MMA: Motion Memory Attention Network for Video Object Detection

Huai Hu, Wenzhong Wang, Aihua Zheng, and Bin Luo(✉)

Anhui University, Hefei, China
huaihu5831@foxmail.com, {wenzhong,luobin}@ahu.edu.cn,
ahzheng214@foxmail.com

Abstract. Modern object detection frameworks such as Faster R-CNN achieve good performance on static images, benefiting from the powerful feature representations. However, it is still challenging to detect tiny, vague and deformable objects in videos. In this paper, we propose a Motion Memory Attention (MMA) network to tackle this issue by considering the motion and temporal information. Specifically, our network contains two main parts: the dual stream and the memory attention module. The dual stream is designed to improve the detection of tiny object, which is composed of an appearance stream and a motion stream. Our motion stream can be embedded into any video object detection framework. In addition, we also introduce the memory attention module to handle the issue of vague and deformable objects by utilizing the temporal information and distinguishing features. Our experiments demonstrate that the detection performance can be significantly improved when integrating the proposed algorithm with Faster R-CNN and YOLO$_{v2}$.

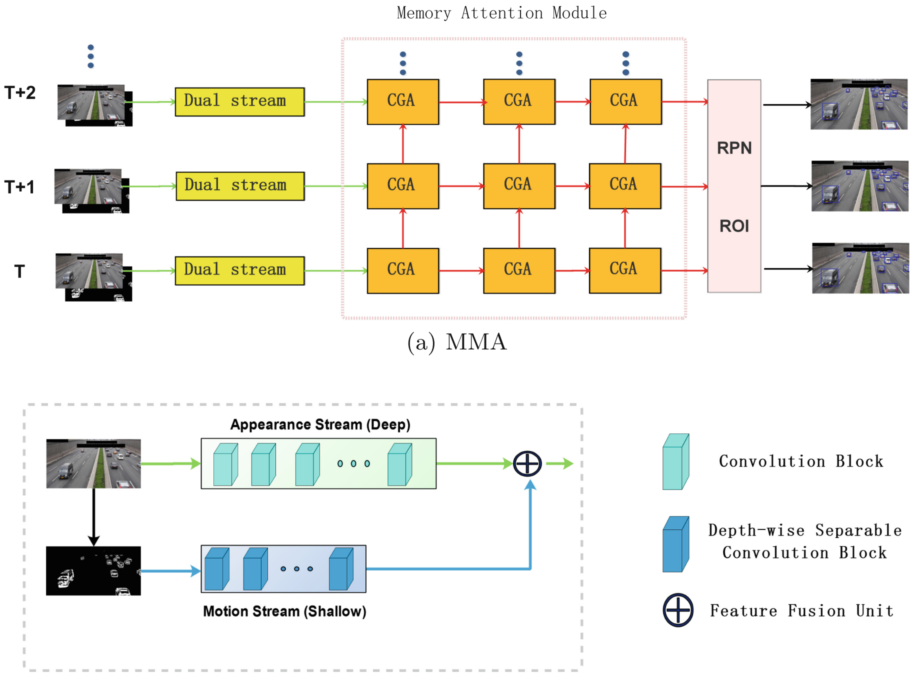
Keywords: Video object detection · Dual stream · Memory attention module

1 Introduction

Object detection is a fundamental task in computer vision. It has been widely used in many applications, such as monitoring system and autonomous driving, etc. In recent years, a lot of detectors based on ConvNets have been proposed to improve the accuracy and speed in object detection task [3,13,16]. Although they have achieved great success of object detection in image, the performance in the video object detection is still not satisfying for tiny, vague and deformable objects. Furthermore, distant objects on RGB frames are usually mixed with the background. The response of these objects on the feature map is not distinguishable enough, which significantly limits the performance of conventional detectors.

The temporal information in video plays an import role in video object detection [21,22]. They usually estimate the optical flow information between consecutive frames to improve the final detection results. However, the estimation of optical flow is time consuming for practical scenarios.

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11902, pp. 167–178, 2019.
https://doi.org/10.1007/978-3-030-34110-7_15



(a) MMA

(b) Dual stream

Fig. 1. (a) **Motion Memory Attention Network (MMA).** Our approach is comprised by dual stream and memory attention module. The memory attention module is shown in Fig. 3; (b) **Dual stream.** The appearance stream is the Faster R-CNN network pre-trained on the COCO dataset. The motion stream is composed of a number of column Depth-wise separable convolution blocks and takes temporal difference frames as input.

To handle aforementioned issues, in this paper, we propose a dual stream video object detection framework which composed of appearance and motion stream, to encode generic appearance and motion cues respectively. The appearance stream is the Faster R-CNN network pre-trained on the COCO dataset. The motion stream is used to mine the motion information. Our motion stream is composed of a number of column depth-wise separable convolution blocks and takes temporal difference frames as input, which greatly reduces computation cost compared to the optical flow based methods. For the tiny and blurry object in the temporal difference frames, the position response of the moving object on the feature map is obvious when the object moves, so our motion streams can capture these objects. Since the temporal difference frames is not valid for stationary objects, the appearance stream can provide complementary cues for object detection.

Furthermore, some detected objects maybe lost in subsequent frames due to occlusion or motion blur. To tackle this issue, we propose to introduce a memory attention module to exploit the temporal correlation in adjacent video frames. Given object states in one frame, we can reliably predict their states in the neighbouring frames using these inter-frame correlations. We use states vectors to describe each video frame and infer the states of any frame from a sequence of the states of its adjacent frames. Specially, the attention model is introduced into the recurrent memory network to refine the states vectors of the video frames. Therefore, the memory attention module can effectively capture the inter-frame correlations. By utilizing the states of adjacent frames, we can improve the detection results of occluded and blurred objects significantly.

The contributions of this paper can be summarized in the following three aspects:

- We propose a dual stream to capture motion information in consecutive video frames, in addition, to the appearance information, the motion information can enhance the response of the moving object in the feature map.
- A memory attention module is proposed to exploit the temporal correlation in adjacent video frames, and refine the states vector of each frame, which can recover the lost object encountering deformation and blurring.
- Our method leads to competitive performance on benchmark video object detection dataset DETRAC [20] across different detectors and backbone networks.

2 Related Work

2.1 Object Detection

Benefiting from the power of Deep ConvNes, object detectors such as Faster R-CNN [16] has shown dramatic improvements in accuracy. Two-stage detectors like R-CNN [4] directly combine the steps of cropping box proposals like Selective Search and classifies them through the CNN model. Compared with the traditional method, it obtains significant precision improvement and opens the deep learning era in object detection. Its descendants like Fast R-CNN [19] performs end-to-end classification and position regression loss training on convolutional neural networks. The Faster R-CNN suggests replacing the selective search with a Regional Recommendation Network (RPN), to generate candidate bounding boxes (anchor boxes) while filtering out background areas. Then it uses another tiny network based on these proposals for classification and bounding box location regression. In recent years, one-stage detectors like SSD [13] and YOLO [15] have been proposed for real-time detection with satisfactory accuracy. However, in contrast to these methods of still-image object detection, our method focuses on object detection in videos.

2.2 Object Detection in Video

Many researchers have focused on more generic categories and realistic videos, but their methods focus on post-processing class scores by static-image detectors to enforce temporal consistency of the scores. MCMOT [12] regards post-processing as a multi-object tracking problem, then uses the tracking target confidence to re-evaluate the confidence of detection. T-CNN [11] propagates the predicted boundary box to adjacent frames according to the pre-computed optical flow, and then uses the tracking algorithm of high confidence boundary box to select multiple candidate frames around the last frame and select the candidate box with the highest score. Han et al. [5] correlated the initial test results into the sequence. The weaker class scores in the same video sequence are improved, and the initial frame-by-frame detection results are improved. In contrast, our approach considers temporal information at the feature layer rather than post-processing the detected object frames. The entire framework completes video object detection via an end-to-end training.

2.3 Long Short Term Memory

LSTM [8] is a structure of Rnn cell that has been proven to be stable and powerful for modeling long-term dependencies, uses three gates (input, output, and forgetting gates) to control the transfer of information between units, and each gate has its own set of weights. The long-term short-term memory (LSTM) and the gated recursive unit (GRU) [2] as the advanced versions of RNN, can alleviate the problem of gradient disappearance to some extent [7,14]. GRU is simpler than LSTM since the output gate is removed from the unit and the output stream is indirectly controlled by the other two gates. Cell memory is also updated in different ways in the GRU. But the traditional GRU are designed to process text data rather than images. Using them on images may causes some problems, such as excessive training parameters to converge. Therefore, we need to convert a gated architecture to a convolutional architecture, replace dot product with convolutions, which effectively utilizes spatial information.

2.4 Attention Modules

Attention module can model long-term dependencies and has been widely used in the Natural Language Processing (NLP) field in recent years. Squeeze-and-Excitation Networks [10] enhance the representational power of the network by modeling channel-wise relationships in an attention mechanism. Chen et al. [1] makes use of several attention masks to fuse feature maps or predictions from different branches. Vaswani et al. [18] applies a self-attention model on machine translation. The attention modules are also increasingly applied in the image vision flied. For example, the work [9] proposes an object relation module to model the relationships among a set of objects, which improves object recognition. Our approach is motivated by the success of attention modules in the above works.

3 The Proposed Approach

Our approach is to detect generic objects in video, calibrate object locations and classify them without any manual intervention. Our method is composed of two modules: the dual stream and the memory attention module, as shown in Fig. 1. Firstly, we pre-train the appearance and motion stream separately to get better feature representation. Then, the output of these two streams are summed together as the encoding results. After that, we utilize the memory attention module to capture temporal information. The augmented features are then fed into the RPN module and the bounding box regression and classification are conducted for object detection.

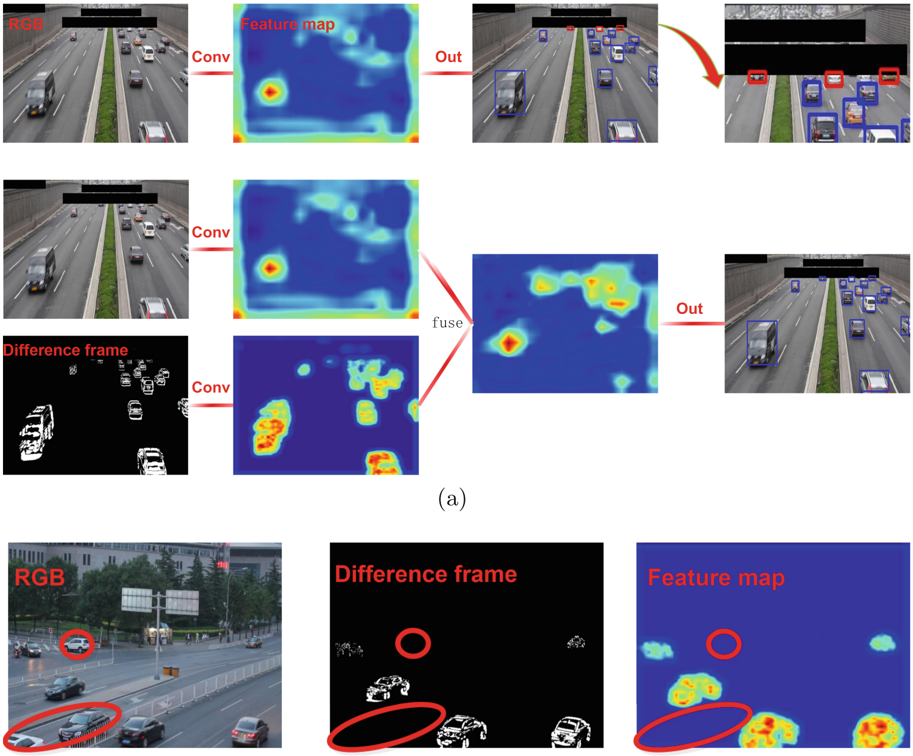
3.1 Dual Stream Architecture

Appearance Stream. Our appearance stream is used to extract the object appearance features, based on Faster R-CNN, which is an advanced method for detection. In order to get a general appearance stream, we use an advanced CNN structure for this stream, such as ResNet50 [6]. It takes RGB frames as input and outputs $H/4 \times W/4$ feature map. It is pre-trained on an object detection dataset, *i.e.*, the COCO dataset, to locate object position.

Motion Stream. It is difficult for our appearance stream to separate tiny objects that are blended with the background. Then our proposed temporal difference frames, as shown in Fig. 2(a), can not only eliminate the background interference, but also enhance the expression of the object in the feature map. After adding the motion stream, the response of object motion feature is significantly enhanced, and the tiny object undetected in original red bounding box can be recovered accurately.

However, our motion stream is invalid for static objects. When an object moves through the scene, motion stream enhances the response of the object position on the feature map. But once it becomes stationary as shown in Fig. 2(b), the motion network can not estimate the object like the appearance stream. Therefore, we leverage this complementary nature to fuse the appearance and motion streams in our pipeline.

For the motion stream, we use Depth-wise separable convolution (DWConv) to reduce computational complexity. Since temporal difference frames is not as complex as RGB frames, shallow Depth-wise separable convolution layers fits them well. This stream decomposes standard convolution into DWConv which can also be called spatial or channel-wise convolution, followed by a 1×1 pointwise convolution layer. Therefore, cross-channel and spatial correlation can be calculated independently, which greatly reduce the number of parameters, and make the structure of the motion network simpler and faster to execute. This method is trained to estimate the location of independently moving objects, based on temporal difference frames calculations from consecutive three frames as input. For the temporal difference method, we set the threshold to 25 and set



(a)

(b)

Fig. 2. (a) Appearance and motion stream feature visualization. It can be clearly seen that, for the feature response of the object position, the effect of the temporal difference frames are much higher than that of the RGB frame. But when we added them together, the objects missed in the red box is restored. (b) Visualization of motion stream features, including stationary objects in red circle. (Color figure online)

the brightness value between 30 and 100 as the background. Then we use some morphology processing (such as corrosion, expansion) to reduce the interference of the motion background. The time cost of obtaining these temporal difference frames is much less than the optical flow picture. We train the motion stream to estimate independently moving objects that produce a $\mathrm{H}/4 \times \mathrm{W}/4$ prediction output, where each value represents the status of the corresponding pixel motion.

3.2 Memory Attention Module

To capture temporal information in the video sequence, we propose a memory attention module which comprised two key components: ConvGRU module and attention module. The ConvGRU module is designed to exploit the temporal correlation in adjacent video frames, and the attention module is to refine the status feature matrix h_t in ConvGRU, as shown in the Fig. 3. Our memory

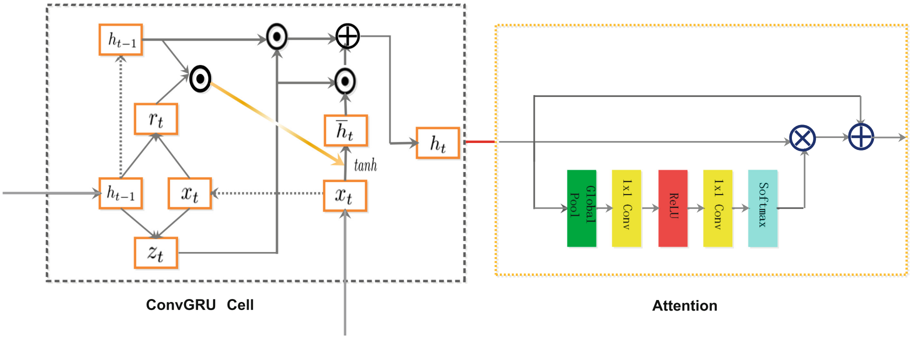


Fig. 3. Memory attention module. Our memory attention module is composed two key components: ConvGRU module and an Attention module.

attention module is computed with convolutional operators and non-linearities as follows.

$$z_t = \sigma(W_{hz} * h_{t-1} + W_{xz} * x_t + b_z) \tag{1}$$

$$r_t = \sigma(W_{hr} * h_{t-1} + W_{xr} * x_t + b_r) \tag{2}$$

$$\overline{h}_t = tanh(W_h * (r_t \odot h_{t-1}) + W_{xr} * x_t + b_r) \tag{3}$$

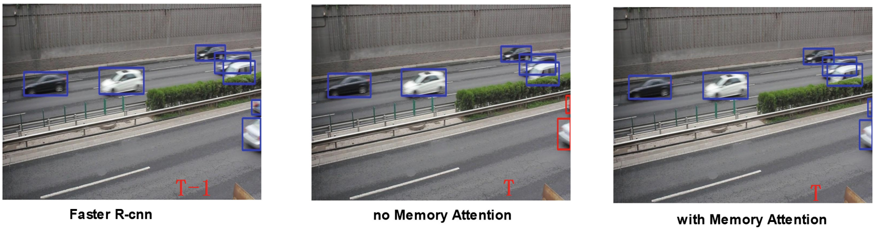
$$h_t = (1 - z_t) \odot h_{t-1} + z \odot \overline{h} \tag{4}$$

$$c_t = Softmax(W_{1*1}(ReLU(W_{1*1} * GAP(h_t)))) \tag{5}$$

$$h_t^{'} = (c_t * h_t + h_t) \tag{6}$$

Firstly, ConvGRU obtains the states of the two control gates (reset gate and update gate) by the last transmitted state h_{t-1} and the input x_t of the current node. As shown in Eq. (1). The state and gate are 3D tensors that characterize the spatiotemporal pattern in the video, effectively remember each object trajectory and their direction. σ stands for the activation function, acting as a gating signal. After getting the gating signal, we use the reset gate to handle the state of the previous frame h_t, and then splice it with the input x_t, and get the implicit state of the current frame through a *tanh* activation function. $\odot$ denotes the multiplication of the corresponding element. The last and most critical step, as we call it the memory update phase, is used to simultaneously forget and remember. According to Eq. (4), we can see that z_t and $1 - z_t$ are interlocked, selectively forgetting or retaining the previous state and the hidden state. Module learning combines the characteristics of the current frame with the video representation of the memory to improve motion predictions, or to fully recover them from previous observations even if the moving objects become stationary.

Our attention module is to refine the state feature h_t in ConvGRU, and to improve the representation of specific context by mining the interrelationships between channels. As shown in Fig. 3, the module is built upon deep CNN features to achieve the feature selection. Detailed steps are shown in Eq. (5). The GAP represents the global average pooling. This descriptor embeds the global distribution of channel-wise feature responses. W is a 1×1 convolution kernel. The *Softmax* value represents the importance of each region in the image feature. Then the output is multiplied by the original features. Finally the extracted features are added to the original features to complete feature enhancement as Eq. (6) shows. The input of attention module is extracted from the state h_t. After the feature selection of the attention module, the refined feature map will continue to be sent to the next cell in the ConvGRU. In this way, we can complement the contents of the current frame with the refined front and rear frame states, which improves the detection ability of blurred, occluded and deformed objects. As shown in Fig. 4. Finally, we feed the augmented features into the RPN and ROI module, and then conduct the bounding box regression and classification for object detection. The details of RPN and ROI module can be referred in [16].



Faster R-cnn

no Memory Attention

with Memory Attention

Fig. 4. Our memory attention module recovers the object in the current frame that was lost due to occlusion and blurring based on the state vector of the previous frame.

4 Experiments

We evaluate our approach on a public dataset DETRAC [20]. We first introduce the dataset and implementation details, followed by a series of ablation experiments. Finally, we present the comparison results with other state-of-the-art methods.

DETRAC: The DETRAC [20] is a large object detection dataset of urban street scene, with 10 h at 24 different locations in Beijing and Tianjin. The frame rate is 25 frames per second with a resolution of 960 × 540. The entire dataset contains 100 videos with 140,000 frames manually labeled with 8,250 vehicles for a total of 1.21 million labeled objects. The training set contains 60 videos, and the rest 40 videos for the test set.

4.1 Implementation Details

In this subsection, we will decompose our approach to verify the contribution of each component. We implement our method based on Pytorch. Our proposed network is based on the ResNet-50 and ResNet-101 pre-trained on ImageNet [17].

Training: We respectively use the Faster R-CNN and YOLO$_{v2}$ as our basic object detection frameworks, most of the parameters are set according to the original publication. SGD training is performed, with 6 image at each mini-batch. 120 K iterations are performed on 4 GPUs, each of which holding two mini-batch. All our experiments are performed on a workstation with Nvidia 1080ti, CUDA 9.0 and cuDNN V7.5.

Table 1. The performance on DETRAC [20] dataset. APP represents appearance stream, MOT represents motion stream, and MA represent memory attention. The Faster R-CNN is based on ResNet-50. The YOLO$_{v2}$ is based on darknet19.

Method	APP	MOT	MA	mAP(%)
Faster R-CNN	✓			71.71
		✓		61.00
	✓	✓		72.83
	✓		✓	72.92
	✓	✓	✓	**73.96**
YOLO$_{v2}$	✓			71.23
		✓		60.16
	✓	✓		72.47
	✓		✓	72.24
	✓	✓	✓	**73.39**

Table 2. Per-class results on DETRAC [20] testing set. MMA net outperforms existing approaches and achieves 74.88% in mAP.

Method	Backbone	mAP (%)	Car	Van	Bus	Others
R-FCN	Res50	71.73	88.42	74.04	90.49	33.97
R-FCN	Res101	73.27	88.63	73.73	90.61	40.11
SSD	Vgg16	70.16	87.29	72.13	87.21	34.02
FSSD	Vgg16	71.75	89.16	73.25	88.46	36.12
YOLO$_{v2}$	darknet19	71.23	89.93	65.84	87.83	41.31
Faster R-CNN	Res50	71.71	88.95	73.08	90.55	34.27
Faster R-CNN	Res101	73.11	88.91	73.23	90.63	39.66
$\mathbf{MMA + YOLO_{v2}}$	darknet19	**73.39**	90.41	69.53	91.23	**42.37**
$\mathbf{MMA + FasterR - CNN}$	Res50	**73.96**	90.25	74.78	93.32	37.48
$\mathbf{MMA + FasterR - CNN}$	Res101	**74.88**	**90.87**	**75.06**	**93.33**	40.26

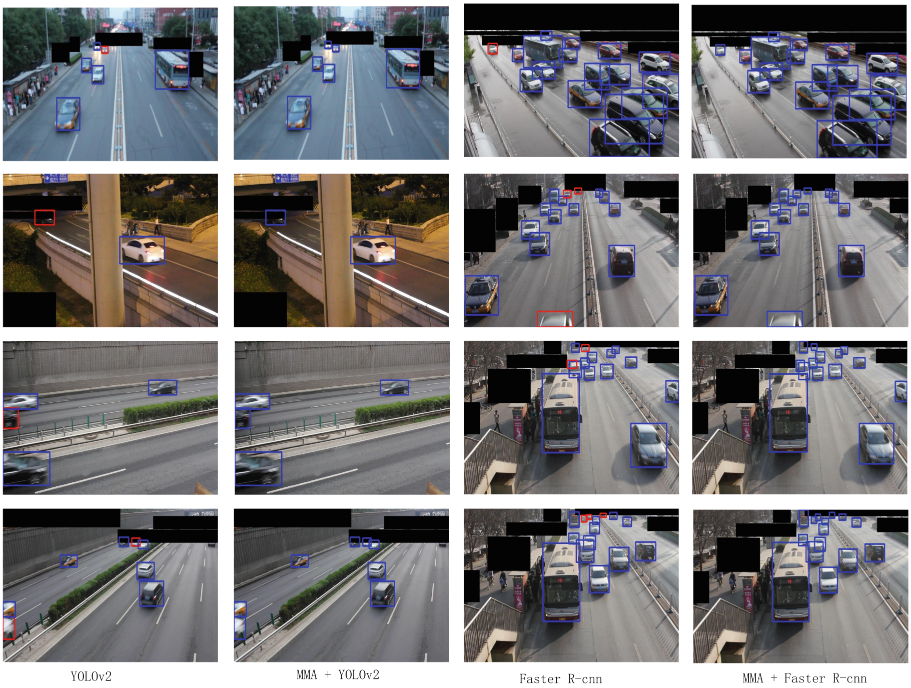


Fig. 5. Detection results of YOLO$_{v2}$, Fater R-cnn and our algorithm. The car in the red box is missed by YOLO$_{v2}$ and FastrRcnn, but detected by our algorithm. (Color figure online)

4.2 Ablation Study

For better understanding MMA net, we investigate the impact of each component in its design. The results are summarized in Table 1.

As show in Table 1, our MMA net improves the performance. Compared with the baseline Faster R-CNN (ResNet-50), our MMA yields to a result of 73.96% in mAP, which brings 2.25% improvement. (1) APP: The appearance stream is the backbone of base detectors like Faster R-CNN. As we can see, our baseline accuracy is 71.71%. (2) MOT: When we add motion steam separately, the accuracy is reduced by a few points compared to the baseline. The reason is that there are many static objects in the dataset. For example, cars stopping at traffic lights are almost equivalent to stationary targets in consecutive frames, which leads to inaccurate temporal difference frame and low precision. Therefore, adding appearance stream can facilitate compensating the ineffective prediction of motion stream, which improves the mAP to 72.83%. (3) MA: Employing memory attention module individually outperforms the baseline by 1.21%. When we integrate these modules together, the performance further achieves to 73.96%. In addition, when we apply a deeper pre-trained network (ResNet-101), our module detection performance is improved to 74.88%. After integrating our components to YOLO$_{v2}$, the performance consistently improves.

4.3 Comparison to the State-of-the-art

Table 2 compares our approach to the state-of-the-art methods on DETRAC [20]. In order to verify the superiority of our approach, we use different backbone networks for verification. The results show that we perform much well than above of method. Figure 5 visualizes some representative results of the Faster R-CNN, YOLO$_{v2}$ baseline and our proposed framework. It is clear that the visualization quality of our method is much better than the baselines.

5 Conclusion

In this paper, we propose a novel module for object detection in video with competitive performance, which introduces a dual stream network with memory attention module. In our network, we make full use of the object motion information and send it into a memory attention module, followed by the refined consecutive frames states for improving detection accuracy. Specifically, the motion stream improves the detection accuracy of the tiny objects, but it can not detect the stationary object, so we merge it with the appearance stream to form a complementary module and memory attention module to recover the lost object due to deformation and blur. Our ablation study shows that our proposed module can achieve competitive results and outperforms other advanced methods. More importantly, our modules can be easily embedded in other object frameworks such as Faster R-CNN and YOLO$_{v2}$, which demonstrates the generality of our method.

Acknowledgements. This work was supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (201900046) and the National Natural Science Foundation of China (61472002).

References

1. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: scale-aware semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint. arXiv:1406.1078 (2014)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems (2016)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)

5. Han, W., et al.: Seq-NMS for video object detection. arXiv preprint. arXiv:1602.08465 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
7. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **6**(02), 107–116 (1998)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
11. Kang, K., et al.: T-CNN: tubelets with convolutional neural networks for object detection from videos. IEEE Trans. Circuits Syst. Video Technol. **28**(10), 2896–2907 (2018)
12. Lee, B., Erdenee, E., Jin, S., Nam, M.Y., Jung, Y.G., Rhee, P.K.: Multi-class multi-object tracking using changing point detection. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 68–83. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_6
13. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
14. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning (2013)
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (2015)
17. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
19. Wang, L., Ouyang, W., Wang, X.: Visual tracking with fully convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
20. Wen, L., et al.: UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. arXiv preprint. arXiv:1511.04136 (2015)
21. Zhu, X., Dai, J., Yuan, L., Wei, Y.: Towards high performance video object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
22. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2017)