Flow-Based Video Recognition

Jifeng Dai

Visual Computing Group, Microsoft Research Asia

Joint work with Xizhou Zhu*, Yuwen Xiong*, Yujie Wang*, Lu Yuan and Yichen Wei (* interns)

Talk pipeline

- Introduction
- Deep Feature Flow for Video Recognition
- Flow-Guided Feature Aggregation for Video Object Detection
- Summary

From image to video

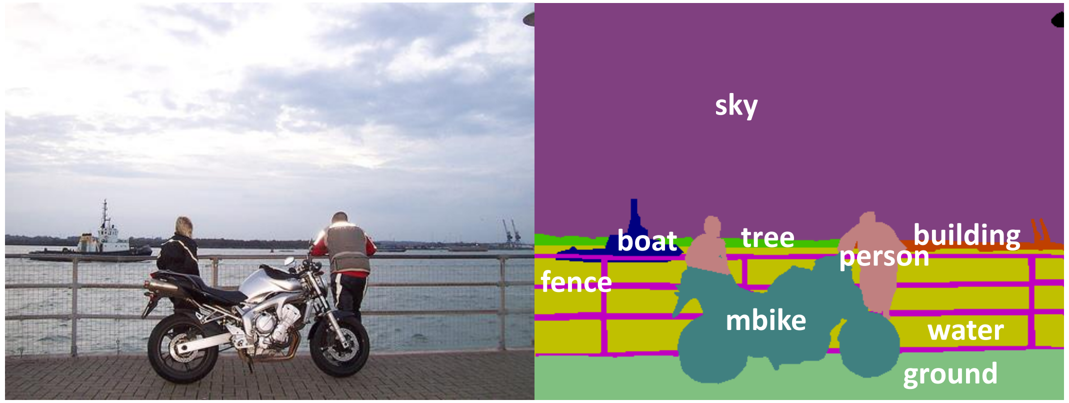


image semantic segmentation

video semantic segmentation

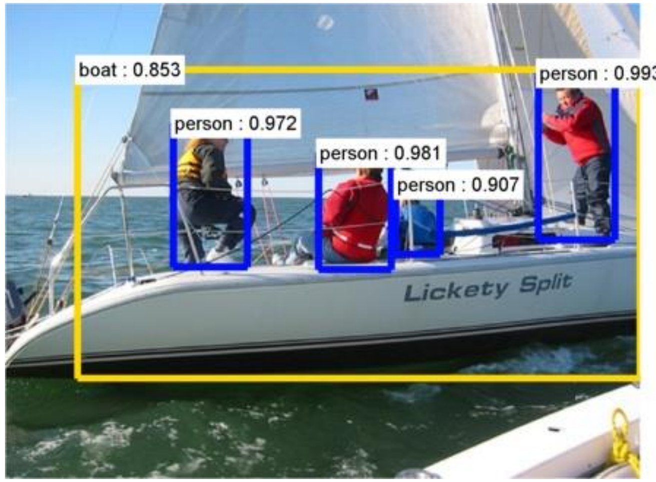
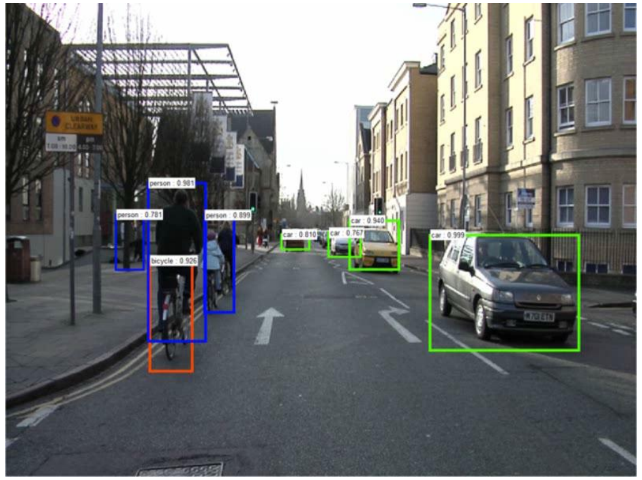


image object detection



video object detection

Per-frame recognition in video is problematic

High Computational Cost

Infeasible for practical needs

Task	Image Size	ResNet-50	ResNet-101
Detection	1000x600	6.27 fps	4.05 fps
Segmentation	2048x1024	2.24 fps	1.52 fps

FPS: frames per second
(NVIDIA K40 and Intel Core i7-4790)

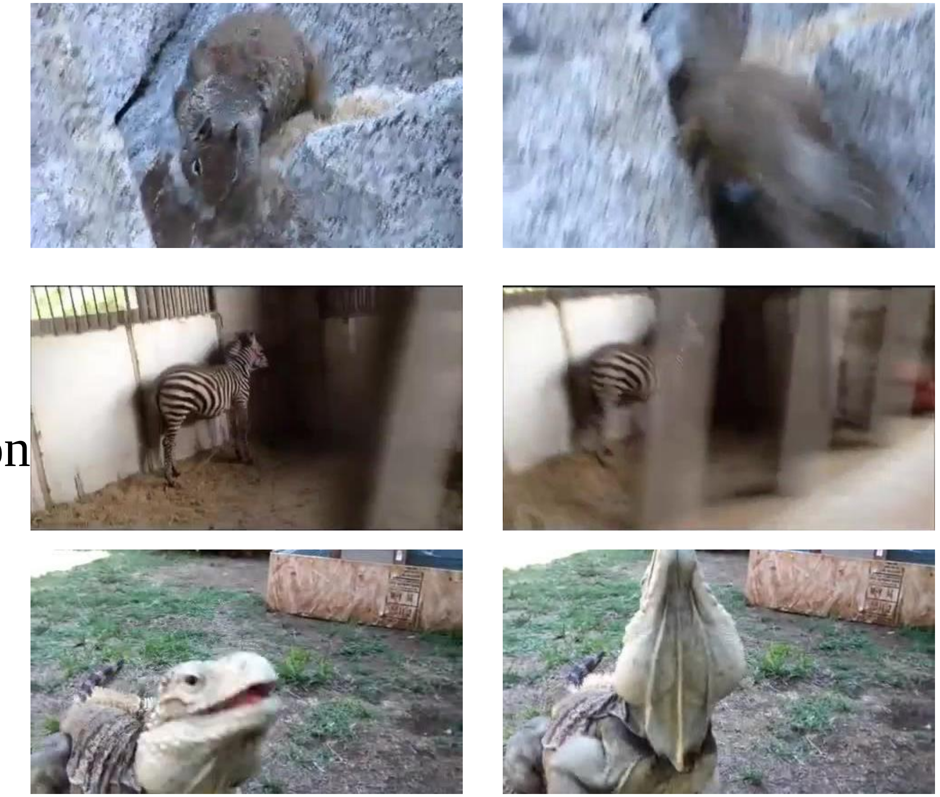
Deteriorated Frame Appearance

Poor feature and recognition accuracy

motion blur

part occlusion

rare poses



Exploit frame motion to do better

- Feature propagation for **speed up** (CVPR 2017)
  - Propagate features on sparse key frames to others
  - Up to **10x** faster at moderate accuracy loss
- Feature aggregation for **better accuracy** (ICCV 2017)
  - Aggregate features on near-by frames to current frame
  - Enhanced feature, better recognition result
- Joint training of flow and recognition in DNN
- Clean, end-to-end, general
- Powering the winner of ImageNet VID 2017

key frame

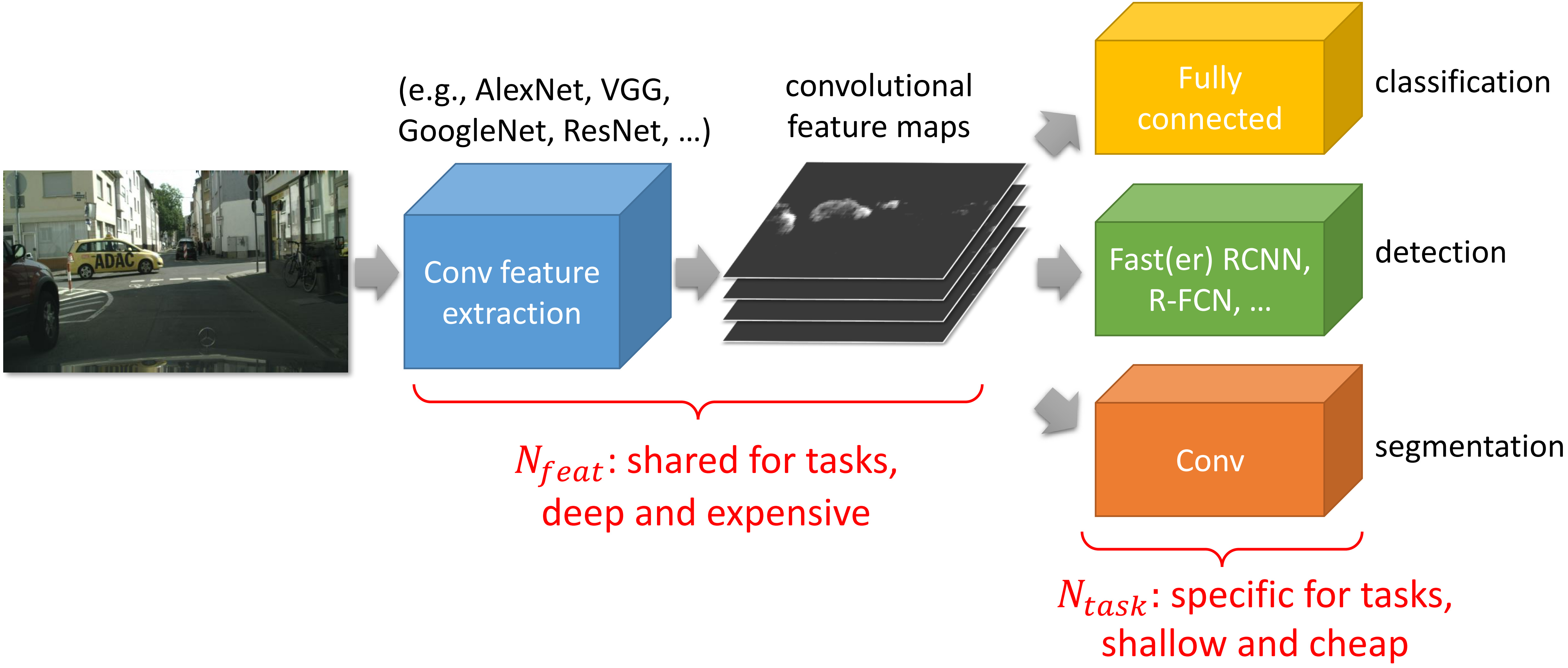
flow field

current frame

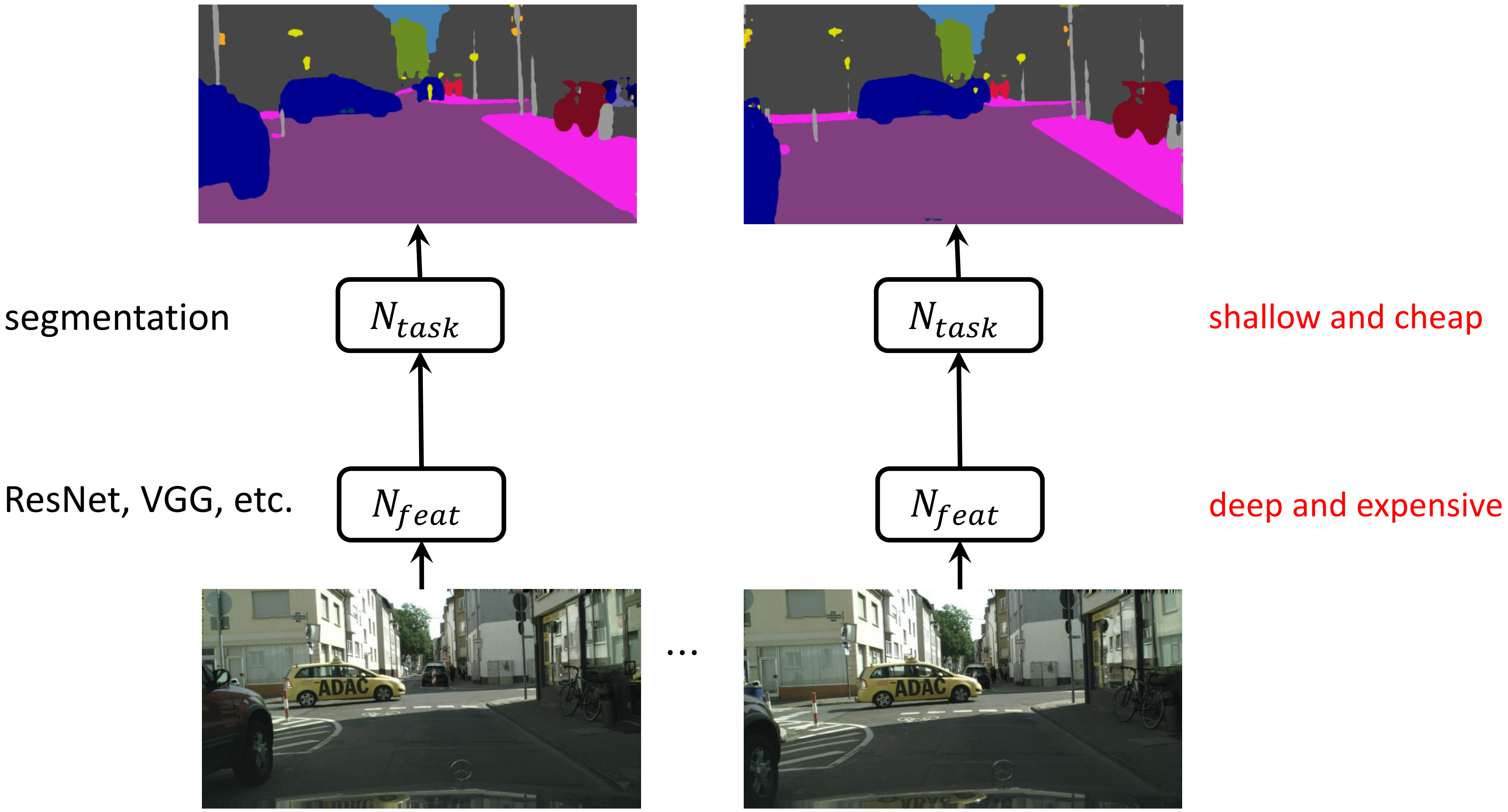
Talk pipeline

- Introduction
- Deep Feature Flow for Video Recognition
- Flow-Guided Feature Aggregation for Video Object Detection
- Summary

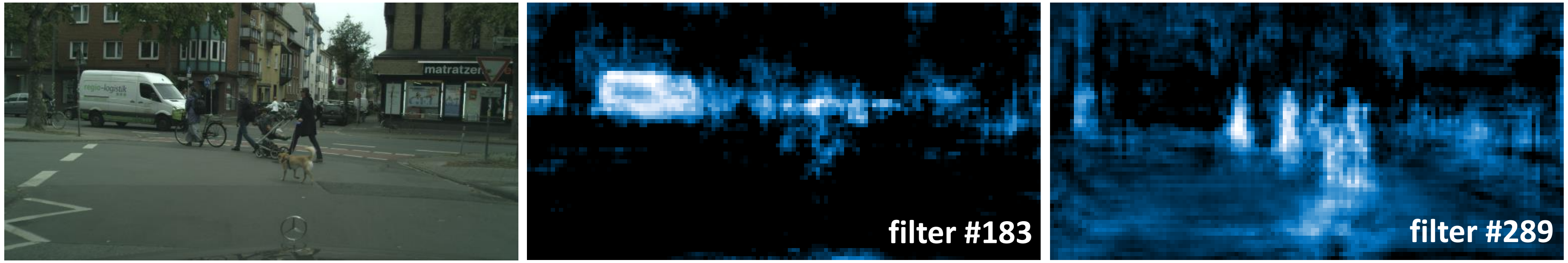
Modern structure for image recognition



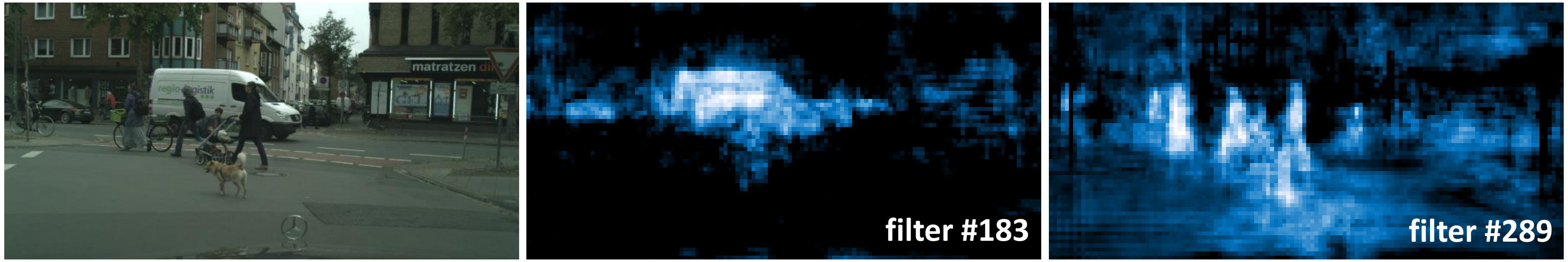
Per-frame baseline



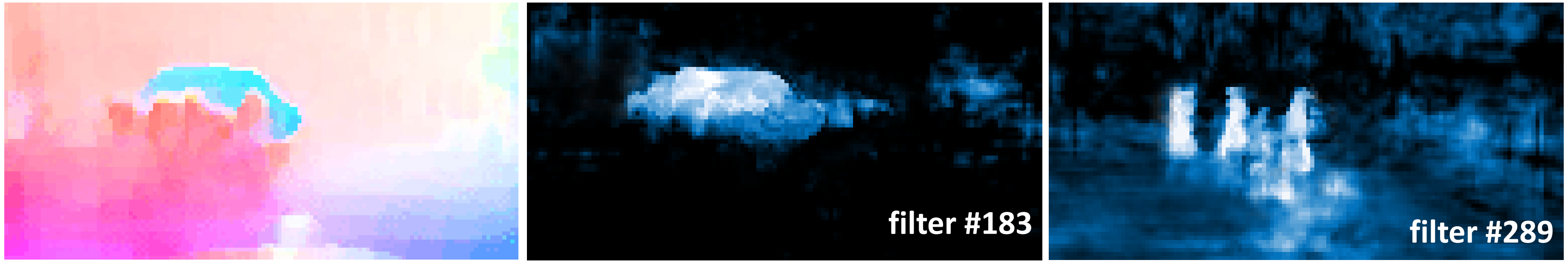
Deep feature flow: key idea



key frame key frame feature maps

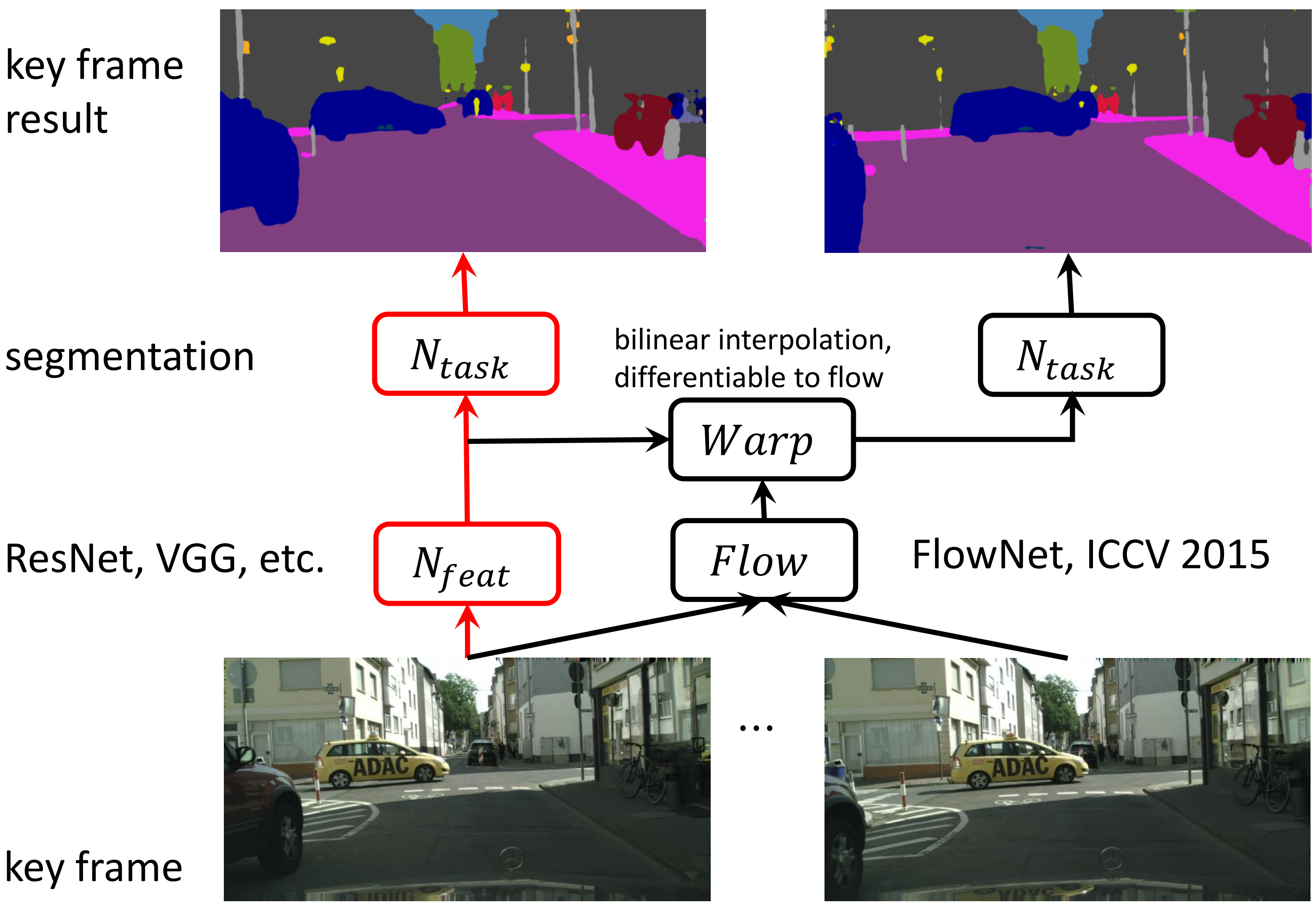


current frame current frame feature maps



flow field warped from key frame to current frame

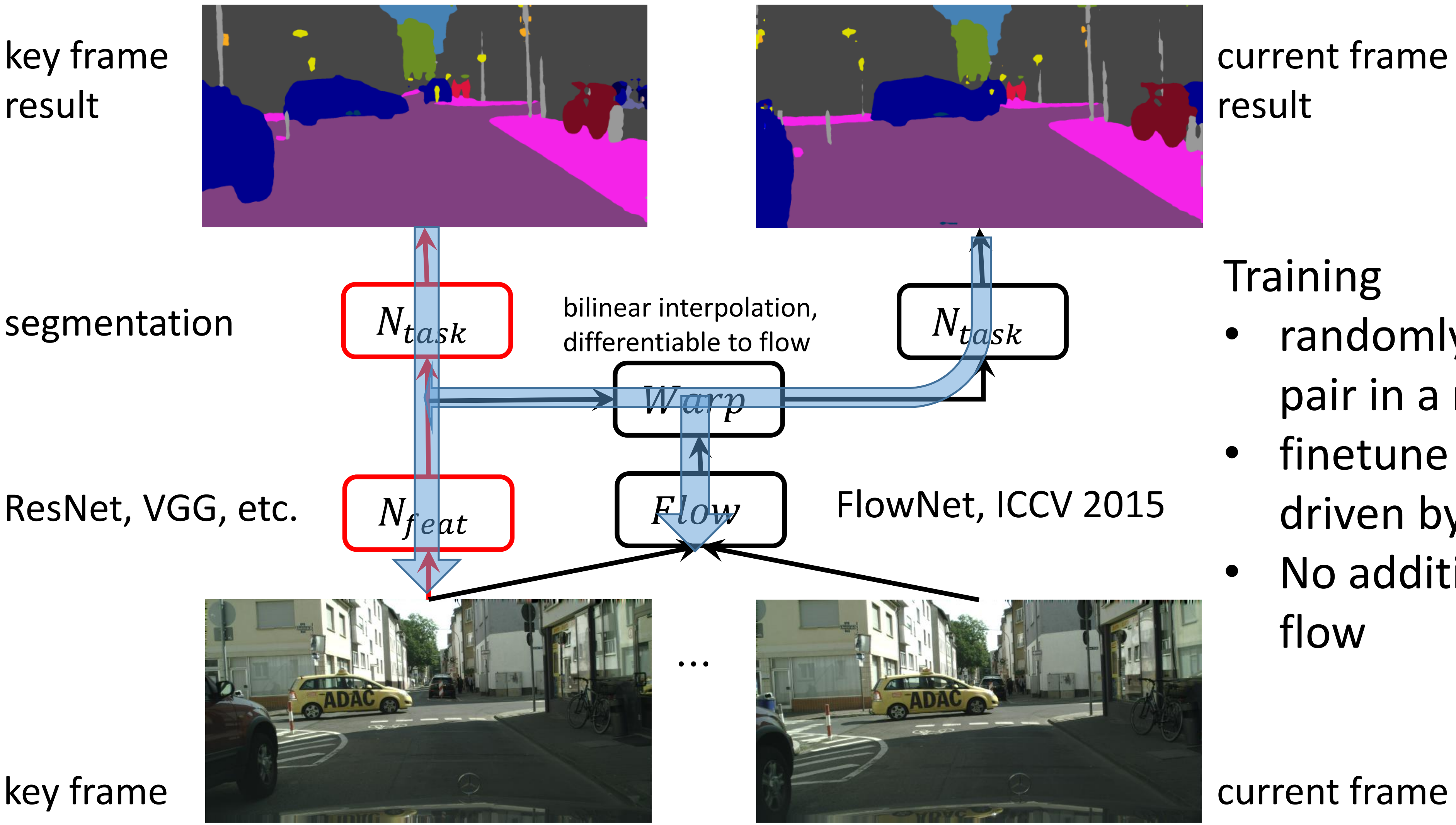
Deep feature flow: network structure



Inference

- run N_{feat} for each key frame
- run flow branch for a few frames after key frame
- key frame is sparse

Feature propagation: training



Training
- randomly sample a frame pair in a minibatch
- finetune all the modules driven by the recognition task
- No additional supervision for flow

Computational complexity analysis

- Per-frame computation ratio $r = \frac{\overbrace{O(F)+O(W)+O(N_{task})}^{\text{propagation from key frame}}}{\underbrace{O(N_{feat})+O(N_{task})}_{\text{computation on key frame}}} \approx \frac{O(F)}{O(N_{feat})} \ll 1$

  W and N_{task} are very cheap

- Flow F is much cheaper than feature extraction N_{feat}

$N_{feat} \setminus F$	FlowNet	FlowNet Half (1/4 of FlowNet)	FlowNet Inception (1/8 of FlowNet)
ResNet-50	9.20	33.56	68.97
ResNet-101	12.71	46.30	95.24

default setting

As $r \ll 1$, here we show $\frac{1}{r}$ for clarify.

Experiment datasets

task	semantic segmentation	object detection
dataset	CityScapes	ImageNet VID
frames per second	17	25 or 30
key frame duration	5	10
#semantic categories	30	30
#videos	train 2975, validation 500, test 1525	train 3862, validation 555, test 937
#frames per video	30	6~5492
annotation	every 20^{th} frame	all frames
evaluation metric	mIoU (mean of Intersection-over-Union)	mAP (mean of Average Precision)

key frame duration is manually chosen to fit the application needs for accuracy-speed trade-off

1. a long duration saves more feature computation but has lower accuracy as flow is less accurate
2. vice versa for a short duration

Ablation study: results on two tasks

method \ task	segmentation	on CityScapes	detection	on ImageNet VID
method \ metric	mIoU (%)	runtime (fps)	mAP (%)	runtime (fps)
Frame (oracle baseline)	71.1	1.52	73.9	4.05
SFF: shallow feature flow (SIFT)				
SFF-slow	67.8	0.08	70.7	0.26
SFF-fast	67.3	0.95	69.7	3.04
DFF: deep feature flow				
DFF	***69.2***	5.60	***73.1***	20.25
DFF fix N	68.8	5.60	72.3	20.25
DFF fix F	67.0	5.60	68.8	20.25
DFF separate	66.9	5.60	67.4	20.25

1. *DFF* is much faster than singe *Frame* baseline at moderate accuracy loss

2. Using off-the-shelf flow algorithm is worse

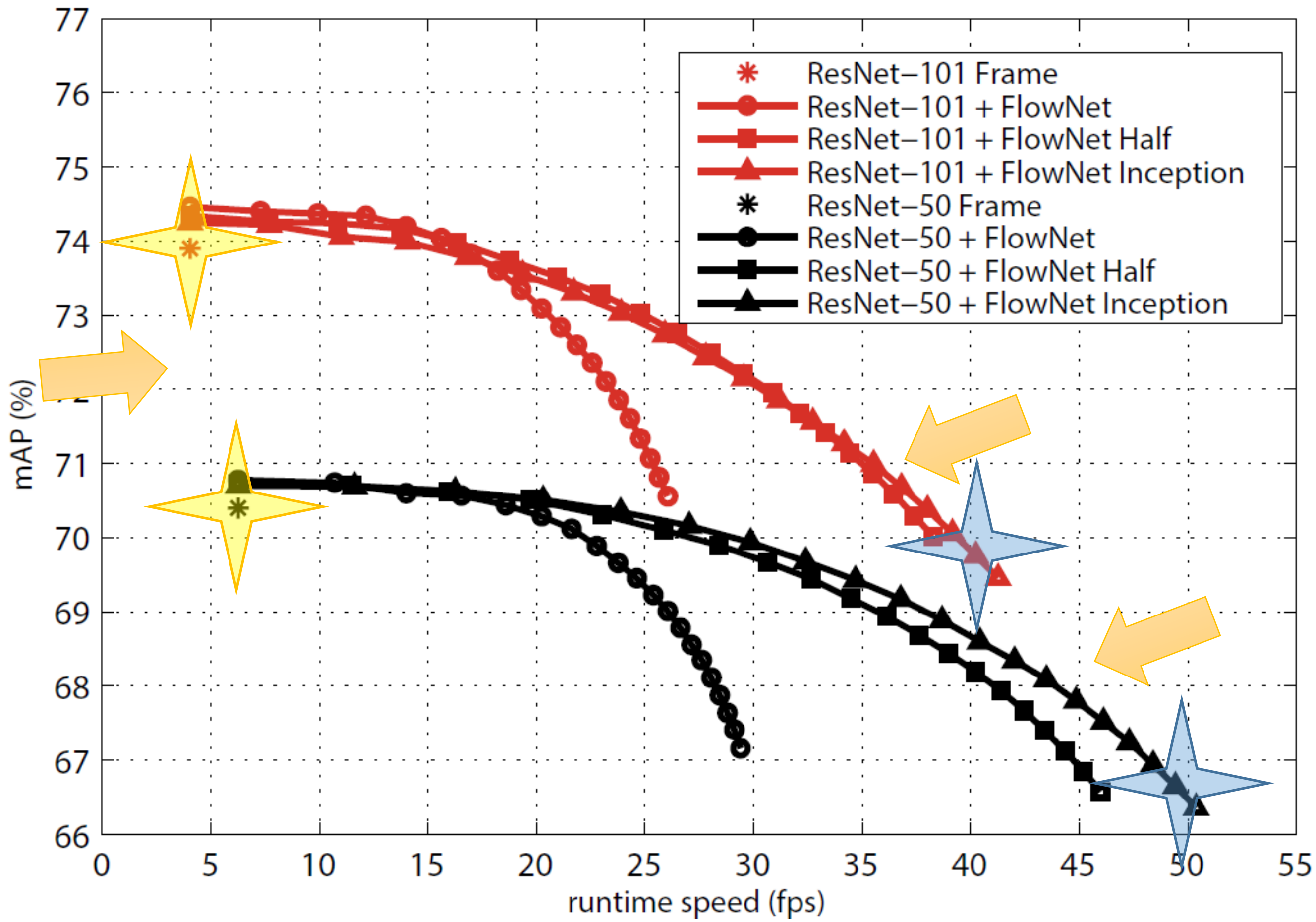
3. Joint end-to-end training is effective

Accuracy-speedup tradeoff by varying N_{feat} and F

- Significant speedup with decent accuracy drop
  (10X faster, 4.4% accuracy drop)

- How to choose flow function?
  - Cheapest FlowNet Inception is the best

- How to choose conv. features?
  - ResNet101 is better



ImageNet VID detection (5354 videos, 25 ~ 30 fps)

Cityscapes Dataset (17 fps, 1024 x 2048)

only **single** frame is **annotated** in each snippets (30 frames)

Ground truth

Our results

Talk pipeline

- Introduction
- Deep Feature Flow for Video Recognition
- Flow-Guided Feature Aggregation for Video Object Detection
- Summary

Deteriorated appearance in videos

How to improve video object detection

Post-processing: box level

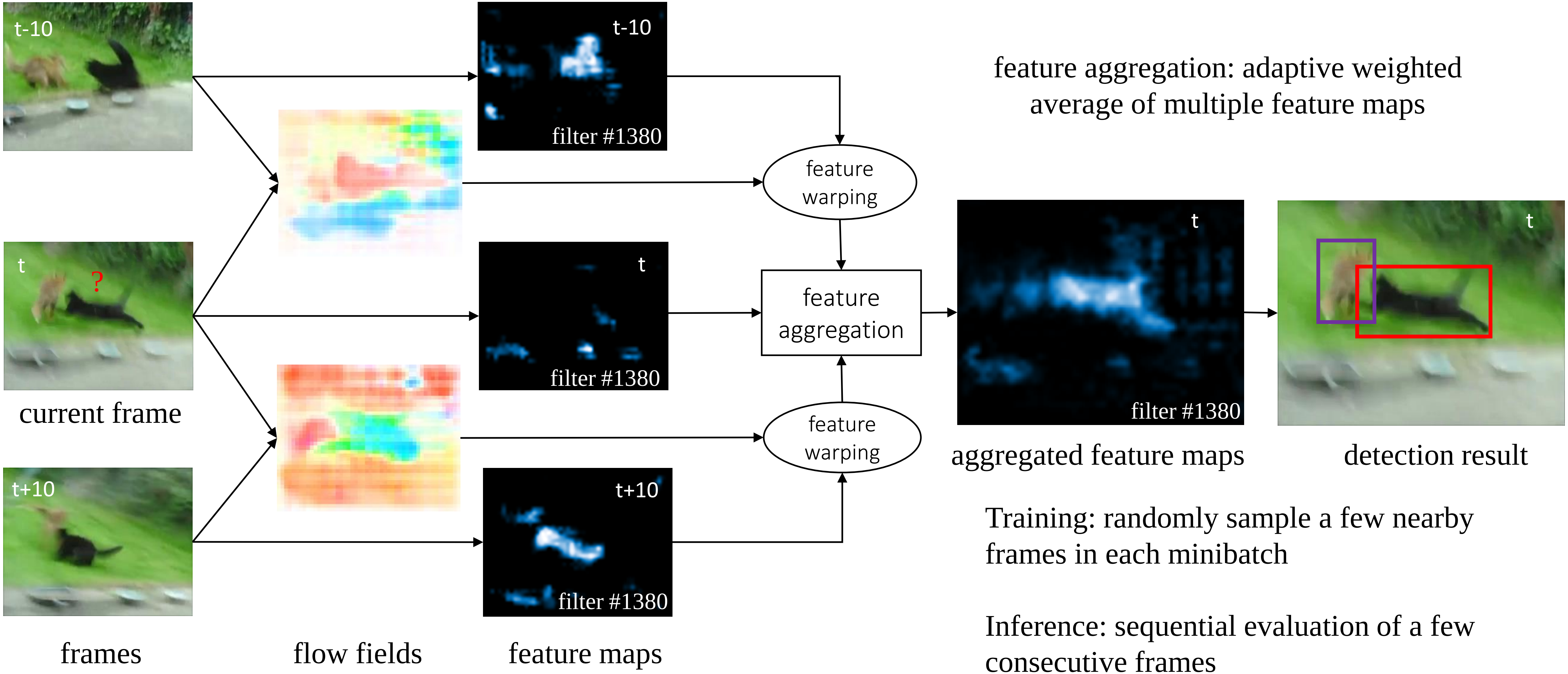
- Manipulation of detected boxes
  - e.g., tracking over multi-frames
- Heuristic, heavily engineered
- Widely used in competition

Better feature learning: feature level

- Enhance deep features
  - learning over multi-frames
- Principled, clean
- Rarely studied

First end-to-end DNN work for video object detection

Flow-guided feature aggregation

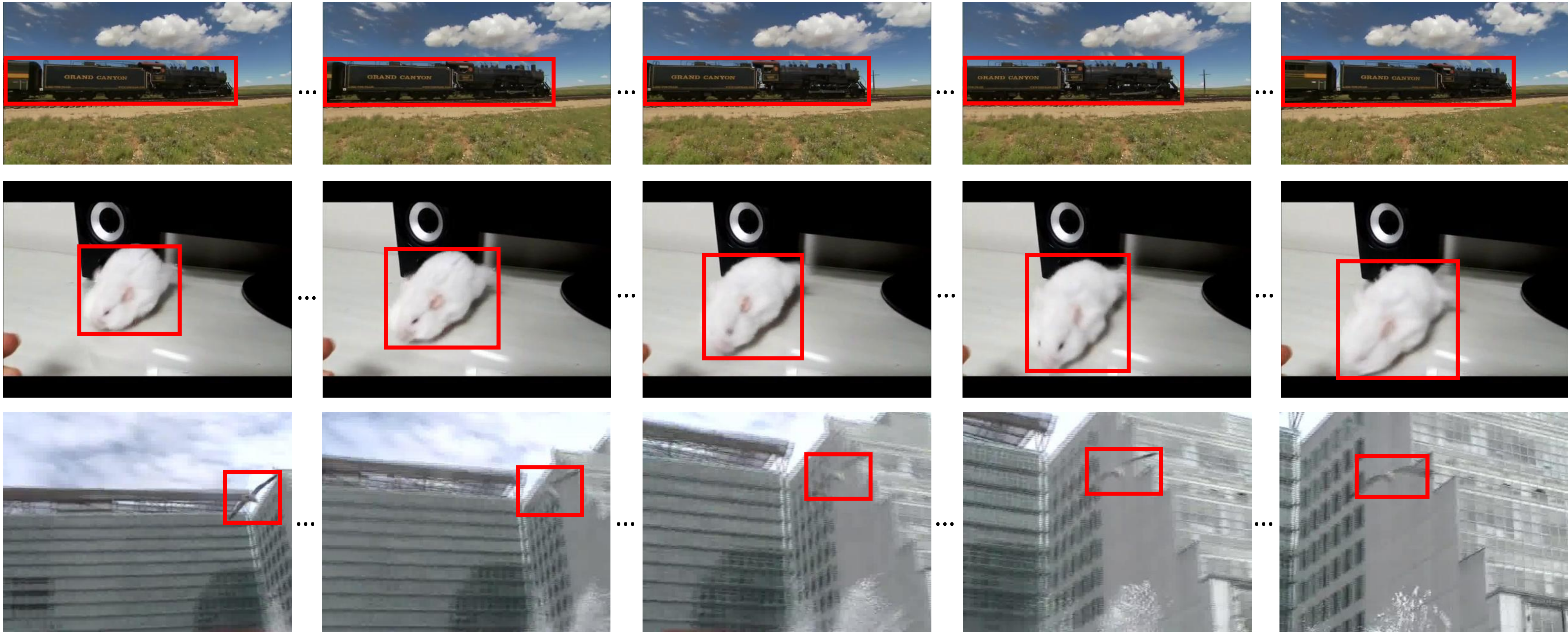


feature aggregation: adaptive weighted average of multiple feature maps

Training: randomly sample a few nearby frames in each minibatch

Inference: sequential evaluation of a few consecutive frames

Use motion IoU to measure object speed



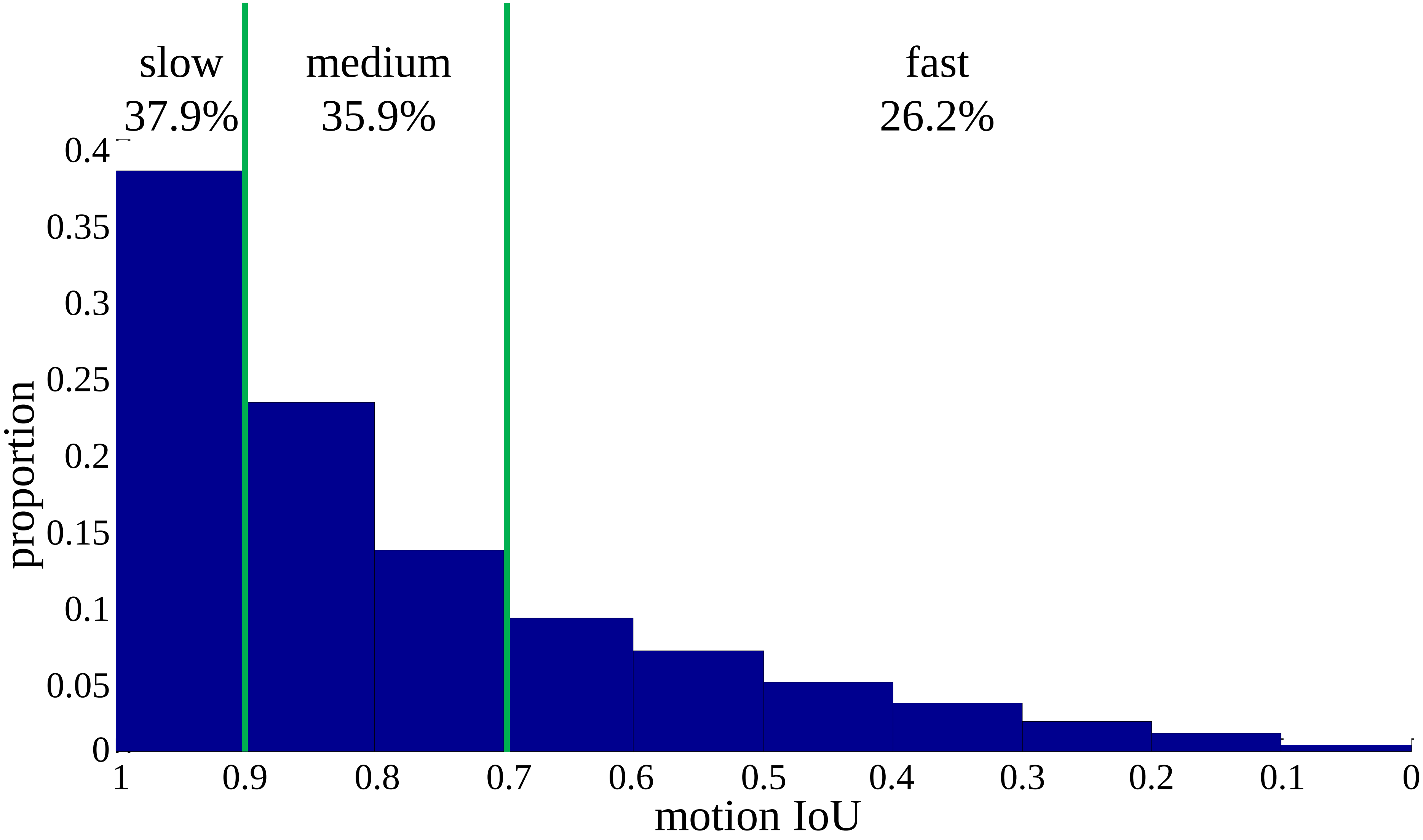
slow

medium

fast

t-10 … t-5 … t … t+5 … t+10

Categorization of object speed



Ablation study results on ImageNet VID

methods	Single frame baseline	Ours (no flow/weights)	Ours (no flow)	***Ours***	Ours (no e2e training)
multi-frame aggregation		√	√	√	√
adaptive weights			√	√	√
flow guided				√	√
end-to-end training		√	√	√	
mAP (%)	73.4	72.0	74.3	***76.3 (↑2.9)***	74.5
mAP (%) (slow)	82.4	82.3	82.2	***83.5 (↑1.1)***	82.5
mAP (%) (medium)	71.6	74.5	74.6	***75.8 (↑4.2)***	74.6
mAP (%) (fast)	51.4	44.6	52.3	***57.6 (↑6.2)***	53.2
runtime (ms)	288	288	305	733	733

1. All components (flow, adaptive weighting, end-to-end learning) are important.

2. Especially effective on fast (difficult) objects

3. Slower as flow computation takes time

#frames in training and inference

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

*: default parameter

- More frames in inference is better (saturated at 21)
- 2 frames in training is sufficient (frame skip randomly sampled)

Integration with post-processing techniques

- Complementary with post-processing techniques
- A clean solution with state-of-the-art performance (80.1 mAP)
  - ImageNet VID 2016 winner: 81.2
  - Highly engineered with various tricks, not used in ours

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		<u>78.4</u>	873*
FGFA	Aligned-Inception-ResNet	77.8	819
+ Seq-NMS		$\boxed{\textbf{80.1}}$	954*

Table 4. Results of baseline method and FGFA before and after combination with box level techniques. As for runtime, entry marked with * utilizes CPU implementation of box-level techniques.

Powering the winner of ImageNet VID 2017

Team name	Entry description	**Number of object categories won**	mean AP
IC&USYD	provide_submission3	15	0.817265
IC&USYD	provide_submission1	6	0.808847
IC&USYD	provide_submission2	4	0.818309
NUS-Qihoo-UIUC_DPNs (VID)	no_extra + seq + mca + mcs	3	0.757772
NUS-Qihoo-UIUC_DPNs (VID)	no_extra + seq + vcm + mcs	1	0.757853
NUS-Qihoo-UIUC_DPNs (VID)	Faster RCNN + Video Context	1	0.748493
THU-CAS	merge-new	0	0.730498
THU-CAS	old-new	0	0.728707
THU-CAS	new-new	0	0.691423
GoerVision	Deformable R-FCN single model+ResNet101	0	0.669631
GoerVision	Ensemble 2 model, use ResNet101 as foundamental classification network and deformable R-FCN to detect video frames, multi-scale testing	0	0.665693
GoerVision	o train the video objectWe use the ResNet101 and Deformable R-FCN for the detection.	0	0.655686
GoerVision	Using R-FCN to detect video object, multi scale testing applied.	0	0.646965
FACEALL_BUPT	SSD based on Resnet101 networks	0	0.195754

[top]

IC&USYD	Jiankang Deng(1), Yuxiang Zhou(1), Baosheng Yu(2), Zhe Chen(2), Stefanos Zafeiriou(1), Dacheng Tao(2), (1)Imperial College London, (2)University of Sydney	Flow acceleration[1,2] is used. Final scores are adaptively chosen between the detector and tracker. [1] Deep Feature Flow for Video Recognition Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. [2] Flow-Guided Feature Aggregation for Video Object Detection, Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Arxiv tech report, 2017.

Video demo

Results

Talk pipeline

- Introduction
- Deep Feature Flow for Video Recognition
- Flow-Guided Feature Aggregation for Video Object Detection
- Summary

Summary

- Exploit motion for video recognition tasks
  - Faster speed or better accuracy
- End-to-end, joint learning of optical flow and recognition tasks
- Feature learning instead of heuristics, general for different tasks
- Code available at
  - https://github.com/msracver/Deep-Feature-Flow
  - https://github.com/msracver/Flow-Guided-Feature-Aggregation

Related work on video semantic segmentation

- Clockwork convnets for video semantic segmentation, ECCV 2016
- Exploiting semantic information and deep matching for optical flow, ECCV 2016
- STFCN: spatio-temporal FCN for semantic video segmentation, arXiv 2016
- Joint optical flow and temporally consistent semantic segmentation, ECCV 2016 workshop
- Feature space optimization for semantic video segmentation, CVPR, 2016
- Optical flow with semantic segmentation and localized layers, CVPR, 2016

- No end-to-end training, only for semantic segmentation

Related work on video object detection

- Seq-nms for video object detection, arXiv 2016
- T-cnn: Tubelets with convolutional neural networks for object detection from videos, CVPR 2016
- Object detection from video tubelets with convolutional neural networks. In CVPR, 2016
- Object detection in videos with tubelet proposal networks. In CVPR, 2017

- No end-to-end training, post processing on box-level instead of feature-level

Future work

- Better flow learning and evaluation

- Better key frame scheduling
  - Better efficiency and accuracy, simultaneously

- Joint learning for detection and tracking
  - new losses (smoothness, box association) on temporal dimension
  - On the stability of video detection and tracking, arXiv 2016

Thanks! Q & A