Flow-Guided Feature Aggregation for Video Object Detection

Hongzhi Liu

August 10, 2018

Abstract

Recent years have witnessed significant progress in object detection. Extending state-of-the-art object detectors from image to video is challenging. Existing work attempts to exploit temporal information on box level, but such methods are not trained end-to-end. During preparation for URPC2018, I read a thesis written by Xizhou Zhu who is from University of Science and Technology of China. This paper presents flow-guided feature aggregation, an accurate and end-to-end learning framework for video object detection. It leverages temporal coherence on feature level instead and improves the per-frame features by aggregation of nearby features along the motion paths, and thus improves the video recognition accuracy. This method significantly improves upon strong single frame baselines in ImageNet VID, especially for more challenging fast moving objects.

1. Overview of FGFA

State-of-the-art methods of object detection share a similar two-stage structure. Deep Convolutional Neural Networks(CNNs) [5, 8] are firstly applied to generate a set of feature maps over the whole input image. A shallow detection-specific network [3, 2, 6] then generates the detection results from the feature maps. These methods achieve excellent results in still images. However, directly applying them for video object detection is challenging.

The recognition accuracy suffers from deteriorated object appearances in videos that are seldom observed in still images, such as motion blur, video defocus, rare poses and *etc.* as shown in Fig. 1 and Fig. 2. Nevertheless, the video has rich information about the same object instance, which is exploited in existing video object detection methods [4] in a simple way usually observed in multiple "nap-shots" in a short time.

Xizhou Zhu and his team attempt to take a deeper look at video object detection. They seek to improve the detection or recognition quality by exploiting temporal information, in a principled way. In this paper, Zhu and his team propose flow-guided feature aggregation (FGFA). As illustrated in

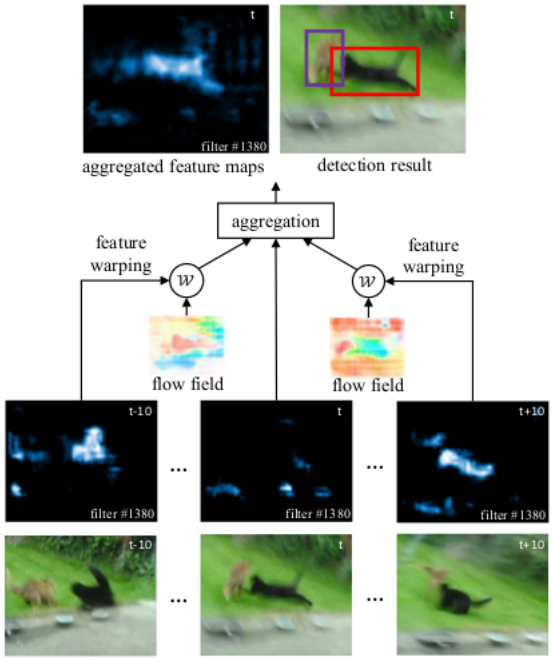

Figure 1. Illustration of FGFA. For each input frame, a feature map sensitive to "cat" is visualized. The feature activations are low at the reference frame t, resulting in detection failure in the reference frame. The nearby frames $t-10$ and $t+10$ have high activations. After FGFA, the feature map at the reference frame is improved and detection on it succeeds.

Fig. 1, the feature extraction network is applied on individual frames to produce the per-frame feature maps. To enhance the features at a reference frame, an optical flow network [1] estimates the motions between the nearby frames and the reference frame. The feature maps from nearby frames are warped to the reference frame according to the flow motion. The warped features maps, as well as its own feature maps on the reference frame, are aggregated according to an adaptive weighting network.

Zhu's approach is evaluated on the large-scale ImageNet VID dataset [7]. Rigorous ablation study verifies that it is effective and significantly improves upon strong single-

Figure 2. Typical deteriorated object appearance in videos.

frame baselines. Combination with box-level methods produces further improvement. Furthermore, they report object detection accuracy on par with the best engineered systems winning the ImageNet VID challenges, without additional bells-and-whistles.

2. Flow Guided Feature Aggregation

In this paper, Xizhou Zhu and his team focus on another aspect of associating and assembling the rich appearance information in consecutive frames to improve the feature representation, and then the video recognition accuracy.

2.1. A Baseline and Motivation

Given the input video frames $I_i, i = 1, \ldots, \infty$, Zhu aims to output object bounding boxes on all the frames, y_i, $i = 1, \ldots, \infty$. A baseline approach is to apply an off-the-shelf object detector to each frame individually.

Two modules are necessary for such feature propagation and enhancement: 1) motion-guided spatial warping. It estimates the motion between frames and warps the feature maps accordingly. 2) feature aggregation module. It figures out how to properly fuse the features from multiple frames. Together with the feature extraction and detection networks, these are the building blocks of Zhu's approach.

2.2. Model Design

Flow-guided warping. As motivated by [9], given a reference frame I_i and a neighbor frame I_j, a flow field $M_{i\to j} = F(I_i, I_j)$ is estimated by a flow network F [1]. The feature maps on the neighbor frame are warped to the reference frame according to the flow. The warping function is defined as Eq. 1.

$$f_{j\to i} = W(f_j, M_{i\to j}) = W(f_j, F(I_i, I, j)) \qquad (1)$$

where W(·) is the bilinear warping function applied on all the locations for each channel in the feature maps, and $f_{j\to i}$ denotes the feature maps warped from frame j to frame i.

Feature aggregation. After feature warping, the reference frame accumulates multiple feature maps from nearby frames. These feature maps provide diverse information of the object instances. For aggregation, the team employ different weights at different spatial locations and let all feature channels share the same spatial weight. The 2D weight maps for warped features $f_{j\to i}$ are denoted as $w_{j\to i}$. The aggregated features at the reference frame $\bar{f}_i$ is then obtained as Eq. 2.

$$\bar{f}_i = \sum_{j=i-K}^{i+K} w_{j\to i} f_{j\to i}, \qquad (2)$$

where K specifies the range of the neighbor frames for aggregation. The equation is similar to the formula of attention models, where varying weights are assigned to the features in the memory buffer. The aggregated features $\bar{f}_i$ are then fed into the detection sub-network to obtain the results as Eq. 3.

$$y_i = \mathcal{N}_{det}(\bar{f}_i). \qquad (3)$$

Compared to the baseline and previous box level methods, their method aggregates information from multiple frames before producing the final detection results.

Adaptive weight. The adaptive weight indicates the importance of all buffer frames $[I_{iK}, \ldots, I_{i+K}]$ to the reference frame I_i at each spatial location. Specifically, at location p, if the warped features $f_{j\to i}(p)$ is close to the features $f_i(p)$, it is assigned to a larger weight. Otherwise, a smaller weight is assigned. Here, Xizhou Zhu uses the cosine similarity metric to measure the similarity between the warped features and the features extracted from the reference frame. Moreover, the team do not directly use the convolutional features obtained from $\mathcal{N}_{feat}(I)$. Instead, they apply a tiny fully convolutional network $\epsilon()$ to features f_i and $f_{j\to i}$, which projects the features to a new embedding

Table 1. Results of using different number of frames in training and inference, using ResNet-50. Default parameters are indicated by.

#training frames	2							5						
# testing frames	1	5	9	13	17	21	25	1	5	9	13	17	21	25
mAP (%)	70.6	72.3	72.8	73.4	73.7	74.0	74.1	70.6	72.4	72.9	73.3	73.6	74.1	74.1
runtime(ms)	203	330	406	488	571	647	726	203	330	406	488	571	647	726

for similarity measure and is dubbed as the embedding subnetwork. They estimate the weight as Eq. 4.

$$w_{j\to i}(p) = \exp(\frac{f^e_{j\to i}(p) \cdot f^e_i(p)}{|f^e_{j\to i}(p)||f^e_i(p)|}). \quad (4)$$

where $f^e = \epsilon(f)$ denotes embedding features for similarity measurement, and the weight $w_{j\to i}$ is normalized for every spatial location p over the nearby frames, $\sum_{j=i-K}^{i+K} w_{j\to i}(p) = 1$. The estimation of weight could be viewed as the process that the cosine similarity between embedding features passes through the SoftMax operation.

3. Experiments

Due to memory issues, Zhu's team use the lightweight ResNet-50 in this experiment. They tried 2 and 5 frames in each mini-batch during SGD training, and 1, 5, 9, 13, 17, 21, and 25 frames in inference. Results in Tab. 1 show that training with 2 and 5 frames achieves very close accuracy. This verifies the effectiveness of their temporal dropout training strategy. In inference, as expected, the accuracy improves as more frames are used. The improvement saturates at 21 frames.

References

[1] A. Dosovitskiy, P. Fischery, and E. Ilg. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1, 2

[2] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1

[4] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016. 1

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 2015. 1

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 1

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 1

[9] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 2