Aggregating Motion and Attention for Video Object Detection

Ruyi Zhang(✉), Zhenjiang Miao, Cong Ma, and Shanshan Hao

Beijing Jiaotong University, No. 3 Shangyuancun, Haidian, Beijing 100044, China
{17120331,zjmiao,13112063,17120308}@bjtu.edu.cn

Abstract. Video object detection plays a vital role in a wide variety of computer vision applications. To deal with challenges such as motion blur, varying viewpoints/poses, and occlusions, we need to solve the temporal association across frames. One of the most typical solutions to maintain frame association is exploiting optical flow between consecutive frames. However, using optical flow alone may lead to poor alignment across frames due to the gap between optical flow and high-level features. In this paper, we propose an Attention-Based Temporal Context module (ABTC) for more accurate frame alignments. We first extract two kinds of features for each frame using the ABTC module and a Flow-Guided Temporal Coherence module (FGTC). Then, the features are integrated and fed to the detection network for the final result. The ABTC and FGTC are complementary to each other and can work together to obtain a higher detection quality. Experiments on the ImageNet VID dataset show that the proposed framework performs favorable against the state-of-the-art methods.

Keywords: Video object detection · Optical flow · Self-attention · End-to-end

1 Introduction

Object detection, aiming at locating and classifying particular objects in an image or throughout an entire video sequence, is a fundamental task in computer vision. In recent years, the development of deep neural networks has contributed a lot to the progress of this task. Yet, many existing object detection methods [1–5] are specially designed for images. Directly applying image-level detecting techniques to the video domain usually fails to get satisfactory performance, since frames tend to be deteriorated by issues such as motion blur, rare poses, and occlusions. Beyond the object detection method for images, temporal information in videos can be exploited to improve the detection performance.

So far, existing video object detection methods [6–12, 17] can be roughly divided into two categories. One category depends on manually-designed post-processing rules [6–9, 12]. These methods first detect each frame independently on a still image detector and then apply hand-crafted rules across the time dimension to refine the final detection results. Generally, the association rules are enforced independently of training. Methods of this type are neither end-to-end nor optimal. By contrast, methods such as FGFA [10]

S. Palaiahnakote et al. (Eds.): ACPR 2019, LNCS 12047, pp. 601–610, 2020.
https://doi.org/10.1007/978-3-030-41299-9_47

and MANet [11] learn to establish temporal consistency by multi-frame aggregation in both training and testing stages. Moreover, they can be trained in an end-to-end manner. In these methods, optical flow is applied to capture temporal information to enhance features in the current frame. However, the optical flow only predicts the displacement of pixels between the original images. Directly applying it to high-level feature may lead to inaccurate spatial correspondences.

To alleviate the issue mentioned above, in this work, we propose an Attention-Based Temporal Context module (ABTC) to enhance frame temporal consistency. This module models pixel-level consistency across frames. Specifically, given features F_t and $F_{t+\tau}$ (or $F_{t-\tau}$) of a reference frame and a neighboring frame, ABTC first computes corresponding weights based on the similarity between any two locations across the two frames. Then, it selectively extracts neighboring spatial information based on the temporal context information. Compared to optical flow-based methods, ABTC can obtain relevant information from high-level features of neighboring frames and bridge the gap between the original frames and high-level features. Finally, the output of the ABTC module is integrated with the output of an optical flow module to form a more effective representation for each frame. This representation is then fed to a detector to get the final detection result.

By incorporating rich temporal information, our model can deal with the challenging issues including appearance changes and occlusions. Extensive experiments conducted on the ImageNet VID dataset demonstrate that our method outperforms state-of-the-art methods in detection accuracy.

2 Related Work

2.1 Object Detection for Still Images

It has been over two decades since the academic community studied object detection. Recently, deep convolutional neural networks have achieved great success on the task of video object detection. Existing object detectors mainly fall into two streams, namely, one-stage methods and two-stage methods. One-stage methods such as YOLO [4] and SSD [5] directly utilize features produced from a feature extraction network to predict class labels and the corresponding locations of objects. On the other hand, two-stage methods such as Fast R-CNN [1], Faster R-CNN [2], and R-FCN [3] need to extract proposals in the first stage and then perform fine-grained object classification and regression based on the proposals. These methods are more flexible for integration and extension. Therefore, we take R-FCN as the basic framework and then extends it for video object detection.

2.2 Object Detection for Videos

Video object detection is increasingly popular in the literature since the introduction of the ImageNet VID dataset. Comparing with images, ample temporal information can be employed to assist object detection in videos. Building relationships in both space and time of objects properties across frames is key to accurate video object detection.

Researchers have designed several video object detectors [6–12, 17] that can be divided into two settings, namely, box-level methods [6–9, 12] and feature-level methods [10, 11, 17]. Seq-NMS [6] links boxes if the IOU of two boxes from consecutive frames is higher than a certain threshold. Then, boxes within the sequences constructed before are rescored and re-ranked through a method named “Seq-NMS”. TPN [7] proposes a novel network to generate high-quality tubelet proposals efficiently and exploits LSTM to construct temporal coherence. TCNN [9] utilizes optical flow to propagate bounding boxes from neighbor frames and also adopts a different strategy for tubelet classification and rescoring.

In feature-level methods, FGFA [10] combines the warped features from adjacent frames to enhance the features of the current frame by using optical flow. MANet [11] employs optical flow information for both pixel-level aggregation and instance-level aggregation to incorporate temporal information in an end-to-end manner. However, the optical flow based feature propagation may fail to align the frames in some cases. To alleviate this issue, we introduce a novel module for dense frame matching in feature space.

2.3 Self-attention Mechanism

The self-attention mechanism has become a hot topic in academia and achieved remarkable success in various tasks since the introduction of [13]. The inspiration of attention mechanism comes from how human perception works to recognize objects across videos. Humans focus attention selectively on parts of the visual space to acquire information when and where they needed other than the whole scene. Specifically, a self-attention module computes the response at a position in a sequence (e.g., a sentence) by attending to all positions and taking their weighted mean values of all positions. [14] aligns the source and the target words by applying soft attention to the task of machine learning. The work [18] applies self-attention to the task of scene segmentation. It aims to capture rich context for powerful feature representation. The work [19] introduces a non-local operator based on self-attention mechanism. Experimental results in [19] show that the performance of both video classification and object detection can be improved via the non-local operator. Unlike previous works, we apply self-attention to the task of video object detection and design a module to select temporal context information for better cross-frame alignment.

3 Approach

3.1 Overview

The full architecture of the proposed approach is shown in Fig. 1. Specifically, given a video sequence X = $\{X_1, \ldots, X_i, \ldots\}$, we aim to use the proposed model to generate the detection results S = $\{S_1, \ldots, S_i, \ldots\}$, where S_i is the detection results corresponding to X_i. All frames are fed forward into a convolutional network N_{feat} to extract the intermediate features F = $\{F_i, \ldots, F_i, \ldots\}$. To find an effective representation of current time t, the intermediate features F_t, $F_{t-\tau}$ (or $F_{t+\tau}$) are taken as the input of two modules, i.e., the Flow-Guided Temporal Coherence module (FGTC) as well as the Attention-Base Temporal Context module (ABTC) to capture temporal information. FGTC exploits optical flow information to propagate features and maintain the temporal coherence across frames. Meanwhile, ABTC learns to selectively extract temporal context information based on self-attention mechanism. The combination of two kinds of features obtained from the two modules is taken as the input of the detection network described in MANet [11]. Both of the two modules allow us to better handle the challenging deteriorated frames commonly seen in videos. We will describe the two modules in detail in the following sections.

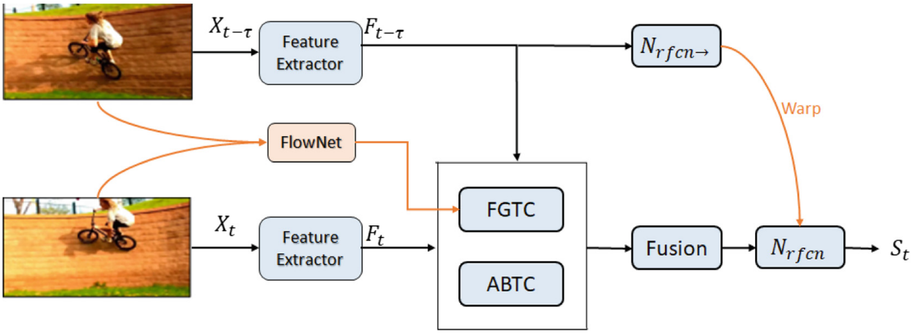

Fig. 1. The full architecture of the proposed approach. Only a neighboring frame $X_{t-\tau}$ and the reference frame X_t are shown for simplicity. Intermediate feature maps $F_{t-\tau}$ and F_t are extracted from a convolutional network and fed to both the FGTC and ABTC modules. Their outputs are then aggregated to obtain the representation of the current frame for the following detection module. The detector is standard but an extra module is added to enhance the features of the region of interests. The output of the Fusion module is fed into the specially designed detector to produce the final results.

3.2 Flow-Guided Temporal Coherence Module

We next explain how the Flow-Guided Temporal Coherence module (FGTC) establishes temporal consistency by using optical flow information. This design is motivated by FGFA [10]. Specifically, at each time step, FGTC takes current frame X_t and a neighbor frame $X_{t-\tau}$ (or $X_{t+\tau}$) as input and computes as follows:

$$F_t = N_{feat}(X_t) \tag{1}$$

$$F_{t-\tau} = N_{feat}(X_{t-\tau}) \tag{2}$$

$$F_{t-\tau\rightarrow t} = W\big(F_{t-\tau}, N_{flow}(X_{t-\tau}, X_t)\big) \tag{3}$$

$$F^e_{t-\tau\rightarrow t}, F^e_t = \varepsilon(F_{t-\tau\rightarrow t}, F_t) \tag{4}$$

$$c_{t-\tau\rightarrow t} = exp\left(\frac{F^e_{t-\tau\rightarrow t}(p)\cdot F^e_t(p)}{\left|F^e_{t-\tau\rightarrow t}(p)\right|\cdot\left|F^e_t(p)\right|}\right) \tag{5}$$

$$\sum\nolimits_{j=i-\tau}^{i+\tau} c_{j\to t} = 1 \tag{6}$$

$$F_i = \sum\nolimits_{j=i-\tau}^{i+\tau} c_{j\to t} F_{j\to t} \tag{7}$$

Here F_t and $F_{t-\tau}$ denote the intermediate features of the reference frame and a neighboring frame extracted from a convolutional network N_{feat}. While $N_{flow}(X_{t-\tau}, X_t)$ indicates a flow field from $X_{t-\tau}$ to X_t estimated by an optical flow network N_{flow}. W(·) is a bilinear warping function exploited on each location of $F_{t-\tau}$. The warped features $F_{t-\tau\to t}$ are computed based on the flow field obtained before as Eq. (3) shows. Next, we embed the two features F_t and $F_{t-\tau\to t}$ with a tiny neural network for similarity measurement. ε(·) in Eq. (4) denotes the embedding function. Then, as Eqs. (5) and (6) shows the cosine similarity metric is applied to measure the relationship between the warped features and the reference features and the measurement result is then normalized and exploited for adaptive feature aggregation. As Eq. (7) shows, F_i is the final flow-guided enhanced feature that incorporates temporal information from time t − τ to time t + τ.

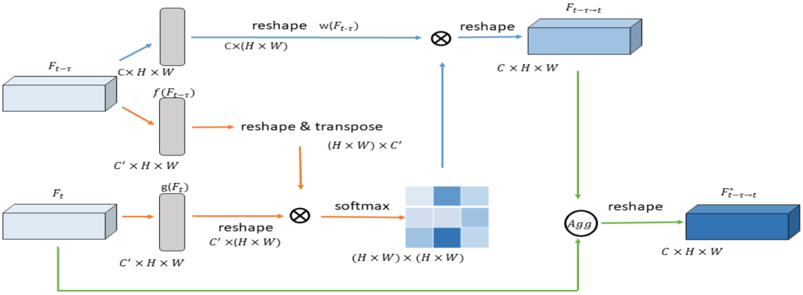

Fig. 2. The detail of Attention-Based Temporal Context module (Color figure online)

3.3 Attention-Based Temporal Context Module

The Flow-Guided Temporal Coherence module propagates temporal information by estimating flow filed between frames. However, only exploiting flow field for feature-level calibration may lead to unsatisfactory spatial correspondence. The reason is that optical flow predicts the displacement of raw pixels and directly using it for the alignment of high-level features may introduce interference. To alleviate this issue, in this section, we introduce a novel ABTC module for feature alignment and enhancement which covers all the space-time pixel locations of feature space. Next, we explain how ABTC incorporates temporal context information using the attention-based temporal context clues.

To capture context information in the time axis, the proposed module compares every space-time locations across frames. Then, the comparison result is utilized to generate a temporal context feature for aligning with the feature of the reference frame. Figure 2 shows the details of the proposed ABTC module. The operation can be summarized into three steps as follows.

The orange lines in the figure represent the first step. This is to generate weights based on feature similarities. Just as Fig. 2 shows, supposing F_t and $F_{t-\tau} \in R^{C\times H\times W}$ are the intermediate features of the reference frame and a neighbor frame, we first embed them into separate convolutional layers to get features with reduced dimensions. Thus we can get $g(F_t) \in R^{C'\times H\times W}$ and $f(F_{t-\tau}) \in R^{C'\times H\times W}$. Then, we perform reshape and transpose operations in turn on $f(F_{t-\tau})$ to $R^{(H\times W)\times C'}$. Meanwhile, we reshape $g(F_t)$ to $R^{C'\times(H\times W)}$, where (H × W) is the number of pixels. The multiplication of the two matrices is the similarity between two feature cells. The above process can be formulated as follows:

$$s_{ij} = [f(F^i_{t-\tau}), g(F^j_t)], \tag{8}$$

where i and j are the locations of the two intermediate feature maps. [·] represents the operation to get similarities s_{ij} between any two cells across features.

Then, we can get normalized correspondence weights by applying a softmax layer:

$$\hat{s}_{ij} = \frac{\exp(s_{ij})}{\sum_{i=1}^{H\times W} s_{ij}}, \tag{9}$$

where $\hat{s}_{ij} \in R^{(H\times W)\times(H\times W)}$ measures the ith position's impact on jth position. This self-attention calculation process simulates the attention mechanism. After that, spatial relations between the two feature maps is established. The resulting $\hat{s}_{ij}$ can be seen as attention maps.

The blue lines indicate the second step. A temporal context feature $F_{t-\tau\to t}$ is produced from step 2. We embed $F^i_{t-\tau}$ to a space that shares the same dimension with $F_{t-\tau}$ and perform reshape on it to get $w(F^i_{t-\tau})$. Equation (9) shows how to get the temporal context feature maps $F_{t-\tau\to t}$. We can infer from the formula that the temporal context feature $F_{t-\tau\to t}$ is a weighted sum of across all positions of the map $w(F^i_{t-\tau})$:

$$F^j_{t-\tau\to t} = \sum_i \hat{s}_{ij} w(F^i_{t-\tau}) \tag{10}$$

$F_{t-\tau\to t}$ is the feature that incorporates temporal context to align with the reference feature F_t.

Then, the procedure of aligning features is represented with green lines as follows:

$$F^*_{t-\tau\to t} = Agg(F_{t-\tau\to t}, F_t) \tag{11}$$

Here, Agg(·) is the function for feature aggregation between $F^*_{t-\tau\to t}$ and F_t. Specifically, the two features are taken as input to a tiny neural network to produce adaptive weights for feature fusion:

$$\text{Agg}(F_{t-\tau\to t}, F_t) = W_{t-\tau\to t} \cdot F_{t-\tau\to t} + W_t \cdot F_t \tag{12}$$

Similarly, the output of this module which absorbs temporal context clues from time t − τ to time t + τ can be formulated as follows:

$$F^*_i = \sum_{j=i-\tau}^{i+\tau} s_{j\to t} F_{j\to t} \tag{13}$$

The outputs of the FGTC and ABTC are summed on element-wise:

$$F^t_{final} = F_i + F^*_i \tag{14}$$

Finally, F^t_{final} is fed into the detection network like detector for the final result.

We name the detector used in the proposed method TR-FCN. R-FCN [3] is a fully convolutional detector. It achieves excellent performance both on speed and accuracy. Based on R-FCN, a tiny neural network is introduced to predict the movement between the proposals among nearby frames to the current frame. We use $N_{rfcn\rightarrow}$ represents the proposals of frame $X_{t-\tau}$. Similar to FGTC described in Sect. 3.2, TR-FCN aligns proposal features by optical flow propagation. The warp operation shows in Fig. 1 indicate the propagation procedure. Such instance-level aggregation further builds temporal information in the detection network, ablative study shows the effectiveness of this detector. Details of the two modules will be further clarified in Sect. 3.4.

3.4 Implementation Details

We take the pre-trained ResNet-101 model [20] as the feature extraction network and make some modifications to it. The specific changes follow the practice of [10]. FlowNet [15] is exploited as the network for optical flow, which is the pioneer of applying deep convolutional networks to optical flow estimation. In order to match the dimension of the intermediate feature, the output of the flow network needs to be downscaled to half.

We use 1 × 1 convolution layers with 256 filters to implement the embedding function f(·) and g(·). For the realization of Agg(·), we use 1 × 1 convolution with 256 filters followed by two 1 × 1 convolution layers with 16 and 2 filters, respectively. The design of the detection network follows MANet [11]. The main difference from a standard detector is that it adds an additional instance-level aggregation module by use of optical flow to make temporal consistency.

4 Experiments

4.1 Dataset and Setup

We evaluate the proposed approach against state-of-the-art video object detectors on the ImageNet VID [13] dataset. ImageNet VID is one of the most popular benchmark datasets for video object detection. It contains 3862 training videos, 555 validation videos and 937 test videos for 30 categories. The annotations for the test set is not released. We report all results on the validation set following the protocols of FGFA [10] and performance is measured in terms of the mean average precision (mAP).

In addition to the ImageNet VID training set, the ImageNet DET training set is also used for training. Note that we only use the 30 categories shared by both of the two datasets. Our model is trained in three stages. We set the frame sampling interval to 15 in the first two stages and 10 in the third stage. The method is implemented with MXNet and trained on a single NVIDIA P40 GPU.

4.2 Ablation Study

First of all, to analyze the effectiveness of various components in the proposed method, we conducted four modifications of our approach in our ablative experiments, the experimental results are shown in Table 1.

Version (a) is the baseline R-FCN with ResNet-101. It obtains a test mAP of 70.9%. For purer analysis, the models listed in Table 2 are all evolved from this strong baseline by applying corresponding temporal module.

Comparing with the baseline R-FCN, version (b) employs FGTC, this module effectively renders temporal information from neighbor frames. It achieves a result of 73.2%, which brings 2.3% improvement.

We then investigate the contribution of ABTC in version (c). When we add it to version (b), we can obtain a 3.5% improvement of test mAP comparing to the single baseline. This module compares every space-time locations across frames in feature space to capture temporal context information.

Table 1. Ablation studies on the ImageNet VID validation set. FGTC represents Flow-Guided Temporal Coherence module, ABTC represents Attention-Based Temporal Context module, and TR-FCN refers to the standard R-FCN with temporal information rendered by instance-level aggregation.

Feature extractor	ResNet-101				
Versions	(a)	(b)	(c)	(d)	(e)
FGTC		√	√	√	√
ABTC			√		√
TR-FCN				√	√
mAP (%)	70.9	73.2	74.1	76.2	77.8

Version (d) exploits TR-FCN instead of R-FCN as the detector. The main difference is that TR-FCN incorporates temporal information by use of optical flow. Specifically, it predicts movements of proposals among nearby frames and aligns them with the proposals obtained from the reference frame.

Version (e) is the proposed method. Comparing to version (d), it improves mAP by 1.6%, indicating that these components are complementary and they can work together to obtain a higher detection quality.

To sum up, the two features produced from both modules can represent useful spatial-temporal information from neighbor frames, and the combination of them is quite necessary for the detection performance. With all the above modules, the overall mAP is improved from 70.9% to 77.8%.

4.3 Detection Results

We compare the proposed method against several existing state-of–the-art methods on the task of video object detection, including Seq-NMS [6], TPN [7], TCN [8], TCNN

[9], FGFA [10], MANet [11] and D&T [12]. The results of Seq-NMS [6], TPN [7], TCN [8], TCNN [9] and D&T [12] are obtained from the original papers. The remaining methods are implemented using the code provided by the authors on a platform with NVIDIA P40 GPU.

As we can see from Table 2, compared with the existing methods for video object detection, the proposed method achieves the best performance. It surpasses the R-FCN based detector by a large margin of ~7 points, proving the effectiveness of our model. Comparing with box-level methods [6–9, 12], feature-level methods [10, 11, 17] exploit temporal information during both training and testing and can be trained end-to-end. Therefore, they usually perform better on accuracy. Comparing with FGFA [10] and MANet [11] which establish temporal information by exploiting optical flow, our method outperforms these methods via the adoption of multi-module collaboration strategy, which greatly improves the results.

Table 2. Quantitative results of our proposed method, comparing with state-of-the-art solutions on ImageNet VID validation set.

Methods	mAP (%)	Backbone
Seq-NMS [6]	52.2	VGGNet
TCN [8]	47.5	GoogLeNet
TPN [7]	68.4	GoogLeNet
R-FCN [3]	70.9	ResNet101
TCNN [9]	73.8	GoogLeNet
DFF [17]	69.9	ResNet101
D(&T loss) [12]	75.8	ResNet101
FGFA [10]	73.2	ResNet101
MANet [11]	76.2	ResNet101
Ours	**77.8**	ResNet101

5 Conclusions

In this paper, we present a unified, end-to-end trainable spatiotemporal CNN model for video object detection. The key components are two modules FGTC and ABTC that extracts two kinds of features for each frame respectively. Specifically, FGTC adaptively propagates features over time via optical flow. To align the frame features more precisely, we propose the ABTC module which aims to render the temporal context for spatial correspondence between features across frames. The two features are combined for the benefit of their complementarity. Experimental results show that the proposed framework achieves 77.8% mAP on ImageNet VID, which outperforms existing state-of-the-art methods. The ablative results show ABTC is complementary to flow-based feature propagation modules, demonstrating the generalization ability of our method.

Acknowledgements. This work is supported by the NSFC 61672089, 61703436, 61572064, 61273274 and CELFA.

References

1. Girshick, R.: Fast R-CNN. In: ICCV (2015)
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: NIPS (2016)
4. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR (2016)
5. 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
6. Han, W., et al.: Seq-NMS for video object detection. arXiv preprint arXiv:1602.08465 (2016)
7. Kang, K., et al.: Object detection in videos with tubelet proposal networks. In: CVPR (2017)
8. Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: CVPR (2016)
9. Kang, K., et al.: T-CNN: tubelets with convolutional neural networks for object detection from videos. In: T-CSVT (2017)
10. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: ICCV (2017)
11. Wang, S., Zhou, Y., Yan, J., Deng, Z.: Fully motion-aware network for video object detection. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 557–573. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_33
12. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: ICCV (2017)
13. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
14. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate arXiv preprint arXiv:1409.0473 (2014)
15. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: ICCV (2015)
16. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. IJCV **115**, 211–252 (2015)
17. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)
18. Fu, J., Liu, J., Tian, H., et al.: Dual Attention Network for Scene Segmentation. arXiv preprint arXiv:1809.02983v4 (2018)
19. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. arXiv preprint arXiv:1711.07971v3 (2018)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)