22	13.12	31.13	12.49	5.27	27.69

TABLE V

COMPARISON OF THE TEST RESULTS OF MODELS WITH DIFFERENT INPUTS ON TWO METRICS OF PA AND mIoU ON THE CITYSCAPES TEST SET

Input type	Pixel Acc. (%)	mIoU (%)
Left image	88.68	40.16
Binocular images	**89.02**	**41.49**

TABLE VI

RESULTS OF SETTING DIFFERENT TOTAL N OF THE HISTORY FRAMES FOR RPPNET ON THE CITYSCAPSES VALIDATION SET. IT IS IMPORTANT TO DETERMINE THE NUMBER OF HISTORICAL FRAMES FOR THE PREDICTIVE PARSING TASK, WE CAN SEE N = 5 WORKS BEST

Method	Pixel Acc. (%)	mIoU (%)
RPPNet (N=3)	89.73	41.76
RPPNet (N=4)	89.78	42.21
RPPNet (N=5)	**89.82**	**42.31**
RPPNet (N=6)	86.81	35.21

listed in Table II, which shows that the results of our RPPNet are significantly better than that of PPNet and 16 out of the 19 classes of our RPPNet achieve higher performance. The qualitative contrast experimental result, as illustrated in Fig. 4, also proves that RPPNet predicts the parsing map of the target frame more accurately than PPNet. The Fig. 4 shows the predictive parsing results of several sequences, note that each result is obtained with only 4 preceding frames. In this figure, the second row and the third row list the predictive parsing results of PPNet and RPPNet respectively.

Overall, our results have three outstanding advantages. First, the parsing results of PPNet are coarse, but our results are smoother. Second, the results of RPPNet contain more accurate and detailed architectures, since it can process the structures of small objects such as poles more elaborately because of the usage of binocular images. Last, RPPNet is more sensitive to dynamic information, such as the example of the second sequence in Fig. 4, the car should appear in the left yellow box is ignored by PPNet, while RPPNet can predict it.

Since RPPNet uses binocular images as input, it is inevitably more time-consuming than PPNet in terms of predicting parsing results, where parsing time for each frame of PPNet is about 0.18s, and that of RPPNet is about 0.33s. Both the models are implemented based on the public deep learning architecture TensorFlow [40] and tested on single NVIDIA GeForce GTX1080. However, if we use more efficient deep learning architectures such as Caffe [42] and adopt more powerful graphic cards, the predictive parsing time for each frame is bound to be greatly reduced.

C. Effect of the Stereo Image Pair

In order to confirm that the stereo image pair is indeed more effective for parsing, we also conduct experiment with our method acting on monocular images. Since ground-truth is labeled for the left images in the Cityscapes dataset, we replace the input of the RPPNet with the left images of the image pairs. Apart from this, the network has the same structure as the original RPPNet. Performance comparisons between predictive parsing results from the models with different inputs are shown in Table V. The quantitative result of the model using the binocular images as input is superior to the result of that using the monocular images as input on both PA(%) and mIOU(%).

D. Hyper-Parameter

It is important to choose the number of preceding frames to predict the parsing result of the current frame carefully, thus in this subsection, we conduct experiment on different previous frames number N used in our method. Before that, we can see that considerable predictive parsing results can be generated when N is equal to 4 by another predictive parsing work PPNet [7]. Based on this, we initially set the number of historical frames around 4 in the experiment. As we have known that if N is too small, RPPNet will not correctly capture the dynamic change between the sequence so that we cannot take advantage of the long term memory of LSTM, but if N is too large, it will lead to big computational burden. Besides, too far frames contribute little to current parsing work. Therefore, after comprehensive consideration, we set N to 3, 4, 5 and 6 respectively in this experiment.

When RPPNet takes different values of N, we train each model on the training set and compare their performance on the validation set so as to choose the most appropriate value for N. Note that for each model with a new N value, we randomly initialize all trainable parameters and train the model from scratch. Two values of PA(%) and mIOU(%) on the validation set are recorded in Table VI. We can observe that both the metrics reach the maximum when N is equal to 5, and when the parameter N equals 6, the performance of RPPNet drops sharply. The reason for this is that over-distant frames bring too much interference information to predictive parsing

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VII
PER-CLASS RESULTS OF THE CONTRAST EXPERIMENT BETWEEN ITNETS AND THEIR RESPECTIVE BASELINES ON THE CITYSCAPES TEST SET

Model	**road(%)**	**swalk(%)**	**build.(%)**	**wall(%)**	**fence(%)**	**pole(%)**	**tlight(%)**	**sign(%)**	**veg.(%)**
VGGNet-baseline	**95.17**	**73.12**	83.44	**53.01**	**56.89**	19.08	**37.47**	40.29	83.03
VGGNet-ITNet	95.12	72.35	**85.46**	51.58	46.22	**23.34**	36.10	**47.30**	**85.49**
terrain(%)	**sky(%)**	**person(%)**	**rider(%)**	**car(%)**	**truck(%)**	**bus(%)**	**train(%)**	**mbike(%)**	**bike(%)**
60.97	79.24	43.28	38.20	79.38	**60.48**	76.38	**77.99**	56.11	43.34
59.94	**81.96**	**53.68**	**43.69**	**84.42**	50.17	**81.22**	77.75	**60.42**	**48.24**

Model	**road(%)**	**swalk(%)**	**build.(%)**	**wall(%)**	**fence(%)**	**pole(%)**	**tlight(%)**	**sign(%)**	**veg.(%)**
ResNet-baseline	95.63	74.17	86.90	**53.85**	**48.24**	27.84	39.09	53.51	87.19
ResNet-ITNet	**97.44**	**79.88**	**89.61**	46.99	46.58	**50.33**	**57.86**	**69.69**	**90.14**
terrain(%)	**sky(%)**	**person(%)**	**rider(%)**	**car(%)**	**truck(%)**	**bus(%)**	**train(%)**	**mbike(%)**	**bike(%)**
61.08	83.19	61.13	**49.27**	87.41	52.38	**83.05**	**78.70**	**61.97**	54.34
58.42	**92.11**	**72.50**	46.01	**92.33**	**60.70**	68.15	46.41	41.92	**67.40**

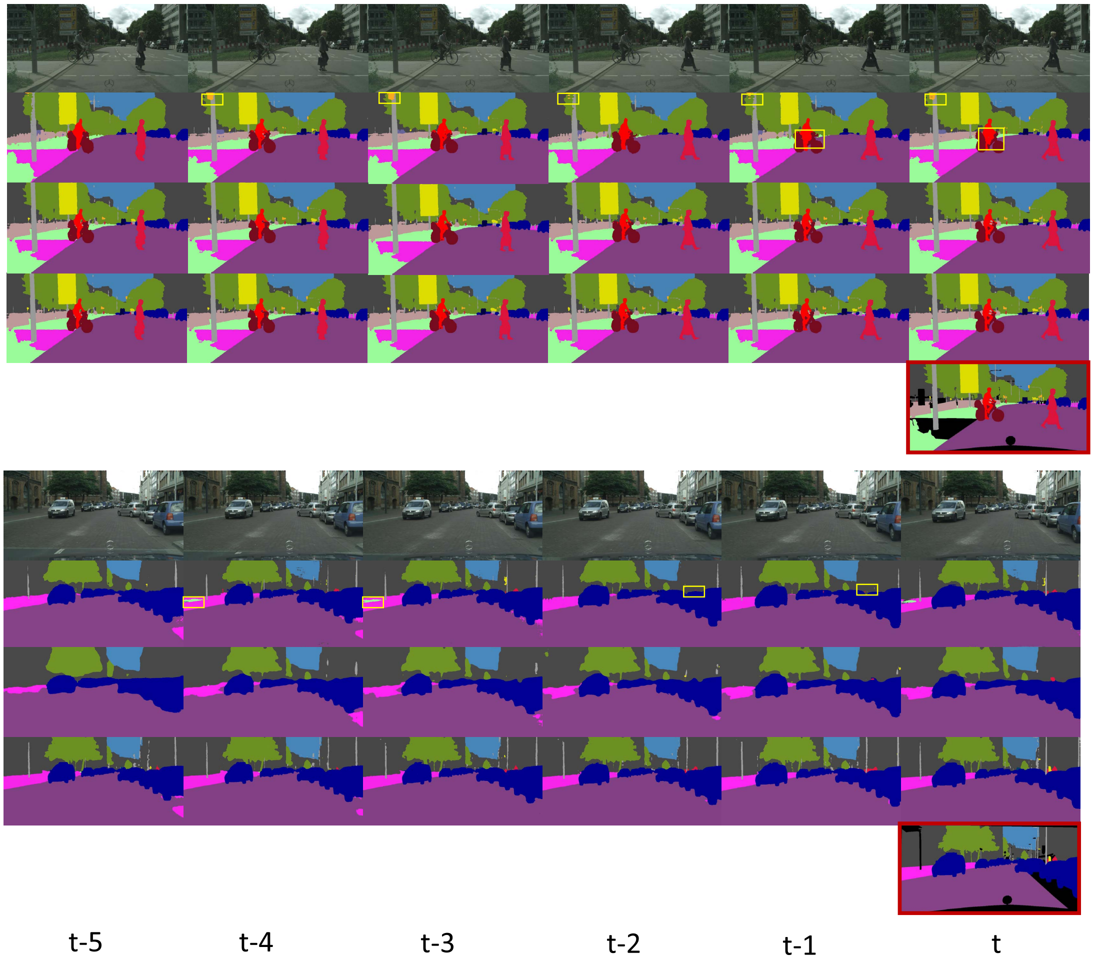


Fig. 5. Visual comparison of the parsing results of VGGNet-baseline and VGGNet-ITNet on Cityscapes test set. **Top row**: Six consecutive left images of a sequence. **Second row**: Parsing maps produced by VGGNet-baseline, and each map is generated using its single corresponding frame. **Third row**: Predictive parsing maps produced by RPPNet, and each map is predicted from 5 preceding frames of current time. **Fourth row**: Parsing results produced by VGGNet-ITNet. **Bottom**: the ground-truth annotation (with red boundary) of the frame t provided by the dataset. Best viewed in color and zoomed pdf.

results. Table IV shows per-class IoU values on different N, and when $N = 5$ we can achieve best performance in most circumstances.

E. Experiments About Integrated Parsing Network (ITNet)

In order to prove that our predictive features of RPPNet can improve the performance of conventional image parsing networks (CIPNets) effectively, we choose two typical network architectures as baselines. One is based on VGGNet, and the other is based on ResNet.

1) VGGNet-ITNet: Fully convolutional network (FCN) based on 16-layer VGGNet has achieved remarkable results in the image parsing task. We make some modifications to VGGNet as our EN_{CIPNet} to fit our task. Concretely, the original stride of the layers 'pool4' and 'pool5' is reduced

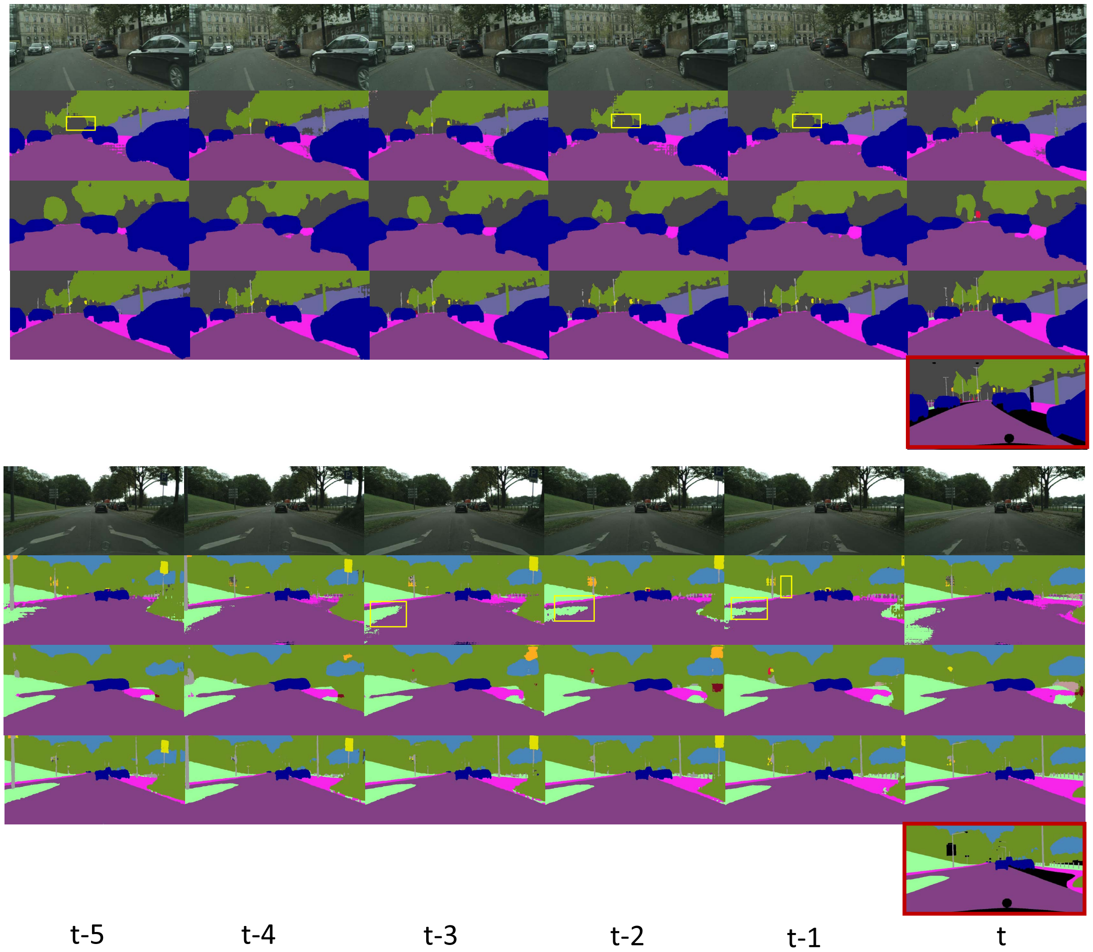


Fig. 6. Visual comparison of the parsing results of ResNet-baseline and ResNet-ITNet on Cityscapes test set. **Top row**: Six consecutive left images of a sequence. **Second row**: Parsing maps produced by ResNet-baseline, and each map is generated using its single corresponding frame. **Third row**: Predictive parsing maps produced by RPPNet, and each map is predicted from 5 preceding frames of current time. **Fourth row**: Parsing results produced by ResNet-ITNet. **Bottom**: the ground-truth annotation (with red boundary) of the frame t provided by the dataset. Best viewed in color and zoomed pdf.

TABLE VIII

Comparison Between IPNets and Their Respective Baselines on Two Indicators of PA and mIOU on the Cityscapes Test Set

Method	**Pixel Acc. (%)**	**mIoU (%)**
VGGNet-baseline	90.64	60.89
VGGNet-ITNet	**91.66**	**62.34**
ResNet-baseline	92.58	65.21
ResNet-ITNet	**94.46**	**67.08**

from 2 to 1, and we replace all layers of ‘conv5’ to be atrous convolutional layers by setting their dilation size to 2 so that its receptive field can be further expanded. Besides, we reduce the original 4096 feature maps in ‘fc6’ to 1024, which greatly reduce the number of parameters. As a result, we get the feature map with the scale of 1/8 according to the input size after ‘fc7’. As for the baseline method, which is named as ‘VGGNet-baseline’ in Tab. VIII, we use the same EN_{CIPNet} as its encoder, and add three deconvolutional layers with a stride of 2, same as decoder of ITNet, and a softmax layer to finish its parsing network.

2) *ResNet-ITNet:* The ITNet based on ResNet is built upon DeepLab [13], which employ the atrous convolution in scene parsing, and can get better results comparing to its previous versions. In this experiment, we don’t make any major change to the whole DeepLab except for some necessary requirements in our task setting. Concretely, the classification layer of ResNet is removed and the left part is set as EN_{CIPNet}, and the decoder is same as that in DeepLab. Besides, for the sake of fairness, the baseline method ‘ResNet-baseline’ in Tab. VIII is exactly the unmodified DeepLab.

Table VIII shows the comparing results of two baselines with their improved networks on the test set. Our two ITNets both achieve better results, where VGGNet-ITNet exceeds VGGNet-baseline by 1.02/1.45 and reaches 91.66/62.24 according to PA(%) and mIOU(%) respectively, meanwhile, ResNet-ITNet yields 94.46/67.08 and exceeds ResNet-baseline by the gap of 1.88/1.87 respectively. More details about the IoU values of each class are displayed in Table VII, from which we can observe that the IoU values of most classes are improved due to the spatial-temporal continuity introduced by the predictive features of RPPNet.

Fig. 5 and Fig. 6 show the visual results of our two ITNet, and demonstrate that our predictive features of RPPNet have strong ability to assist CIPNets to improve the performance of scene parsing by applying the predictive features. In these figures, the second row of every instance shows the parsing

TABLE IX
THE TRANSFER RESULTS OF RPPNET AND RESNET-ITNET ON THE KITTI DATASET

Method	Pixel Acc. (%)	mIoU (%)
RPPNet (N=5)	84.71	33.81
ResNet-ITNet	87.67	52.82

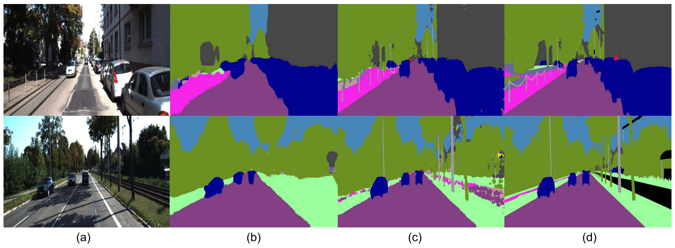


Fig. 7. Visual transfer results of RPPNet and ResNet-ITNet on the KITTI dataset. For each column we list the images of to be parsed (a), RPPNet (b), ResNet-ITNet (c), and Ground Truth (d).

results produced by CIPNet, the third row illustrates the result of the RPPNet, and the fourth row is the result of ITNet, which can be considered as the fused results from both CIPNet and RPPNet. From the figures, we can discover that the continuity of parsing maps between frames in the second row is significantly improved, especially the discontinuous parts in the second row highlighted in yellow are compensated in the fourth row.

F. The Transferability of RPPNet and ITNet

We test the transferability of the models we proposed on the KITTI [43] dataset. The data format and metrics for semantic segmentation in the KITTI dataset are conform with the Cityscapes dataset. We test the model on the KITTI dataset directly which is trained on the Cityscapes dataset. The test results on the KITTI dataset are shown in Table IX. We can see that the values of PA and mIOU on the KITTI dataset are a little lower because of no transfer learning or fine-tune conducted on the KITTI dataset. However, Qualitative results in Fig. 7 reflect the parsing results have certain reference significance.

V. CONCLUSION

This paper proposed a novel video scene predictive parsing model RPPNet, which can not only predict the image parsing results of the future frames, but also be used to assist the conventional image parsing of current frame. The superiority of RPPNet benefits from two strategies applied in our method. First, the adoption of binocular stereo images, which can mine 3D structural information, has greatly improve the performance of finding small objects such as poles. Second, the LSTM network is used skillfully to predict the features of the target frame, which can discover the spatial-temporal consistence between the preceding frames and current target frame, and make the parsing results more accuracy for moving objects, such as vehicles and pedestrians. In addition, the predictive features of RPPNet can be further used to assist the traditional image parsing networks to parse video scenes. The experiments on the Cityscapes prove that the proposed RPPNet can well solve the both tasks of predictive image parsing and conventional image parsing, and show the superiority comparing to some state-of-the-art methods.

REFERENCES

[1] T. Akilan, Q. M. J. Wu, W. Jiang, A. Safaei, and J. Huo, “New trend in video foreground detection using deep learning,” in *Proc. IEEE 61st Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2018, pp. 889–892.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[3] 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, Dec. 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[5] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[6] L. Li, B. Qian, J. Lian, W. Zheng, and Y. Zhou, “Traffic scene segmentation based on RGB-D image and deep learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1664–1669, May 2018.

[7] X. Jin *et al.*, “Video scene parsing with predictive feature learning,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5580–5588.

[8] A. Graves, *Long Short-Term Memory*. Berlin, Germany: Springer, 2012.

[9] M. Wollmer *et al.*, “Online driver distraction detection using long short-term memory,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 574–582, Jun. 2011.

[10] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, “A 3D CNN-LSTM-based image-to-image foreground segmentation,” *IEEE Trans. Intell. Transp. Syst.*, to be published.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Simultaneous detection and segmentation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zurich, Switzerland: Springer, 2014, pp. 297–312.

[13] L.-C. 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2881–2890.

[15] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[16] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. (2017). “Rethinking atrous convolution for semantic image segmentation.” [Online]. Available: https://arxiv.org/abs/1706.05587

[17] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Indoor semantic segmentation using depth information,” in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–8.

[18] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.

[19] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: Edge-preserving interpolation of correspondences for optical flow,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1164–1172.

[20] S. Gupta, P. Arbelaez, and J. Malik, “Perceptual organization and recognition of indoor scenes from RGB-D images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 564–571.

[21] M. Schwarz, H. Schulz, and S. Behnke, “RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features,” in *Proc. Int. Conf. Robot. Automat.*, May 2015, pp. 1329–1335.

[22] D. Lin, G. Chen, D. Cohen-Or, P.-A. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of RGB-D images," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1311–1319.

[23] X. Jin, Y. Chen, J. Feng, Z. Jie, and S. Yan, "Multi-path feedback recurrent neural network for scene parsing," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4096–4102.

[24] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7361–7369.

[25] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 2392–2396.

[26] S.-H. Chen, S.-H. Hwang, and Y.-R. Wang, "An RNN-based prosodic information synthesizer for Mandarin text-to-speech," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 3, pp. 226–239, May 1998.

[27] R. Wang, M. Panju, and M. Gohari. (2017). "Classification-based RNN machine translation using GRUs." [Online]. Available: https://arxiv.org/abs/1703.07841

[28] B. Zhang, D. Xiong, and J. Su. (2018). "Cseq2seq: Cyclic sequence-to-sequence learning." [Online]. Available: https://arxiv.org/abs/1607.08725

[29] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098.

[30] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1134–1142.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

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

[33] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[34] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–11.

[35] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.

[36] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[37] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D scene analysis from a moving vehicle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[38] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.

[39] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Proc. Int. Symp. Vis. Comput.*, 2016, pp. 234–244.

[40] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

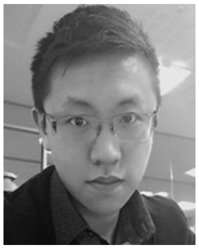
[41] J. Deng, W. Dong, R. Socher, L.-J. Li, L. Kai, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[42] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

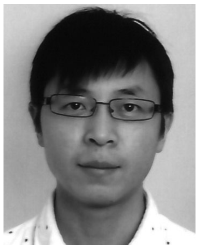
[43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

Lingli Zhou received the B.S. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2016, where she is currently pursuing the master's degree with the School of Computer Science and Engineering. Her current research interests include semantic segmentation, stereo vision, and deep learning.

Haofeng Zhang received the B.Eng. and Ph.D. degrees from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2007, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include computer vision and robotics.



Yang Long received the M.Sc. and Ph.D. degrees in computer vision and machine learning from the Department of Electronic and Electrical Engineering, the University of Sheffield, U.K., in 2014 and 2017, respectively. He is currently a Research Fellow with Open Laboratary, School of Computing, University of Newcastle. His research interests include artificial intelligence, machine learning, computer vision, and deep learning, zero-shot learning, with focus on transparent AI for healthcare data science.



Ling Shao (M'09–SM'10) is currently the CEO and a Chief Scientist of the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging. He is an Associate Editor of the IEEE Transactions on Image Processing, the IEEE Transactions on Neural Networks and Learning Systems, and several other journals. He is a fellow of the International Association of Pattern Recognition, the Institution of Engineering and Technology, and the British Computer Society.

Jingyu Yang received the B.Sc. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China. From 1982 to 1984, he was a Visiting Scientist with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994, he was a Visiting Professor with the Department of Computer Science, Missourian University. In 1998, he was a Visiting Professor with Concordia University, Canada. He is currently a Professor and the Chairman with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He has authored over 100 scientific papers in computer vision, pattern recognition, and artificial intelligence. His current research interests include image processing, robot vision, pattern recognition, and artificial intelligence. He received over 20 provincial awards and national awards.