Flow-Guided Feature Aggregation for Video Object Detection

Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, Yichen Wei

Motivation

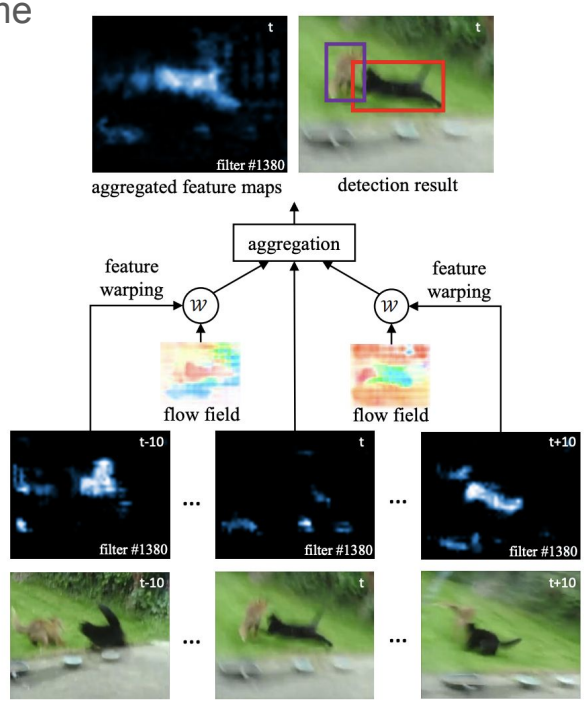
- Video object detection suffers from degenerated object appearances (e.g., motion blur, video defocus, rare poses).
  - Temporal information from neighboring frames can help.

Motivation

- Video object detection suffers from degenerated object appearances (e.g., motion blur, video defocus, rare poses).
  - Temporal information from neighboring frames can help.
- Previous work: first apply object detectors in single frames, then in the post processing step, assemble the detected bounding boxes across temporal dimension.
  - Not end-to-end training
  - Rely on heuristic post-processing

Method

- Improve the per-frame feature learning by temporal aggregation
  - E.g. Propagate features from nearby frames to reference frame
  - Simply averaging the feature will deteriorate the performance
- Two modules:
  - Motion-guided spatial warping
  - Feature aggregation module



Method -- Motion-guided spatial warping

- Estimate flow field between reference frame i and neighbor frame j with a flow network. $$\mathbf{M}_{i \to j} = \mathcal{F}(I_i, I_j)$$
- Warp the feature map on the neighbor frame according to the flow:

$$f_{j \to i} = \mathcal{W}(f_j, \mathbf{M}_{i \to j}) = \mathcal{W}(f_j, \mathcal{F}(I_i, I_j)),$$

- W: bilinear warping function

Method - Feature aggregation

- The reference frame accumulates multiple feature maps from nearby frames.

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

- Adaptive weight: indicate the importance of nearby frame to the reference frame at each spatial location p.

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

- a convolutional network is applied on feature $f_{j\to i}(p)$ to get $f^e_{j\to i}(p)$

Inference Time

Algorithm 1 Inference algorithm of flow guided feature aggregation for video object detection.

input: video frames $\{I_i\}$, aggregation range K
for $k = 1$ **to** $K + 1$ **do** ▷ initialize feature buffer
  $f_k = \mathcal{N}_{\text{feat}}(I_k)$
end for
for $i = 1$ **to** ∞ **do** ▷ reference frame
  for $j = \max(1, i - K)$ **to** $i + K$ **do** ▷ nearby frames
    $f_{j\to i} = \mathcal{W}(f_j, \mathcal{F}(I_i, I_j))$ ▷ flow-guided warp
    $f^e_{j\to i}, f^e_i = \mathcal{E}(f_{j\to i}, f_i)$ ▷ compute embedding features
    $w_{j\to i} = \exp(\frac{f^e_{j\to i} \cdot f^e_i}{|f^e_{j\to i}||f^e_i|})$ ▷ compute aggregation weight
  end for
  $\bar{f}_i = \sum_{j=i-K}^{i+K} w_{j\to i} f_{j\to i}$ ▷ aggregate features
  $y_i = \mathcal{N}_{\text{det}}(\bar{f}_i)$ ▷ detect on the reference frame
  $f_{i+K+1} = \mathcal{N}_{\text{feat}}(I_{i+K+1})$ ▷ update feature buffer
end for
output: detection results $\{y_i\}$

$$r = 1 + \frac{(2K+1) \cdot (\mathcal{O}(\mathcal{F}) + \mathcal{O}(\mathcal{E}) + \mathcal{O}(\mathcal{W}))}{\mathcal{O}(\mathcal{N}_{\text{feat}}) + \mathcal{O}(\mathcal{N}_{\text{det}})},$$

$$1 + \frac{(2K+1) \cdot \mathcal{O}(\mathcal{F})}{\mathcal{O}(\mathcal{N}_{\text{feat}})}$$

Training

- Temporal dropout: use a large K in inference but a small K in training
- The neighbor K frames are randomly sampled from a large range that is equal to the one during inference.

# 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

Implementation Details

- Flow network: FlowNet (pretrained on Flying Chairs dataset)
- Feature network: ResNet-50, ResNet-101, Aligned-Inception-ResNet (pretrained on ImageNet classification)
- Detection network: R-FCN for object detection and RPN for region proposal
- Dataset: ImageNet VID dataset
- Two-phase training: first train on ImageNet DET and then train on ImageNet VID.

Results

instance size	small	middle	large
mAP (%)	24.2	49.5	83.2
mAP (%) (slow)	36.7	56.4	86.9
mAP (%) (medium)	32.4	51.4	80.9
mAP (%) (fast)	24.9	43.7	67.5

Detecting fast moving objects is very challenging, irrespective to how large the object is.

Results

$\mathcal{N}_{\text{feat}}$	ResNet-50					ResNet-101				
methods	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
multi-frame feature aggregation?		✓	✓	✓	✓		✓	✓	✓	✓
adaptive weights?			✓	✓	✓			✓	✓	✓
flow-guided?				✓	✓				✓	✓
end-to-end training?		✓	✓	✓			✓	✓	✓	
mAP (%)	70.6	$69.6_{\downarrow 1.0}$	$71.8_{\uparrow 1.2}$	$\mathbf{74.0}_{\uparrow \mathbf{3.4}}$	$72.1_{\uparrow 1.5}$	73.4	$72.0_{\downarrow 1.4}$	$74.3_{\uparrow 0.9}$	$\mathbf{76.3}_{\uparrow \mathbf{2.9}}$	$74.5_{\uparrow 1.1}$
mAP (%) (slow)	79.3	$81.4_{\uparrow 2.1}$	$81.5_{\uparrow 2.2}$	$\mathbf{82.4}_{\uparrow \mathbf{3.1}}$	$81.3_{\uparrow 2.0}$	82.4	$82.3_{\downarrow 0.1}$	$82.2_{\downarrow 0.2}$	$\mathbf{83.5}_{\uparrow \mathbf{1.2}}$	$82.5_{\uparrow 0.1}$
mAP (%) (medium)	68.6	$71.4_{\uparrow 2.8}$	$71.4_{\uparrow 2.8}$	$\mathbf{72.6}_{\uparrow \mathbf{4.0}}$	$71.5_{\uparrow 2.9}$	71.6	$74.5_{\uparrow 2.9}$	$74.6_{\uparrow 3.0}$	$\mathbf{75.8}_{\uparrow \mathbf{4.2}}$	$74.6_{\uparrow 3.0}$
mAP (%) (fast)	50.1	$42.5_{\downarrow 7.6}$	$50.4_{\uparrow 0.3}$	$\mathbf{55.0}_{\uparrow \mathbf{4.9}}$	$51.2_{\uparrow 1.1}$	51.4	$44.6_{\downarrow 6.8}$	$52.3_{\uparrow 0.9}$	$\mathbf{57.6}_{\uparrow \mathbf{6.2}}$	$53.2_{\uparrow 1.8}$
runtime (ms)	203	204	220	647	647	288	288	305	733	733

Simply aggregating the feature from nearby frames will harm the performance.

Results

$\mathcal{N}_{\text{feat}}$	ResNet-50					ResNet-101				
methods	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
multi-frame feature aggregation?		✓	✓	✓	✓		✓	✓	✓	✓
adaptive weights?			✓	✓	✓			✓	✓	✓
flow-guided?				✓	✓				✓	✓
end-to-end training?		✓	✓	✓			✓	✓	✓	
mAP (%)	70.6	$69.6_{\downarrow 1.0}$	$71.8_{\uparrow 1.2}$	$\mathbf{74.0}_{\uparrow \mathbf{3.4}}$	$72.1_{\uparrow 1.5}$	73.4	$72.0_{\downarrow 1.4}$	$74.3_{\uparrow 0.9}$	$\mathbf{76.3}_{\uparrow \mathbf{2.9}}$	$74.5_{\uparrow 1.1}$
mAP (%) (slow)	79.3	$81.4_{\uparrow 2.1}$	$81.5_{\uparrow 2.2}$	$\mathbf{82.4}_{\uparrow \mathbf{3.1}}$	$81.3_{\uparrow 2.0}$	82.4	$82.3_{\downarrow 0.1}$	$82.2_{\downarrow 0.2}$	$\mathbf{83.5}_{\uparrow \mathbf{1.2}}$	$82.5_{\uparrow 0.1}$
mAP (%) (medium)	68.6	$71.4_{\uparrow 2.8}$	$71.4_{\uparrow 2.8}$	$\mathbf{72.6}_{\uparrow \mathbf{4.0}}$	$71.5_{\uparrow 2.9}$	71.6	$74.5_{\uparrow 2.9}$	$74.6_{\uparrow 3.0}$	$\mathbf{75.8}_{\uparrow \mathbf{4.2}}$	$74.6_{\uparrow 3.0}$
mAP (%) (fast)	50.1	$42.5_{\downarrow 7.6}$	$50.4_{\uparrow 0.3}$	$\mathbf{55.0}_{\uparrow \mathbf{4.9}}$	$51.2_{\uparrow 1.1}$	51.4	$44.6_{\downarrow 6.8}$	$52.3_{\uparrow 0.9}$	$\mathbf{57.6}_{\uparrow \mathbf{6.2}}$	$53.2_{\uparrow 1.8}$
runtime (ms)	203	204	220	647	647	288	288	305	733	733

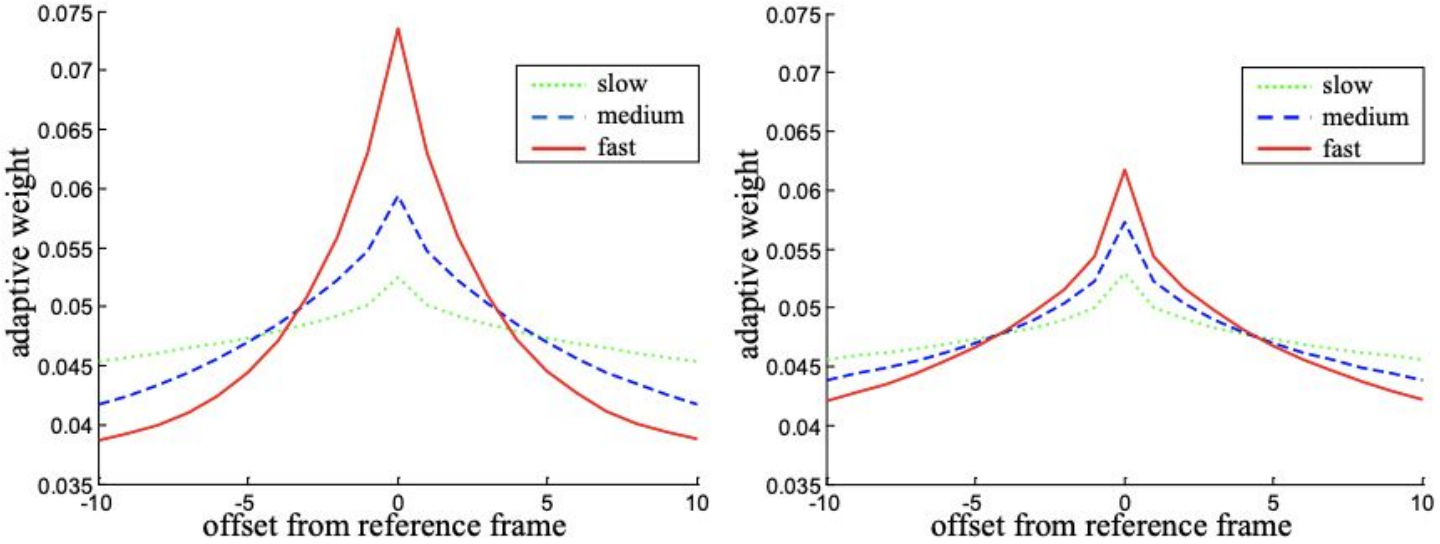
Adding adaptive weight is little help for slow and medium moving instances compared with simply averaging the features, but is important for fast moving instances.

Results

$\mathcal{N}_{\text{feat}}$	ResNet-50					ResNet-101				
methods	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
multi-frame feature aggregation?		✓	✓	✓	✓		✓	✓	✓	✓
adaptive weights?			✓	✓	✓			✓	✓	✓
flow-guided?				✓	✓				✓	✓
end-to-end training?		✓	✓	✓			✓	✓	✓	
mAP (%)	70.6	$69.6_{\downarrow 1.0}$	$71.8_{\uparrow 1.2}$	$\mathbf{74.0}_{\uparrow \mathbf{3.4}}$	$72.1_{\uparrow 1.5}$	73.4	$72.0_{\downarrow 1.4}$	$74.3_{\uparrow 0.9}$	$\mathbf{76.3}_{\uparrow \mathbf{2.9}}$	$74.5_{\uparrow 1.1}$
mAP (%) (slow)	79.3	$81.4_{\uparrow 2.1}$	$81.5_{\uparrow 2.2}$	$\mathbf{82.4}_{\uparrow \mathbf{3.1}}$	$81.3_{\uparrow 2.0}$	82.4	$82.3_{\downarrow 0.1}$	$82.2_{\downarrow 0.2}$	$\mathbf{83.5}_{\uparrow \mathbf{1.2}}$	$82.5_{\uparrow 0.1}$
mAP (%) (medium)	68.6	$71.4_{\uparrow 2.8}$	$71.4_{\uparrow 2.8}$	$\mathbf{72.6}_{\uparrow \mathbf{4.0}}$	$71.5_{\uparrow 2.9}$	71.6	$74.5_{\uparrow 2.9}$	$74.6_{\uparrow 3.0}$	$\mathbf{75.8}_{\uparrow \mathbf{4.2}}$	$74.6_{\uparrow 3.0}$
mAP (%) (fast)	50.1	$42.5_{\downarrow 7.6}$	$50.4_{\uparrow 0.3}$	$\mathbf{55.0}_{\uparrow \mathbf{4.9}}$	$51.2_{\uparrow 1.1}$	51.4	$44.6_{\downarrow 6.8}$	$52.3_{\uparrow 0.9}$	$\mathbf{57.6}_{\uparrow \mathbf{6.2}}$	$53.2_{\uparrow 1.8}$
runtime (ms)	203	204	220	647	647	288	288	305	733	733

Flow-guided feature aggregation show relatively large improvement for detecting slow, medium and fast moving instances.

Results



Flow-guided feature aggregation effectively promotes the information from nearby frames.

Results

$\mathcal{N}_{\text{feat}}$	ResNet-50					ResNet-101				
methods	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
multi-frame feature aggregation?		✓	✓	✓	✓		✓	✓	✓	✓
adaptive weights?			✓	✓	✓			✓	✓	✓
flow-guided?				✓	✓				✓	✓
end-to-end training?		✓	✓	✓			✓	✓	✓	
mAP (%)	70.6	$69.6_{\downarrow 1.0}$	$71.8_{\uparrow 1.2}$	$\mathbf{74.0}_{\uparrow \mathbf{3.4}}$	$72.1_{\uparrow 1.5}$	73.4	$72.0_{\downarrow 1.4}$	$74.3_{\uparrow 0.9}$	$\mathbf{76.3}_{\uparrow \mathbf{2.9}}$	$74.5_{\uparrow 1.1}$
mAP (%) (slow)	79.3	$81.4_{\uparrow 2.1}$	$81.5_{\uparrow 2.2}$	$\mathbf{82.4}_{\uparrow \mathbf{3.1}}$	$81.3_{\uparrow 2.0}$	82.4	$82.3_{\downarrow 0.1}$	$82.2_{\downarrow 0.2}$	$\mathbf{83.5}_{\uparrow \mathbf{1.2}}$	$82.5_{\uparrow 0.1}$
mAP (%) (medium)	68.6	$71.4_{\uparrow 2.8}$	$71.4_{\uparrow 2.8}$	$\mathbf{72.6}_{\uparrow \mathbf{4.0}}$	$71.5_{\uparrow 2.9}$	71.6	$74.5_{\uparrow 2.9}$	$74.6_{\uparrow 3.0}$	$\mathbf{75.8}_{\uparrow \mathbf{4.2}}$	$74.6_{\uparrow 3.0}$
mAP (%) (fast)	50.1	$42.5_{\downarrow 7.6}$	$50.4_{\uparrow 0.3}$	$\mathbf{55.0}_{\uparrow \mathbf{4.9}}$	$51.2_{\uparrow 1.1}$	51.4	$44.6_{\downarrow 6.8}$	$52.3_{\uparrow 0.9}$	$\mathbf{57.6}_{\uparrow \mathbf{6.2}}$	$53.2_{\uparrow 1.8}$
runtime (ms)	203	204	220	647	647	288	288	305	733	733

Based on (a), only train the embedding sub-network (the network to convert features to compute adaptive weight). The large decrease in performance shows that end-to-end training is important.

Results

method	feature network	mAP (%)	runtime (ms)
single-frame baseline	ResNet-101	73.4	288
+ MGP		74.1	574*
+ Tubelet rescoring		75.1	1662
+ Seq-NMS		76.8	433*
FGFA	ResNet-101	76.3	733
+ MGP		75.5	1019*
+ Tubelet rescoring		76.6	1891
+ Seq-NMS		78.4	873*
FGFA	Aligned-Inception-ResNet	77.8	819
+ Seq-NMS		**80.1**	954*

All three post-processing methods can improve the baseline performance, but only Seq-NMS shows significant improvement with proposed FGFA method.

Using Aligned-Inception ResNet as feature network, and use Seq-NMS as post-processing, the proposed method could reach near SOTA performance (81.2).


baseline
fox: 0.98
fox: 0.99
fox: 0.97
lion: 0.32
dog: 0.22
antelope: 0.65
FGFA
fox: 0.99
fox: 0.95
fox: 0.94
fox: 0.80
fox: 0.50
baseline
horse: 0.74
dog: 0.81
dog: 0.65
horse: 0.99
bicycle: 0.66
horse: 0.99
FGFA
horse: 0.84
horse: 0.62
horse: 0.84
horse: 0.99
horse: 0.99
baseline
horse: 0.22
horse: 0.32
horse: 0.89
car: 0.25
FGFA
horse: 0.87
horse: 0.90
horse: 0.28
horse: 0.16
horse: 0.38

Contributions

- Achieves competitive results without techniques like ensemble.
- Proposes an end-to-end framework that incorporates temporal information for video object detection
- Extensive ablation studies demonstrating the effectiveness of their approach, and also demonstrating that the instance moving speed makes video object detection more challenging

Questions

Questions

- What do you think of the proposed motion-guided spatial warping and feature aggregation module?
- Are there any other ways to incorporate temporal information (e.g., optical flow features) or utilize nearby frames?