MEGVII 旷视

Object Detection in Recent 3 Years

Beyond RetinaNet and Mask R-CNN

Gang Yu

旷视研究院

Schedule of Tutorial

- Lecture 1: Beyond RetinaNet and Mask R-CNN (Gang Yu)
- Lecture 2: AutoML for Object Detection (Xiangyu Zhang)
- Lecture 3: Finegrained Visual Analysis (Xiu-shen Wei)

Outline

- Introduction to Object Detection
- Modern Object detectors
  - One Stage detector vs Two-stage detector
- Challenges
  - Backbone
  - Head
  - Pretraining
  - Scale
  - Batch Size
  - Crowd
  - NAS
  - Fine-Grained
- Conclusion

Outline

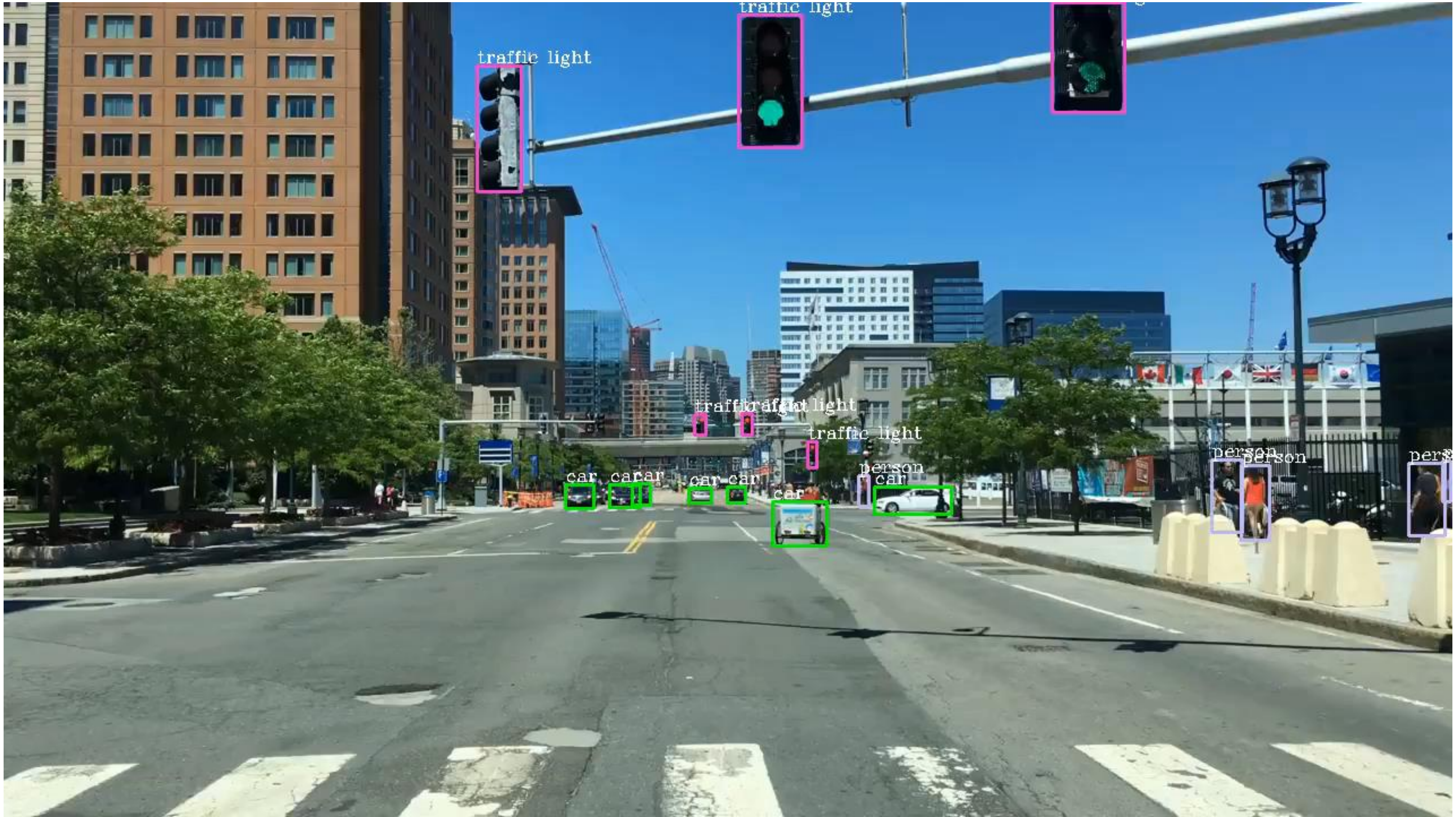
MEGVII 旷视

- Introduction to Object Detection
- Modern Object detectors
  - One Stage detector vs Two-stage detector
- Challenges
  - Backbone
  - Head
  - Pretraining
  - Scale
  - Batch Size
  - Crowd
  - NAS
  - Fine-Grained
- Conclusion

What is object detection?

MEGVII 旷视

What is object detection?

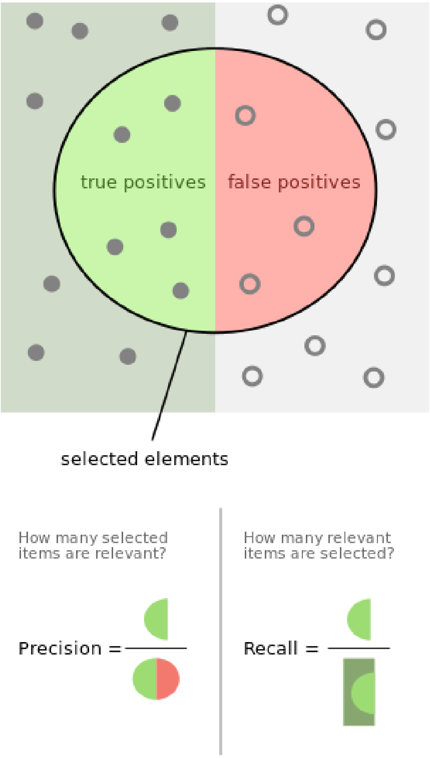


Detection - Evaluation Criteria

Average Precision (AP) and mAP

Precision and recall are single-value metrics based on the whole list of documents returned by the system. For systems that return a ranked sequence of documents, it is desirable to also consider the order in which the returned documents are presented. By computing a precision and recall at every position in the ranked sequence of documents, one can plot a precision-recall curve, plotting precision $p(r)$ as a function of recall r. Average precision computes the average value of $p(r)$ o
the interval from $r = 0$ to $r = 1$:[9]

$$\operatorname{AveP} = \int_0^1 p(r)dr$$



Figures are from wikipedia

Detection - Evaluation Criteria

MEGVII 旷视

mmAP

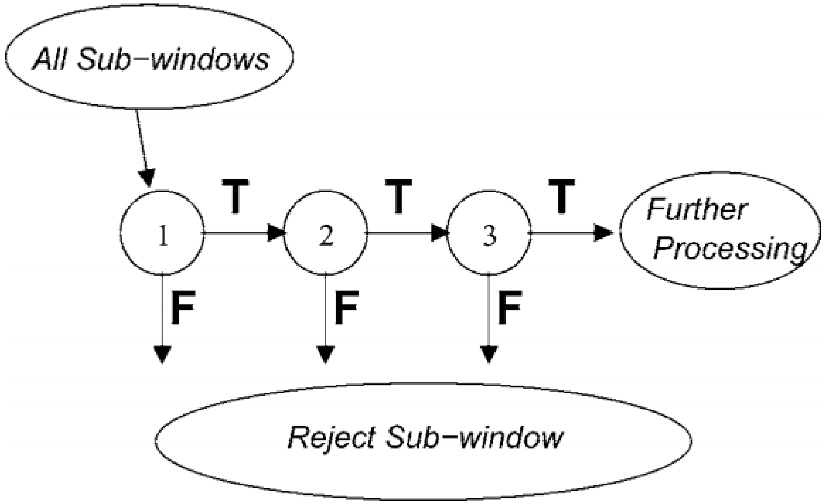
```
Average Precision (AP):
  AP             % AP at IoU=.50:.05:.95 (primary challenge metric)
  AP^IoU=.50     % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75     % AP at IoU=.75 (strict metric)
AP Across Scales:
  AP^small       % AP for small objects: area < 32^2
  AP^medium      % AP for medium objects: 32^2 < area < 96^2
  AP^large       % AP for large objects: area > 96^2
Average Recall (AR):
  AR^max=1       % AR given 1 detection per image
  AR^max=10      % AR given 10 detections per image
  AR^max=100     % AR given 100 detections per image
AR Across Scales:
  AR^small       % AR for small objects: area < 32^2
  AR^medium      % AR for medium objects: 32^2 < area < 96^2
  AR^large       % AR for large objects: area > 96^2
```

1. Unless otherwise specified, *AP and AR are averaged over multiple Intersection over Union (IoU) values*. Specifically we use 10 IoU thresholds of .50:.05:.95. This is a break from tradition, where AP is computed at a single IoU of .50 (which corresponds to our metric $AP^{IoU=.50}$). Averaging over IoUs rewards detectors with better localization.
2. AP is averaged over all categories. Traditionally, this is called "mean average precision" (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context.
3. AP (averaged across all 10 IoU thresholds and all 80 categories) will determine the challenge winner. This should be considered the single most important metric when considering performance on COCO.
4. In COCO, there are more small objects than large objects. Specifically: approximately 41% of objects are small (area < 32^2), 34% are medium (32^2 < area < 96^2), and 24% are large (area > 96^2). Area is measured as the number of pixels in the segmentation mask.
5. AR is the maximum recall given a fixed number of detections per image, averaged over categories and IoUs. AR is related to the metric of the same name used in proposal evaluation but is computed on a per-category basis.
6. All metrics are computed allowing for at most 100 top-scoring detections per image (across all categories).
7. The evaluation metrics for detection with bounding boxes and segmentation masks are identical in all respects except for the IoU computation (which is performed over boxes or masks, respectively).

Figures are from http://cocodataset.org

How to perform a detection?

- Sliding window: enumerate all the windows (up to millions of windows)
  - VJ detector: cascade chain
- Fully Convolutional network
  - shared computation

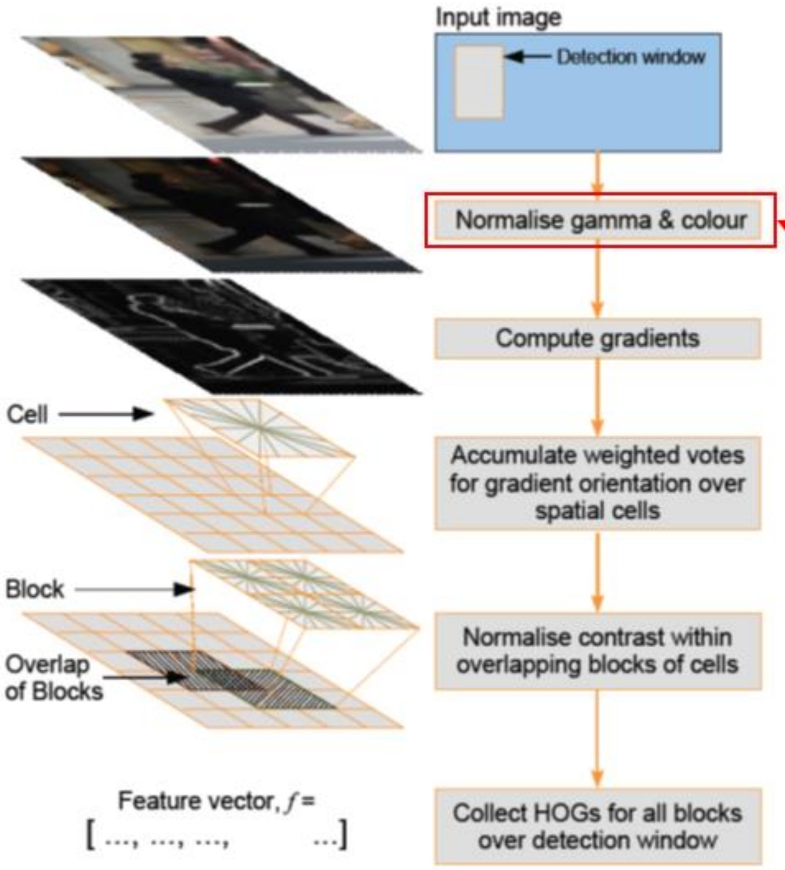


Robust Real-time Object Detection; Viola, Jones; IJCV 2001
http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf

General Detection Before Deep Learning

- Feature + classifier
- Feature
  - Haar Feature
  - HOG (Histogram of Gradient)
  - LBP (Local Binary Pattern)
  - ACF (Aggregated Channel Feature)
  - …
- Classifier
  - SVM
  - Bootsing
  - Random Forest

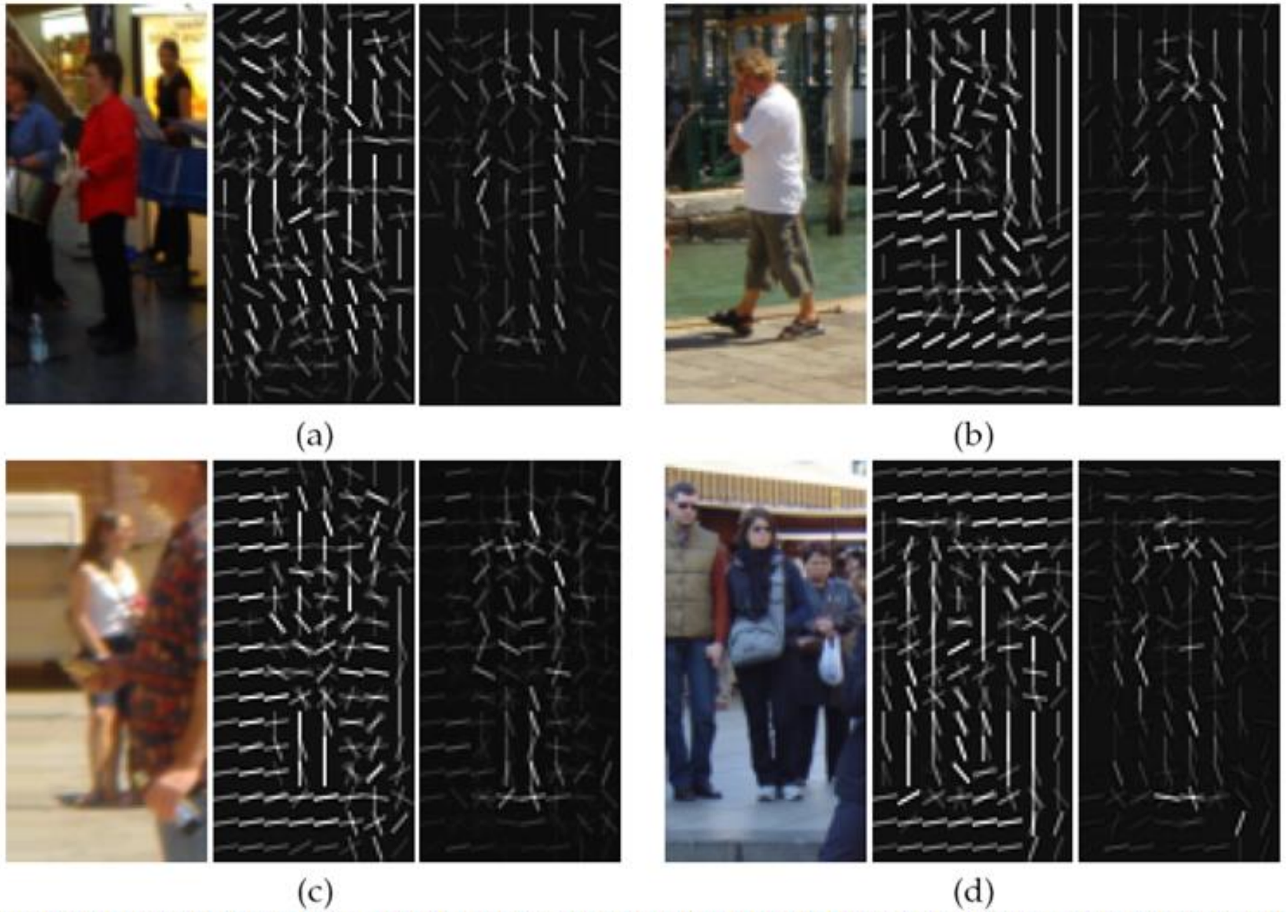
Traditional Hand-crafted Feature: HoG



In practice, effect is very small (about 1%) while some computational time is required*

*Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, SanDiego, USA, June 2005. Vol. II, pp. 886-893.

Traditional Hand-crafted Feature: HoG



In each triplet: (1) the input image, (2) the corresponding R-HOG feature vector (only the dominant orientation of each cell is shown), (3) the dominant orientations selected by the SVM (obtained by multiplying the feature vector by the corresponding weights from the linear SVM).

General Detection Before Deep Learning

MEGVII 旷视

Traditional Methods

- Pros
  - Efficient to compute (e.g., HAAR, ACF) on CPU
  - Easy to debug, analyze the bad cases
  - reasonable performance on limited training data
- Cons
  - Limited performance on large dataset
  - Hard to be accelerated by GPU

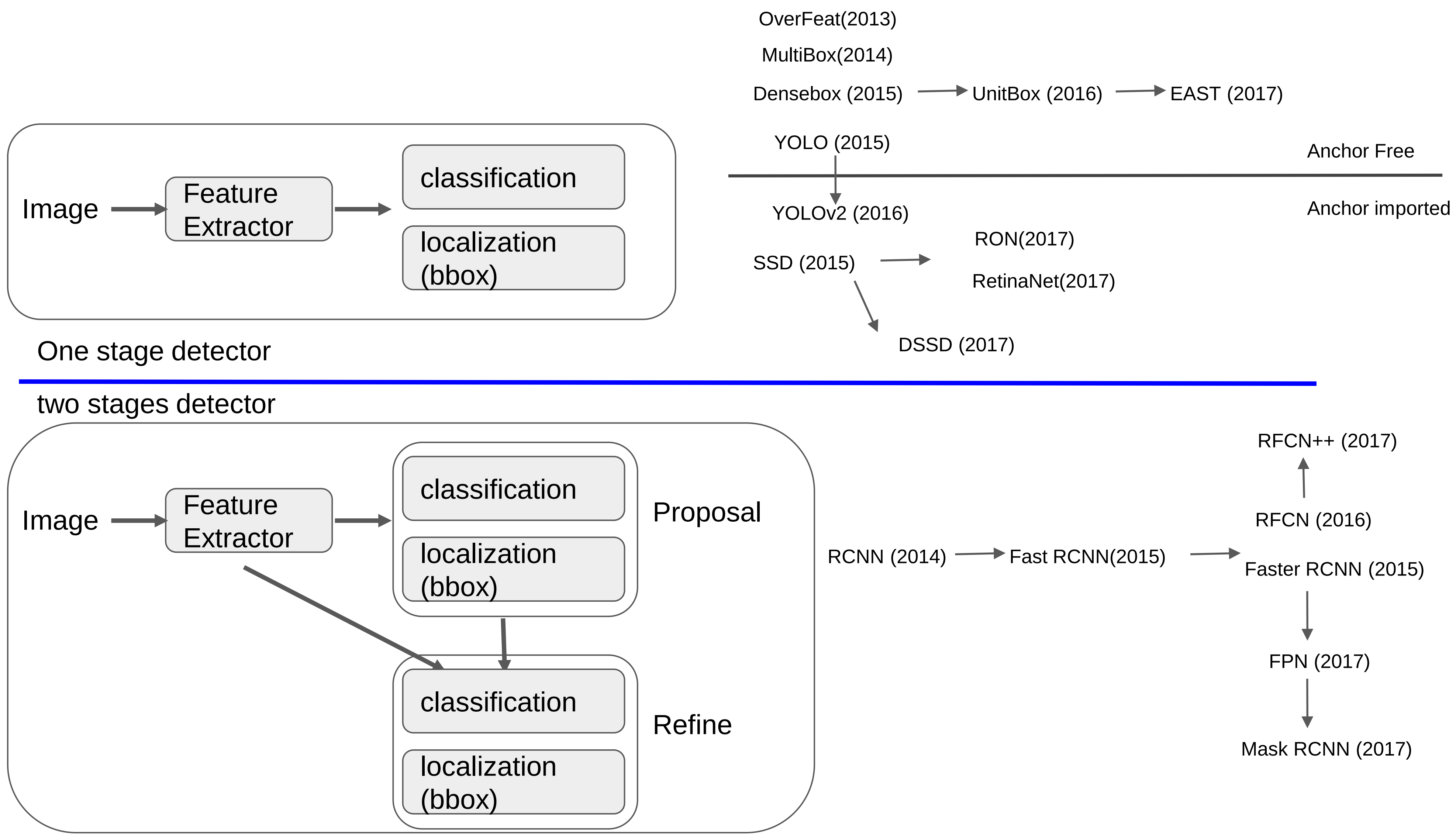
Deep Learning for Object Detection

MEGVII 旷视

Based on the whether following the “proposal and refine”

- One Stage
  - Example: Densebox, YOLO (YOLO v2), SSD, Retina Net
  - Keyword: Anchor, Divide and conquer, loss sampling
- Two Stage
  - Example: RCNN (Fast RCNN, Faster RCNN), RFCN, FPN, MaskRCNN
  - Keyword: speed, performance

A bit of History



Outline

MEGVII 旷视

- Introduction to Object Detection
- Modern Object detectors
  - One Stage detector vs Two-stage detector
- Challenges
  - Backbone
  - Head
  - Pretraining
  - Scale
  - Batch Size
  - Crowd
  - NAS
  - Fine-Grained
- Conclusion

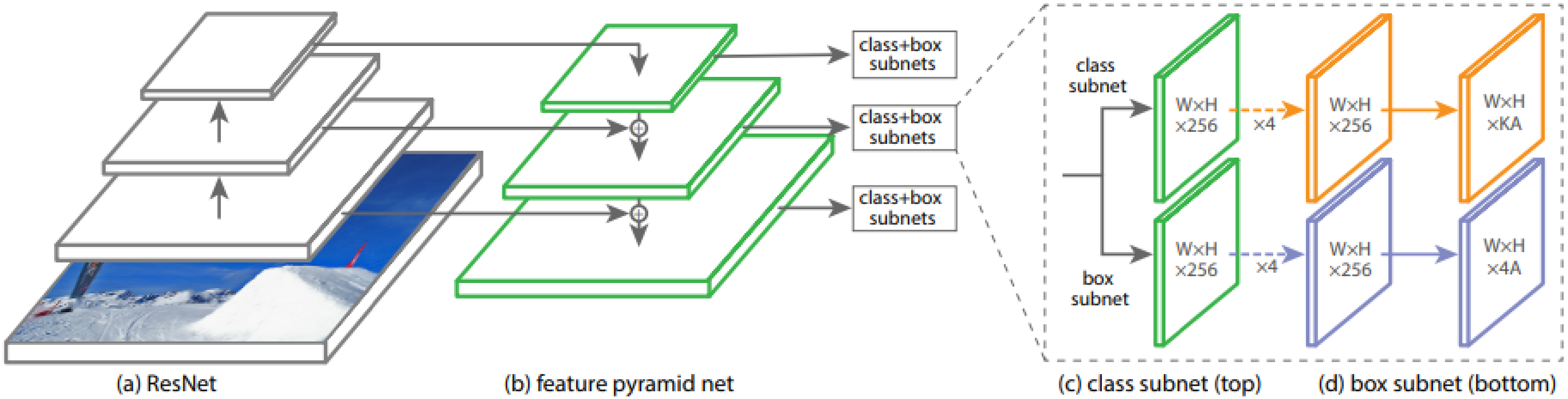
Modern Object detectors

Backbone → Head → Postprocess NMS

- Modern object detectors
  - RetinaNet
    - f1-f7 for backbone,   f3-f7 with 4 convs for head
  - FPN with ROIAlign
    - f1-f6 for backbone, two fcs for head
  - Recall vs localization
    - One stage detector: Recall is high but compromising the localization ability
    - Two stage detector: Strong localization ability

One Stage detector: RetinaNet

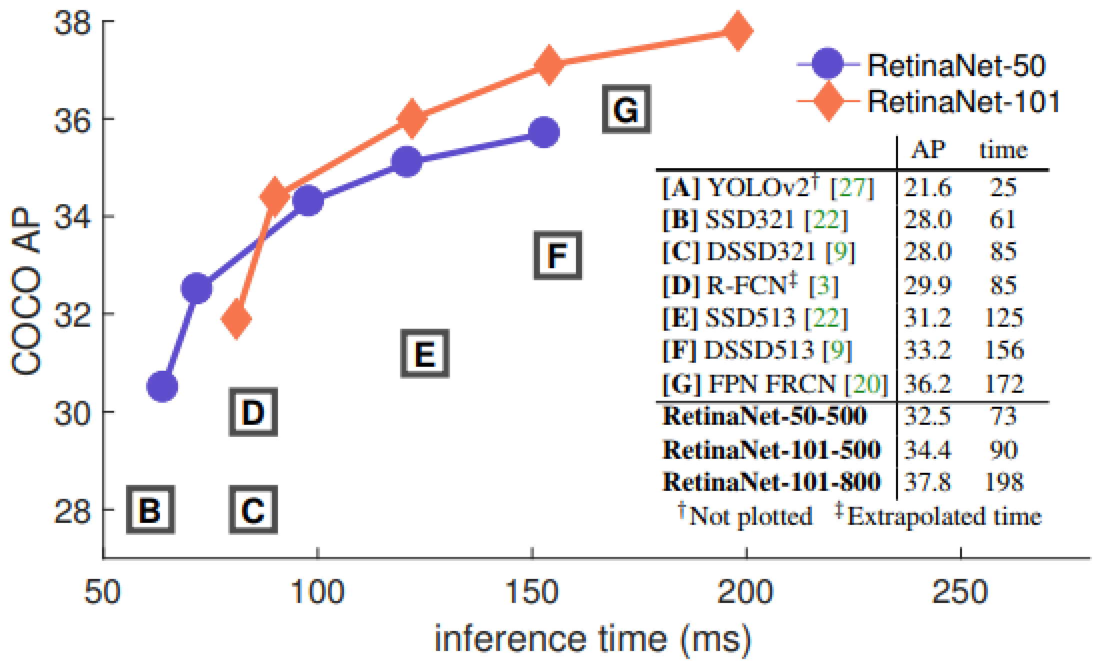
- FPN Structure
- Focal loss



Focal Loss for Dense Object Detection， Lin etc, ICCV 2017 Best student paper

One Stage detector: RetinaNet

- FPN Structure
- Focal loss



Focal Loss for Dense Object Detection， Lin etc, ICCV 2017 Best student paper

Two-Stage detector: FPN/Mask R-CNN

MEGVII 旷视

- FPN Structure
- ROIAlign

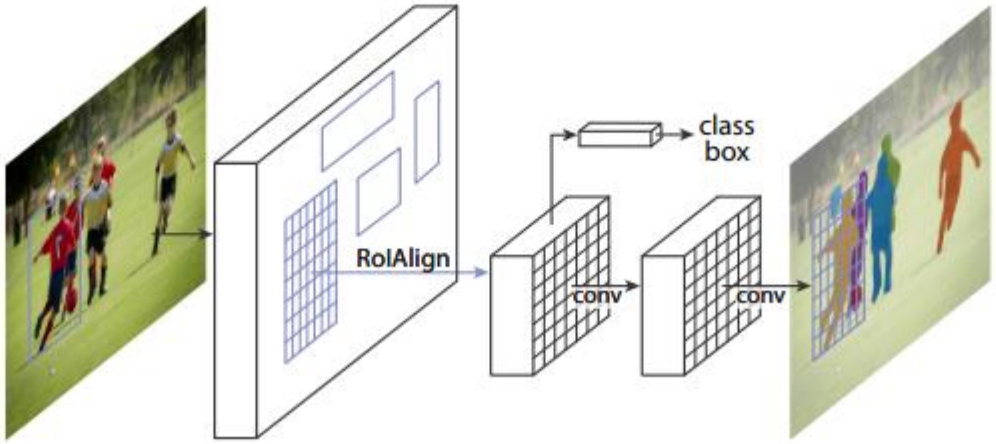


Figure 1. The **Mask R-CNN** framework for instance segmentation.

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [39]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

Mask R-CNN， He etc, ICCV 2017 Best paper

What is next for object detection?

- The pipeline seems to be mature
- There still exists a large gap between existing state-of-arts and product requirements
- The devil is in the detail

Outline

- Introduction to Object Detection
- Modern Object detectors
  - One Stage detector vs Two-stage detector
- Challenges
  - Backbone
  - Head
  - Pretraining
  - Scale
  - Batch Size
  - Crowd
  - NAS
  - Fine-Grained
- Conclusion

Challenges Overview

- Backbone
- Head
- Pretraining
- Scale
- Batch Size
- Crowd
- NAS
- Fine-grained

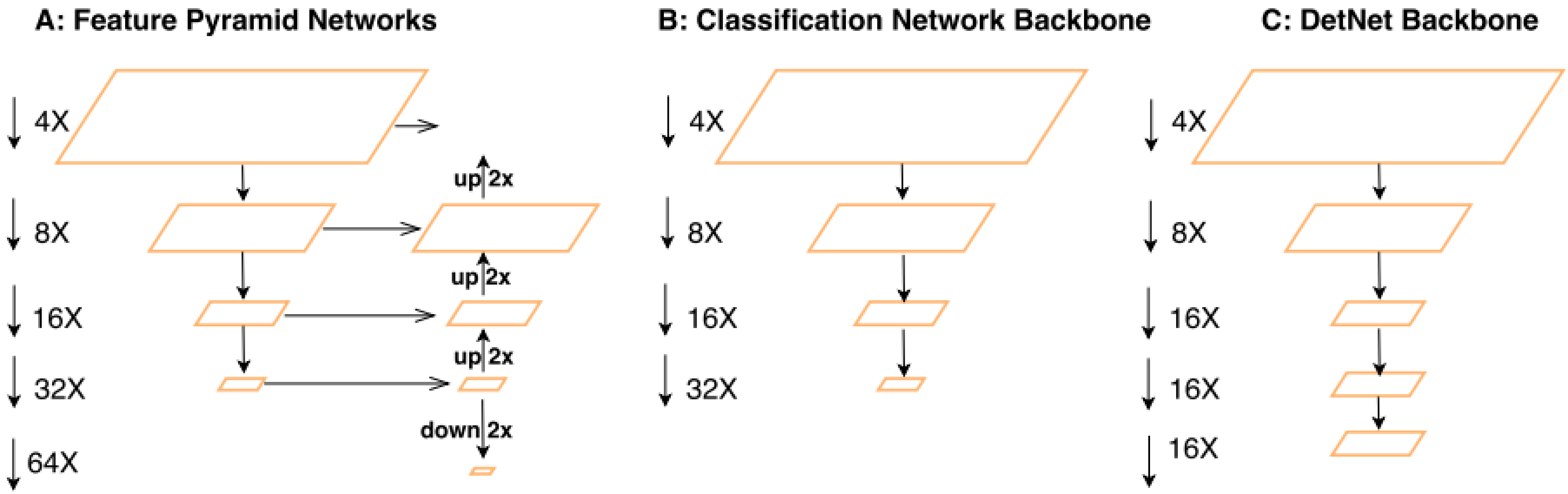
→ Backbone → Head → Postprocess NMS

Challenges - Backbone

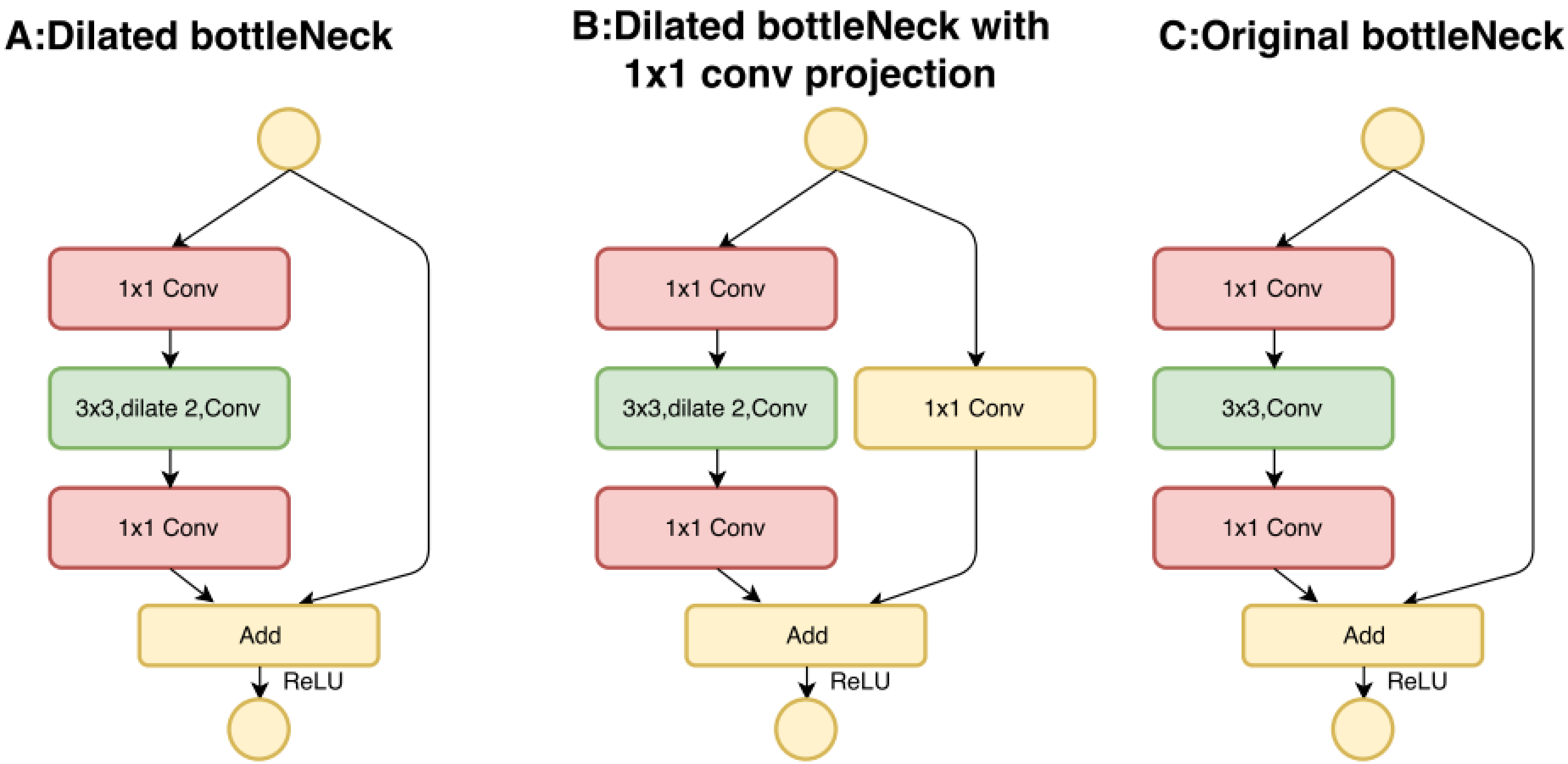
- Backbone network is designed for classification task but not for localization task
  - Receptive Field vs Spatial resolution
- Only f1-f5 is pretrained but randomly initializing f6 and f7 (if applicable)

Backbone - DetNet

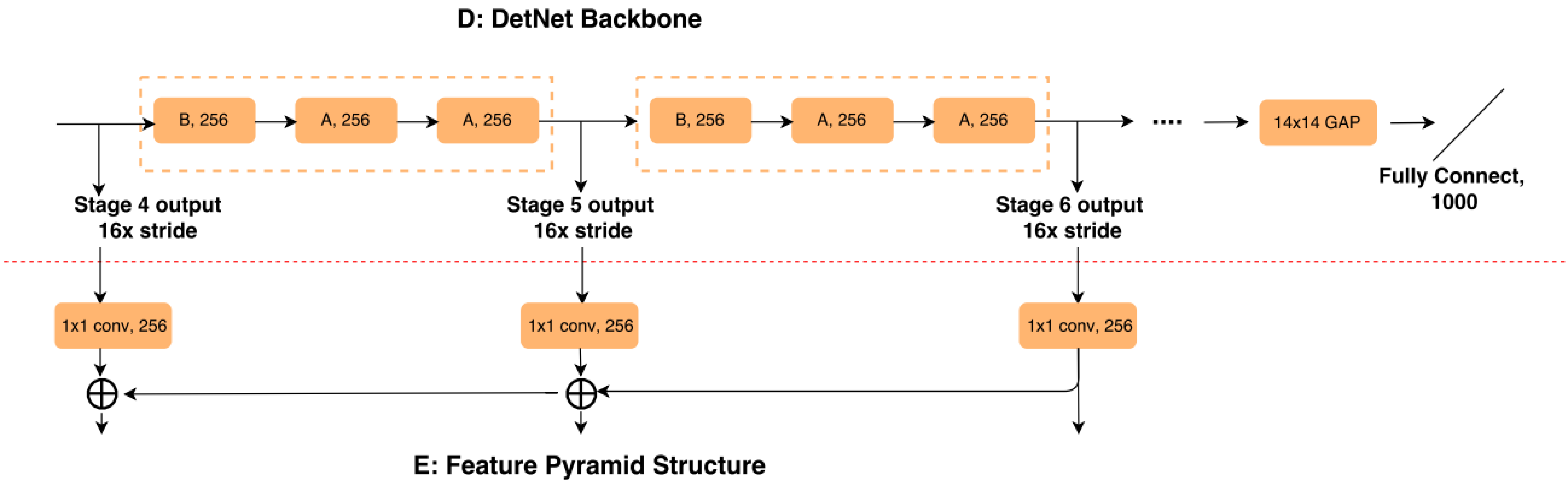
- DetNet: A Backbone network for Object Detection, Li etc, 2018, https://arxiv.org/pdf/1804.06215.pdf



Backbone - DetNet



Backbone - DetNet



Backbone - DetNet

MEGVII 旷视

bacbone	Classification		FPN on COCO minival						FPN on COCO test-dev					
	Err	FLOPs	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
D-59	23.5	4.8G	**40.2**	61.7	43.7	23.9	43.2	52.0	**40.3**	62.1	43.8	23.6	42.6	50.0
R-62	23.4	4.7G	38.8	60.6	42.4	22.6	41.6	51.6	39.0	61.0	42.3	21.9	41.2	49.7
R-50	24.1	3.8G	37.9	60.0	41.2	22.9	40.6	49.2	38.4	60.4	41.6	22.5	40.7	47.9
D-101	23.0	7.9G	**41.9**	62.8	45.7	25.4	45.2	55.1	**42.2**	63.2	45.8	24.5	44.8	53.1
R-101	22.9	7.8G	39.9	62.0	43.7	24.1	43.4	52.0	40.3	62.5	44.0	23.3	43.1	50.6

Table 1. Comparison of 'D' DetNet and 'R' ResNet. We report both results on ImageNet classification (Top1 Error) and FPN COCO detection. Results validate that DetNet is more suitable for object detection. Keeping same model size, DetNet consistently outperform ResNet.

Backbone - DetNet

MEGVII 旷视

Models	scales	mAP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{85}
ResNet-50	over all scales	37.9	60.0	55.1	47.2	33.1	22.1
	small	22.9	40.1	35.5	28.0	17.5	10.4
	middle	40.6	63.9	59.0	51.2	35.7	23.3
	large	49.2	72.2	68.2	60.8	46.6	34.5
DetNet-59	over all scales	40.2	61.7	57.0	49.6	36.2	25. 8
	small	23.9	41.8	36.8	29.8	17.7	10.5
	middle	43.2	65.8	61.2	53.6	39.9	27.3
	large	52.0	73.1	69.5	63	51.4	40.0

Models	scales	mAR	AR_{50}	AR_{60}	AR_{70}	AR_{80}	AR_{85}
ResNet-50	over all scales	52.8	80.5	74.7	64.3	46.8	34.2
	small	35.5	60.0	53.8	43.3	28.7	18.7
	middle	56.0	84.9	79.2	68.7	50.5	36.2
	large	67.0	95.0	90.9	80.3	63.1	50.2
DetNet-59	over all scales	56.1	83.1	77.8	67.6	51.0	38.9
	small	39.2	66.4	59.4	47.3	29.5	19.6
	middle	59.5	87.4	82.5	72.6	55.6	41.2
	large	70.1	95.4	91.8	82.9	69.1	56.3

Backbone - DetNet

MEGVII 旷视

Models	Backbone	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
SSD513 [3]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3,37]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
Faster R-CNN +++ [11]	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN G-RMI [2] [38]	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
RetinaNet [4]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
FPN [33]	ResNet-101	37.3	59.6	40.3	19.8	40.2	48.8
FPN	**DetNet-59**	**40.3**	62.1	43.8	23.6	42.6	50.0

Models	Backbone	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
MNC [39]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [40] + OHEM [41]	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [40] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN [33]	ResNet-101	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	**DetNet-59**	**37.1**	60.0	39.6	18.6	39.0	51.3

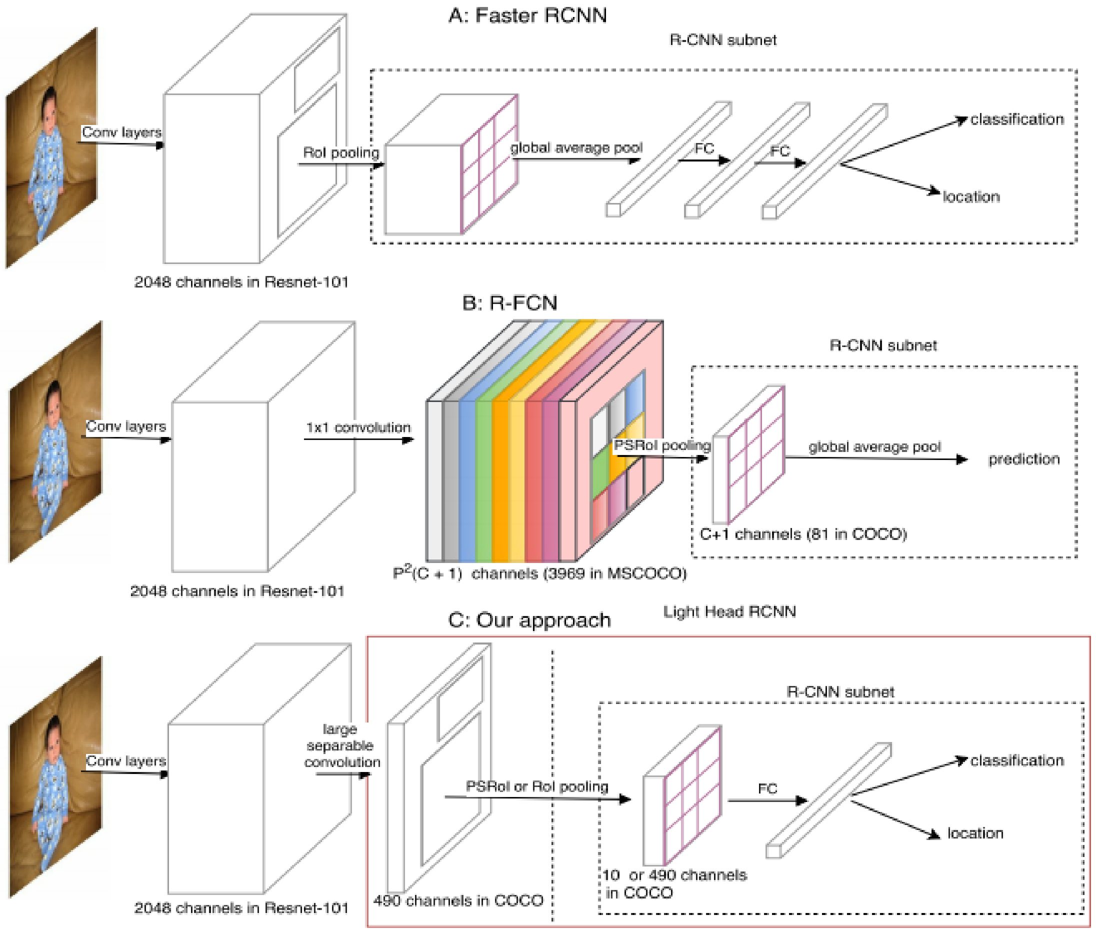
Challenges - Head

- Speed is significantly improved for the two-stage detector
  - RCNN - > Fast RCNN -> Faster RCNN - > RFCN
- How to obtain efficient speed as one stage detector like YOLO, SSD?
  - Small Backbone
  - Light Head

Head – Light head RCNN

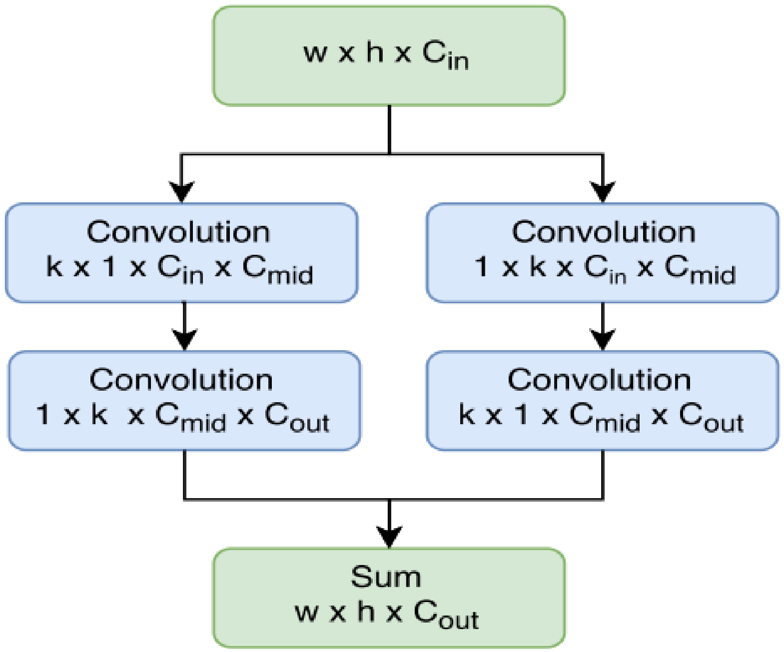
- Light-Head R-CNN: In Defense of Two-Stage Object Detector, 2017, https://arxiv.org/pdf/1711.07264.pdf

  Code: https://github.com/zengarden/light_head_rcnn



Head – Light head RCNN

- Backbone
  - L: Resnet101
  - S: Xception145
- Thin Feature map
  - L:C_{mid} = 256
  - S: C_{mid} =64
  - C_{out} = 10 * 7 * 7
- R-CNN subnet
  - A fc layer is connected to the PS ROI pool/Align

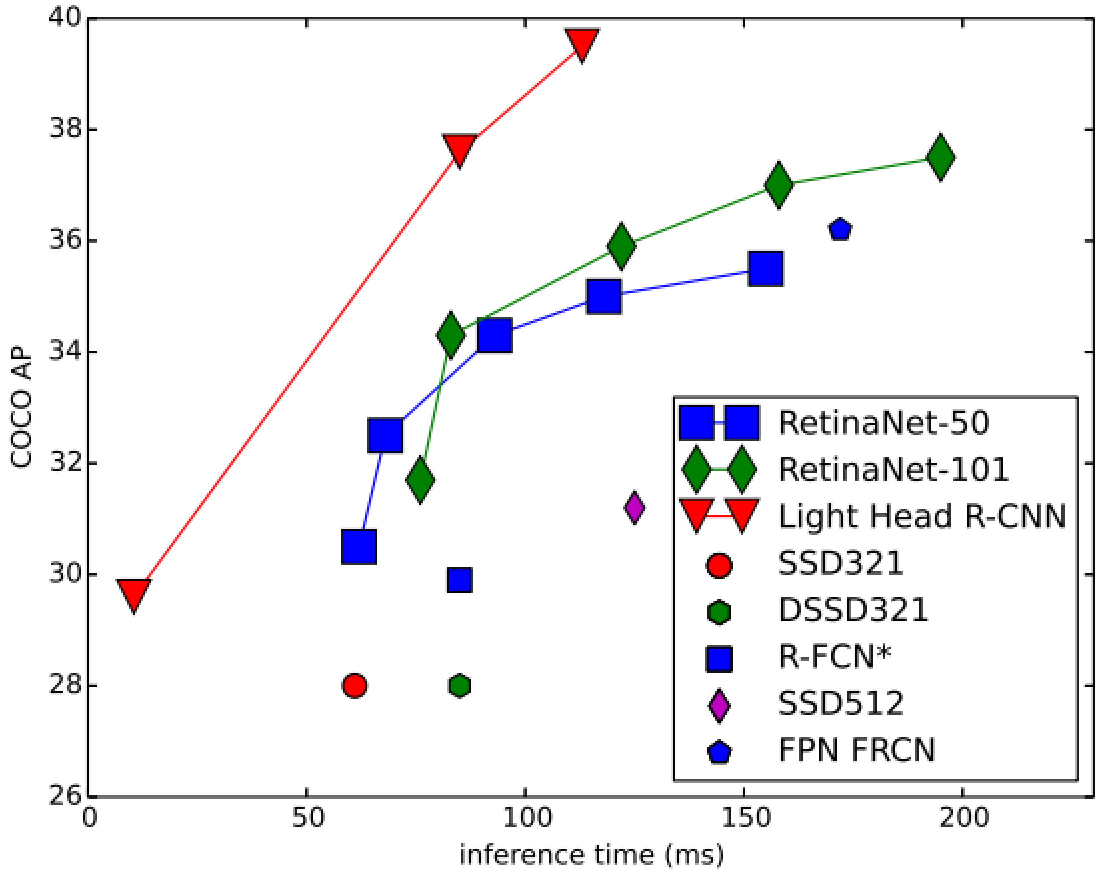


Head – Light head RCNN

Models	mAP@[0.5:0.95]
baseline Faster R-CNN	35.6
B2 (R-FCN)	35.2
+Large Kernel	35.9
+Large Kernel and Light-Head R-CNN	37.7

method	test size shorter edge/max size	feature pyramid	align	mAP@[0.5:0.95]	AP_s	AP_m	AP_l
R-FCN [17]	600/1000			32.1	12.8	34.9	46.1
Faster R-CNN (2fc)	600/1000			30.3	9.9	32.2	47.4
Deformable [3]	600/1000		√	34.5	14.0	37.7	50.3
G-RMI [13]	600/1000			35.6	-	-	-
FPN [19]	800/1200	√		36.2	18.2	39.0	48.2
Mask R-CNN [7]	800/1200	√	√	38.2	20.1	41.1	50.2
RetinaNet [20]	800/1200	√		37.8	20.2	41.1	49.2
RetinaNet ms-train [20]	800/1200	√		39.1	21.8	42.7	50.2
Light head R-CNN	800/1200		√	**39.5**	21.8	43.0	50.7
Light head R-CNN ms-train	800/1200		√	**40.8**	22.7	44.3	52.8
Light head R-CNN	800/1200	√	√	**41.5**	25.2	45.3	53.1

Head – Light head RCNN



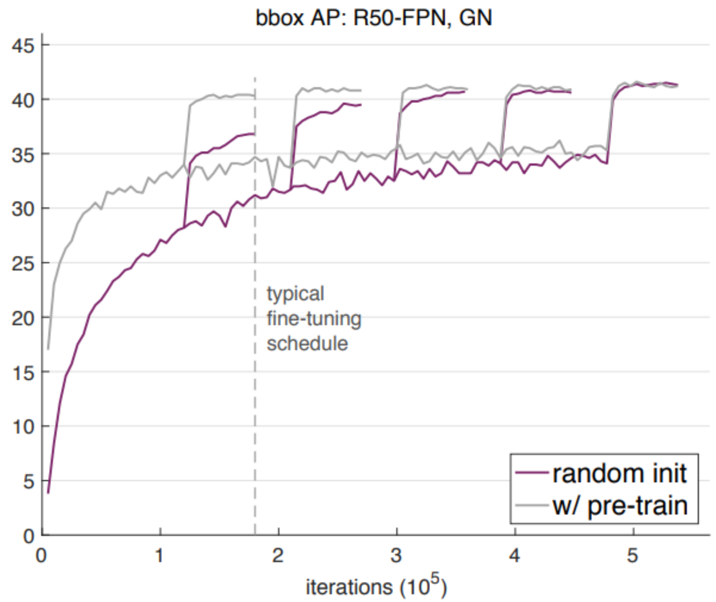
Model	backbone	test size short/max edge	speed (fps)	mAP @[0.5:0.95]
YOLO V2	Darknet	448/448	40	21.6
SSD	VGG	300/300	58	25.1
SSD	Resnet101	300/300	16	28.0
SSD	Resnet101	500/500	8	31.2
DSSD [4]	Resnet101	300/300	8	28.0
DSSD	Resnet101	500/500	6	33.2
R-FCN	Resnet101	600/1000	11	29.9
DeNet	Resnet34	512/512	83	29.4
Light Head R-CNN	xception*	800/1200	**95**	**31.5**
Light Head R-CNN	xception*	700/1100	**102**	**30.7**

Head – Light head RCNN

- Mobile Version
  - ThunderNet: Towards Real-time Generic Object Detection, Qin etc, Arxiv 2019
  - https://arxiv.org/abs/1903.11752

Pretraining – Objects365

- ImageNet pretraining is usually employed for backbone training
- Training from Scratch
  - Scratch Det claims GN/BN is important
  - Rethinking ImageNet Pretraining validates that training time is important



Pretraining – Objects365

- Objects365 Dataset

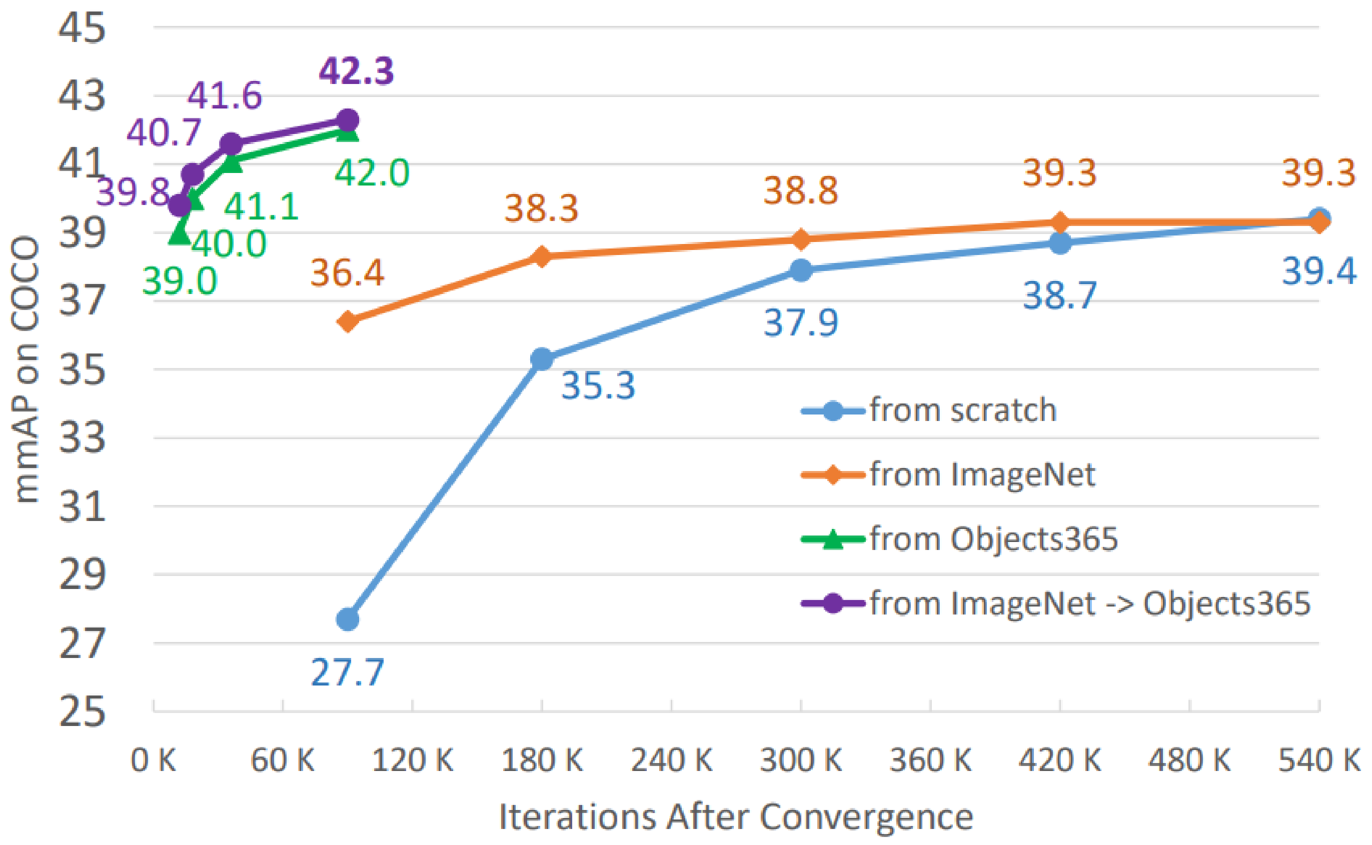
Dataset	Images	Boxes	Categories	Boxes/img
Pascal VOC	11.5k	27k	20	2.4
ImageNet All	477k	534k	200	1.1
ImageNet Dense	80k	186k	200	2.3
COCO	123k	896k	80	7.3
Objects365	**638k**	**10,101k**	**365**	**15.8**

Table 1. Comparison of the dataset statistics with existing fully annotated object detection benchmarks. The table includes statistics for training and validation sets.

Pretraining – Objects365

MEGVII 旷视

- Pretraining with Objects365 vs ImageNet vs from Sctratch



Pretraining – Objects365

MEGVII 旷视

- Pretraining on *Backbone* or Pretraining on *both backbone and head*

Method	Pretrain Part	Iters	mmAP
ImageNet	Backbone	90K	36.4
ImageNet	Backbone	180K	38.3
Objects365	Backbone	90K	37.8
Objects365	Backbone	180K	39.4
Objects365	Backbone & Head	90K	**42.0**

Pretraining – Objects365

- Results on VOC Detection & VOC Segmentation

Pretraining Dataset	mAP
None	63.4
ImageNet	80.2
ImageNet -> COCO 540K iters	85.1
OpenImages	82.4
Objects365	86.2
ImageNet -> Objects365	**86.7**

Pretraining Dataset	mIOU
None	58.3
ImageNet	74.5
ImageNet -> COCO 540K iters	74.9
OpenImages	74.1
Objects365	**76.7**
ImageNet -> Objects365	76.6

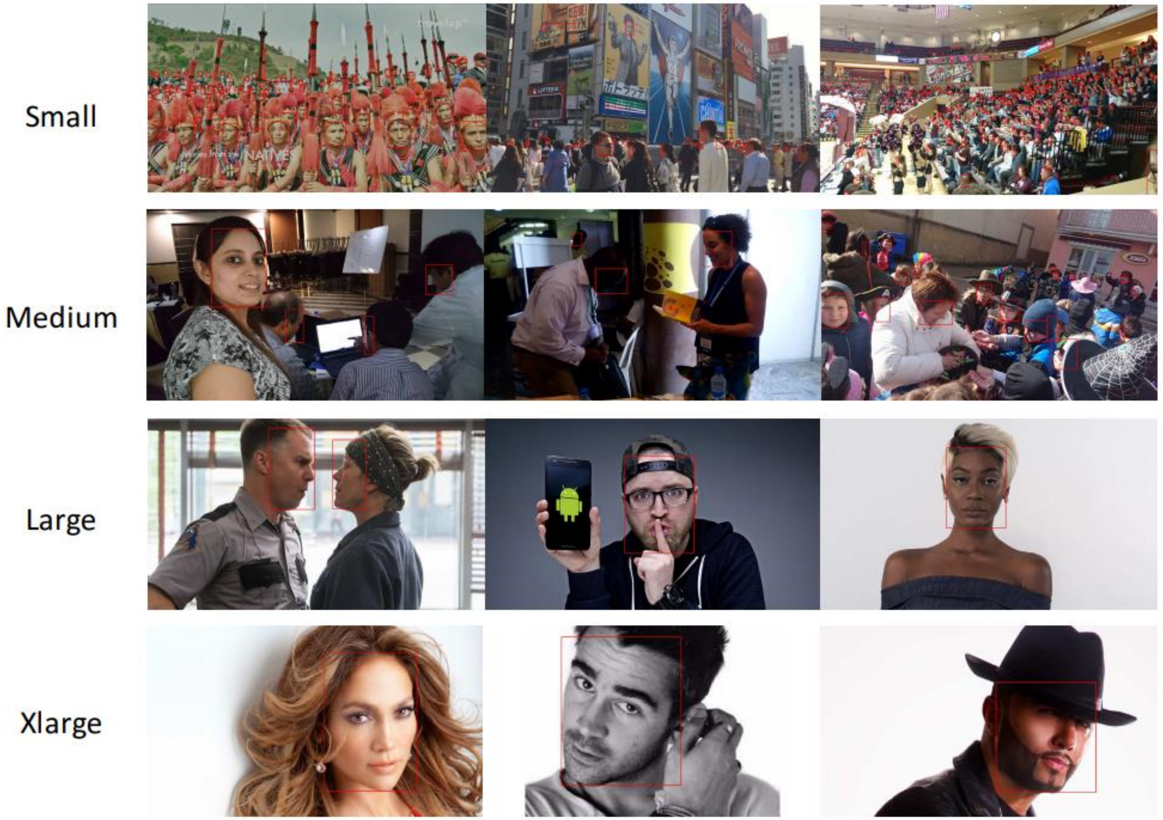
Pretraining – Objects365

- Summary
  - Pretraining is important to reduce the training time
  - Pretraining with a large dataset is beneficial for the performance

Challenges - Scale

MEGVII 旷视

- Scale variations is extremely large for object detection



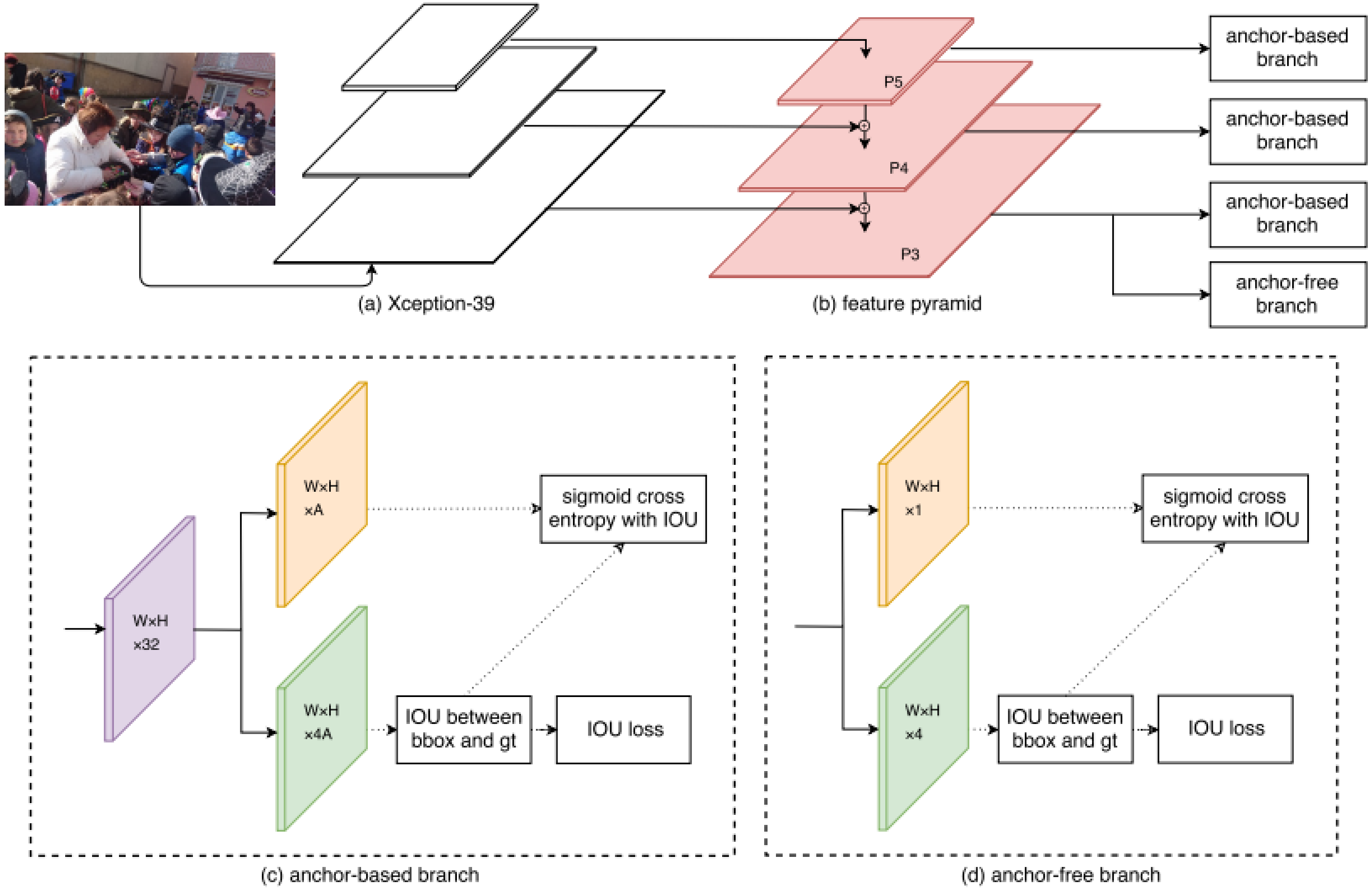
Challenges - Scale

- Scale variations is extremely large for object detection
- Previous works
  - Divide and Conquer: SSD, DSSD, RON, FPN, …
    - Limited Scale variation
  - Scale Normalization for Image Pyramids, Singh etc, CVPR2018
    - Slow inference speed
- How to address extremely large scale variation without compromising inference speed?

Scale - SFace

- SFace: An Efficient Network for Face Detection in Large Scale Variations, 2018, http://cn.arxiv.org/pdf/1804.06559.pdf
  - Anchor-based:
    - Good localization for the scales which are covered by anchors
    - Difficult to address all the scale ranges of faces
  - Anchor-free:
    - Able to cover various face scales
    - Not good for the localization ability

Scale - SFace



anchor-based branch
anchor-based branch
anchor-based branch
anchor-free branch
P5
P4
P3
(a) Xception-39
(b) feature pyramid
W×H ×32
W×H ×A
W×H ×4A
sigmoid cross entropy with IOU
IOU between bbox and gt
IOU loss
(c) anchor-based branch
W×H ×1
W×H ×4
sigmoid cross entropy with IOU
IOU between bbox and gt
IOU loss
(d) anchor-free branch

Scale - SFace

BaseNet	P3-P5 layer	re-score	Anchor-based Branch	Anchor-free Branch	AP (easy)	AP (medium)	AP (hard)
RetinaNet					**92.6**	**91.2**	65.0
RetinaNet(multi-scale)					90.7	90.3	75.2
RetinaNet	✓				43.8	64.9	74.7
UnitBox					70.6	76.0	67.8
SFace	✓		✓		43.5	64.4	73.7
SFace	✓			✓	71.6	78.1	73.7
SFace	✓		✓	✓	71.6	78.1	73.8
SFace	✓	✓	✓		39.5	62.4	72.9
SFace	✓	✓		✓	90.0	88.8	78.8
SFace	✓	✓	✓	✓	89.8	88.7	**80.7**

BaseNet	P3-P5 layer	re-score	Anchor-based Branch	Anchor-free Branch	AP(< 32)	AP(32 – 256)	AP(256 – 512)	AP(> 1024)	AP(all)
RetinaNet					1.22	54.62	74.51	47.74	53.34
RetinaNet	✓				**29.23**	49.72	0.00	0.00	32.73
UnitBox					3.27	61.09	81	50.97	63.82
SFace	✓		✓		26.86	46.51	0.00	0.00	30.72
SFace	✓			✓	3.41	57.35	77.80	53.82	61.49
SFace	✓		✓	✓	3.51	57.38	77.81	53.80	61.60
SFace	✓	✓	✓		23.62	48.03	0.00	0.00	31.50
SFace	✓	✓		✓	4.07	62.93	**81.38**	**60.48**	64.30
SFace	✓	✓	✓	✓	6.70	**63.09**	**81.35**	**60.48**	**65.39**

Scale - SFace

MEGVII 旷视

Method	AP(easy)	AP (medium)	AP(hard)
ACF [31]	69.5	58.8	29.0
Faceness [32]	71.6	60.4	31.5
LDCF+ [33]	79.7	77.2	56.4
MT-CNN [23]	85.1	82.0	60.7
CMS-RCNN [34]	90.2	87.4	64.3
ScaleFaces [35]	86.7	86.6	76.4
SFace	**89.1**	**87.9**	**80.4**

Min size	1080	1200	1500	2160
Time	12.46ms	14.30ms	21.53ms	41.13ms
AP (WIDER FACE hard)	76.7	78.4	80.7	78.8

Scale - SFace

MEGVII 旷视

- Summary:
  - Integrate anchor-based and anchor-free for the scale issue
  - A new benchmark for face detection with large scale variations: 4K Face

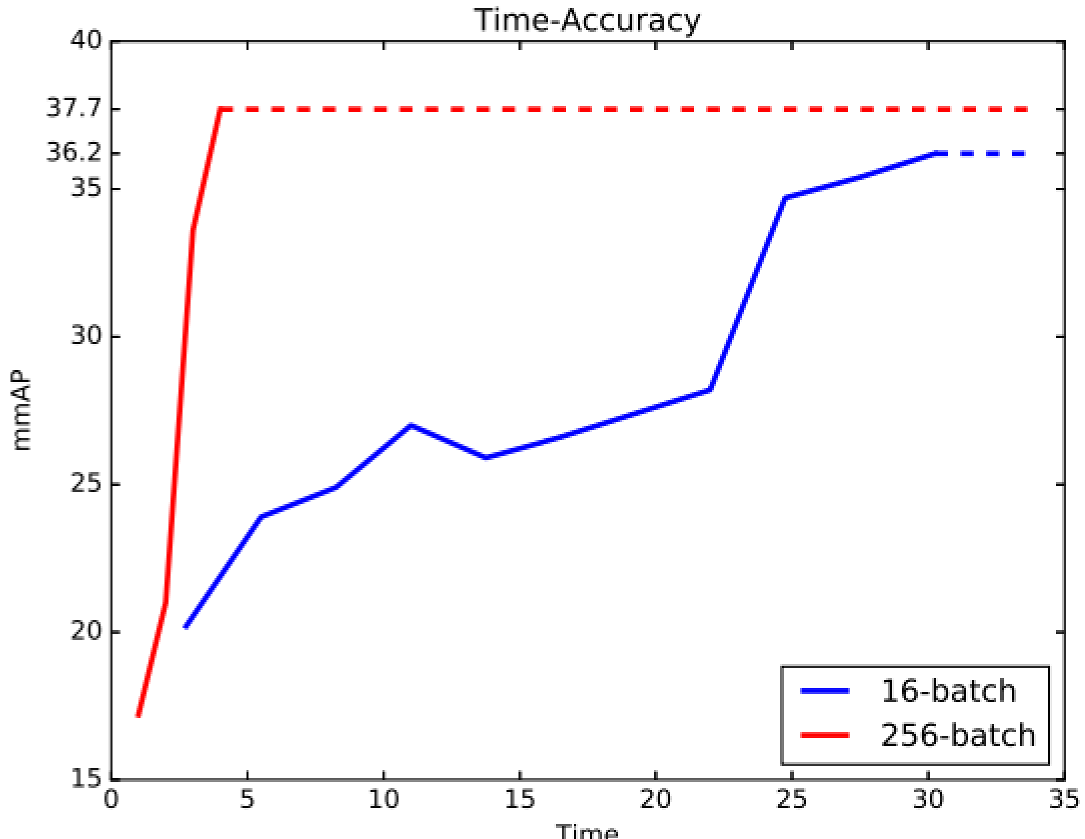
Challenges - Batchsize

MEGVII 旷视

- Small mini-batchsize for general object detection
  - 2 for R-CNN, Faster RCNN
  - 16 for RetinaNet, Mask RCNN
- Problem with small mini-batchsize
  - Long training time
  - Insufficient BN statistics
  - Inbalanced pos/neg ratio

Batchsize – MegDet

- MegDet: A Large Mini-Batch Object Detector, CVPR2018, https://arxiv.org/pdf/1711.07240.pdf



Batchsize – MegDet

- Techniques
  - Learning rate warmup
  - Cross-GPU Batch Normalization

name	mmAP	mmAR
DANet	45.7	62.7
Trimps-Soushen+QINIU	48.0	65.4
bharat_umd	48.1	64.8
FAIR Mask R-CNN [14]	50.3	66.1
MSRA	50.4	69.0
UCenter	51.0	67.9
MegDet (Ensemble)	**52.5**	**69.0**

Challenges - Crowd

- NMS is a post-processing step to eliminate multiple responses on one object instance
  - Reasonable for mild crowdness like COCO and VOC
  - Will Fail in the case when the objects are in a crowd

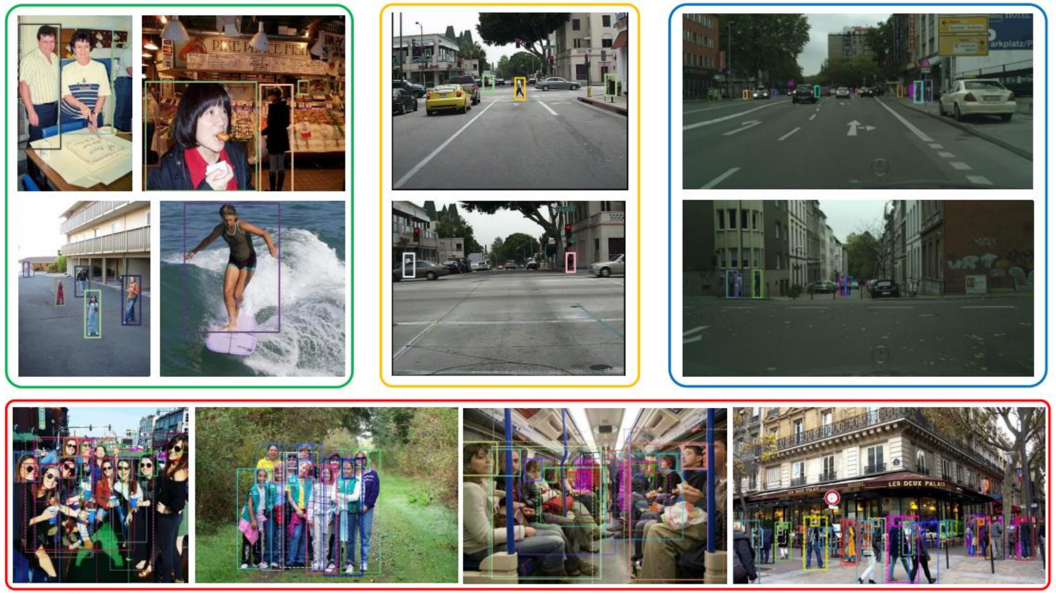


Figure 1. Illustrative examples from different human dataset benchmarks. The images inside the green, yellow, blue boxes are from the COCO [17], Caltech [6], and CityPersons [31] datasets, respectively. The images from the second row inside the red box are from our CrowdHuman benchmark with full body, visible body, and head bounding box annotations for each person.

Challenges - Crowd

MEGVII 旷视

- A few works have been devoted to this topic
  - Softnms, Bodla etc, ICCV 2017, http://www.cs.umd.edu/~bharat/snms.pdf
  - Relation Networks, Hu etc, CVPR 2018, https://arxiv.org/pdf/1711.11575.pdf
- Lacking a good benchmark for evaluation in the literature

Crowd - CrowdHuman

- CrowdHuman: A Benchmark for Detecting Human in a Crowd, 2018, https://arxiv.org/pdf/1805.00123.pdf, http://www.crowdhuman.org/
  - A benchmark with Head, Visible Human, Full body bounding-box
  - Generalization ability for other head/pedestrian datasets
  - **Crowdness**

Crowd - CrowdHuman

	Caltech	KITTI	CityPersons	COCOPersons	CrowdHuman
# images	$42,782$	$3,712$	$2,975$	$\mathbf{64,115}$	$15,000$
# persons	$13,674$	$2,322$	$19,238$	$257,252$	$\mathbf{339,565}$
# ignore regions	$50,363$	45	$6,768$	$5,206$	$\mathbf{99,227}$
# person/image	0.32	0.63	6.47	4.01	$\mathbf{22.64}$
# unique persons	$1,273$	$< 2,322$	$19,238$	$257,252$	$\mathbf{339,565}$

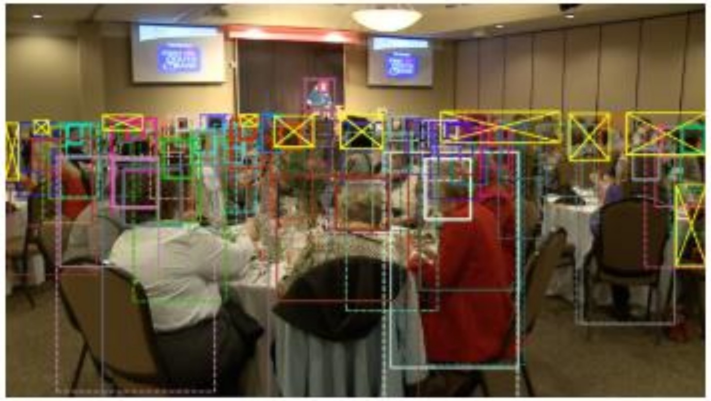
pair/img	Cal	City	COCO	CrowdHuman
iou>0.3	0.06	0.96	0.13	9.02
iou>0.4	0.03	0.58	0.05	4.89
iou>0.5	0.02	0.32	0.02	2.40
iou>0.6	0.01	0.17	0.01	1.01
iou>0.7	0.00	0.08	0.00	0.33
iou>0.8	0.00	0.02	0.00	0.07
iou>0.9	0.00	0.00	0.00	0.01

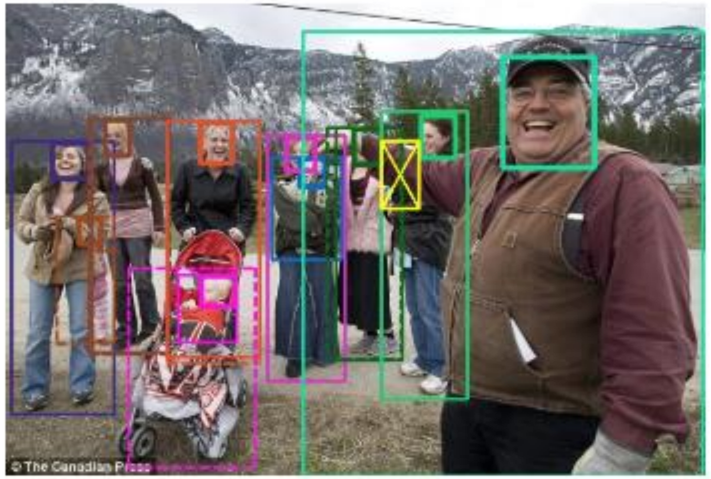
Table 4. Comparison of pair-wise overlap between two human instances.

pair/img	Cal	City	COCO	CrowdHuman
iou>0.1	0.02	0.30	0.02	8.70
iou>0.2	0.00	0.11	0.00	2.09
iou>0.3	0.00	0.04	0.00	0.51
iou>0.4	0.00	0.01	0.00	0.12
iou>0.5	0.00	0.00	0.00	0.03

Table 5. Comparison of high-order overlaps among three human instances.

Crowd-CrowdHuman





Crowd-CrowdHuman

- Generalization
  - Head
  - Pedestrian
  - COCO

Table 11. Experimental reslts on CityPersons

Train-set	Recall	AP	mMR
Caltech	87.21	65.87	45.52
CityPersons	97.97	94.35	14.81
CrowdHuman	98.73	98.10	21.18
Crowd⇒City	97.78	95.58	**10.67**
CityPersons [31]	-	-	14.8
Repulsion [26]	-	-	13.2

Table 10. Experimental results on Caltech dataset.

Train-set	Recall	AP	mMR
Caltech	99.76	89.95	10.08
CityPersons	99.05	85.81	14.69
CrowdHuman	99.88	90.58	8.81
Crowd⇒Calt	99.88	95.69	**3.46**
CityPersons⇒Calt [31]	-	-	5.1
Repulsion [26]	-	-	4.0
[18]	-	-	5.5

Table 12. Experimental results on Brainwash.

Train-set	Recall	AP	mMR
Brainwash	98.52	95.74	19.77
Crowd⇒Brain	98.66	96.15	**17.24**
[23]	-	78.0	-

Table 9. Experimental results on COCOPersons.

Train-set	Recall	AP	mMR
COCOPersons	95.57	83.83	41.89
Crowd⇒COCO	95.87	85.02	**39.79**

Conclusion

- The task of object detection is still far from solved
- Details are important to further improve the performance
  - Backbone
  - Head
  - Pretraining
  - Scale
  - Batchsize
  - Crowd
- The improvement of object detection will be a significantly boost for the computer vision industry

广告部分

MEGVII 旷视

- Megvii Detection 知乎专栏

Email: yugang@megvii.com

MEGVII 旷视