Modeling deep structures for 3D scene understanding

Wanli Ouyang （欧阳万里）

The Chinese University of Hong Kong

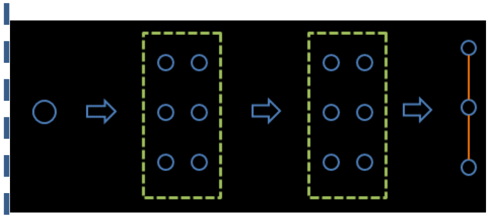
The University of Sydney

Outline

Introduction

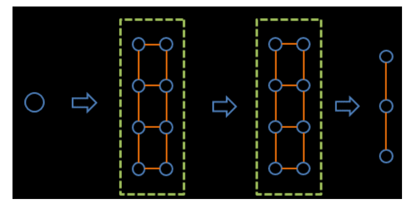
Structured deep learning

Structured output



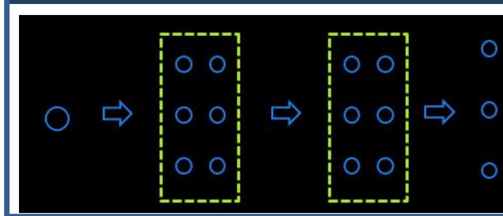
Depth Estimation
(TPAMI'18)

Structured features



Scene Graph Generation
(ICCV'17, ECCV'18)

Back-bone model design



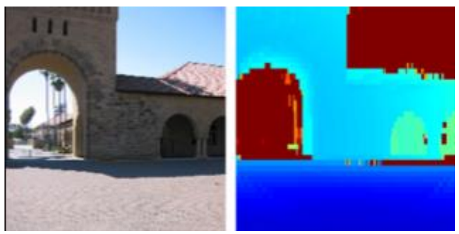
Conclusion

Outline

Introduction

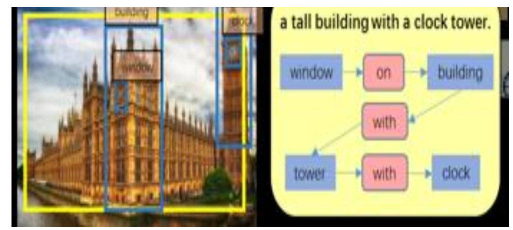
Structured deep learning

Structured output



Depth Estimation
(TPAMI'18)

Structured features



Scene Graph Generation
(ICCV'17, ECCV'18)

Back-bone model design

Image/video classification
(CVPR'18, NIPS'18)

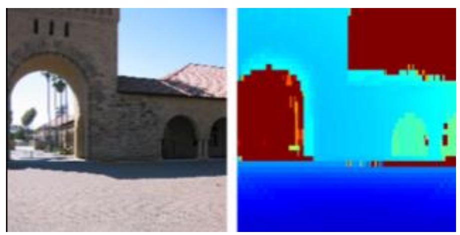
Conclusion

Outline

Introduction

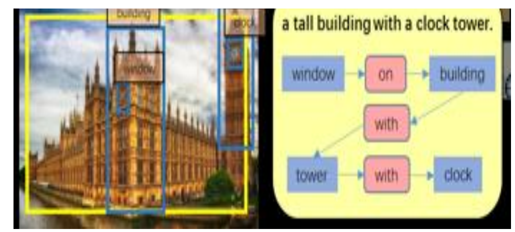
Structured deep learning

Structured output



Depth Estimation
(TPAMI'18)

Structured features



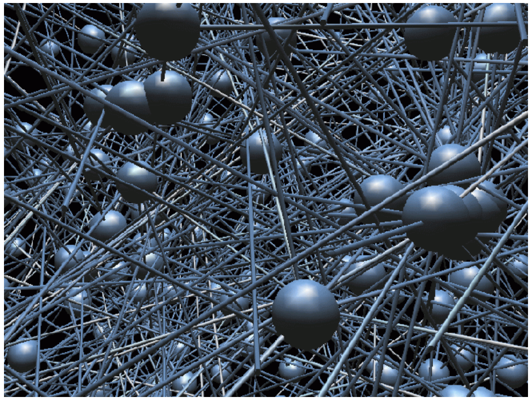
Scene Graph Generation
(ICCV'17, ECCV'18)

Back-bone model design

Image/video classification
(CVPR'18, NIPS'18)

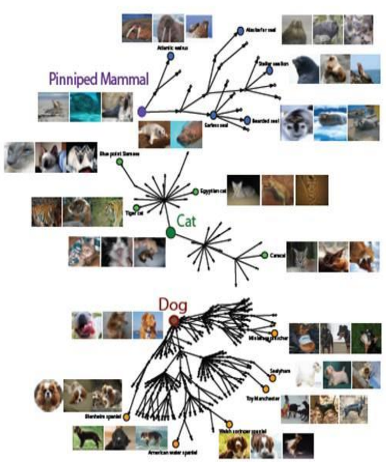
Codes available!

Conclusion




Simulate brain activities and employ millions of neurons to fit billions of training samples. Deep neural networks are trained with GPU clusters with tens of thousands of processors

Hinton won ImageNet competition
Classify 1.2 million images into 1,000 categories
Beating existing computer vision methods by 20+%
Surpassing human performance

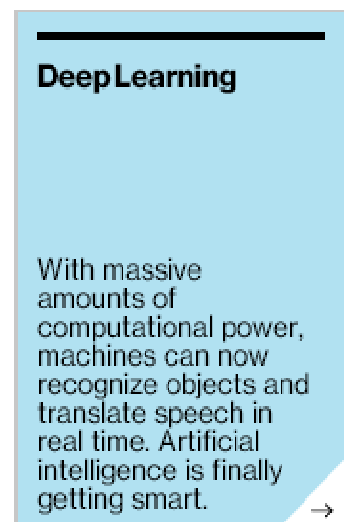


Deep learning

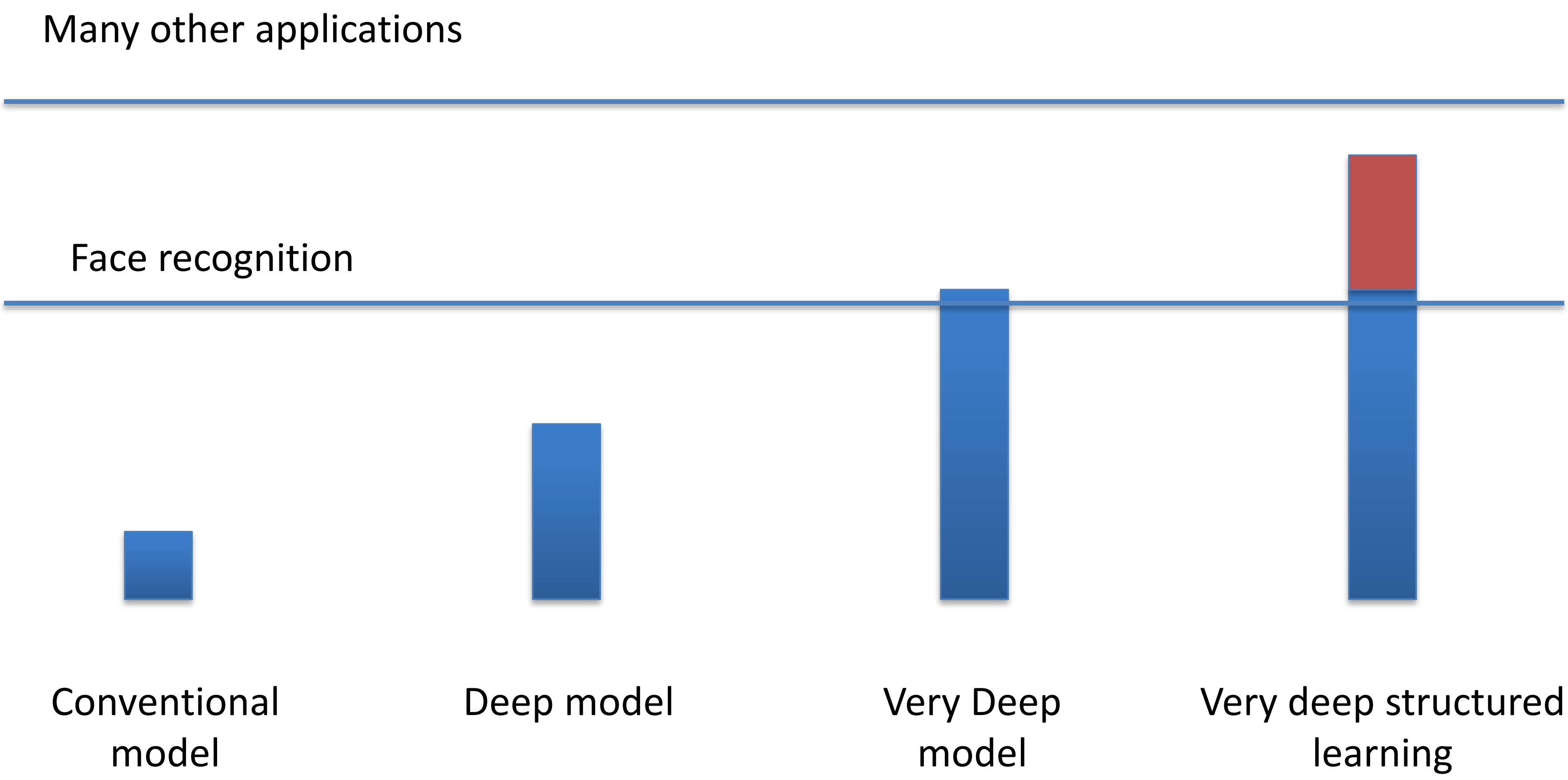
REVOLUTIONARY

Web-scale visual search, self-driving cars, surveillance, multimedia
…
Hold records on most of the computer vision problems

MIT Tech Review
Top 10 Breakthroughs 2013
Ranking No. 1

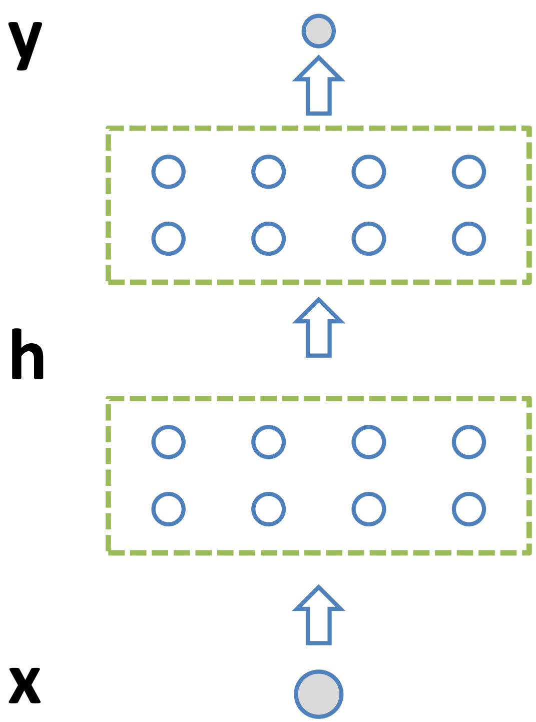


Performance vs practical need



Structure in neurons

- Conventional neural networks
  - Neurons in the same layer have no connection
  - Neurons in adjacent layers are fully connected, at least within a local region



Structure exists in brain

Structure in data

Structure in data

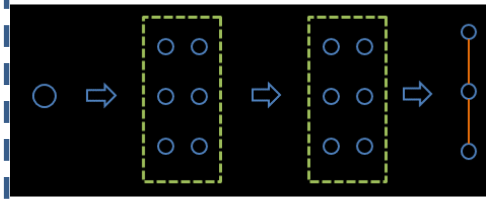
Correlation

Outline

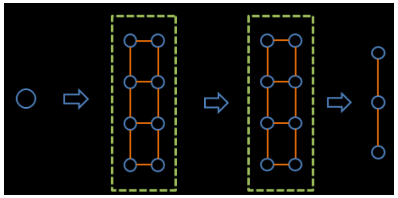
Introduction

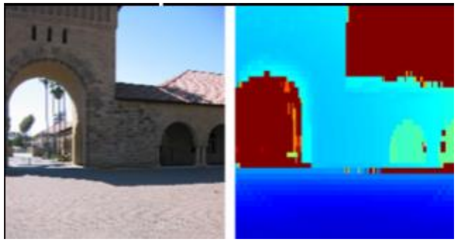
Structured deep learning

Structured output

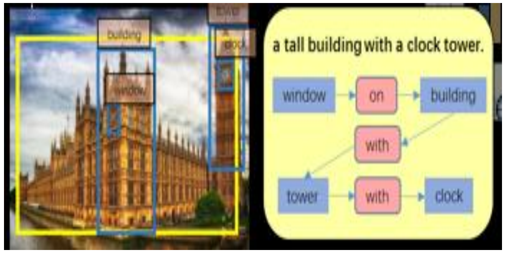


Structured features





Depth Estimation
(TPAMI'18)



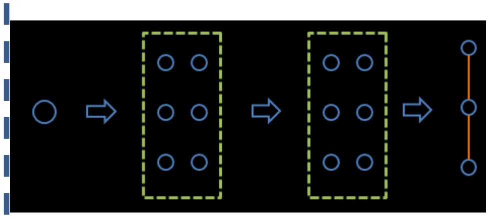
Scene Graph Generation
(ICCV'17, ECCV'18)

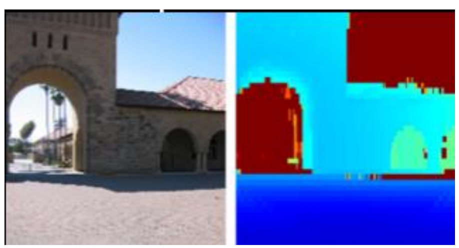
Outline

Introduction

Structured deep learning

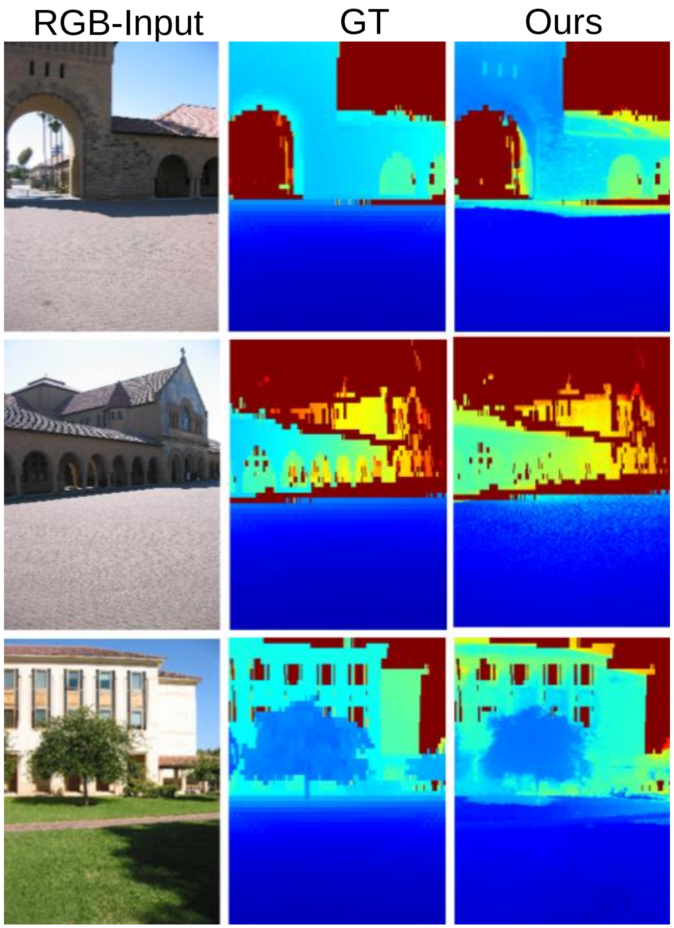
Structured output





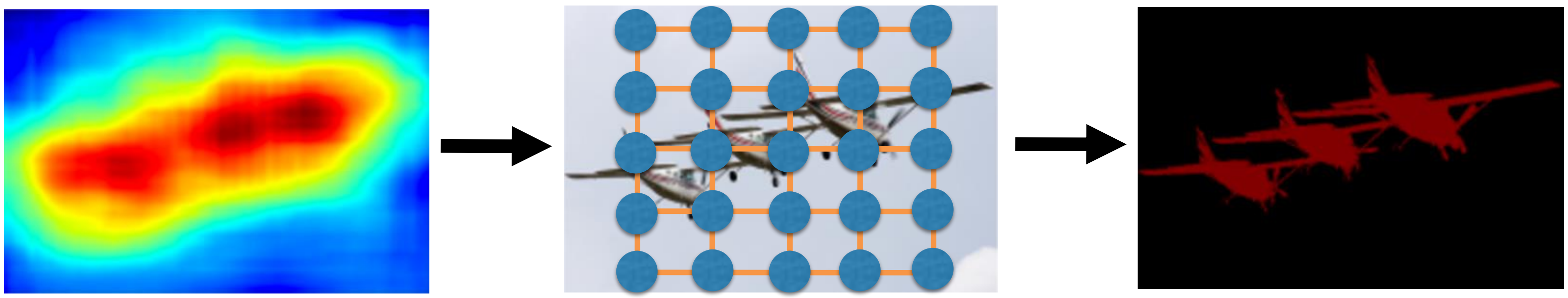
Depth Estimation
(TPAMI'18)

Monocular depth estimation



Motivation

- Deep structured dense pixel-level prediction:



CNN coarse output | CRF-modeling | Inference

Representative works:

- CRF-RNN:
  - S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

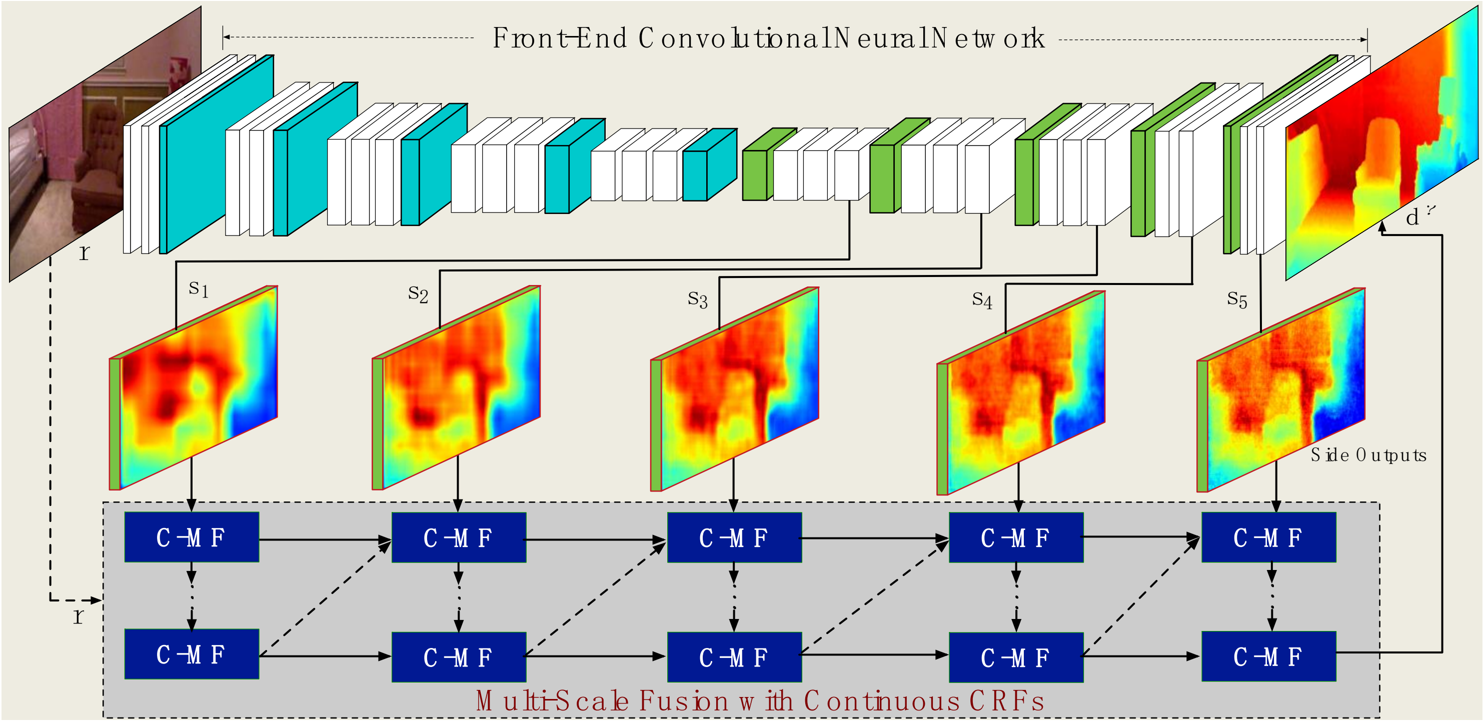
In Discrete Domain

- Deep convolutional neural field:
  - F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039, 2016.

In single scale with patch-level refinement due to the $O(n^3)$ complexity of closed-form solution

Ours: In Multi-scale with pixel-level dense refinement with $O(n)$ complexity

Approach



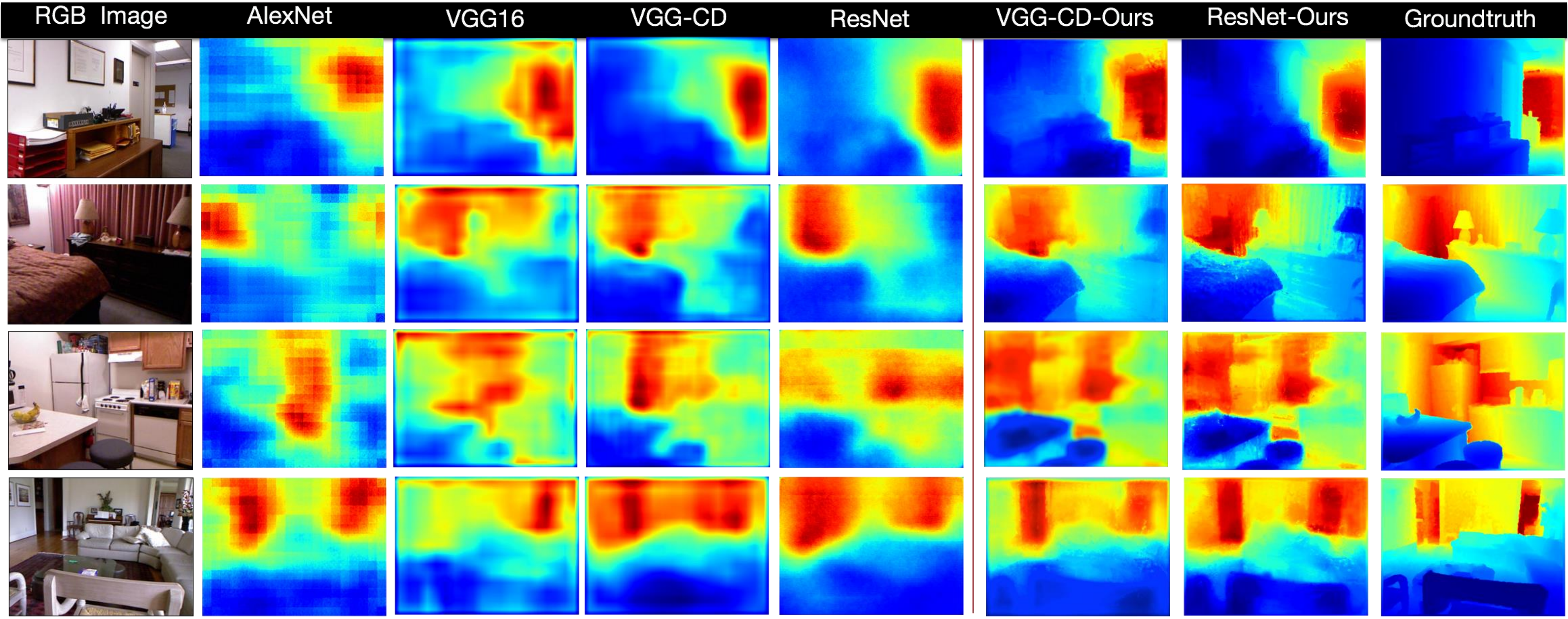
Multi-Scale Deep Structured Fusion & Prediction + In Continuous Domain + Within a Joint CNN-CRF Framework

Spotlight Oral, TPAMI18

Message passing using CNN-CRF

- 20 minutes to illustrate CRF …

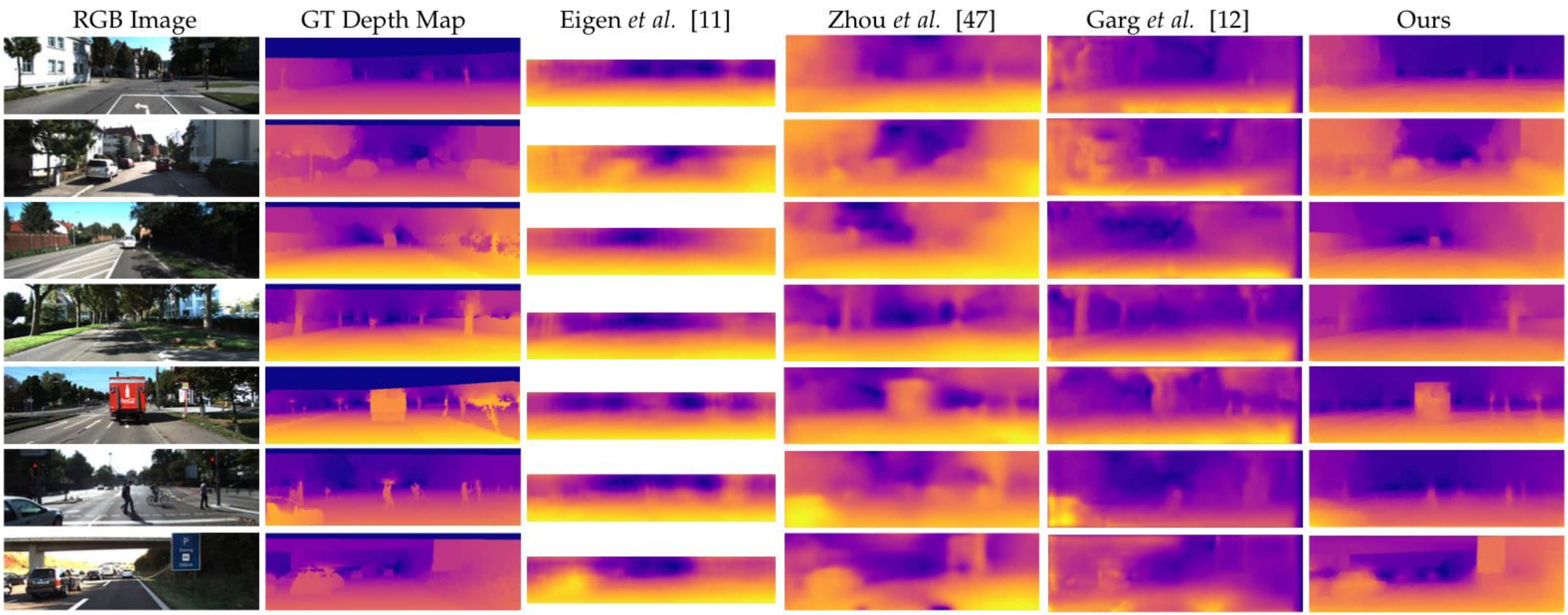
Results



Qualitative results on NYUD-V2: significant improvement over the pretrained front-end CNNs

Code:

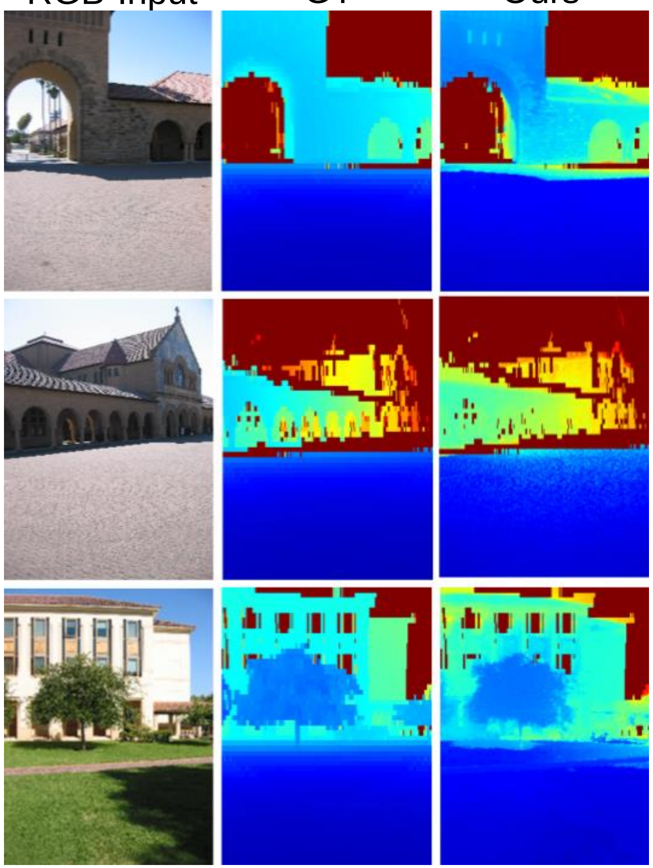
Results



Results on KITTI: ours achieved the best performance compared with the state-of-the-art

Code:

Results



Qualitative results on Make3D

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
HED [42]	0.185	0.077	0.723	0.678	0.918	0.980
Hypercolumn [13]	0.189	0.080	0.730	0.667	0.911	0.978
C-CRF	0.193	0.082	0.742	0.662	0.909	0.976
Ours (single-scale)	0.187	0.079	0.727	0.674	0.916	0.980
Ours - cascade (3-scale)	0.176	0.074	0.695	0.689	0.920	0.980
Ours - cascade (5-scale)	0.169	0.071	0.673	0.698	0.923	**0.981**
Ours - unified (3-scale)	0.172	0.072	0.683	0.691	0.922	**0.981**
Ours - unified (5-scale)	**0.163**	**0.069**	**0.655**	**0.706**	**0.925**	**0.981**

More effective than the classic multi-scale fusion schemes

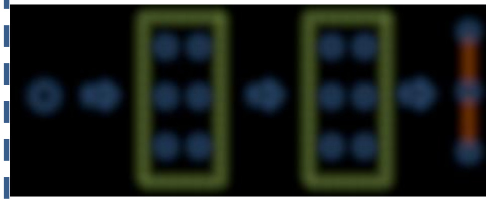
Method	C1 Error			C2 Error		
	rel	log10	rms	rel	log10	rms
Karsch *et al.* [17]	0.355	0.127	9.20	0.361	0.148	15.10
Liu et al. [28]	0.335	0.137	9.49	0.338	0.134	12.60
Liu et al. [26]	0.314	0.119	8.60	0.307	0.125	12.89
Li et al. [24]	0.278	0.092	7.19	0.279	0.102	10.27
Laina *et al.* [23] (ℓ_2 loss)	0.223	0.089	4.89	-	-	-
Laina *et al.* [23] (Huber loss)	0.176	0.072	4.46	-	-	-
Ours (ResNet-50-cascade)	0.213	0.082	4.67	0.221	4.79	8.81
Ours (ResNet-50-unified)	0.206	0.076	4.51	0.212	4.71	8.73
Ours (ResNet-50-unified-10K)	0.184	**0.065**	**4.38**	**0.198**	**4.53**	**8.56**

Achieved the best performance on most of the metrics.

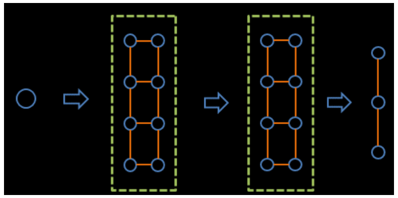
Outline

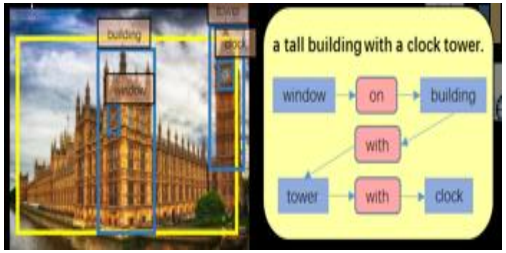
Introduction

Structured deep learning



Structured features

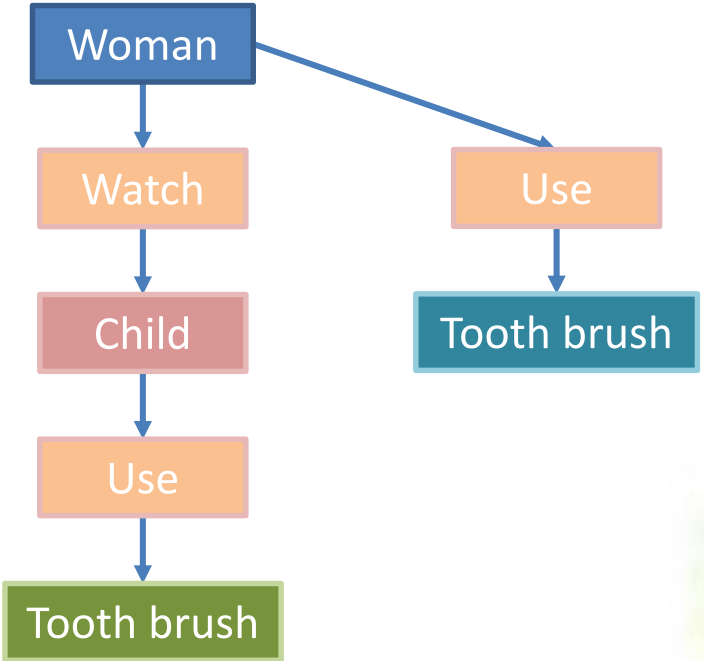




Scene Graph Generation
(ICCV’17, ECCV’18)

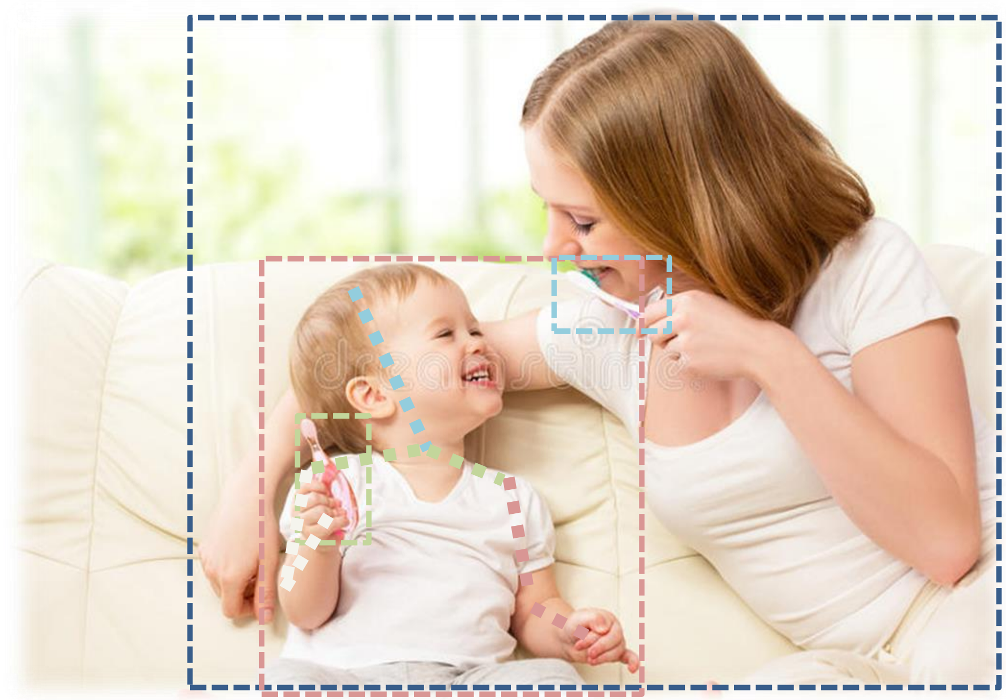
Mom and her cute baby are brushing teeth

Region captioning



Scene graph generation

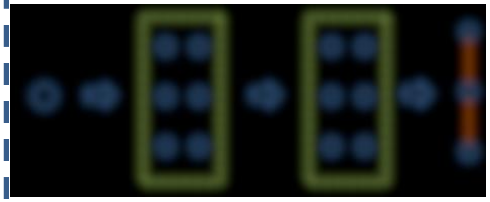
Object detection



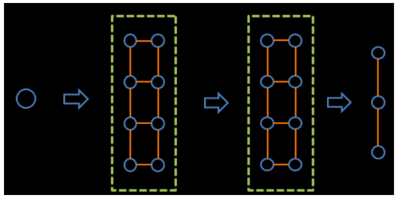
Outline

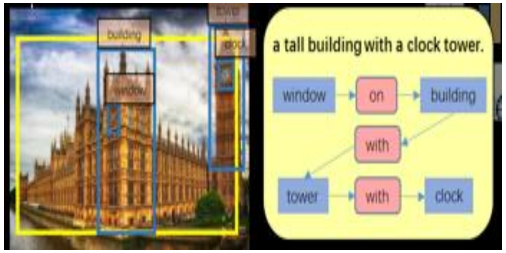
Introduction

Structured deep learning



Structured features

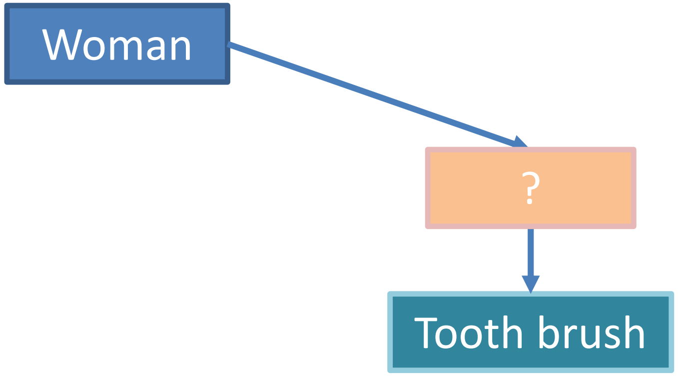


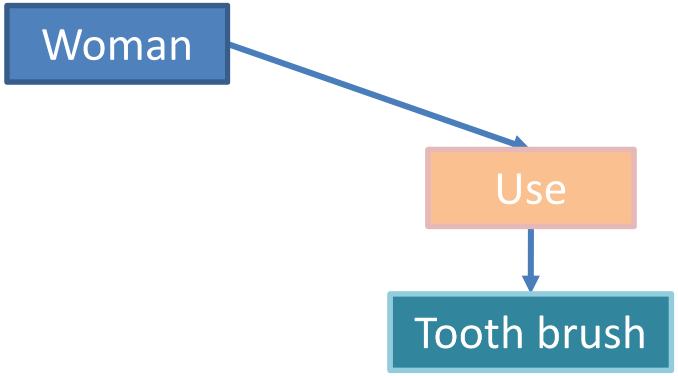


Scene Graph Generation (ICCV’17, ECCV’18)

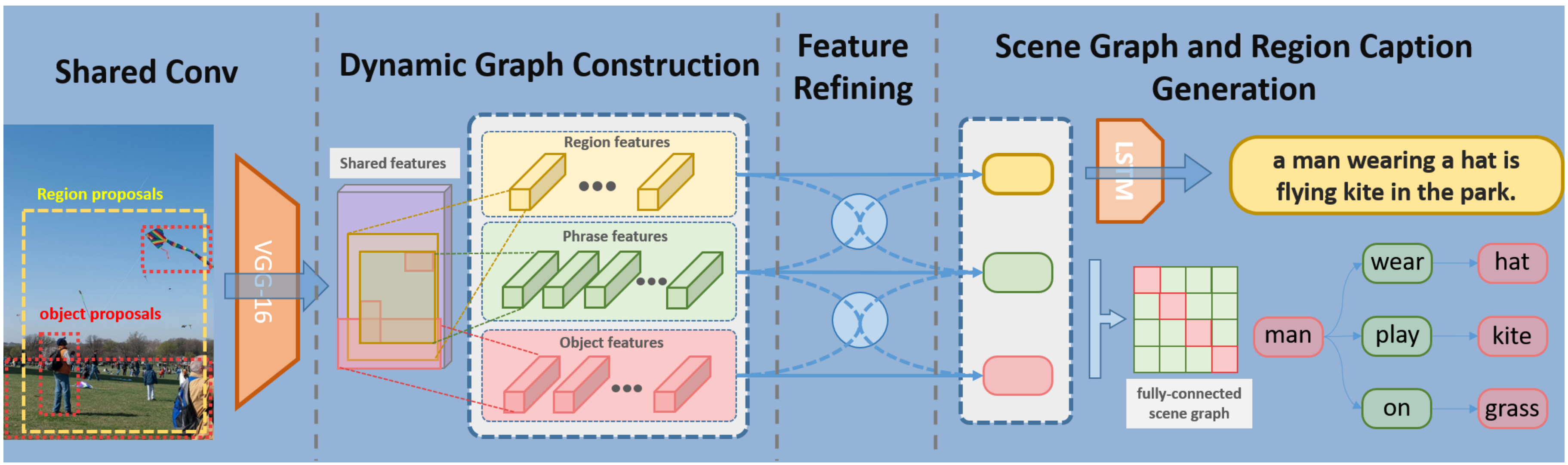
Why structured features?

Contains rich visual information

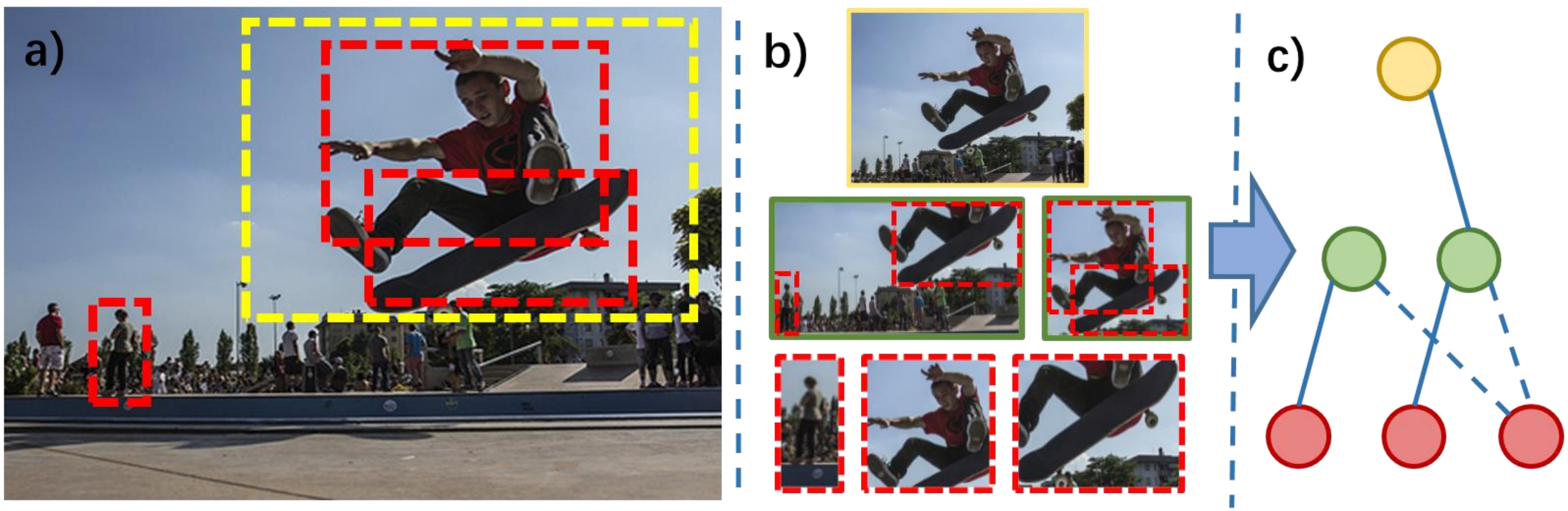

Woman
?
Tooth brush


Woman
Use
Tooth brush

Overview our proposed **Multi-level Scene Description Network (MSDN)**

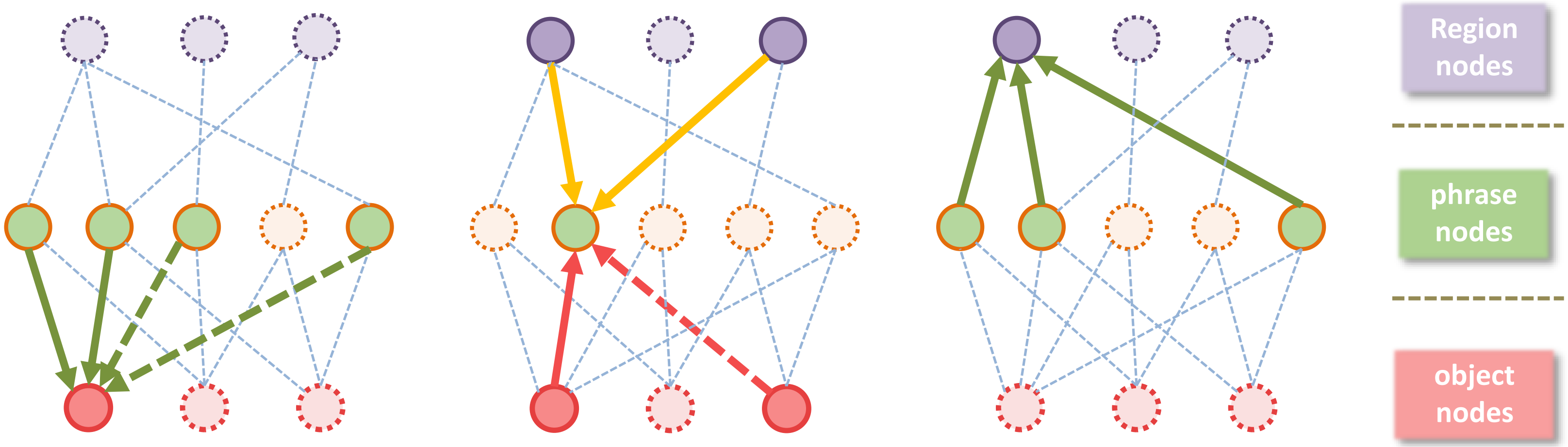


Methodology: Dynamic Graph Construction

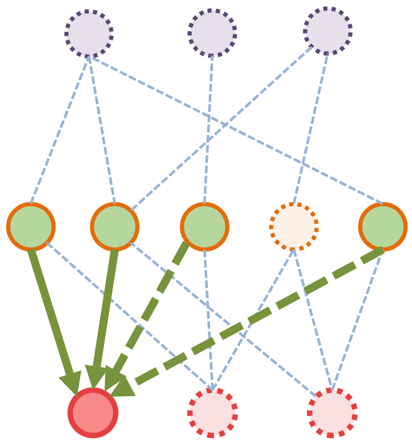


Methodology: Feature Refining

(a) Object Updating **(b) Phrase Updating** **(c) Region Updating**



Methodology: Object feature updating



- **Phrase feature merge:** Since the features from different phrases have different importance factors for refining objects, we use a gate function to determine weights.

$$\tilde{\boldsymbol{x}}_i^{(p\to s)} = \frac{1}{\|\boldsymbol{E}_{i,p}\|} \sum_{(i,j)\in \boldsymbol{E}_{s,p}} \sigma_{\langle o,p\rangle} \left(\boldsymbol{x}_i^{(o)}, \boldsymbol{x}_j^{(p)}\right) \boldsymbol{x}_j^{(p)}$$

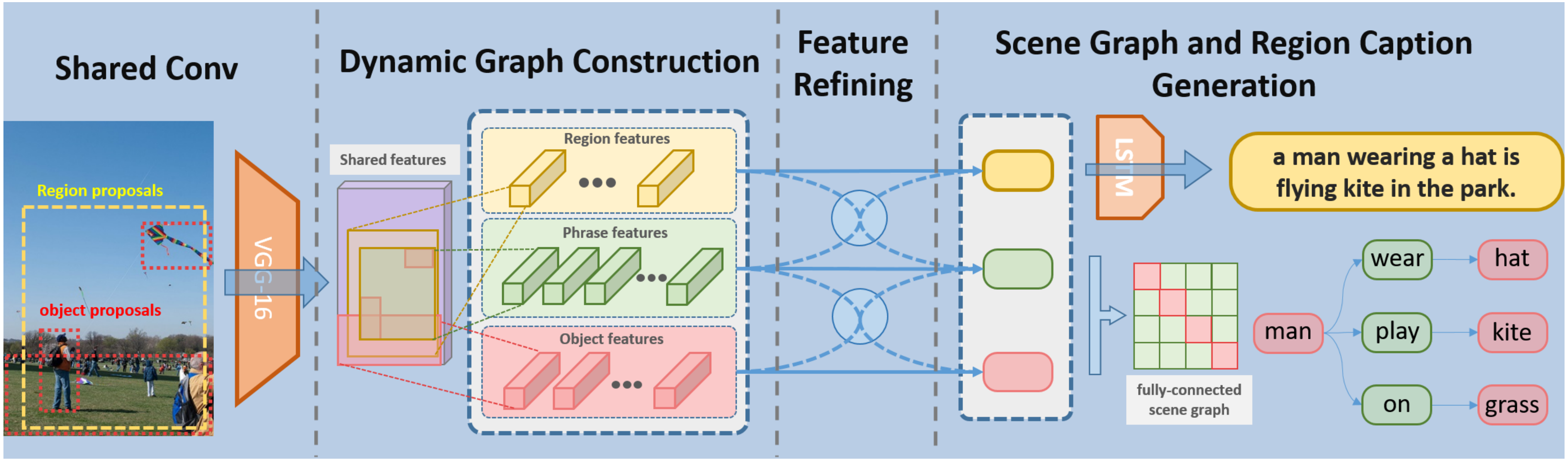
The gate function is defined as:

$$\sigma_{\langle o,p\rangle}\left(\boldsymbol{x}_i^{(o)}, \boldsymbol{x}_j^{(p)}\right) = \sum_{g=1}^{G} \mathrm{sigmoid}\left(\boldsymbol{w}_{\langle o,p\rangle}^{(g)} \cdot \left[\boldsymbol{x}_i^{(o)}, \boldsymbol{x}_j^{(p)}\right]\right),$$

- **Refine object features**: For the *i*-th object, there are two merged features:

$$\boldsymbol{x}_{i,t+1}^{(o)} = \boldsymbol{x}_{i,t}^{(o)} + \boldsymbol{F}^{(p\to s)}\left(\tilde{\boldsymbol{x}}_i^{(p\to s)}\right) + \boldsymbol{F}^{(p\to o)}\left(\tilde{\boldsymbol{x}}_i^{(p\to o)}\right)$$

Overview our proposed **Multi-level Scene Description Network (MSDN)**



Code:

Quantitative Results

Comparison with existing works:

- *LP: Visual Relationship detection using word embeddings as language prior (Lu, Cewu, et al., ECCV 2016)*
- *ISGG: Scene graph generation using iterative message passing (Xu, Danfei, et al. arXiv:1701.02426)*

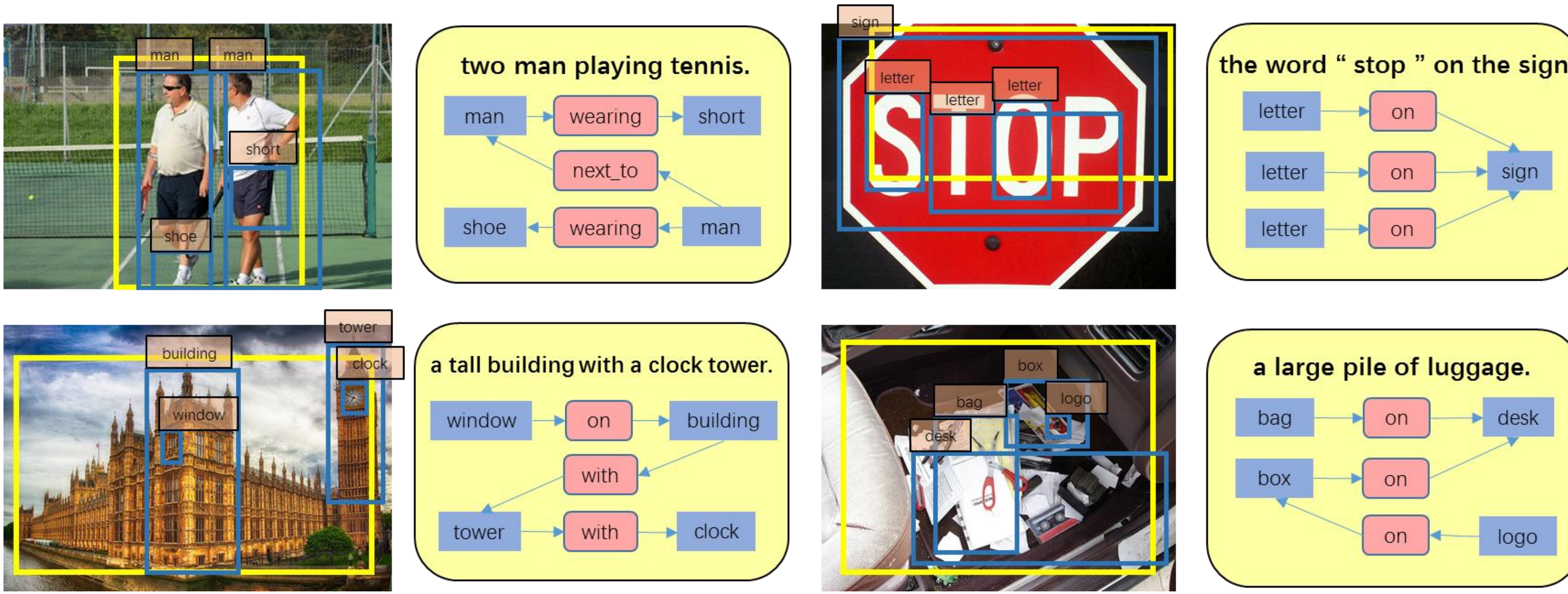
Task		LP [23]	ISGG [33]	Ours
PredCls	R@50	26.67	58.17	**67.03**
	R@100	33.32	62.74	**71.01**
PhrCls	R@50	10.11	18.77	**24.34**
	R@100	12.64	20.23	**26.50**
SGGen	R@50	0.08	7.09	**10.72**
	R@100	0.14	9.91	**14.22**

Experiment on object detection & captioning:

- *FRCNN: Faster R-CNN (Girshick, Ross., ICCV 2015) with the same number of potential object proposals as used at our MSDN.*
- *Baseline-3-bran.: the baseline model with 3 branches but the feature refining structure removed.*

Object Det.	FRCNN [31]	Baseline-3-bran.	Ours
mean AP(%)	6.72	6.70	**7.43**
Acc. Top-1(%)	53.57	53.14	**61.12**
Acc. Top-5(%)	83.50	83.25	**89.86**
Region Caption	Baseline	Baseline-3-bran.	Ours
AP [18](%)	3.98	3.68	**5.07**

Qualitative Results



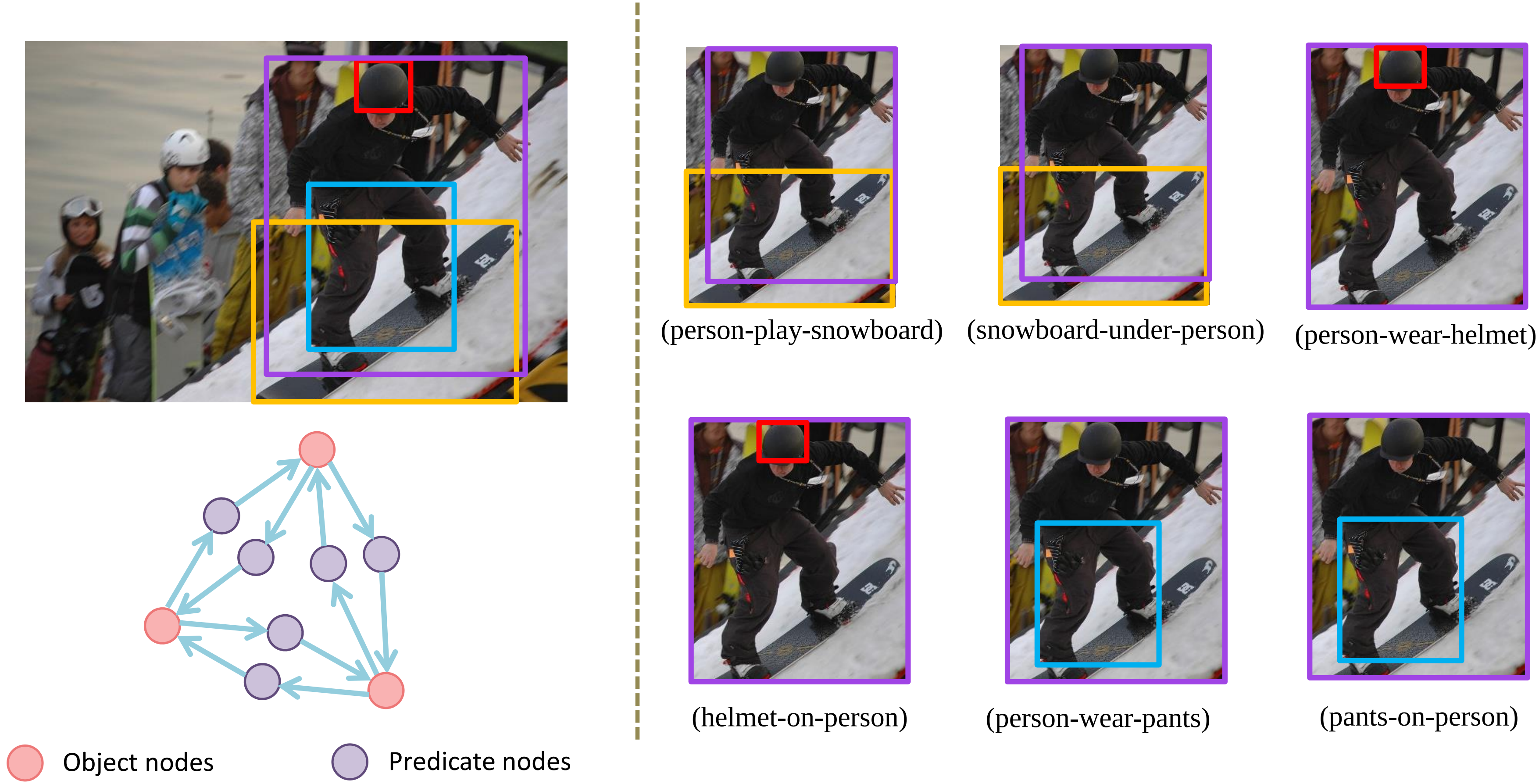
Top-1 region captioning results with detected objects and corresponding relationships are visualized.

2.45s/img

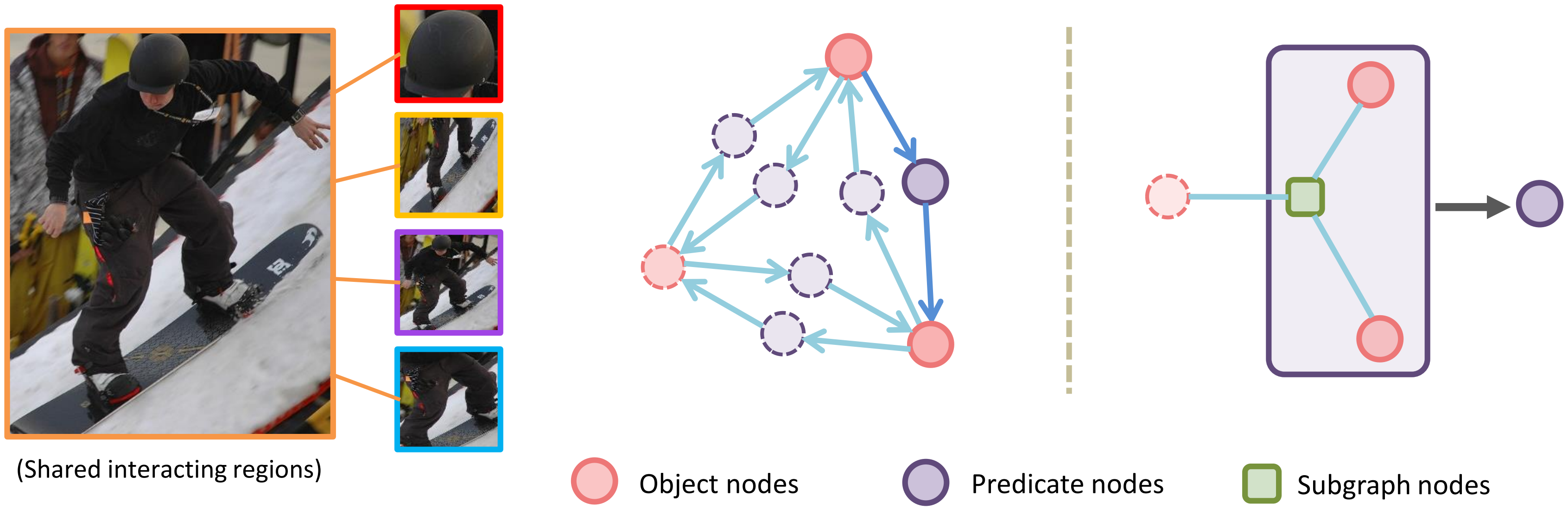
Slow inference speed due to large number of phrase proposals

Factorizable Net

An Efficient Subgraph-based Framework for Scene Graph Generation

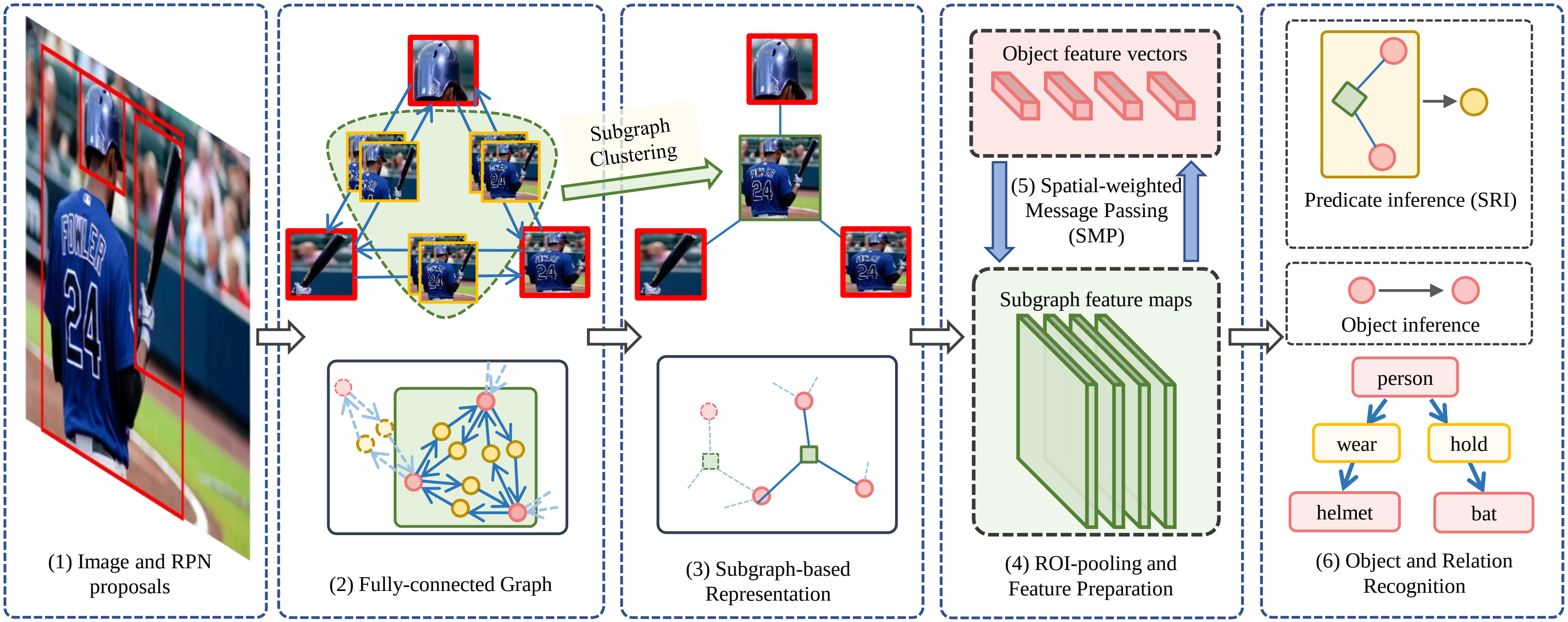


[8] Li, Yikang, et al. “Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation .” *ECCV 2018*.



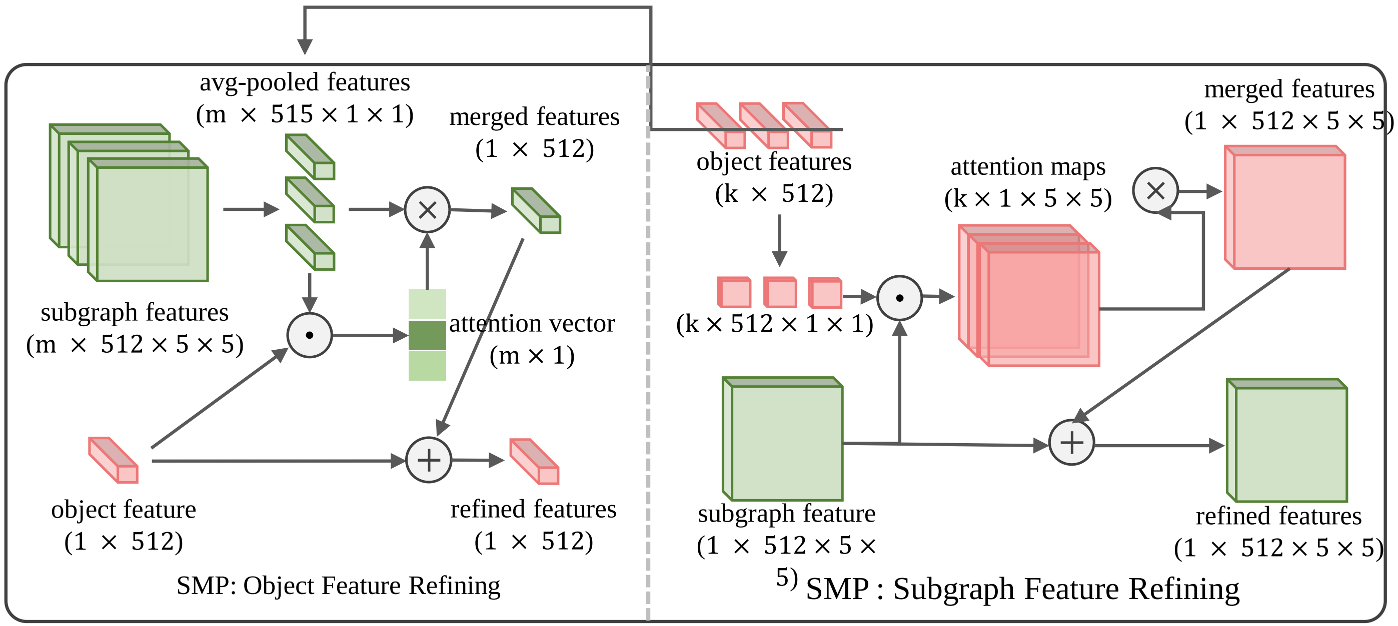
[8] Li, Yikang, et al. "Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation ." *ECCV 2018*.

Factorizable Net



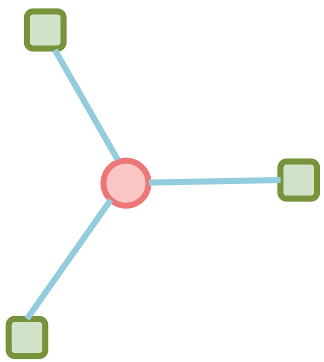
[8] Li, Yikang, et al. "Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation ." *ECCV 2018*.

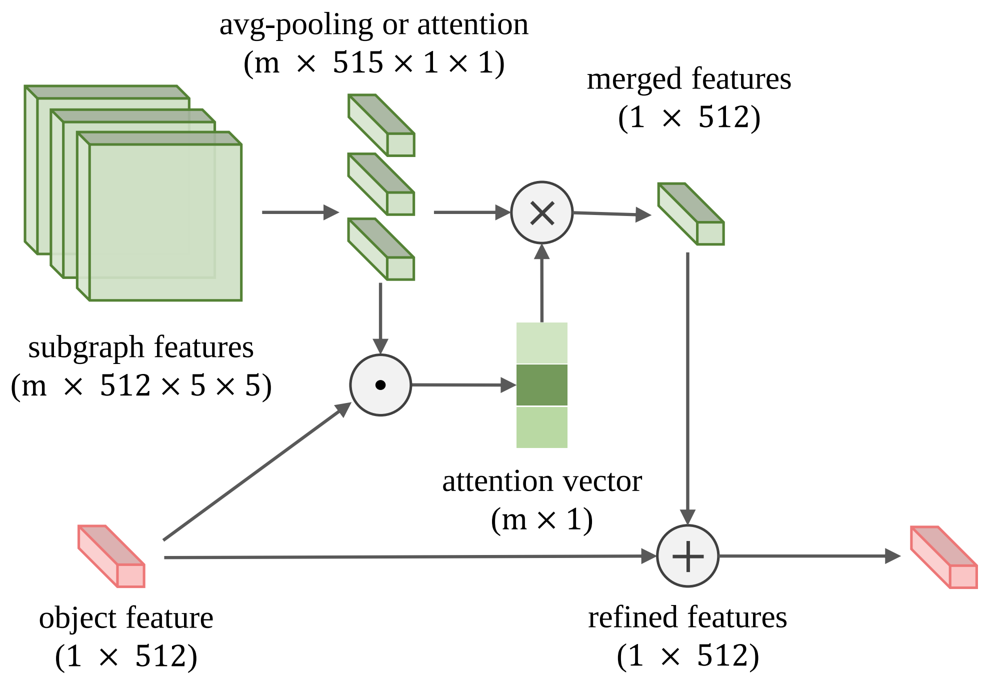
SMP: Spatial-weighted Message Passing



[8] Li, Yikang, et al. "Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation ." *ECCV 2018*.

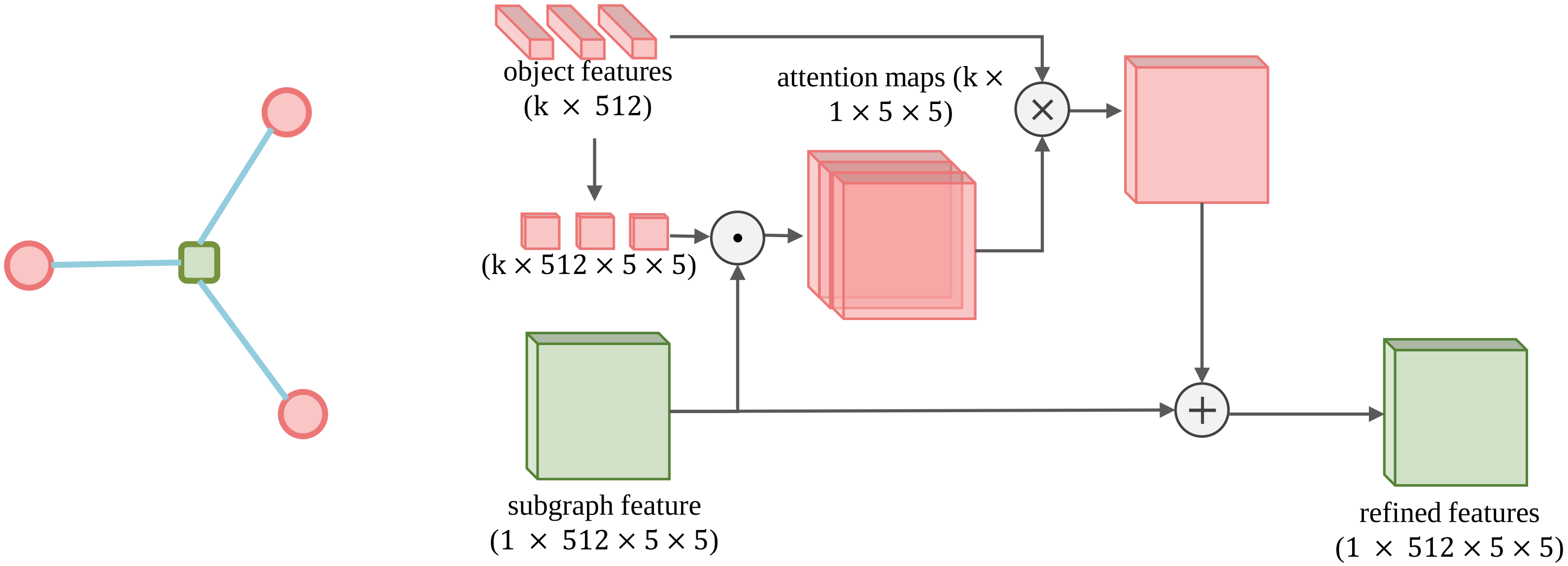
SMP: subgraph to object





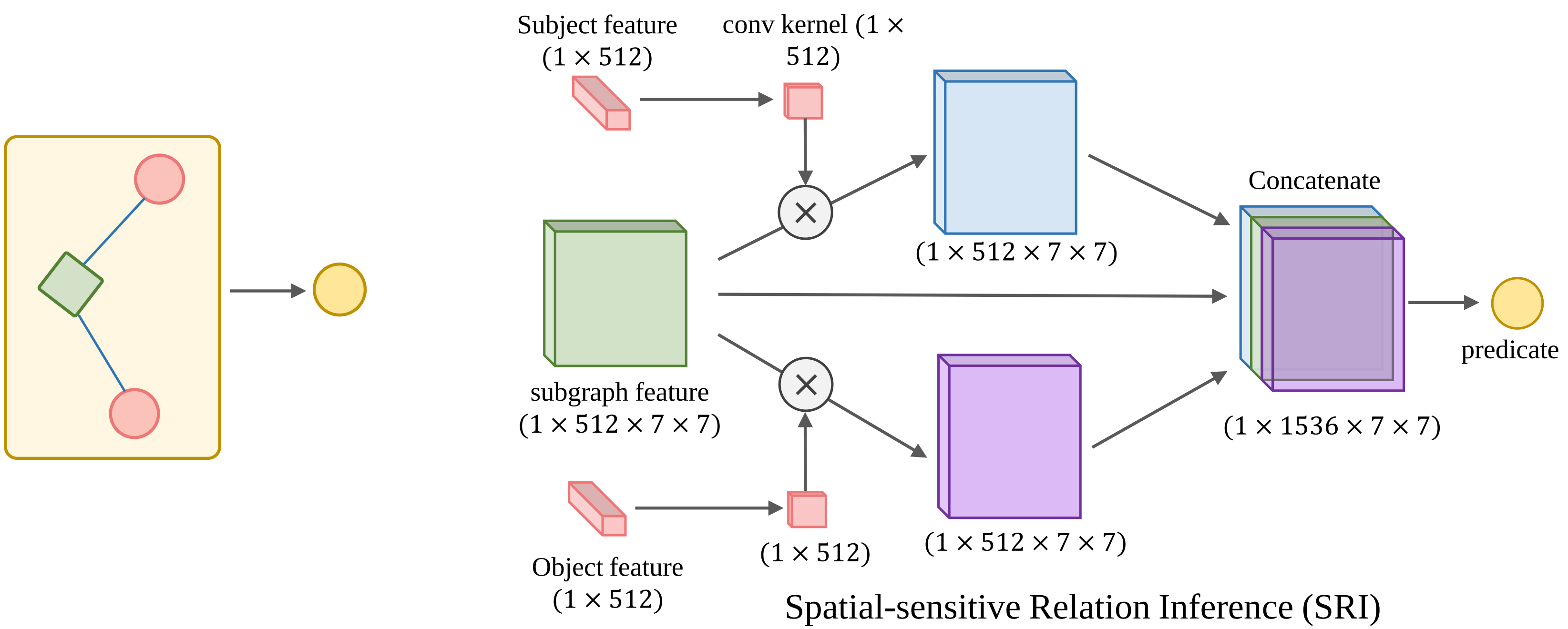
[8] Li, Yikang, et al. “Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation .” *ECCV 2018*.

SMP: object to subgraph



[8] Li, Yikang, et al. “Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation .” *ECCV 2018*.

SRI: Spatial-sensitive Relation Inference



[8] Li, Yikang, et al. "Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation ." *ECCV 2018*.

Comparison with Existing Methods

Dataset	Model	PhrDet		SGGen		Speed
		Rec@50	Rec@100	Rec@50	Rec@100	
VRD [1]	LP [1]	16.17	17.03	13.86	14.70	1.18*
	ViP-CNN [3]	22.78	27.91	17.32	20.01	0.78
	DR-Net [6]	19.93	23.45	17.73	20.88	2.83
	ILC [54]	16.89	20.70	15.08	18.37	2.70**
	Ours Full:1-SMP	25.90	30.52	18.16	21.04	**0.45**
	Ours Full:2-SMP	**26.03**	**30.77**	**18.32**	**21.20**	0.55
VG-MSDN [2, 4]	ISGG [5]	15.87	19.45	8.23	10.88	1.64
	MSDN [4]	19.95	24.93	10.72	14.22	3.56
	Ours-Full: 2-SMP	**23.34**	**28.53**	**13.75**	**16.81**	**0.55**
VG-DR-Net [2, 6]	DR-Net [6]	23.95	27.57	20.79	23.76	2.83
	Ours-Full: 2-SMP	**26.71**	**31.33**	**21.44**	**24.90**	**0.55**

* Only consider the post-processing time given the CNN features and object detection results. ** As reported in [54], it takes about 45 minutes to test 1000 images on single K80 GPU.

[8] Li, Yikang, et al. "Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation ." *ECCV 2018*.

Evaluation on Object Detection

Model	FRCNN-64 [55]	FRCNN-300 [55]	MSDN [4]	Ours-w/o-Rel	Ours
mean AP(%)	6.72	10.21	7.43	13.02	**15.70**

- **FRCNN-64:** Faster RCNN with 64 object proposals (experiment settings in [7])
- **FRCNN-300**: Faster RCNN with 300 object proposals (experiment settings in [9])
- **MSDN**: Our proposed Multilevel Scene Description Network in [4]
- **Ours-w/o-Rel**: Adopt the subgraph-based framework but without relationship supervision
- **Ours**: Our Factorizable Net with 1 SMP (model 5 in Ablation Study)

[4] Li, Yikang, et al. “Scene graph generation from objects, phrases and region captions.” *ICCV 2017.*

[8] Li, Yikang, et al. “Factorazable Net: An Efficient Subgraph-based framework for Scene Graph Generation .” *2018*.

[9] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *NIPS 2015*.

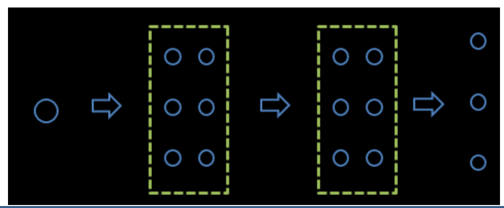
Does structure only exist for specific task?

Outline

Introduction

Structured deep learning

Back-bone model design



FishNet
(NIPS 18)

Optical guided feature
(CVPR18)

Current CNN Structures

Called U-Net, Hourglass, or Conv-deconv

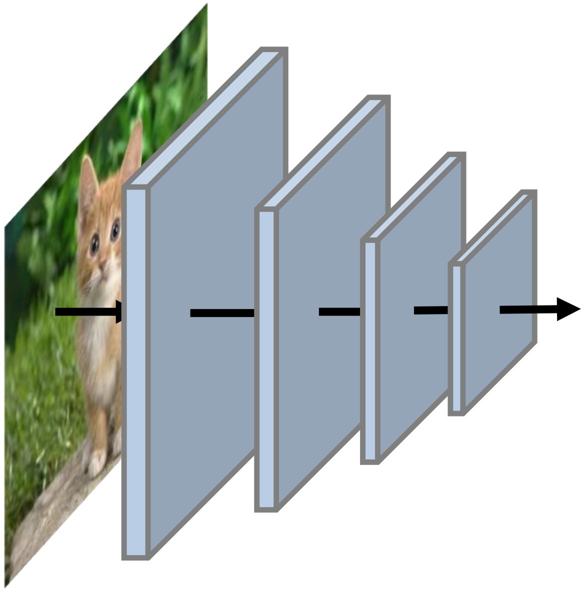
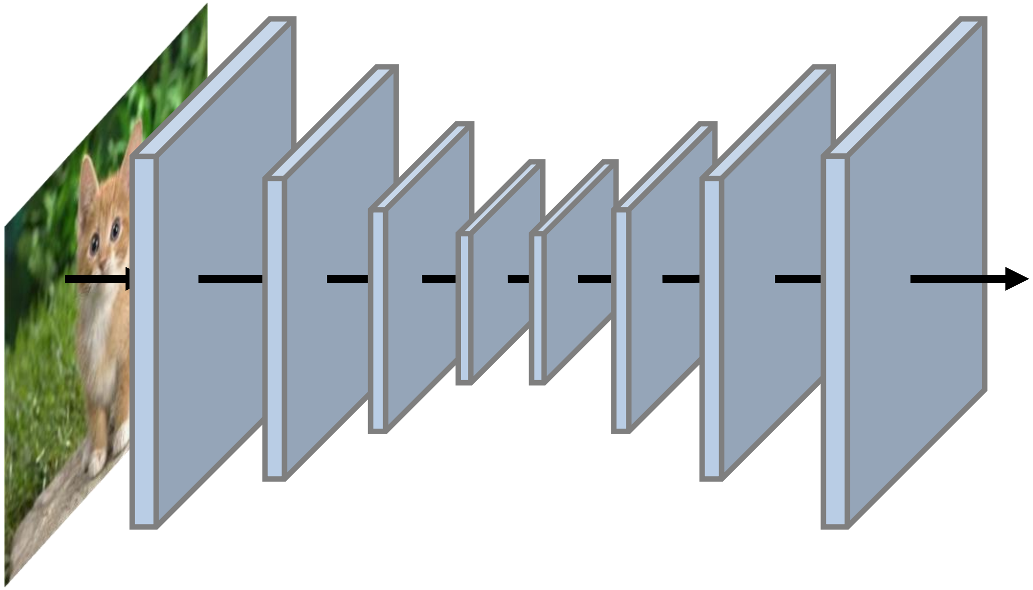


Image Classification:
Summarize high-level semantic information of the whole image.



Detection/Segmentation:
High-level semantic meaning with high spatial resolution

Architectures designed for different granularities are **DIVERGING**

Unify the advantages of networks for pixel-level, region-level, and image-level tasks

Hourglass for Classification

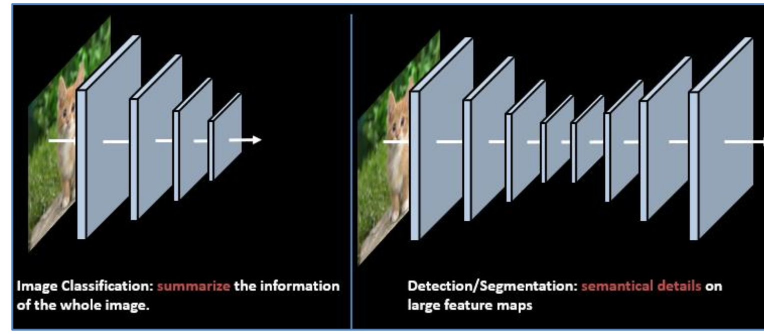
Features with high-level semantics and high resolution is good

Directly applying hourglass for classification?

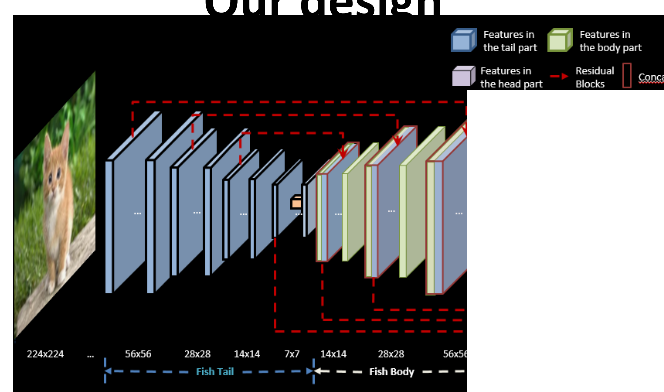
Poor performance.

So what is the **problem**?

- **Different tasks require different resolutions of feature**
- **Down sample high-level features with high resolution**

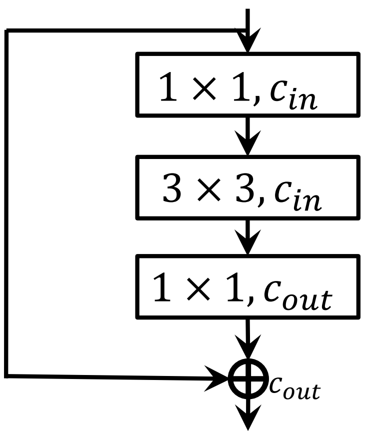


Our design

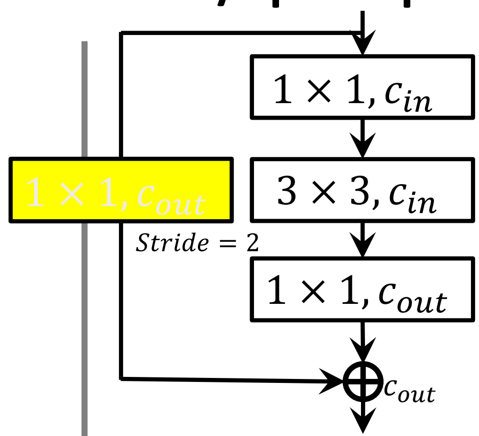


Hourglass for Classification

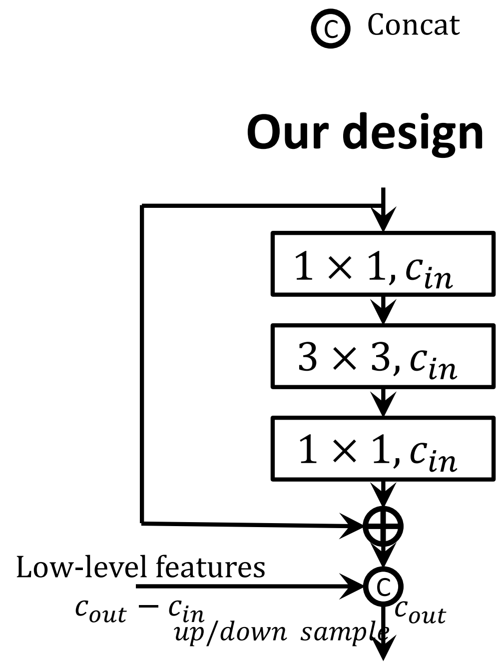
Normal Res-Block



Res-Block for down/up sampling



Our design



- **Different tasks require different resolutions of feature**
- **Hourglass may bring more isolated convolutions than ResNet**

The 1×1 convolution layer in yellow indicates the Isolated convolution.

Observation and design

Our observation

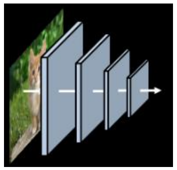
1. Diverged structures for tasks requiring different resolutions.
2. Isolated Conv blocks the direct back-propagation
3. Features with different depths are not fully explored, or mixed but not preserved

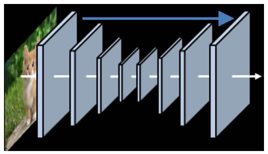
Solution

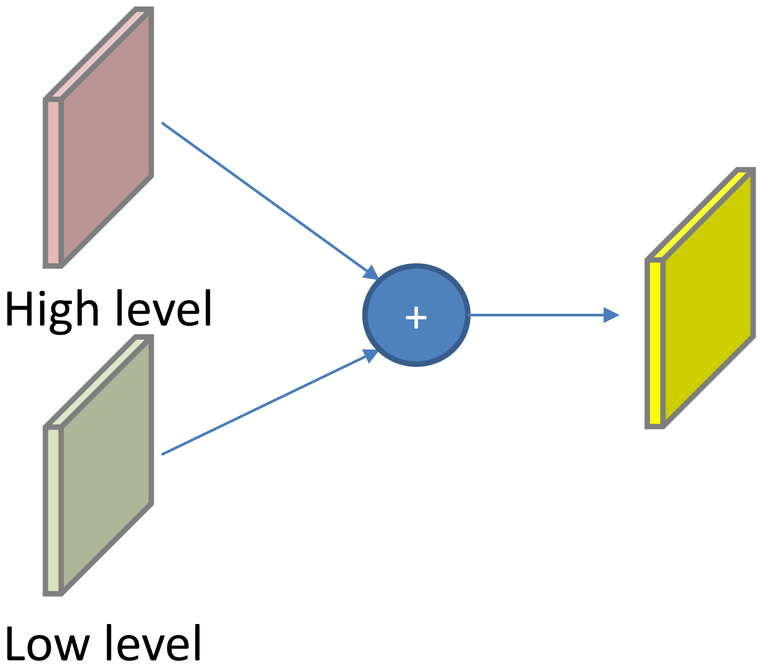
1. Unify the advantages of networks for pixel-level, region-level, and image-level tasks.
2. Design a network that does not need isolated convolution
3. Features from varying depths are **preserved and refined** from each other.

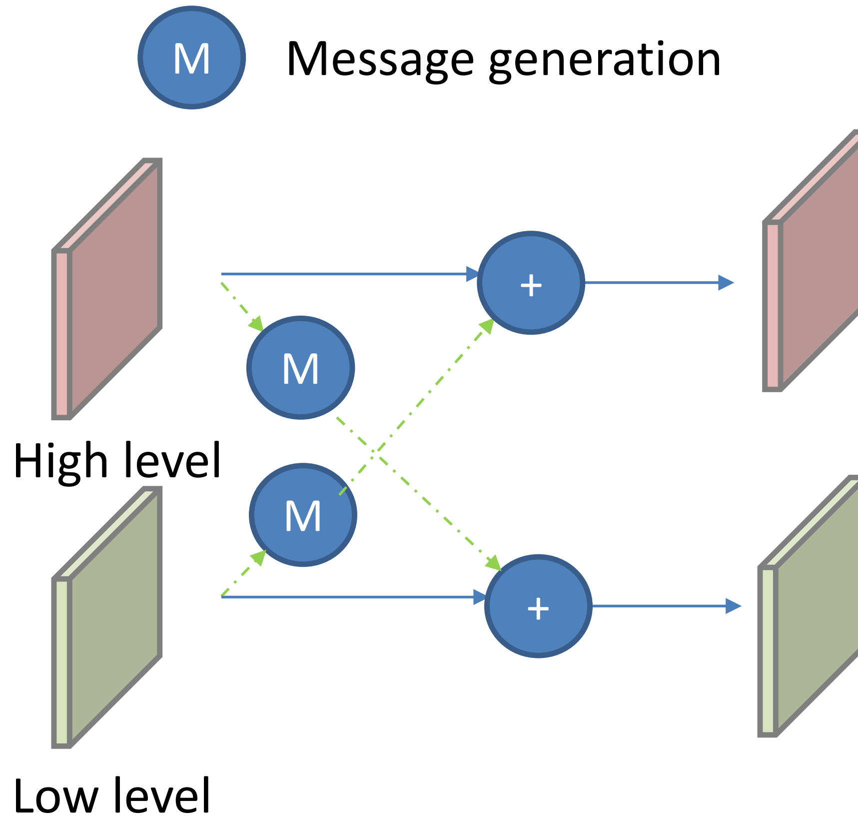
Bharath Hariharan, et al. "Hypercolumns for object segmentation and fine-grained localization." *CVPR'15.*
Newell, Alejandro, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation." *ECCV'*16.

Difference between mix and preserve and refine





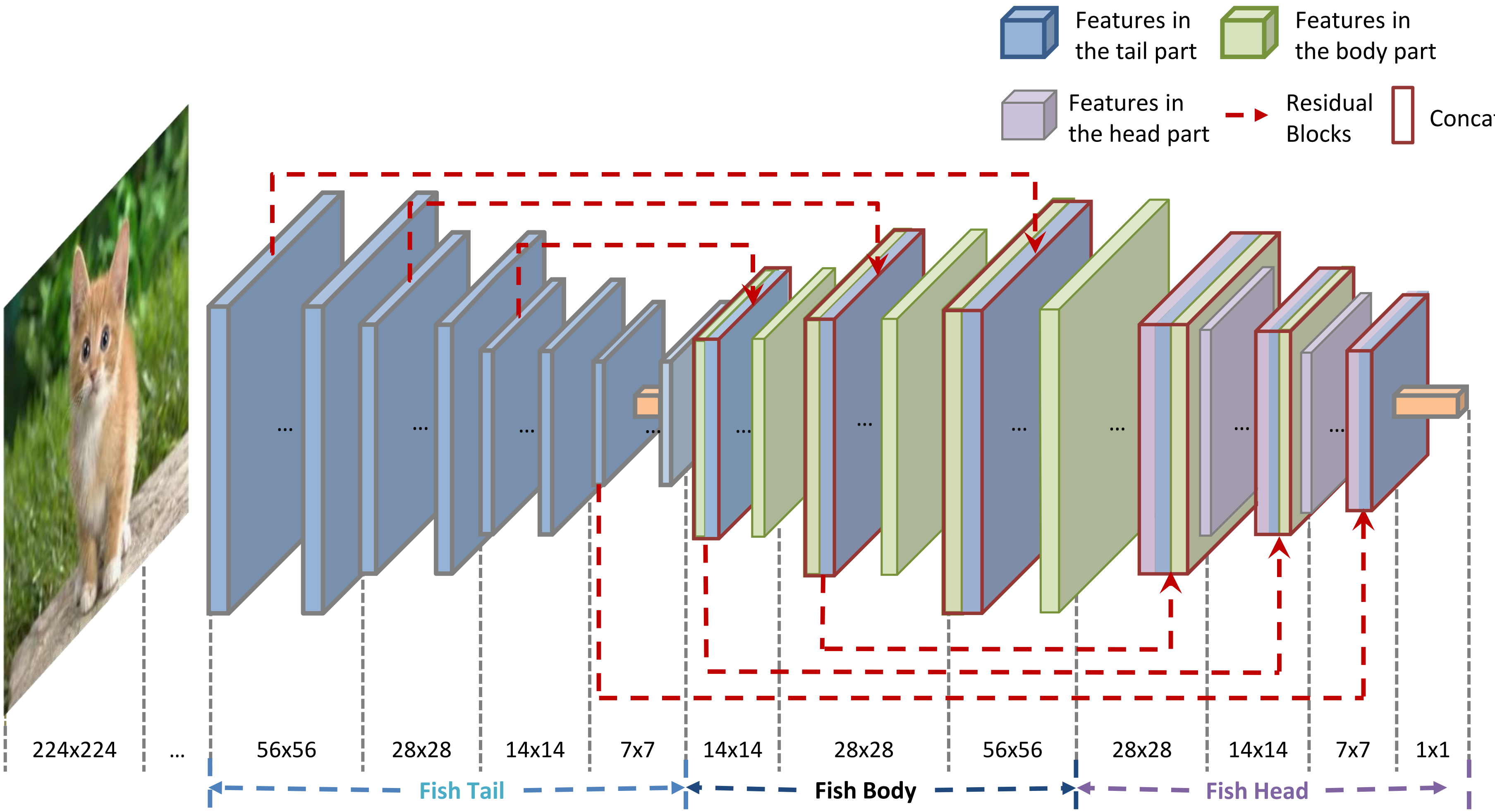




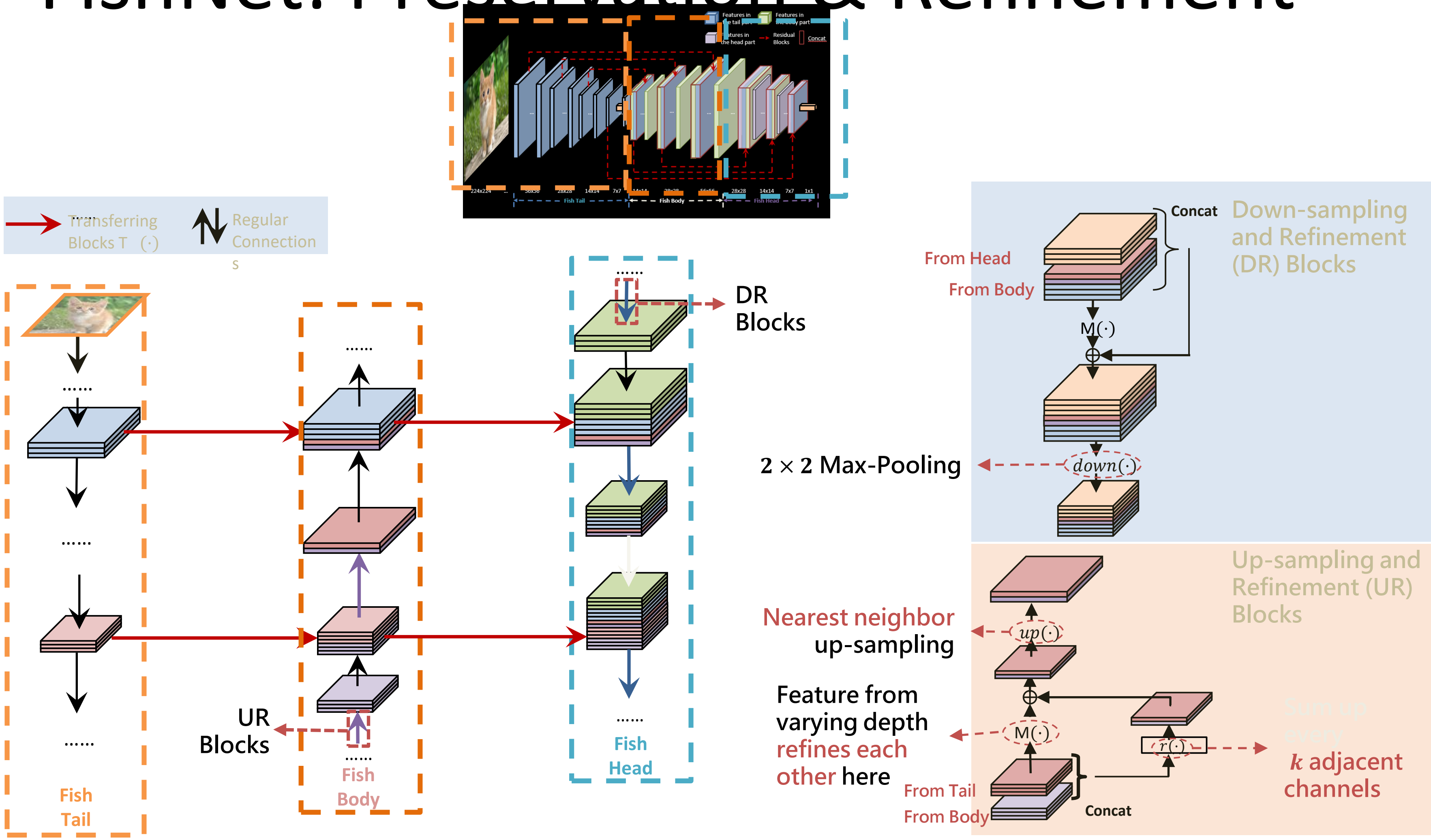
Mixed features

Preserve and refine

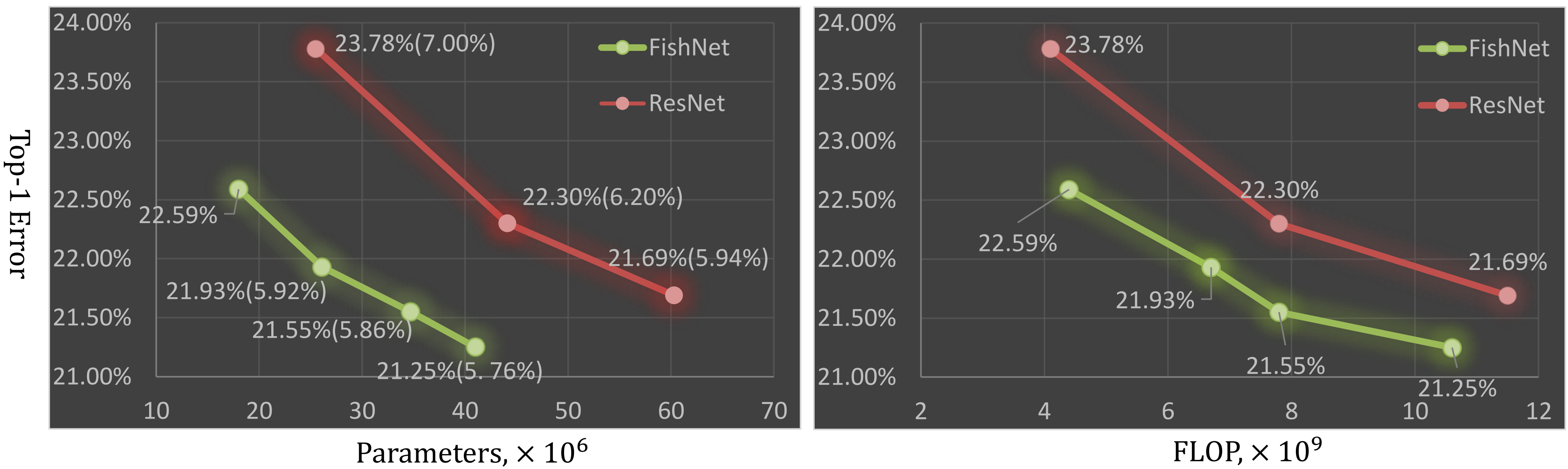
FishNet: Overview



FishNet: Preservation & Refinement

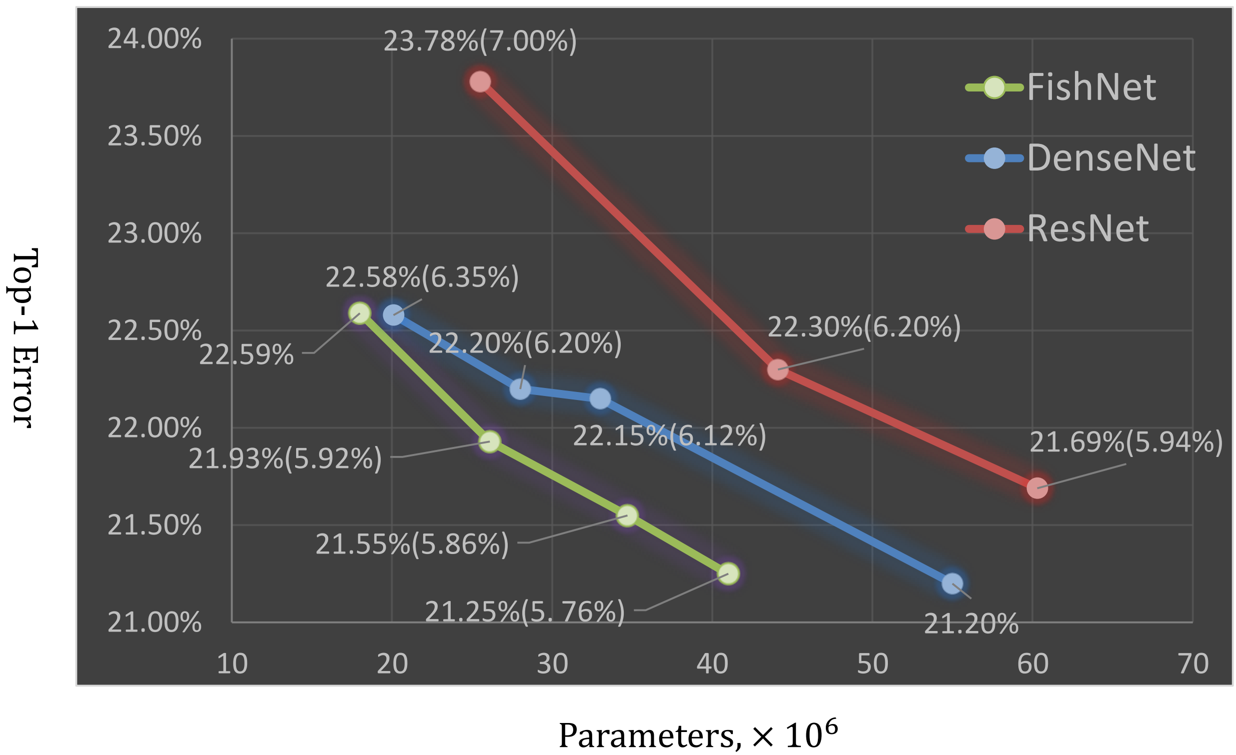


FishNet: Performance-ImageNet



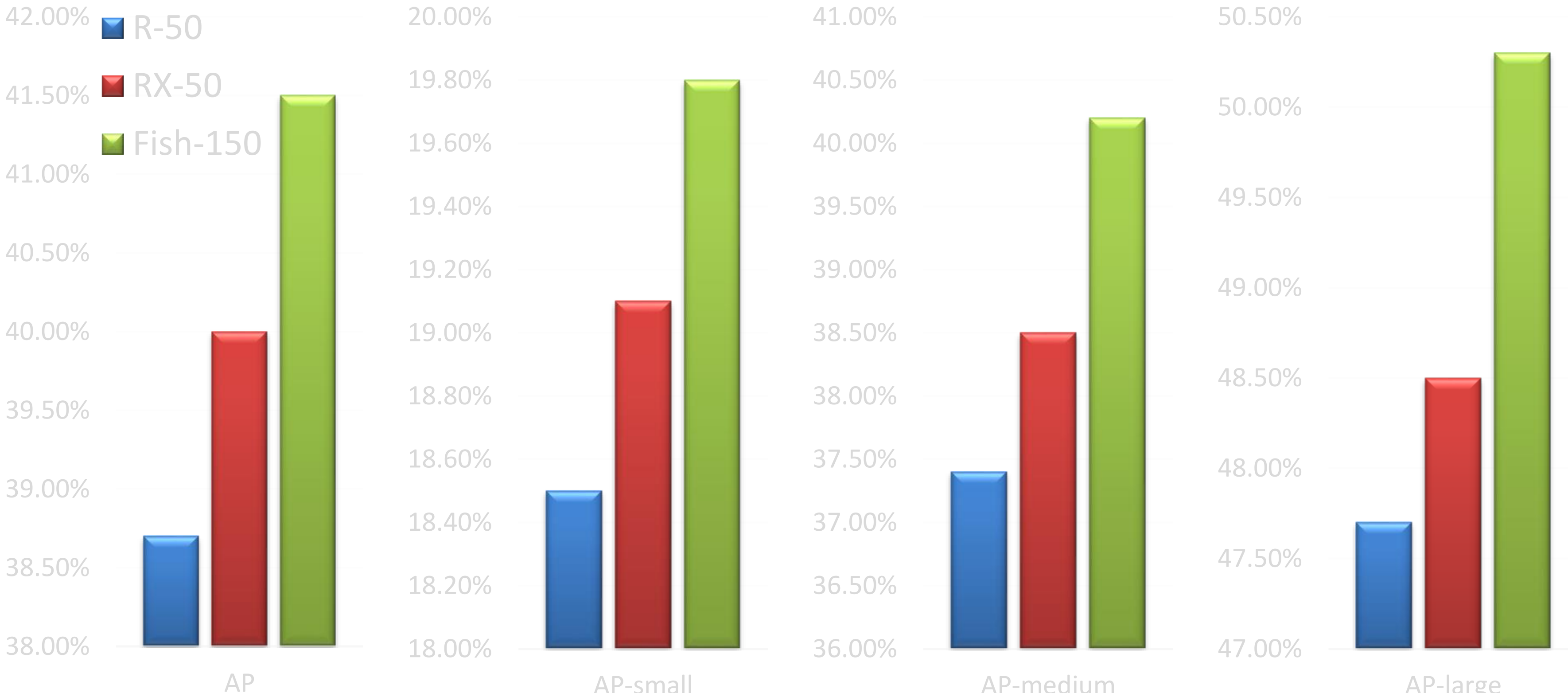
Code
https://github.com/kevin-ssy/FishNet

FishNet: Performance-ImageNet



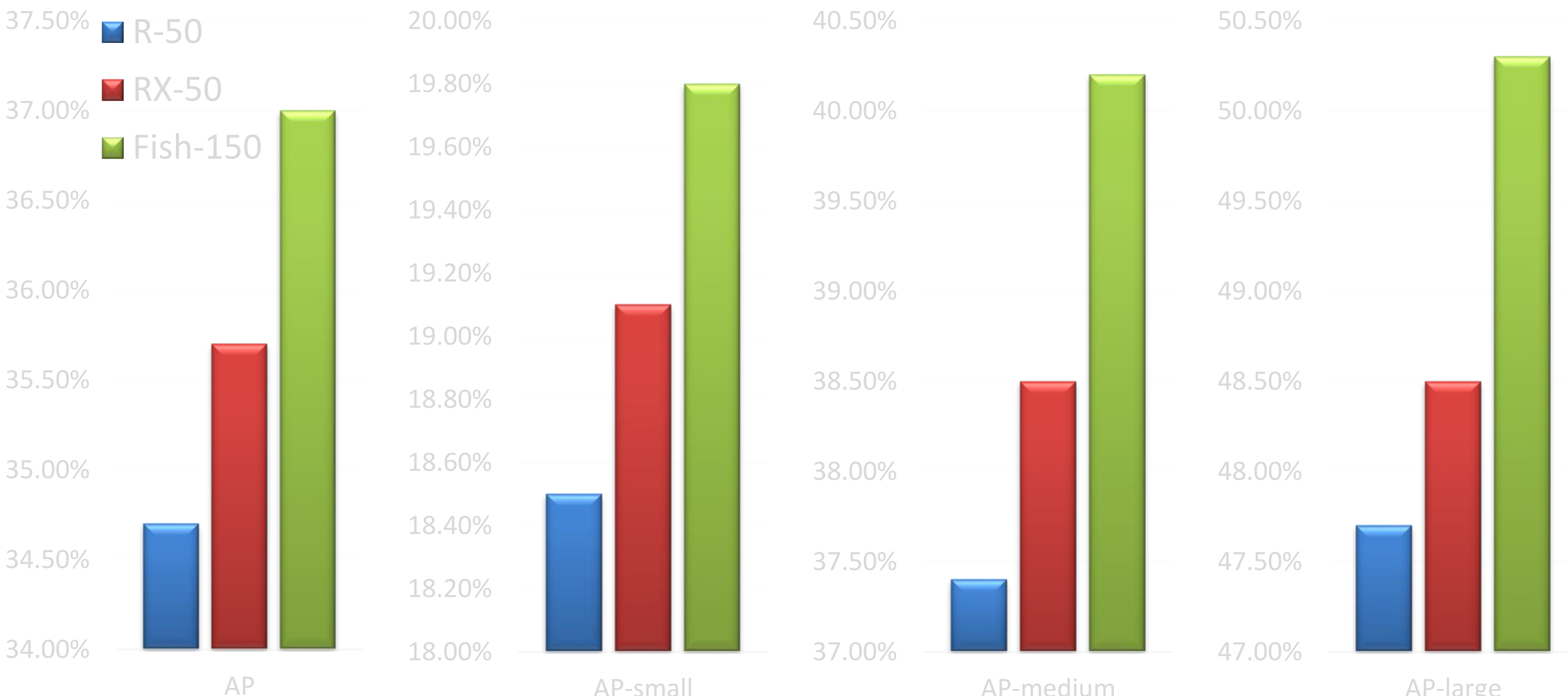
Code
https://github.com/kevin-ssy/FishNet

FishNet: Performance on COCO Detection



Code
https://github.com/kevin-ssy/FishNet

FishNet: Performance on COCO Instance Segmentation

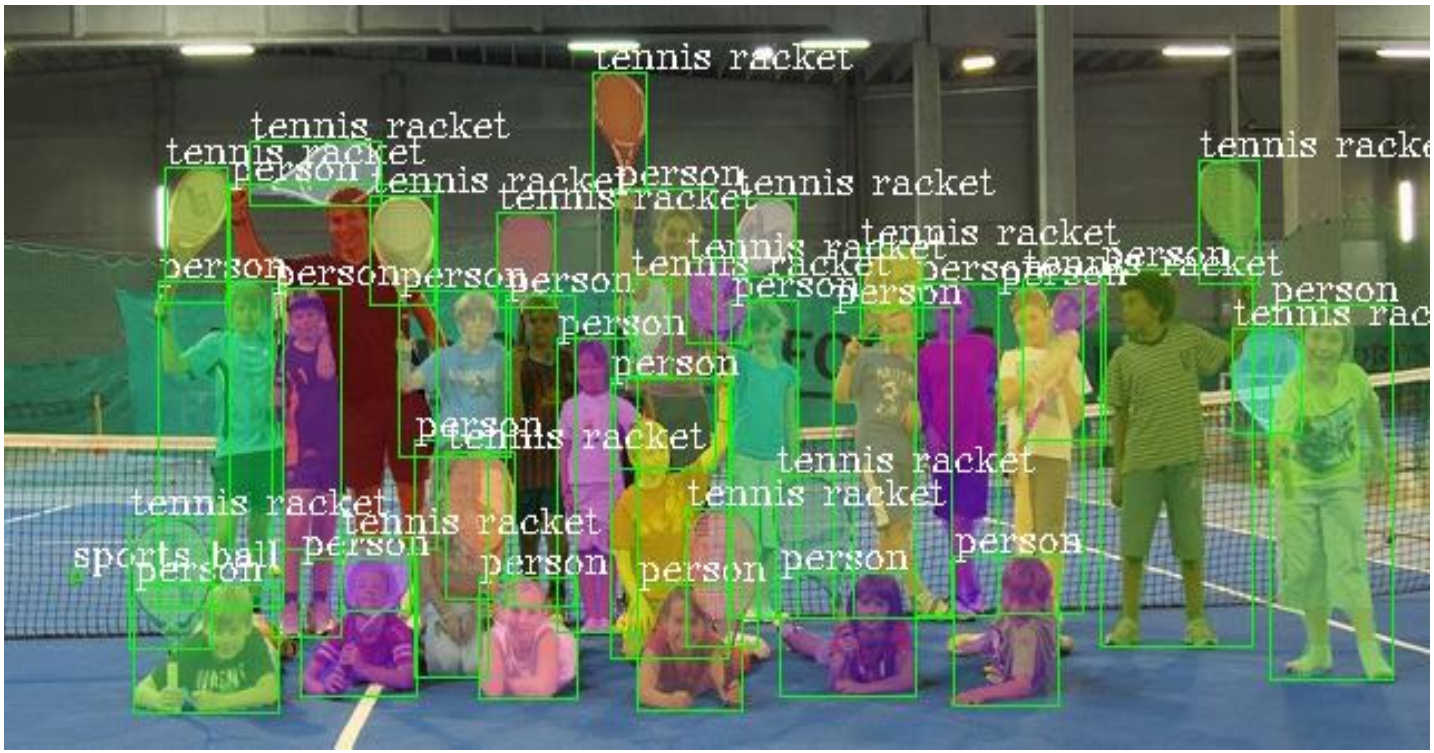


Code
https://github.com/kevin-ssy/FishNet

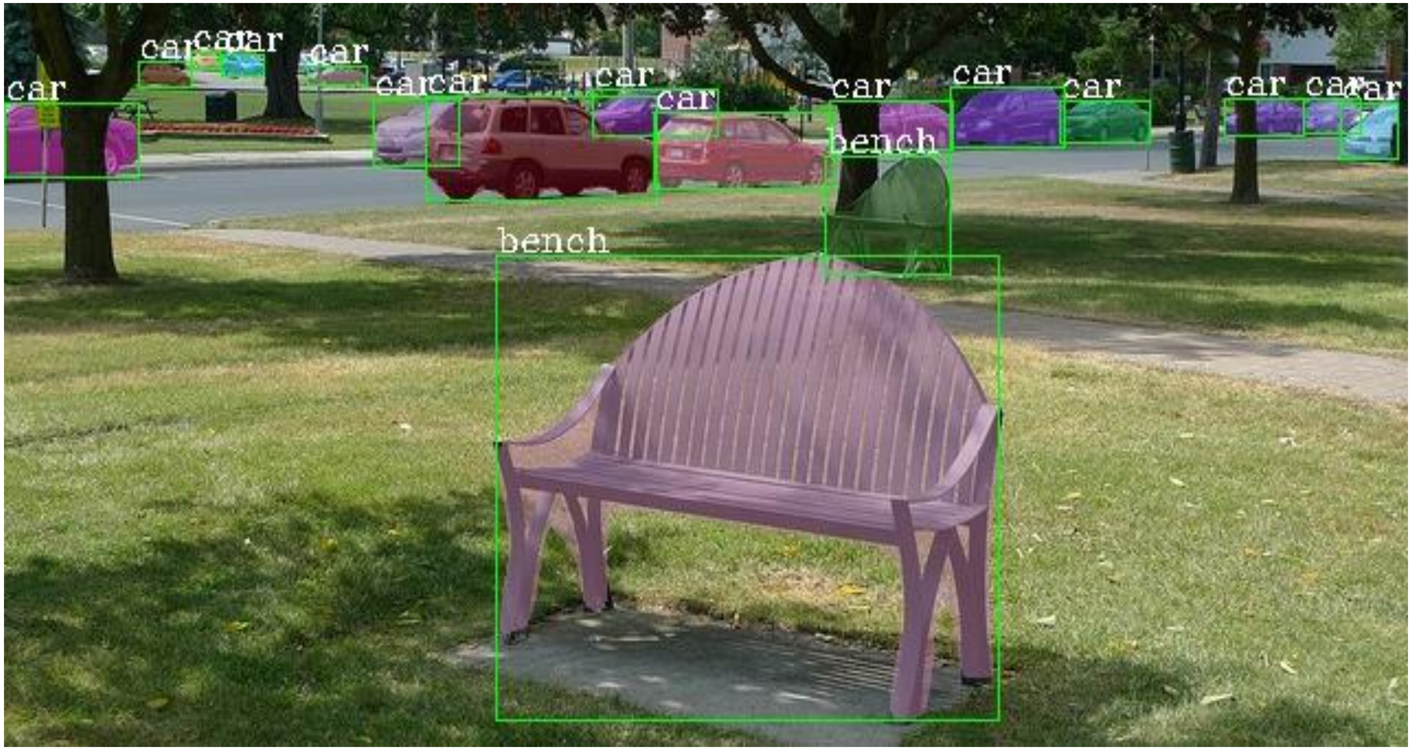
Winning COCO 2018 Instance Segmentation Task

	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}	date
MMDet	0.486	0.730	0.530	0.339	0.520	0.602	0.368	0.593	0.632	0.464	0.665	0.777	2018-08-18
Megvii (Face++)	0.485	0.737	0.532	0.298	0.507	0.641	0.369	0.594	0.630	0.474	0.659	0.767	2018-08-18
FirstShot	0.463	0.681	0.508	0.258	0.483	0.636	0.359	0.580	0.622	0.445	0.655	0.776	2018-08-17

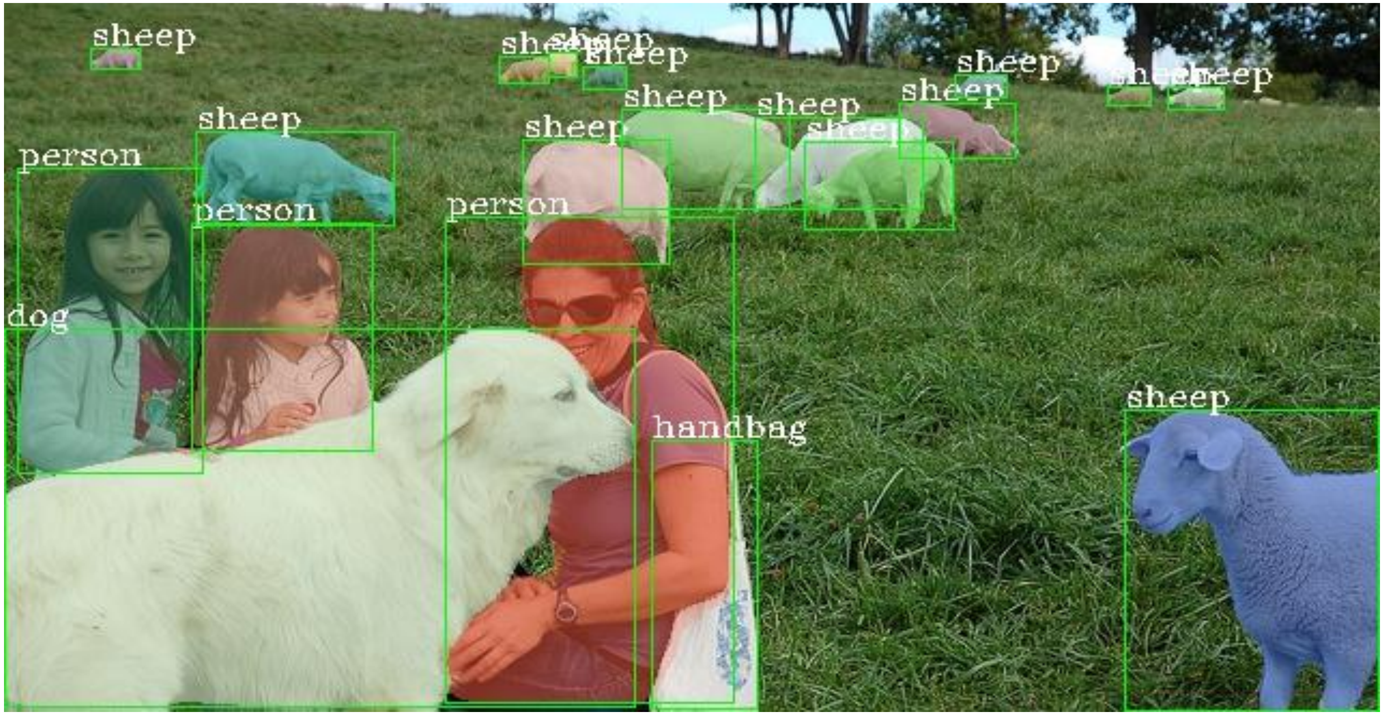
Visualization



Visualization

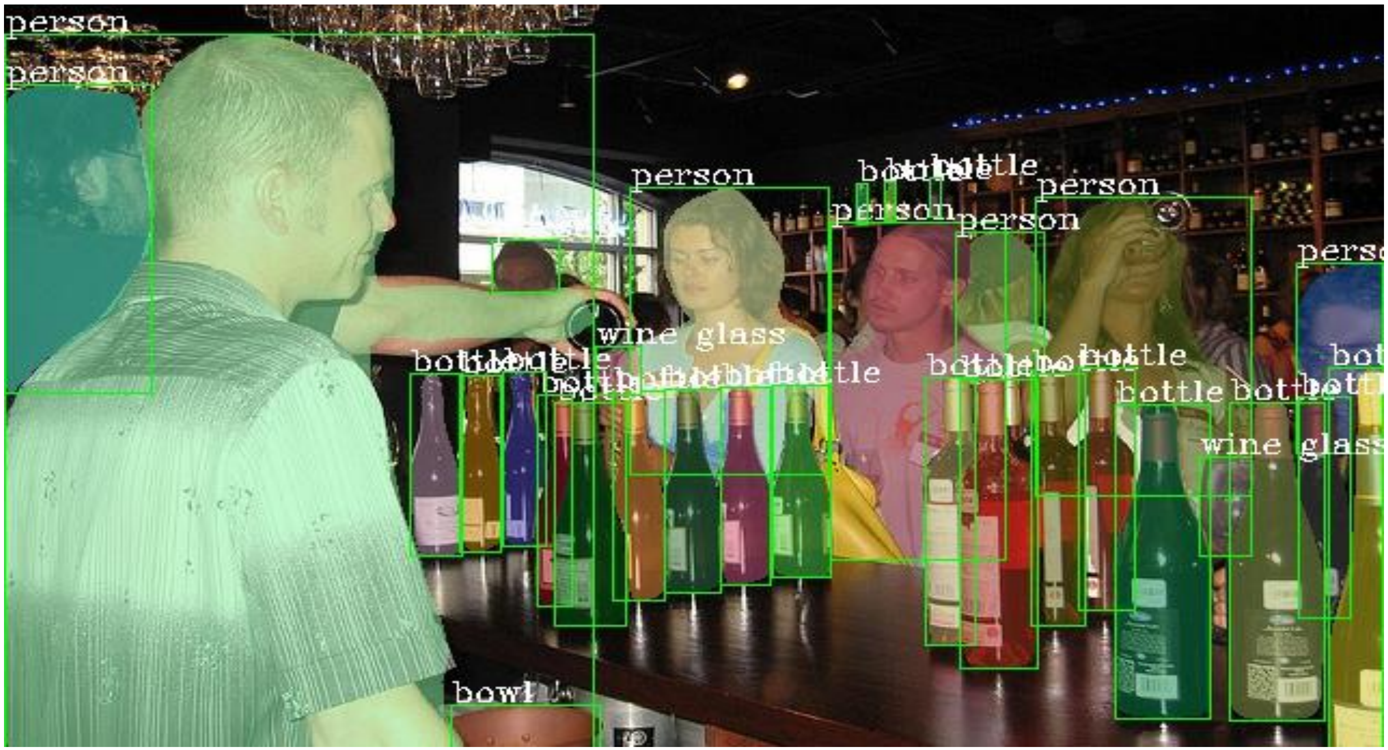


Visualization



Visualization

Visualization



Codebase

- Comprehensive
  - [x] RPN
  - [x] Fast/Faster R-CNN
  - [x] Mask R-CNN
  - [x] FPN
  - [x] Cascade R-CNN
  - [x] RetinaNet
  - [ ] More … …

- **High performance**
  - [x] Better performance
  - [x] Optimized memory consumption
  - [x] Faster speed

- **Handy to develop**
  - [x] Written with PyTorch
  - [x] Modular design

GitHub: mmdet

FishNet: Advantages

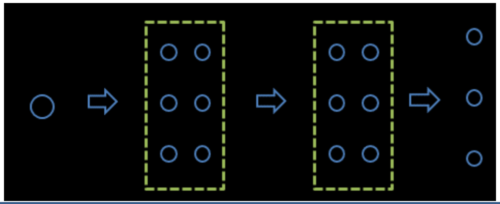
- Better gradient flow to shallow layers
- High-resolution features contain rich low-level and high-level semantics
- Build up correlation among features with different semantic information
  - They are preserved and refined from each other

Outline

Introduction

Structured deep learning

Back-bone model design



FishNet
(NIPS 18)

Optical guided feature
(CVPR18)

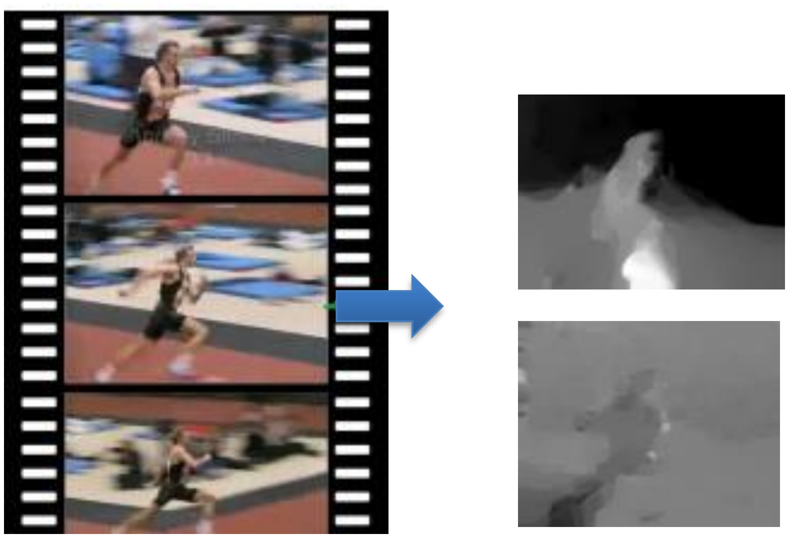
Action Recognition

- Recognize action from videos

Optical flow in Action Recognition

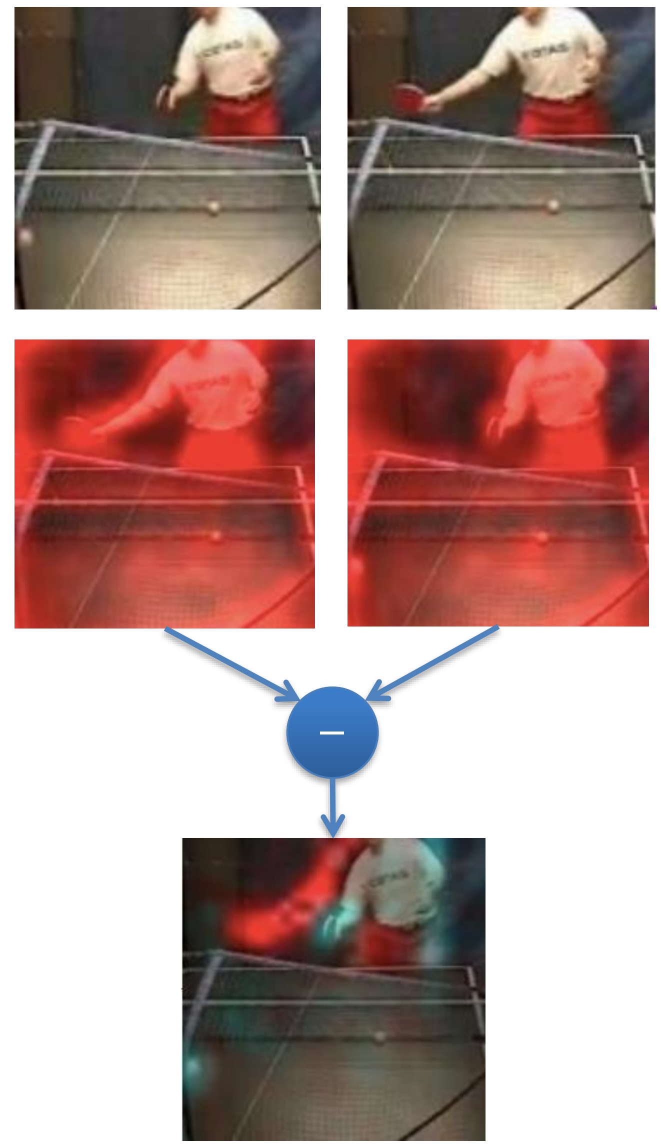
- Motion is the important information
- Optical flow
  - Effective
  - Time consuming

We need a better motion representation



Modality	Acc.
RGB	85.5%
RGB+Optical Flow	94.0%

Optical flow guided feature



Optical flow guided feature

Optical flow:

$$I(x,y,t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

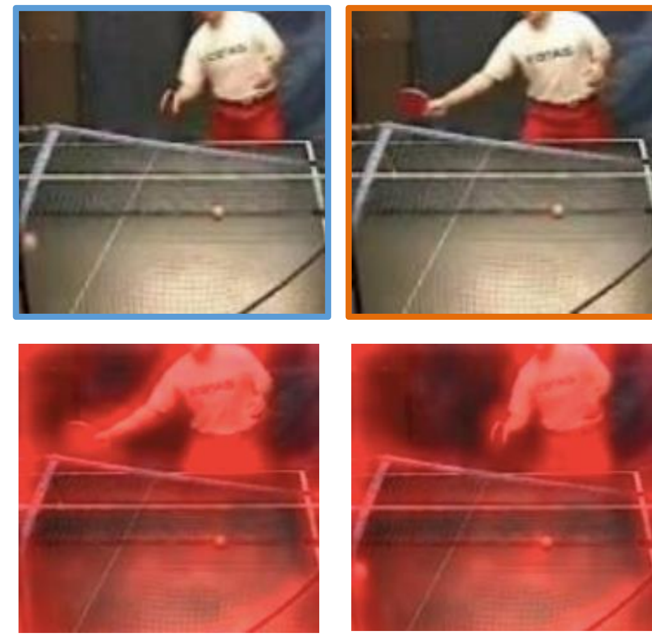
$$\frac{\partial I(x,y,t)}{\partial x} v_x + \frac{\partial I(x,y,t)}{\partial y} v_y + \frac{\partial I(x,y,t)}{\partial t} = 0$$

$\{v_x, v_y\}$ = optical flow

Intuitive Inspiration

Coefficient for optical flow:

$$\{\frac{\partial I(x,y,t)}{\partial x}, \frac{\partial I(x,y,t)}{\partial y}, \frac{\partial I(x,y,t)}{\partial t}\}$$



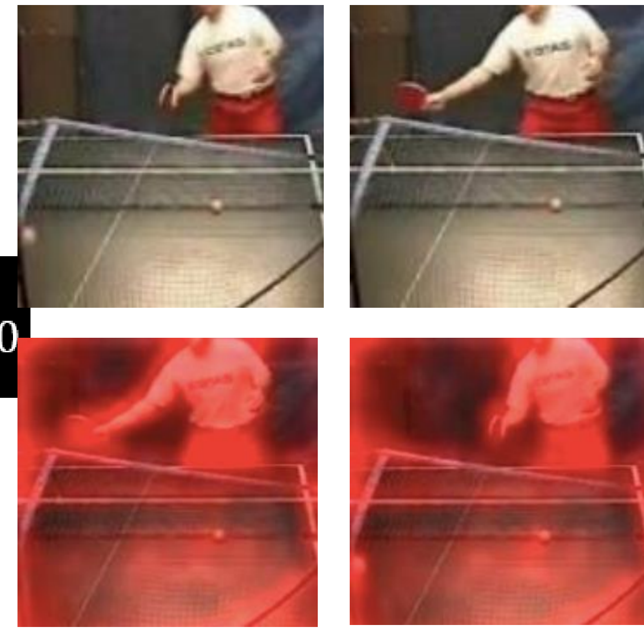
Optical flow guided feature

Feature flow:

$$f(I(x,y,t))=f(I(x+\Delta x,y+\Delta y,t+\Delta t))$$

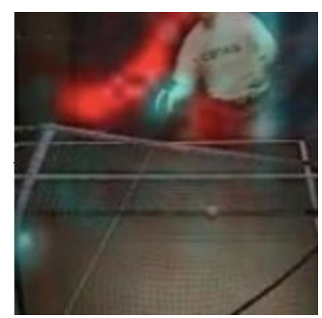
$$\frac{\partial f(I(x,y,t))}{\partial x}\tilde{v}_x+\frac{\partial f(I(x,y,t))}{\partial y}\tilde{v}_y+\frac{\partial f(I(x,y,t))}{\partial t}=0$$

$\{\tilde{v}_x,\tilde{v}_y\}$ = feature flow

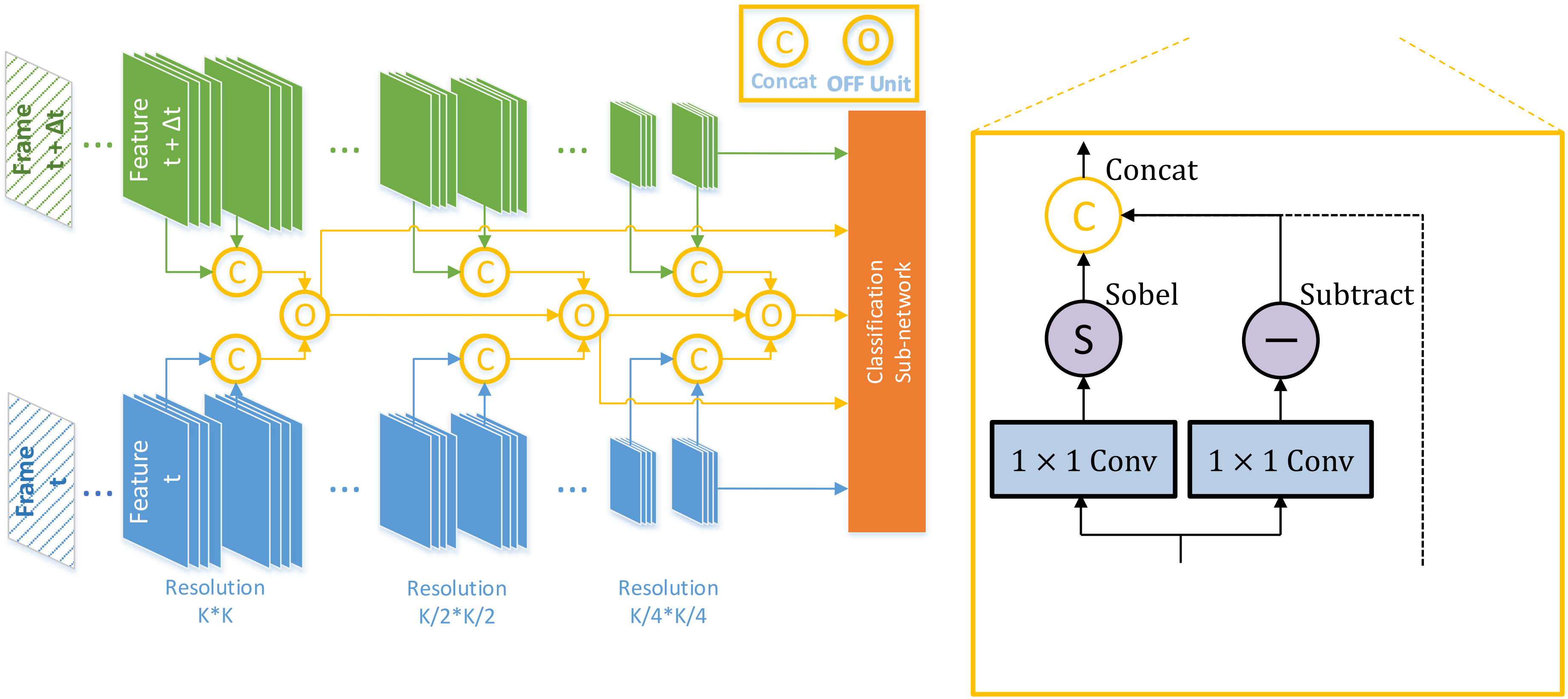


Optical flow guided feature (OFF):

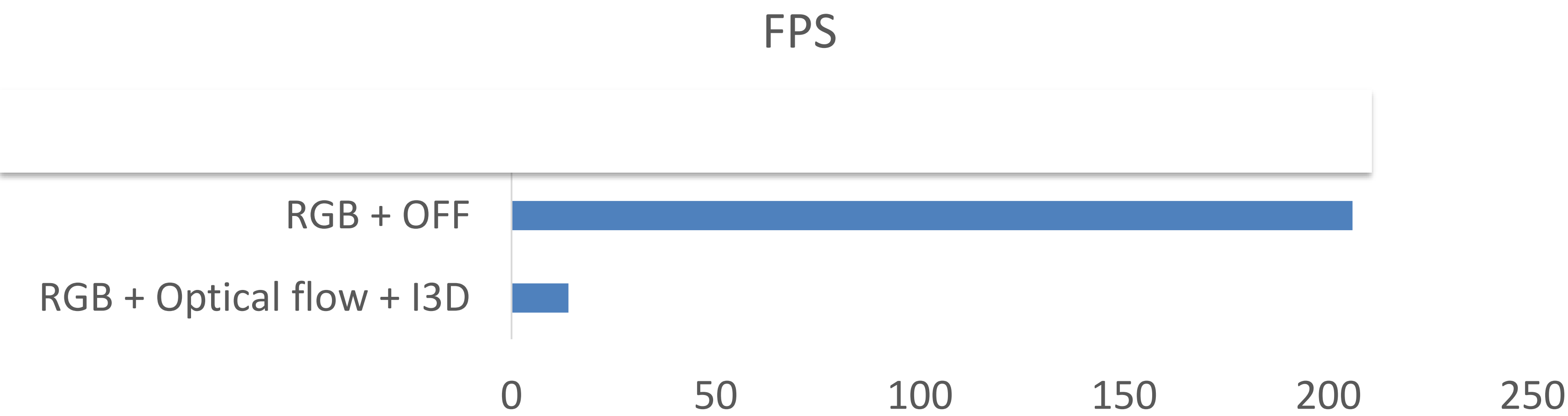
$$\left\{\frac{\partial f(I(x,y,t);w)}{\partial x},\frac{\partial f(I(x,y,t);w)}{\partial y},\frac{\partial f(I(x,y,t);w)}{\partial t}\right\}$$

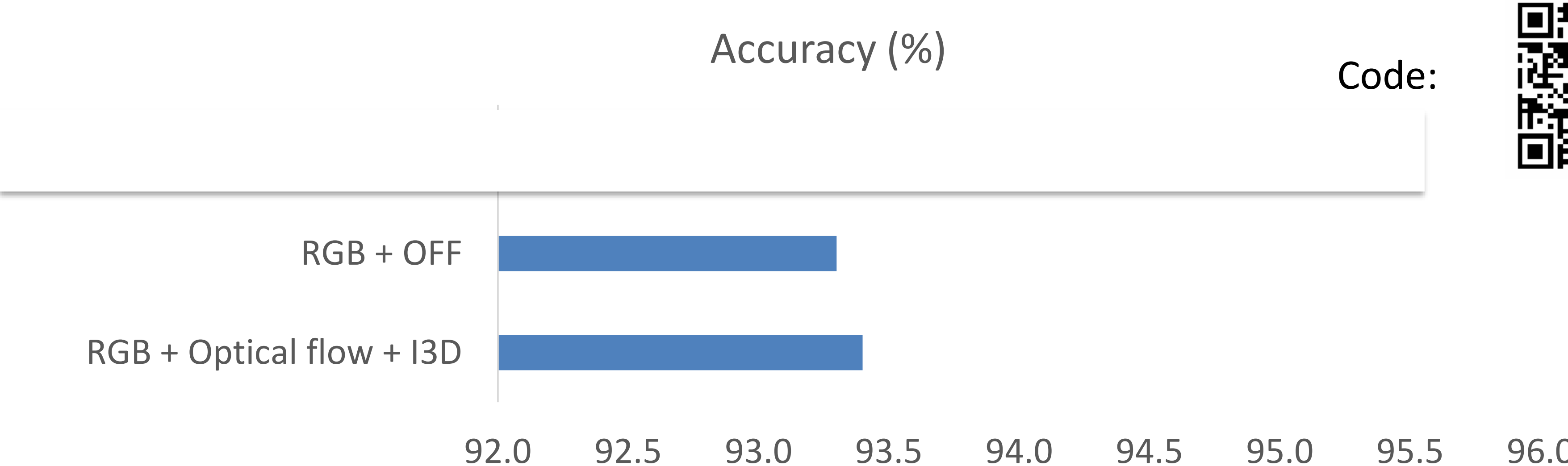


Optical flow guided feature



Optical Flow Guided Feature (OFF): Experimental results



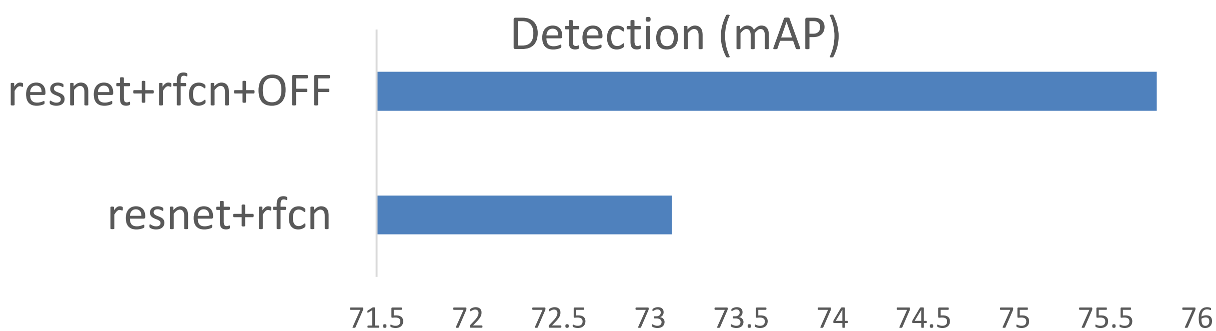


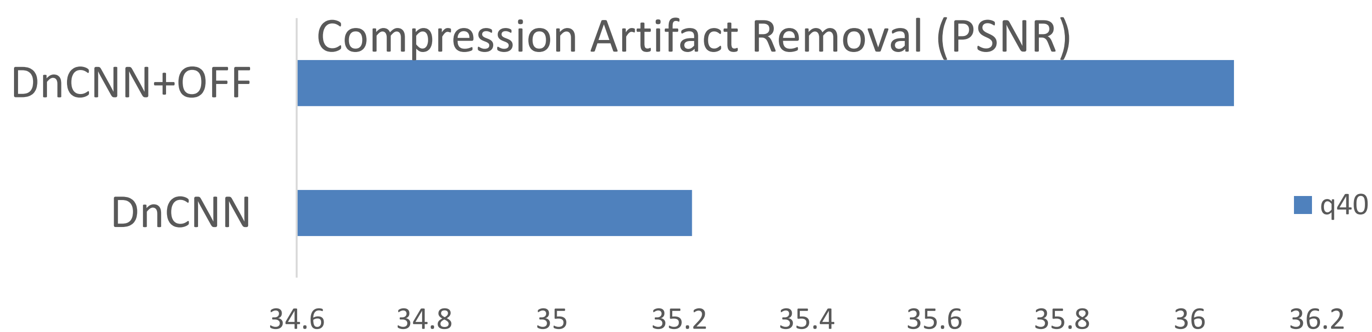
Code:

1. OFF with only RGB inputs is **comparable** with the other state-of-the-art methods using optical flow as input.

Not only for action recognition

- Also effective for
  - Video object detection
  - Video compression artifact removal



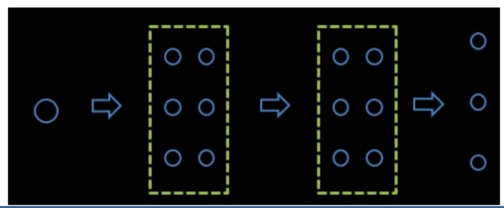


Outline

Introduction

Structured deep learning

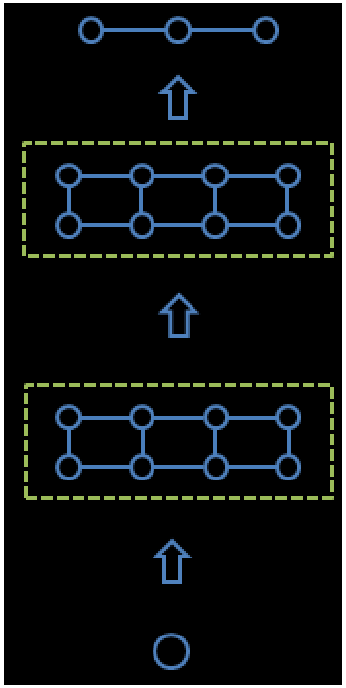
Back-bone model design



FishNet
(NIPS 18)

Optical guided feature
(CVPR18)

Take home message



- Structured deep learning is
  - effective
  - for output, features
  - from observation
- End-to-end joint training bridges the gap between structure modeling and feature learning

Thank you!