MA-ResNet50: A General Encoder Network for Video Segmentation

Xiaotian Liu, Lei Yang, Xiaoyu Zhang and Xiaohui Duan
School of Electronics Engineering and Computer Science, Peking University, Beijing, China

Keywords: Video Segmentation, Attention Mechanism, Encoder Network.

Abstract: To improve the performance of segmentation networks on video streaming, most researchers now use optical-flow based method and non optical-flow CNN based method. The former suffers from heavy computational cost and high latency while the latter suffers from poor applicability and versatility. In this paper, we design a Partial Channel Memory Attention module (PCMA) to store and fuse time series features from video sequences.Then, we propose a Memory Attention ResNet50 network (MA-ResNet50) by combining the PCMA module with ResNet50, making it the first video based feature extraction encoder appliable for most of the currently proposed segmentation networks. For experiments, we combine our MA-ResNet50 with four acknowledged per-frame segmentation networks: DeeplabV3P, PSPNet, SFNet, and DNLNet. The results show that our MA-ResNet50 outperforms the original ResNet50 generally in these 4 networks on VSPW and CamVid. Our method also achieves state-of-the-art accuracy on CamVid. The code is avilable at *https://github.com/xiaotianliu01/MA-Resnet50*.

1 INTRODUCTION

As a video scene analysis technology, video segmentation is to assign pixel-wise labels for voxels (pixels from Spatial-Temporal viewpoint) (Qiu et al., 2018).As video becomes the main medium of information transmission,video segmentation now plays an increasingly important role in many cutting-edge technologies such as autonomous driving (Zhang et al., 2013; Teichmann et al., 2018), augmented reality (Miksik et al., 2015), robotic vision (Vineet et al., 2015), and so on.

In recent years, with the development of deep convolutional neural networks, some non-sequential segmentation networks (Chen et al., 2018; Zhao et al., 2017; Lee et al., 2019; Yin et al., 2020) have already achieved relatively high performance on several per-frame annotated datasets. However, the direct application of these non-sequential models on video steaming always brings about two problems, i.e., unstable infer results and redundant calculations, mainly because of the disability of integrating spatial and temporal relations between consecutive frames.

Nowadays, researchers on video feature integration have mainly developed three methods, i.e., optical-flow based method, spatial and temporal CNN based method, and non-CNN algorithms based method. For the first method, optical-flow can measure the apparent motion of pixels between consecutive frames, so it is used in keyframe mechanism to reduce calculations (Xu et al., 2018; Zhu et al., 2017; Li et al., 2018) and fed into CNN as extraneous information to improve accuracy (Ding et al., 2020; Gadde et al., 2017; Nilsson and Sminchisescu, 2018). However, calculating optical-flow (by algorithms or CNN) can be computationally expensive, which affects models' speed and latency. For the second method, researchers design some special spatial-temporal CNN networks for video sequence and conduct end-to-end training progress to extract spatial and temporal features simultaneously (Qiu et al., 2018; Siam et al., 2017; Siam et al., 2016; Hu et al., 2020). Although this method doesn't introduce extra calculations, every structure proposed can only be applied in one specified network, which impairs its applicability and versatility. For the last method, to achieve feature fusion outside CNN, researchers propose some non-CNN algorithms which won't introduce too much computational cost and can be modularized to different models (Lin et al., 2019; Wang et al., 2021a; Wang et al., 2021b). Our proposed method belongs to the third category illustrated above.

In contrast to the aforementioned methods, our method achieves a trade-off between computational cost and model's versatility. Our contributions can be summarized as follows: (1) We design a novel Partial Channel Memory Attention module (PCMA)

Liu, X., Yang, L., Zhang, X. and Duan, X.
MA-ResNet50: A General Encoder Network for Video Segmentation.
DOI: 10.5220/0010800800003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 79-86
ISBN: 978-989-758-555-5; ISSN: 2184-4321

based on attention mechanism to capture relations between frames outside CNN, which won't introduce so much computational cost as optical-flow based method. (2) We propose a general Memory Attention ResNet50(MA-ResNet50) encoder network for video sequence feature extraction, which covers the deficiency of applicability and versatility for spatial and temporal CNN based method. (3) We apply our proposed MA-ResNet50 to four acknowledged per-frame segmentation networks. The experiments show that our MA-ResNet50 is superior to the original ResNet50 in accuracy on two video segmentation datasets, namely CamVid and VSPW, for video semantic segmentation task and video object segmentation task. Also, our method achieves state-of-the-art accuracy on CamVid.

2 RELATED WORKS

2.1 Attention Mechanism

Originated from Machine Translation and Natural Language Processing (Bahdanau et al., 2016), the attention mechanism was designed to adaptively allocate limited computing resources on different parts of input data according to their contributions to the final result. In computer vision area, attention mechanism specifically contains channel attention mechanism (Hu et al., 2018), spatial attention mechanism (Jaderberg et al., 2016), and channel-spatial attention mechanism (Woo et al., 2018).

There are two major tasks in the video segmentation area: video semantic segmentation (Garcia-Garcia et al., 2018), which requires the identification of every pixel's class in a video, and video object segmentation (Caelles et al., 2017), which requires the separation of an object from the background in a video. For video semantic segmentation task, the memory attention mechanism was firstly introduced by (Wang et al., 2021b), who uses a key-value memory method to calculate attention matrix with historical features, retrieving information from the previous frames to enhance the representation of the current frame. For video object segmentation task, (Wang et al., 2021a) uses the memory attention mechanism to renew non-redundant information between frames, and (Oh et al., 2019) leverages a similar method to match features between memory and current frames. Also based on attention mechanism, our PCMA module improves the integrality of the temporal features by exploiting features of different depths and reduce computational cost at the same time.

2.2 ResNet

Winner of ILSVRC-2016 with 96.4% accuracy, ResNet (He et al., 2016) is well known for its introduction of residual blocks, which makes training deep neural networks easier by inserting skip connections among neural networks. Since its emergence, many newly proposed segmentation networks (Chen et al., 2018; Zhao et al., 2017; Lee et al., 2019; Yin et al., 2020) have used it as a well pretrained encoder network, and achieved satisfactory performance on many different datasets by combining it with different proposed decoder networks. Developed from ResNet by using its 50 layers version, our MA-ResNet50 makes it possible for encoder network to extract temporal features and achieve better performance on video segmentation task.

3 METHODOLOGY

3.1 Overview

An overview of our Memory Attention ResNet50 is illustrated in Figure 1. The whole model mainly contains three elements, i.e., ResNet50 blocks, Partial Channel Memory Attention module (PCMA) and memory M_n where $n \in \{1,2,3,4\}$. For ResNet50, with 4 convolutional blocks, it extracts 4 high-dimensional features C_n where $n \in \{1,2,3,4\}$ from the input image I_i. For the PCMA module, it obtains the long-range temporal context information from M_n and uses it to enhance the presentation of current feature C_n, which we will explain in more detail in the next part. For the memory M_n, it is a key-value structure data, where the key is used to generate attention matrix by calculating pixel-wise feature correlation between consecutive frames and the value is used to generate enhanced features. Specially, memory M_n, contains the feature information for historical T frames and both its key M_n^K and value M_n^V are generated by concatenating T enhanced features from PCMA module in channel dimension.

For the whole segmentation process, our MA-ResNet50 works in a cyclic updating way. For every image fed into the network, it is extracted firstly by ResNet50, thus producing four features of different sizes. Along with their corresponding memory, the features are then fed into PCMA to generate temporal enhanced features. On one hand, the enhanced features are sent to segmentation head network to generate segmentation results. On the other hand, the enhanced features are used to update memory on a First In First Out(FIFO) basis.

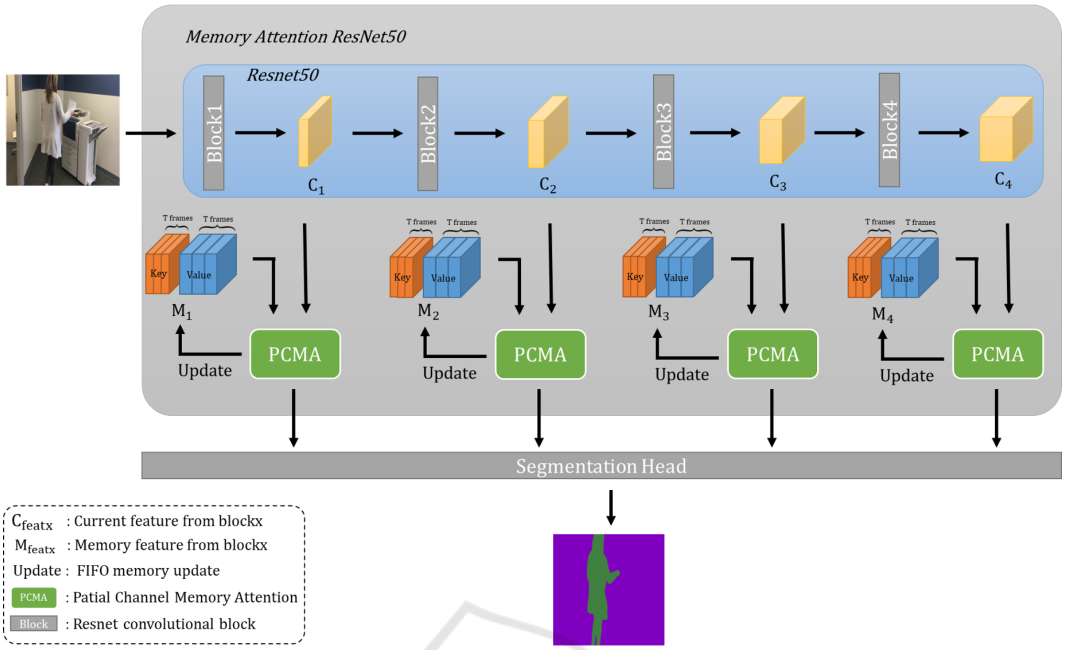
Memory Attention ResNet50
Resnet50
Block1
Block2
Block3
Block4
C_1
C_2
C_3
C_4
T frames
T frames
Key
Value
M_1
M_2
M_3
M_4
PCMA
Update
Segmentation Head
C_{featx} : Current feature from blockx
M_{featx} : Memory feature from blockx
Update : FIFO memory update
PCMA : Patial Channel Memory Attention
Block : Resnet convolutional block

Figure 1: Overview of our Partial Memory Attention ResNet50.The input current image is firstly extracted by 4 convolutional blocks from ResNet50. Along with their corresponding memory M_1, M_2, M_3, M_4, the output features C_1, C_2, C_3, C_4 are then fed into PCMA module for feature enhancement. Output enhanced features from PCMA module are then sent to segmentation head for results generation and to memory for updating.

3.2 Partial Channel Memory Attention Module

The intuition of PCMA module comes from human visual system. In a continuous observation process, human visual system forms a short-time memeory which contains past semantic information about the observation tatrget. When understanding the current view, human visual system exploits this memeory as reference to allocate attention on different parts of the view. In PCMA module, we apply the similar mechanism by using feature matrices to store semantic information from past frames and using correlation calculation to enhance attention allocation on current frame.

Figure 2 shows the pipeline of our PCMA module. As mentioned in 3.1, the input of PCMA module is the current feature C_n and key-value data M_n^K , M_n^V from memory feature M_n. The output is the enhanced feature, namely $Enhanced - Feat$, for generating results and key-value data, namely $Update - Key$ and $Update - Value$, for memory updating.

To reduce the computational cost of generating memory attention, we adopt a 2D convolution layer to compress the input feature by halving the number of its channels. Specially, instead of feeding the whole $C_n \in \mathcal{R}^{c\times h\times w}$ into the 2D convolution layer, we innovatively select partial channels of C_n to be calculated and enhanced, namely $C_n^{selected}$, which balances the contributions of historical and current information to the final results and also saves calculation resources. To adjust the selection strategy, we introduce an exogenous ratio, namely $r \in [0,1]$, whose effect is tested in the experiments part. The selection strategy can be illustrated as:

$$C_n^{selected} = \Theta\left(C_n^{(1-r)*c}, C_n^{(1-r)*c+1}, \dots, C_n^{c-1}, C_n^{c}\right) \quad (1)$$

Here C_n^i donates the 2D tensor in i^{th} channel of C_n, Θ donates concatenate operation and c donates the channel number of C_n.After the key encoder progress, we flatten the output tensor $C_n^K \in \mathcal{R}^{c*r*0.5\times h\times w}$ to a 2D matrix, namely $C_n^{K'} \in \mathcal{R}^{c*r*0.5\times h*w}$, for later calculation.

3.2.1 Enhanced Feature Generation

After flattening the $M_n^K \in R^{T*c*r*0.5\times h\times w}$ to a 2D matrix, namely $M_n^{K'} \in R^{c*r*0.5\times T*h*w}$, we conduct matrix multiplication on $C_n^{K'}$ and $M_n^{K'}$ to generate memory attention A_n, which can be illustrated as:

$$A_n(i,j) = \sum_{l=1}^{c*r*0.5} M_n^{K'}(l,i) * C_n^{K'}(l,j) \quad (2)$$

For $A_n \in R^{T*h*w\times h*w}$, $A_n(i,j)$ denotes the correlation between i^{th} pixel in M_n^K and j^{th} pixel in C_n^K,

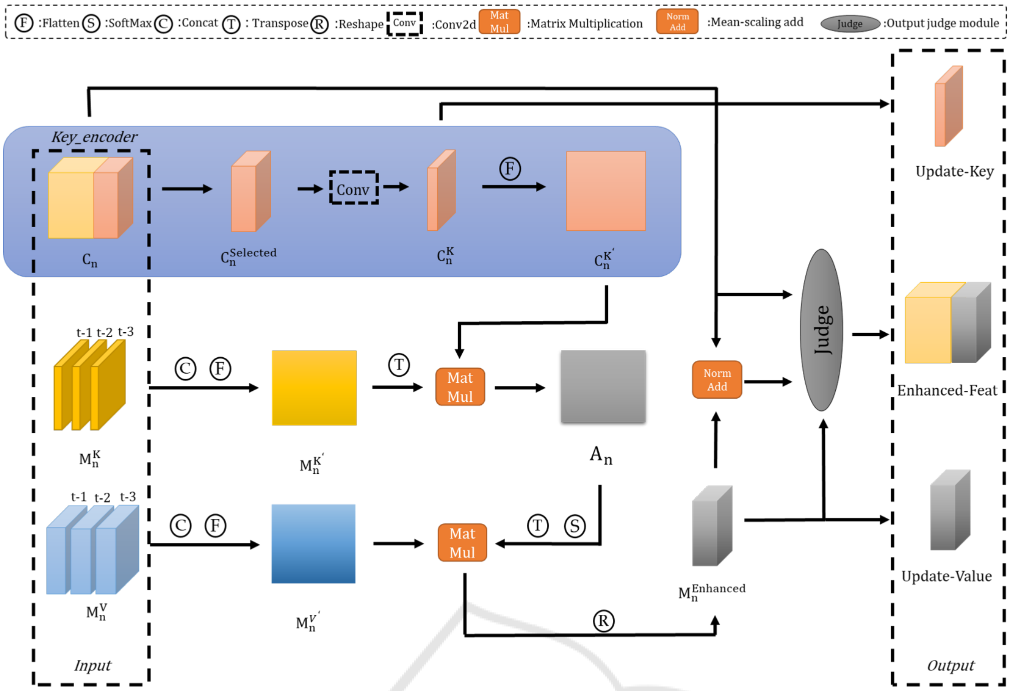
F :Flatten
S :SoftMax
C :Concat
T : Transpose
R :Reshape
Conv :Conv2d
Mat Mul :Matrix Multiplication
Norm Add :Mean-scaling add
Judge :Output judge module
Key_encoder
C_n
C_n^{Selected}
Conv
C_n^K
F
C_n^{K'}
t-1 t-2 t-3
M_n^K
C F
M_n^{K'}
T
Mat Mul
A_n
Norm Add
Judge
t-1 t-2 t-3
M_n^V
C F
M_n^{V'}
Mat Mul
T S
M_n^{Enhanced}
R
Input
Update-Key
Enhanced-Feat
Update-Value
Output

Figure 2: Pipeline of Partial Channel Memory Attention module. After Key Encoder, the current feature C_n is transferred to 2D matrix $C_n^{K'}$. The key-value memory data M_n^K, M_n^V are also transferred to 2D matrix $M_n^{K'}$, $M_n^{V'}$. Memory attention matrix A_n is obtained by conducting matrix multiplication on $C_n^{K'}$ and $M_n^{K'}$. Similarly, enhanced feature $M_n{}^{enhanced}$ is obtained by conducting matrix multiplication on A_n and $M_n^{V'}$.After feature aggregation and the Judge module, the enhanced feature of full channels is output for results generation. Also, the intermediate result C_n^K and $M_n{}^{enhanced}$ are output for memory updating.

which indicates the pixel-wise matching between historical features and current features. Then we conduct a softmax operation on A_n's first dimension for normalization and flatten $M_n^V \in R^{T*c*r \times h*w}$ to $M_n^{V'} \in R^{c*r \times T*h*w}$. Using normalized A_n as weight, we conduct matrix multiplication similarly on A_n and $M_n^{V'}$, which produces the final temporal enhanced feature, namely $M_n^{enhanced}$, after the reshape operation.

3.2.2 Output Judge Module

3.2.3 Partial Channel Key Encoder

For better feature aggregation, we adopt a mean-scaling add method, which is:

$$F_n = M_n{}^{enchanced} * \frac{mean(C_n^{seclected})}{mean(M_n^{enhanced})} + C_n^{seclected} \quad (3)$$

Here F_n donates the fusion feature.

In the actual test process, we find that when a temporal sequence has low consistency, mainly because of the rapid change of scenes and the low fps of input video, the relations between frames can be inapparent, which makes the F_n tend to be equalized. This happens more when the number of channels for the enhanced feature increases. Therefore, to avoid the equalized F_n from affecting the final results as noise, we design a judge module to output the final feature. The principle of the judge module can be shown as:

$$a = \frac{mean(F_n) - min(F_n)}{max(F_n) - min(F_n)} \quad (4)$$

$$O_n = \begin{cases} C_n^{selected} & \text{a <0.1 or a >0.9} \\ F_n & \text{0.1<a <0.9} \end{cases} \quad (5)$$

Here O_n donates the output embedding feature. Then, we combine O_n with the unselected channels from C_n to output the full channels enhanced feature to generate segmentation results. Specially, C_n^K and $M_n{}^{enhanced}$ are also output as key-value data to update memory.

4 EXPERIMENTS

4.1 Datasets

To verify the validity of our MA-ResNet50 on video semantic segmentation task and video object segmentation task, we conduct experiments on two datasets,

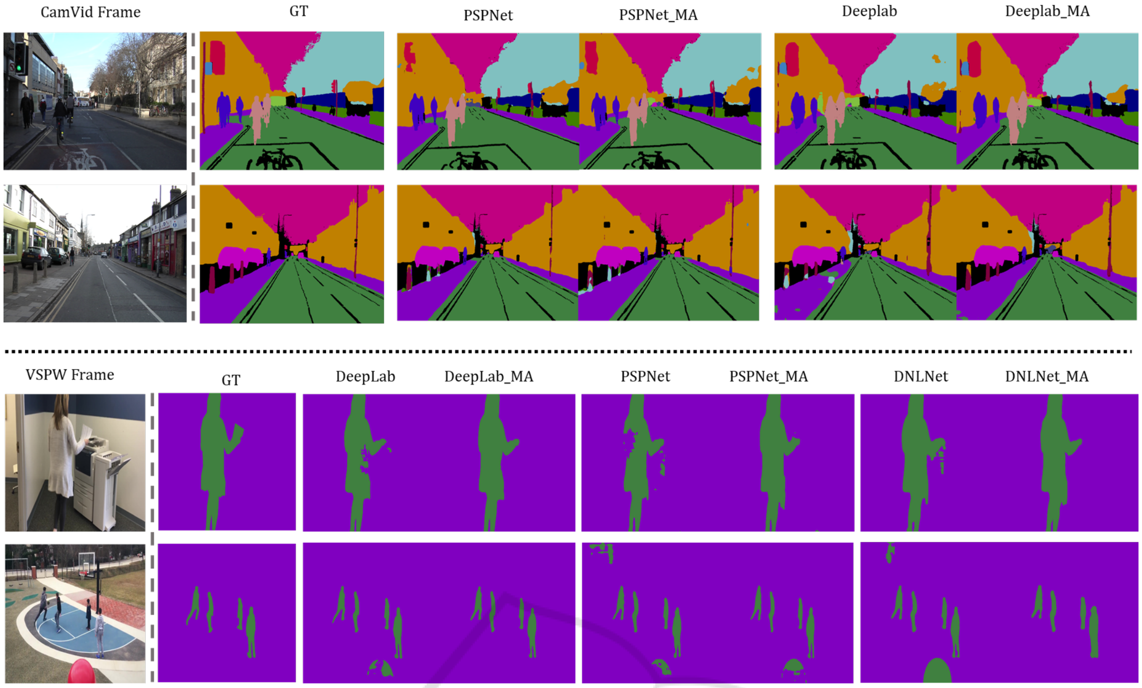

Figure 3: Segmentation results visualization. Specially, model and model_MA separately stand for results of model based on original ResNet50 and our MA-ResNet50.

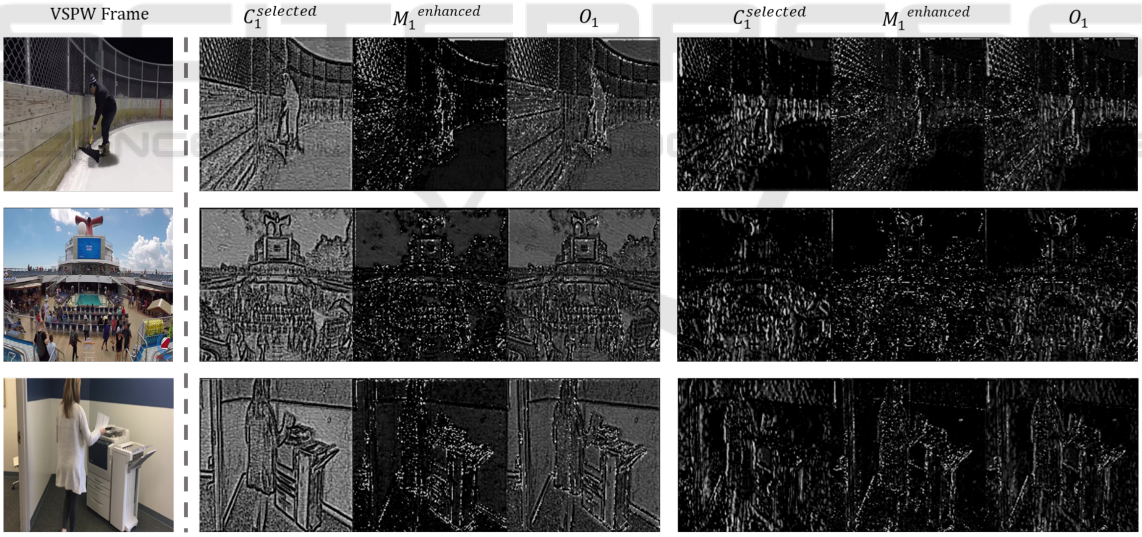

Figure 4: 2D depth maps visualization. $C_1^{selected}$ stands for the original feature from Block1 of ResNet50. $M_1^{enhanced}$ stands for the enhanced memory feature of $C_1^{selected}$. O_1 stands for the final output enhanced feature, which is $C_1^{selected}$+$M_1^{enhanced}$ in most cases.

i.e., CamVid (Brostow et al., 2009) and VSPW (Miao et al., 2021). With 11 categories labeled, CamVid is a video semantic segmentation dataset for traffic scenes which contains 4 videos and each video is annotated at 1 fps. For VSPW, it contains 3536 long-temporal clips for various real-world scenarios with dense pixel-wise annotations at 15fps. To conduct experiments on video object segmentation task, instead of using VSPW's whole 124 categories, we only select the category with the highest frequency as the detection target, which is the Person category. Besides, we adopt mean Intersection-over-Union(mIoU) as our

Table 1: Comparison of ResNet50 baseline and our MA-ResNet50 on CamVid and VSPW.

Dataset	Model	Encoder	mIoU%	fps	Δ **mIoU%**
CamVid (Less Temporal)	DeeplabV3P	ResNet50	73.77	45.78	**+1.25**
		MA-ResNet50	**75.02**	38.60	
	PSPNet	ResNet50	69.61	46.14	**+1.78**
		MA-ResNet50	**71.39**	42.60	
	SFNet	ResNet50	75.55	12.99	**+1.50**
		MA-ResNet50	**77.05**	11.55	
VSPW (More Temporal)	DeeplabV3P	ResNet50	81.79	43.45	**+1.31**
		MA-ResNet50	**83.10**	37.27	
	PSPNet	ResNet50	75.95	45.35	**+2.57**
		MA-ResNet50	**78.52**	42.08	
	DNLNet	ResNet50	78.94	11.79	**+1.88**
		MA-ResNet50	**80.82**	10.88	

revaluation metric, which can be illustrated as:

$$mIoU = \frac{1}{k+1}\sum_{i=1}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \quad (6)$$

Here k donates the number of segmentation classes and p_{ij} donates the number of pixels whose ground truth class is i and prediction class is j.

4.2 Training

To ensure the rigor of the controlled trial, we train all the networks with the same setting on Nvidia TeslaV100 GPU. For loss and optimizer, we use cross-entropy loss and SGD with its weight decay set to 4e-5. We employ a poly learning rate policy to adjust the learning rate every iteration from 0.1 to 0.001. For resolution, images from CamVid are resized to 960×720 while images from VSPW are resized to 640×640. Besides, the total iteration is set to 60000 for CamVid and 120000 for VSPW and the batch size is set to 2. For data augmentation, we adopt resize, random horizontal flip and normalize without shuffling.

4.3 Improving ResNet50 Baseline

To prove that our MA-ResNet50 is superior to the original ResNet50 baseline in segmentation accuracy, we employ 4 acknowledged per-frame segmentation models, i.e., DeeplabV3P (Chen et al., 2018), PSPNet (Zhao et al., 2017), SFNet (Lee et al., 2019) and DNLNet (Yin et al., 2020). We respectively replace the encoder networks in these 4 models with the original ResNet50 and our MA-ResNet50. After the training process in the same setting illustrated above, the comparison of results is shown in Table 1.

The comparison implies that our MA-ResNet50 outperforms the original ResNet50 generally on these two datasets by 1% - 3% mIoU. Specially, in contrast to CamVid, the improvement is more significant for more temporal data from VSPW.

To make the comparison more intuitive, we visualize some of the segmentation results. Also, we visualize some 2D depth maps from $C_n^{selected}$ and $M_n^{enhanced}$ to show the effect of enhancement. The visualization results are shown in Figure 3 and Figure 4.

4.4 Ablation Study

For ablation study, we test the effect of two variables in our model, i.e., T, the number of stored features in memory and r, the channel select ratio. Experiment results for ablation study are illustrated in Table 2 and Table 3.

Table 2: Effect of the T variable on VSPW for three models based on MA-ResNet50.

Model	mIoU%		
	T=1	T=2	T=3
DeepLabv3p	**83.1**	83.09	83.09
PSPNet	78.47	**78.52**	78.48
DNLNet	80.81	**80.82**	80.81

The results imply that the optimal T tends to be 2 for more temporal data from VSPW and the optimal r tends to be 0.25 for less temporal data from CamVid. It should be mentioned, however, that greater T and r will result in an increase in computational cost. To illustrate this numerically, we calculate FLOPS of MA-ResNet50 with different combinations of T and r, and the results are shown in Table 4.

Table 3: Effect of the r variable on CamVid for two models based on MA-ResNet50.

Model	mIoU%		
	r=0.125	r=0.25	r=0.5
PSPNet	71.3	**71.39**	71.25
SFNet	77	**77.05**	76.27

Table 4: FLOPS for different combination of r and T.Specially, the last row stands for $\frac{FLOPS\ of\ MA-ResNet50}{FLOPS\ of\ ResNet50} * 100\%$.

r	T	FLOPS(B)	/ResNet50(%)
0.125	1	4.11	102.59
	2	6.82	104.43
	3	9.52	106.01
0.25	1	11.07	107.08
	2	16.47	110.38
	3	21.88	113.8
0.5	1	33.47	121.13
	2	44.29	128.03
	3	55.1	134.8

4.5 State-of-the-Art Comparison

Applying MA-ResNet50 to SFNet (Lee et al., 2019), our method achieves better accuracy than other state-of-the-art methods for video segmentation task on CamVid. The comparison between our method and other optical-flow based and non optical-flow based methods is shown in Table 5.

Table 5: Comparison of our method with other state-of-the-art methods on CamVid.

Method	mIoU%
GRFP (Nilsson and Sminchisescu, 2018)	66.10
Netwarp(Gadde et al., 2017)	67.10
TDNet(Hu et al., 2020)	76.00
TMANet(Wang et al., 2021b)	76.50
Ours	**77.05**

5 CONCLUSIONS

In this paper, we propose a Memory Attention ResNet50 encoder network for video sequence feature extraction. Specially, we design a Partial Channel Memory Attention module to integrate long-term temporal relations in consecutive frames. Experiments imply that our method outperforms the original ResNet50 in 4 per-frame segmentation networks. Our method also achieves state-of-the-art accuracy on CamVid. In future work, we will mainly work on new correlation calculation algorithms to reduce computational cost and improve enhancement effectiveness.

REFERENCES

Bahdanau, D., Cho, K., and Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate.

Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.*, 30:88–97.

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Gool, L. V. (2017). One-shot video object segmentation.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ding, M., Wang, Z., Zhou, B., Shi, J., Lu, Z., and Luo, P. (2020). Every frame counts: Joint learning of video segmentation and optical flow. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):10713–10720.

Gadde, R., Jampani, V., and Gehler, P. V. (2017). Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., and Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, P., Caba, F., Wang, O., Lin, Z., Sclaroff, S., and Perazzi, F. (2020). Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2016). Spatial transformer networks.

Lee, J., Kim, D., Ponce, J., and Ham, B. (2019). Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y., Shi, J., and Lin, D. (2018). Low-latency video semantic segmentation. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR).

Lin, J., Gan, C., and Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Miao, J., Wei, Y., Wu, Y., Liang, C., Li, G., and Yang, Y. (2021). Vspw: A large-scale dataset for video scene parsing in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Miksik, O., Vineet, V., Lidegaard, M., Prasaath, R., Nießner, M., Golodetz, S., Hicks, S. L., Pérez, P., Izadi, S., and Torr, P. H. (2015). The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3317–3326, New York, NY, USA. Association for Computing Machinery.

Nilsson, D. and Sminchisescu, C. (2018). Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oh, S. W., Lee, J.-Y., Xu, N., and Kim, S. J. (2019). Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Qiu, Z., Yao, T., and Mei, T. (2018). Learning deep spatio-temporal dependence for semantic video segmentation. *IEEE Transactions on Multimedia*, 20(4):939–949.

Siam, M., Valipour, S., Jägersand, M., and Ray, N. (2016). Convolutional gated recurrent networks for video segmentation. *CoRR*, abs/1611.05435.

Siam, M., Valipour, S., Jagersand, M., and Ray, N. (2017). Convolutional gated recurrent networks for video segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3090–3094.

Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., and Urtasun, R. (2018). Multinet: Real-time joint semantic reasoning for autonomous driving.

Vineet, V., Miksik, O., Lidegaard, M., Nießner, M., Golodetz, S., Prisacariu, V. A., Kähler, O., Murray, D. W., Izadi, S., Pérez, P., and Torr, P. H. S. (2015). Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 75–82.

Wang, H., Jiang, X., Ren, H., Hu, Y., and Bai, S. (2021a). Swiftnet: Real-time video object segmentation. *CoRR*, abs/2102.04604.

Wang, H., Wang, W., and Liu, J. (2021b). Temporal memory attention for video semantic segmentation. *CoRR*, abs/2102.08643.

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xu, Y.-S., Fu, T.-J., Yang, H.-K., and Lee, C.-Y. (2018). Dynamic video segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., and Hu, H. (2020). Disentangled non-local neural networks. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 191–207, Cham. Springer International Publishing.

Zhang, H., Geiger, A., and Urtasun, R. (2013). Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, X., Xiong, Y., Dai, J., Yuan, L., and Wei, Y. (2017). Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.