M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network

Qijie Zhao[1], Tao Sheng[1], Yongtao Wang[1]*, Zhi Tang[1], Ying Chen[2], Ling Cai[2] and Haibin Ling[3]
[1]Institute of Computer Science and Technology, Peking University, Beijing, P.R. China
[2]AI Labs, DAMO Academy, Alibaba Group
[3]Computer and Information Sciences Department, Temple University
{zhaoqijie, shengtao, wyt, tangzhi}@pku.edu.cn,
{cailing.cl, chenying.ailab}@alibaba-inc.com, {hbling}@temple.edu

Qijie Zhao
12, Nov, 2019
Visual **D**ata **I**nterpreting and **G**eneration Lab(VDIG)
Institute of Computer Science and Technology, Peking University
Supervisor: Associate Professor Yongtao Wang
Homepage: qijiezhao.github.io
Mail: zhaoqijie@pku.edu.cn

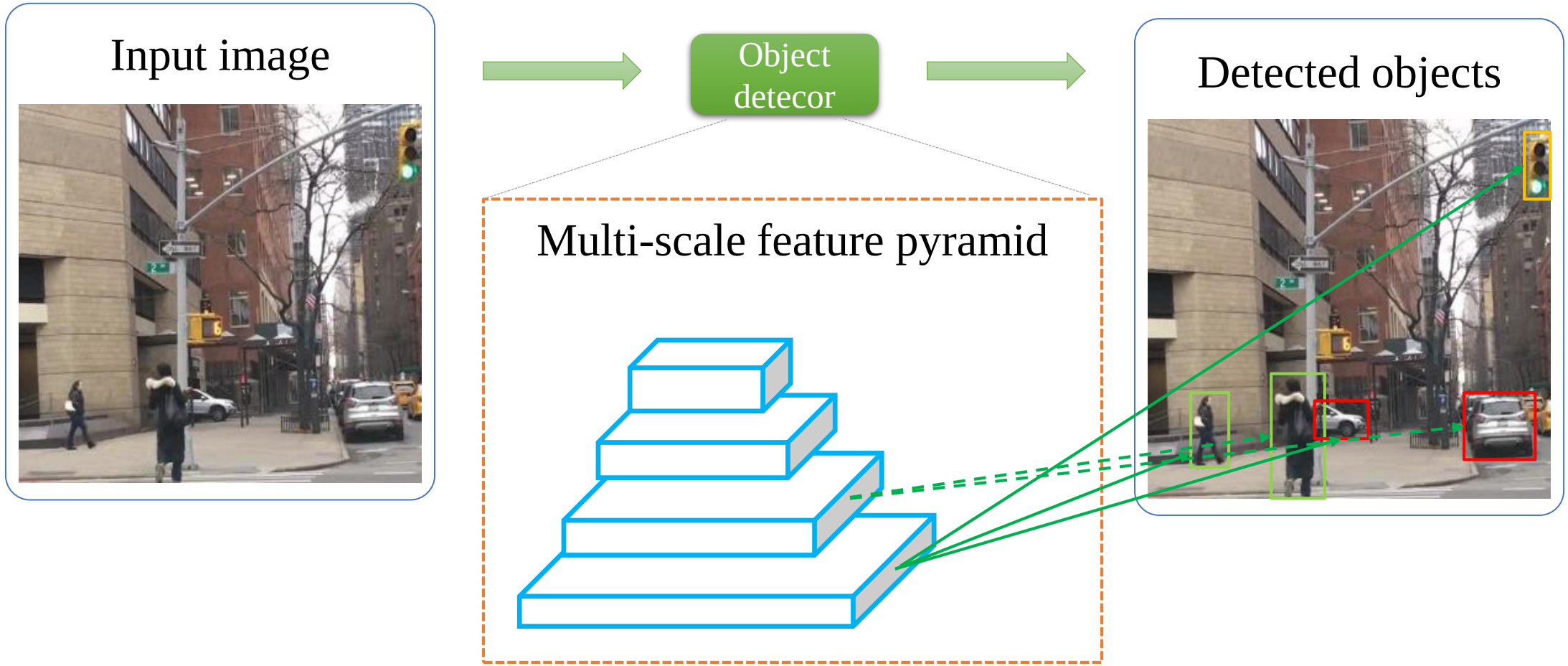
Content

- Introduction
- Proposed method
- Experiments
- Summary

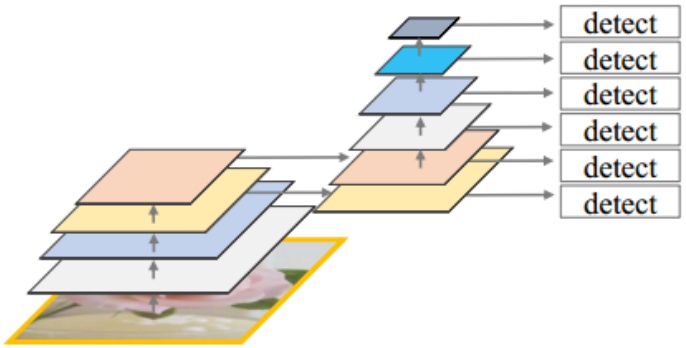
Content

- Introduction
- Proposed method
- Experiments
- Summary

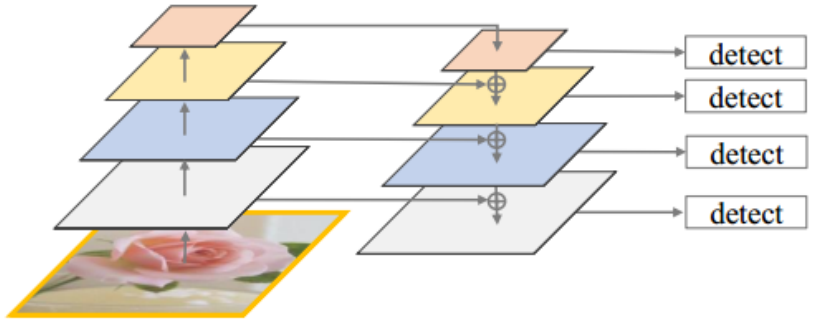
➢ Object detection based on Multi-scale features



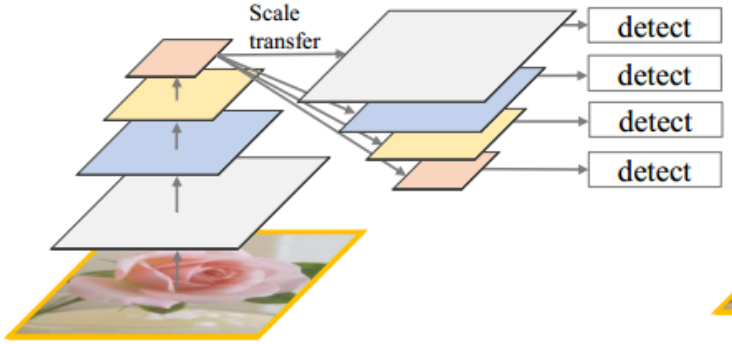
➢ Feature Pyramid Networks



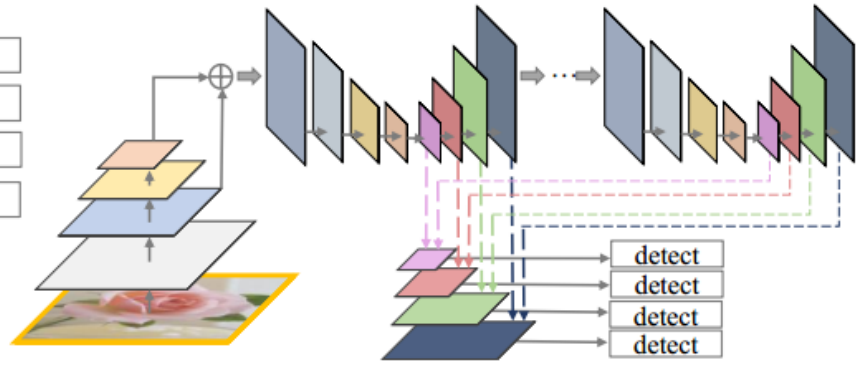
(a) SSD-style feature pyramid



(b) FPN-style feature pyramid



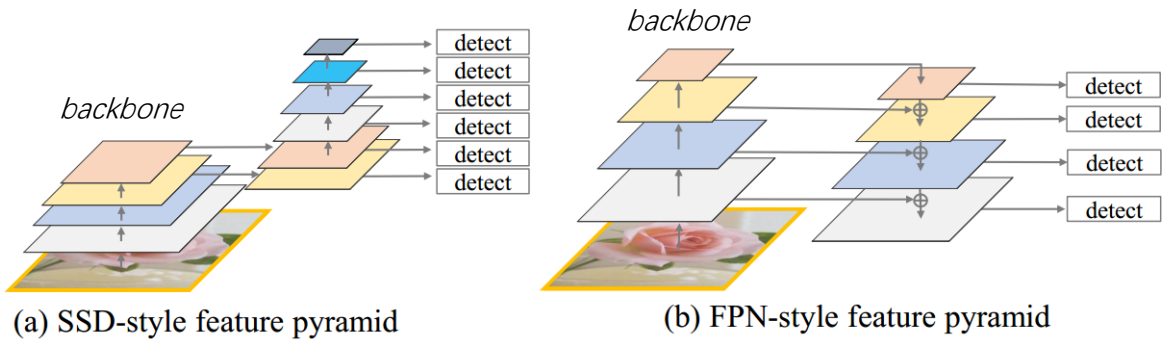
(c) STDN-style feature pyramid



(d) Our multi-level feature pyramid

➢ Motivations

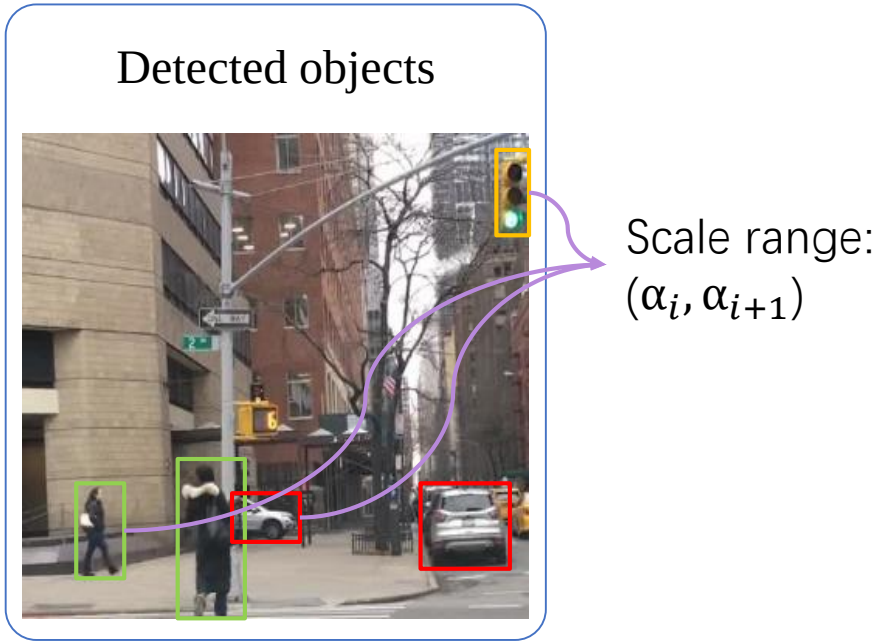
- State-of-the-art detectors that based on multi-scale feature pyramid network are limited with the backbone that pre-trained on image classification task.



(a) SSD-style feature pyramid

(b) FPN-style feature pyramid

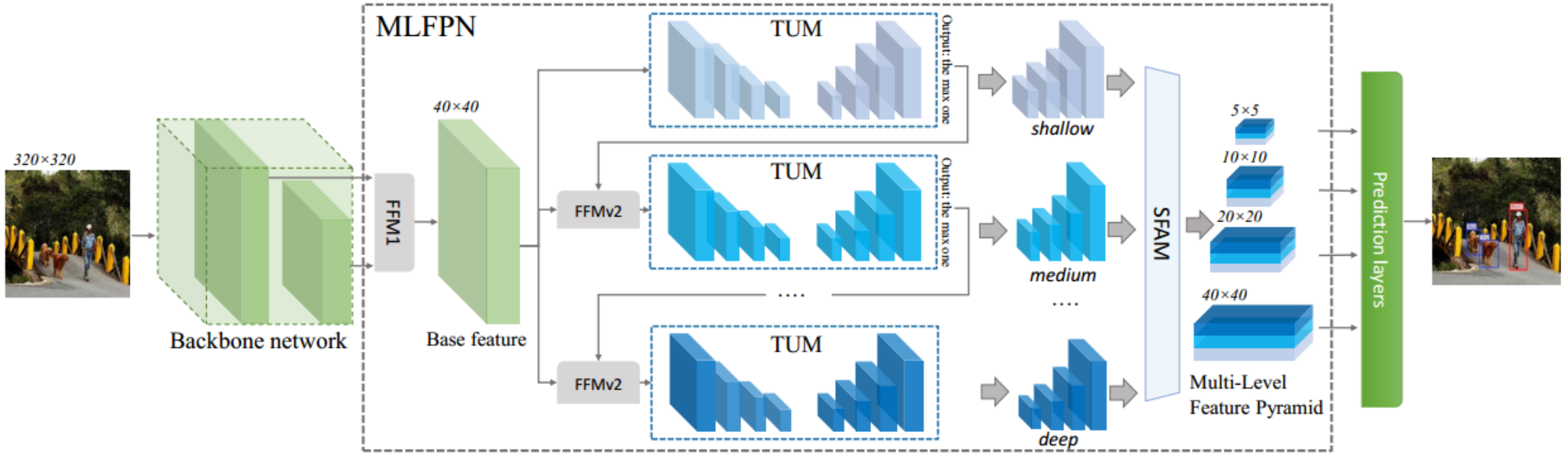
- Widely used feature pyramid networks always ignore the complex appearance variation across object instances with equivalent scale.



Content

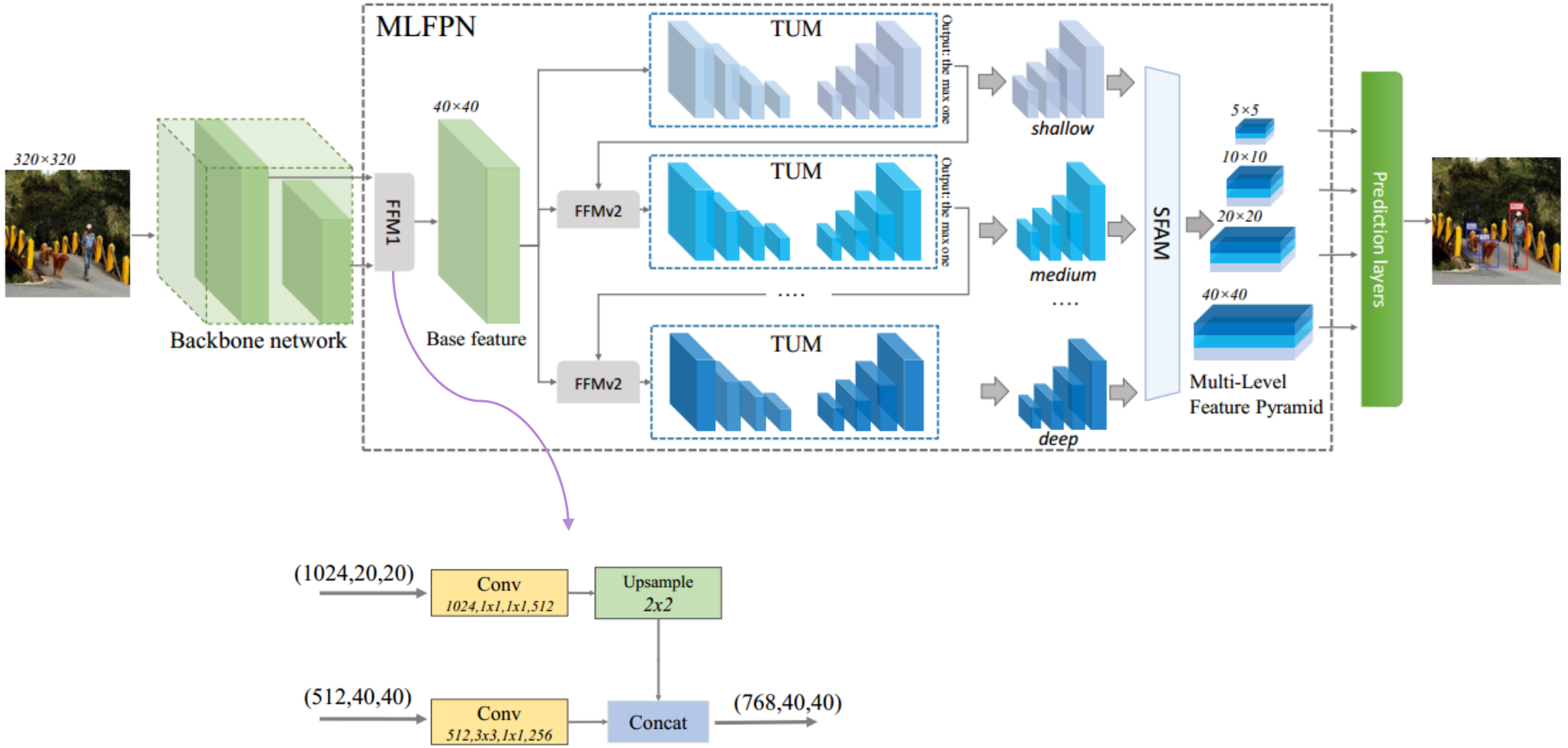
- Introduction
- Proposed method
- Experiments
- Summary

➢ Architecture of M2Det

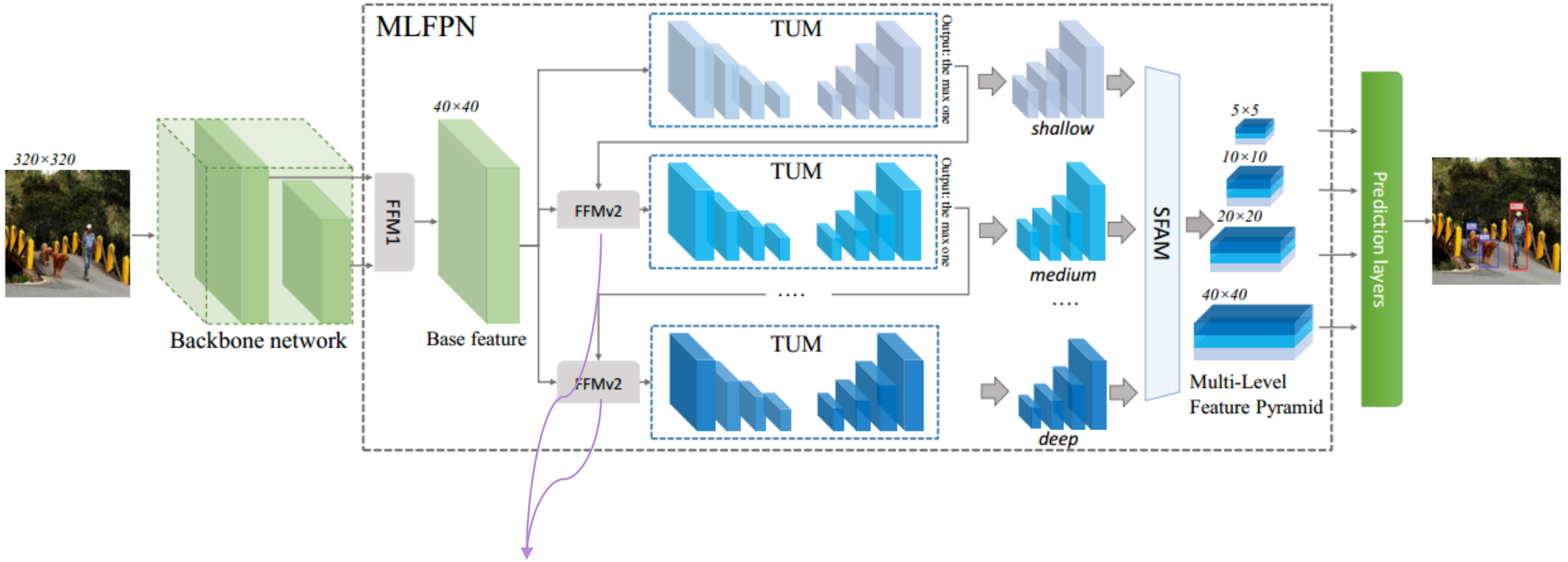


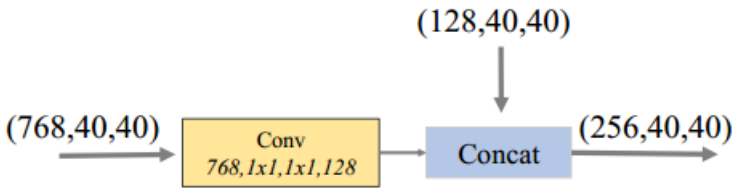
Multi-level Feature Pyramid Network(MLFPN) contains **FFMv1, FFMv2, multiple TUMs and SFAM**.

➢ Architecture of M2Det



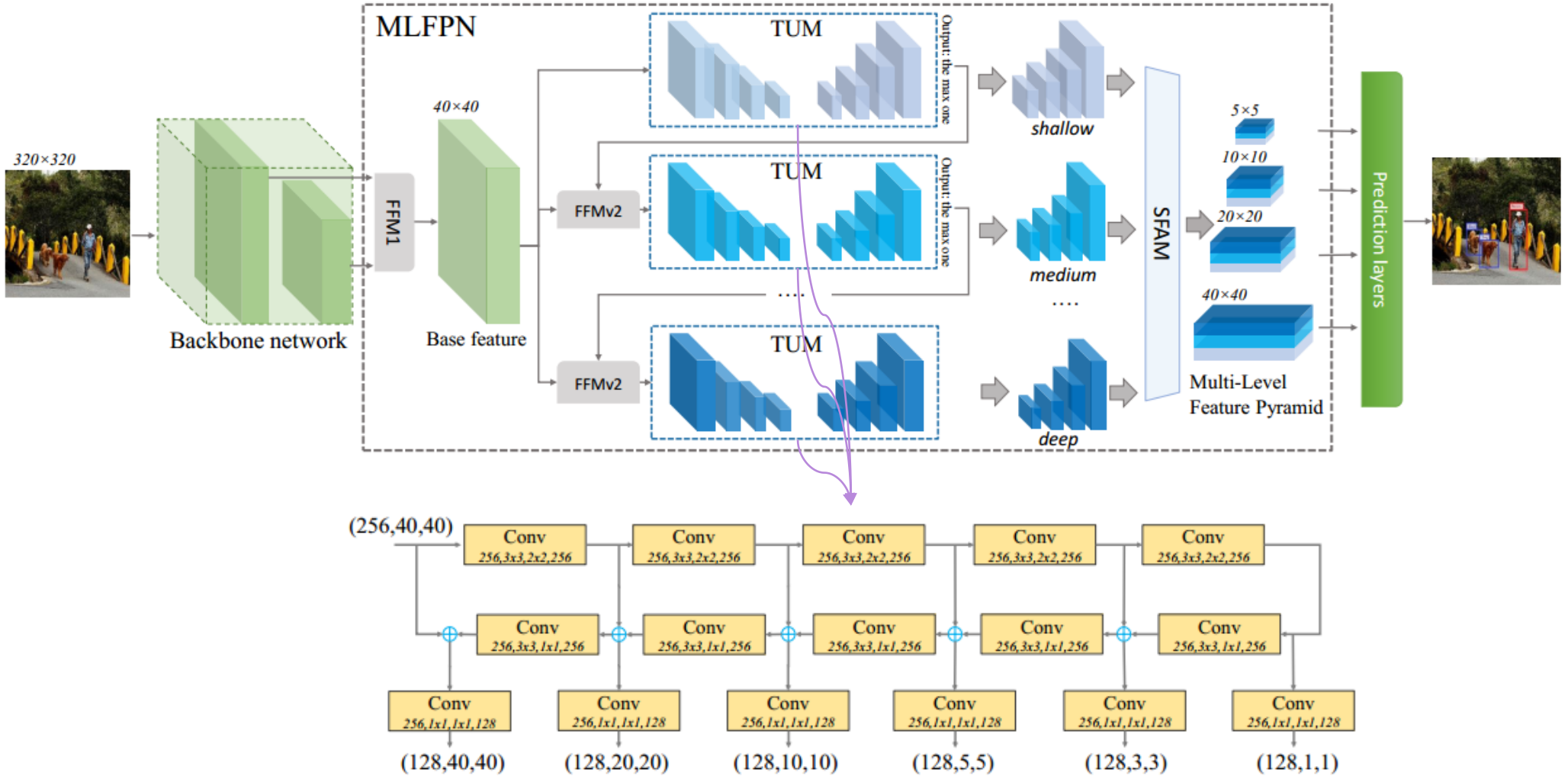
➢ Architecture of M2Det



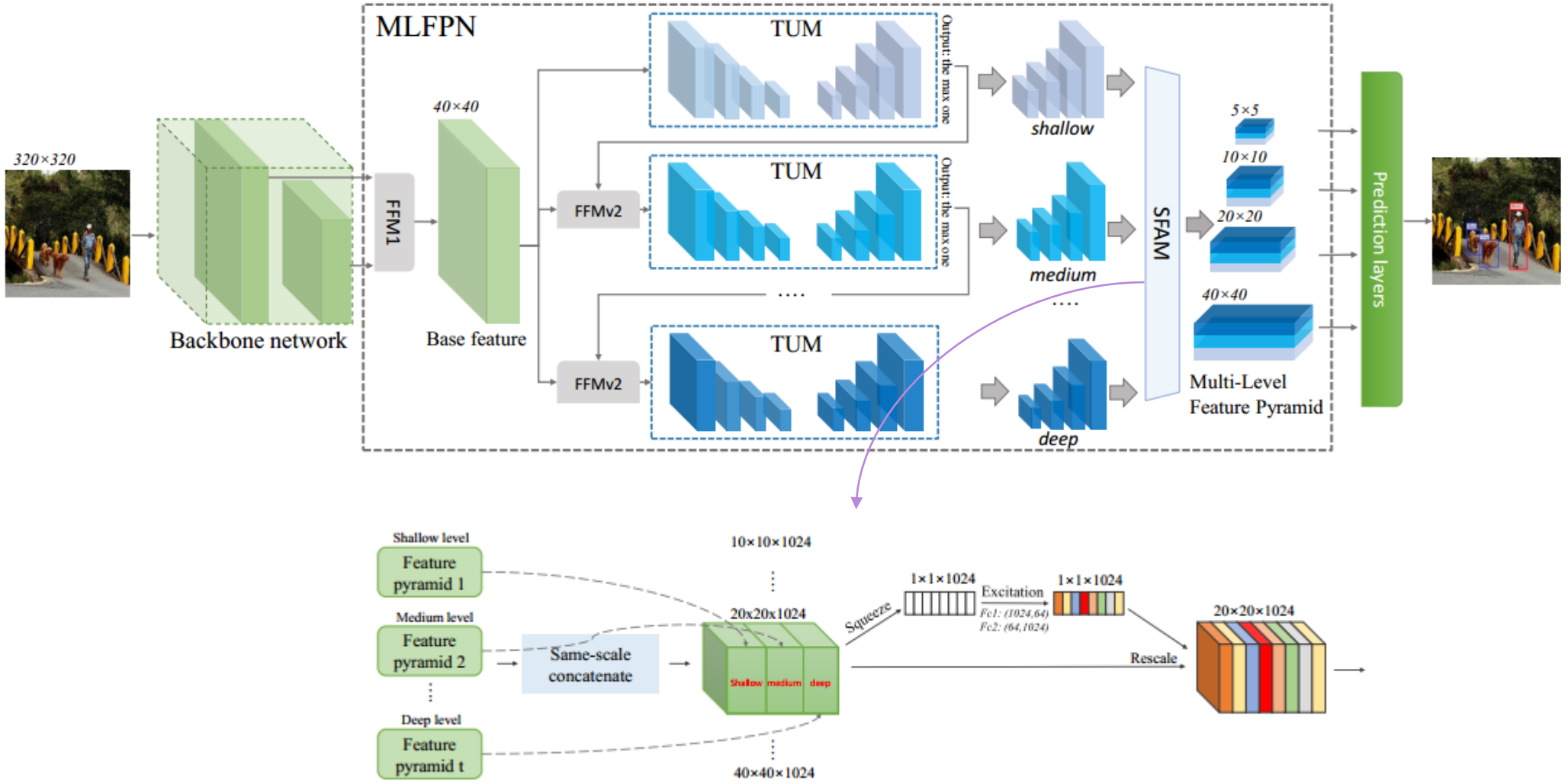


$$[\mathbf{x}_1^l, \mathbf{x}_2^l, ..., \mathbf{x}_i^l] = \begin{cases} \mathbf{T}_l(\mathbf{X}_{base}), & l = 1 \\ \mathbf{T}_l(\mathbf{F}(\mathbf{X}_{base}, \mathbf{x}_i^{l-1})), & l = 2...L \end{cases}$$

➢ Architecture of M2Det



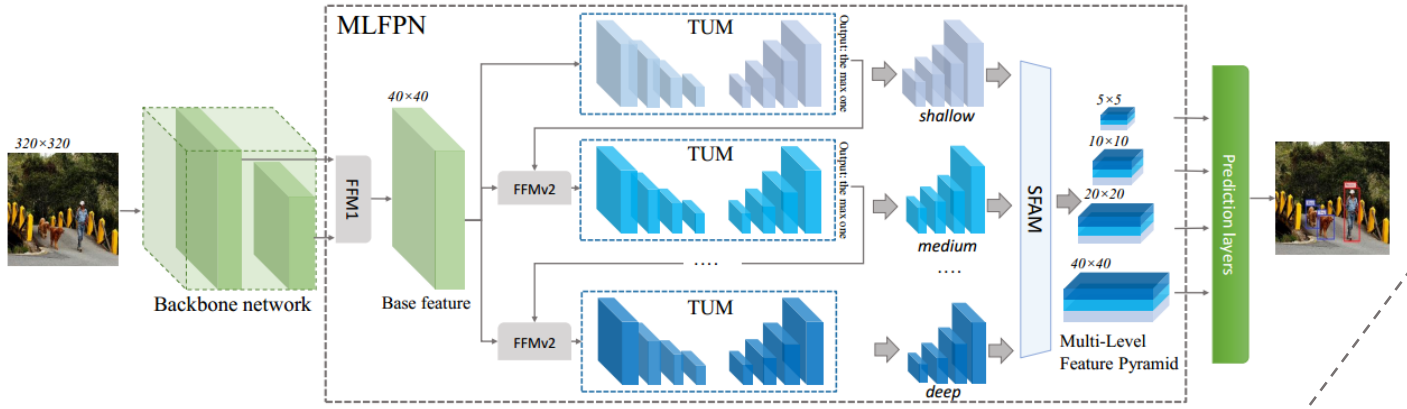
➢ Architecture of M2Det



Content

- Introduction
- Proposed method
- Experiments
- Summary

➢ Ablation Study



	(1)	(2)	(3)	(4)	(5)	(6)	(7)
+ 1 s-TUM		✓					
+ 8 s-TUM			✓				
+ 8 TUM				✓	✓	✓	✓
+ Base feature					✓	✓	✓
+ SFAM						✓	✓
VGG16 ⇒ Res101							✓
AP	25.8	27.5	30.6	30.8	32.7	33.2	**34.1**
AP_{50}	44.7	45.2	50.0	50.3	51.9	52.2	**53.7**
AP_{small}	7.2	7.7	13.8	13.7	13.9	15.0	**15.9**
AP_{medium}	27.4	28.0	35.3	35.3	37.9	38.2	**39.5**
AP_{large}	41.4	47.0	44.5	44.8	48.8	49.1	**49.3**

COCO benchmark.

Validation set: 2014minival

(1) baseline: simple SSD, vgg, 320×320

(2) Construct base feature with FFMv1, connect a TUM without 1x1 convs(s-TUM)

(3) Connect 8 s-TUMs

(4) Change to 8 TUMs

(5) Feed Base feature with FFMv2

(6) Add SFAM

(7) Change backbone to ResNet101

➢ Different Configurations of MLFPN

TUMs	Channels	Params(M)	AP	AP_{50}	AP_{75}
2	256	40.1	30.5	50.5	32.0
2	512	106.5	32.1	51.8	34.0
4	128	34.2	29.8	49.7	31.2
4	256	60.2	31.8	51.4	33.0
4	512	192.2	33.4	52.6	34.2
8	128	47.5	31.8	50.6	33.6
8	256	98.9	33.2	52.2	35.2
8	512	368.8	34.0	52.9	36.4
16	128	73.9	32.5	51.7	34.4
16	256	176.8	33.6	52.6	35.7

Default settings: VGG, 320×320

We configure two hyper parameters:

a. Number of TUMs

b. Number of Channels

Given the baseline: ***(2, 256)***

Compare (8, 128) and (4, 256)

(16, 128) and (2, 512)

Conclusion:

Although both dimensions can benefit the detection accuracy, depth is better than width

➢ Compare with State-of-the-art

MS-COCO, test-dev detection results, ~300

one-stage:										
SSD300* (Liu et al. 2016)	VGG-16	300×300	False	43	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ (Kong et al. 2017)	VGG-16	384×384	False	15	27.4	49.5	27.1	-	-	-
DSSD321 (Fu et al. 2017)	ResNet-101	321×321	False	9.5	28.0	46.1	29.2	7.4	28.1	47.6
RetinaNet400 (Lin et al. 2017b)	ResNet-101	~640×400	False	12.3	31.9	49.5	34.1	11.6	35.8	48.5
RefineDet320 (Zhang et al. 2018)	VGG-16	320×320	False	38.7	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet320 (Zhang et al. 2018)	ResNet-101	320×320	True	-	38.6	59.9	41.7	21.1	41.7	52.3
M2Det (Ours)	VGG-16	320×320	False	33.4	33.5	52.4	35.6	14.4	37.6	47.6
M2Det (Ours)	VGG-16	320×320	True	-	38.9	59.1	42.4	24.4	41.5	47.6
M2Det (Ours)	ResNet-101	320×320	False	21.7	34.3	53.5	36.5	14.8	38.8	47.9
M2Det (Ours)	ResNet-101	320×320	True	-	**39.7**	**60.0**	**43.3**	**25.3**	**42.5**	**48.3**

MS-COCO, test-dev detection results, ~512

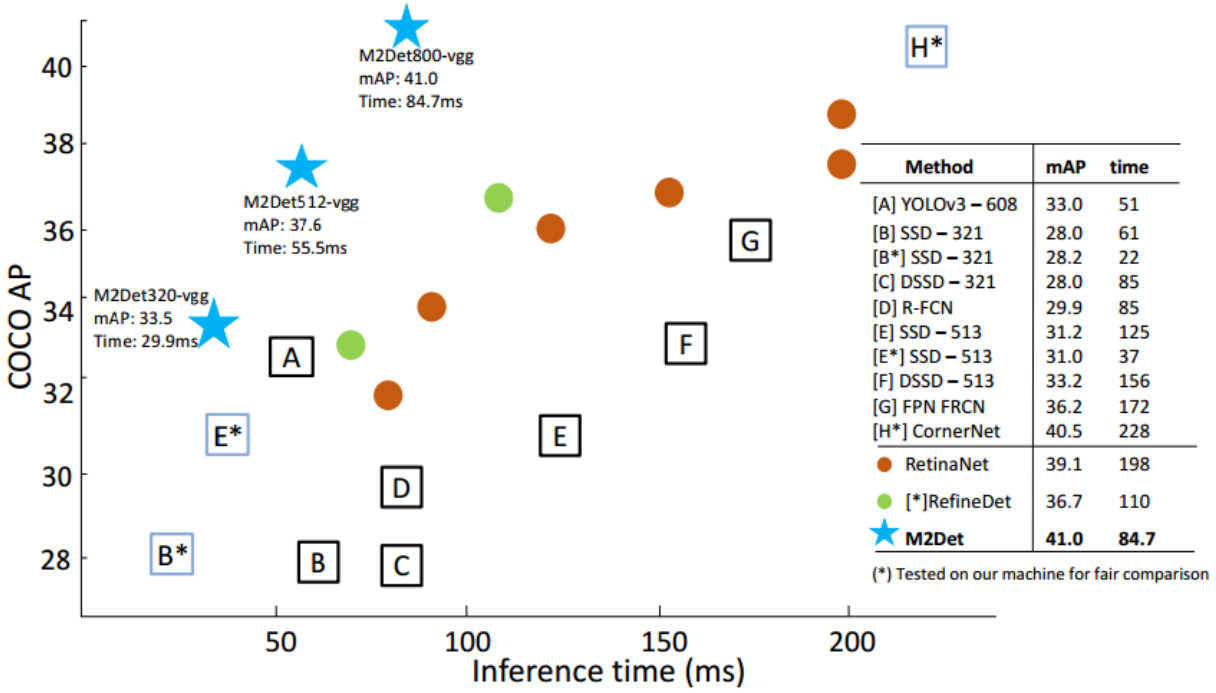
YOLOv3 (Redmon and Farhadi 2018)	DarkNet-53	608×608	False	19.8	33.0	57.9	34.4	18.3	35.4	41.9
SSD512* (Liu et al. 2016)	VGG-16	512×512	False	22	28.8	48.5	30.3	10.9	31.8	43.5
DSSD513 (Fu et al. 2017)	ResNet-101	513×513	False	5.5	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet500 (Lin et al. 2017b)	ResNet-101	~832×500	False	11.1	34.4	53.1	36.8	14.7	38.5	49.1
RefineDet512 (Zhang et al. 2018)	VGG-16	512×512	False	22.3	33.0	54.5	35.5	16.3	36.3	44.3
RefineDet512 (Zhang et al. 2018)	ResNet-101	512×512	True	-	41.8	62.9	45.7	25.6	45.1	54.1
CornerNet (Law and Deng 2018)	Hourglass	512×512	False	4.4	40.5	57.8	45.3	20.8	44.8	56.7
CornerNet (Law and Deng 2018)	Hourglass	512×512	True	-	42.1	57.8	45.3	20.8	44.8	56.7
M2Det (Ours)	VGG-16	512×512	False	18.0	37.6	56.6	40.5	18.4	43.4	51.2
M2Det (Ours)	VGG-16	512×512	True	-	42.9	62.5	47.2	28.0	47.4	52.8
M2Det (Ours)	ResNet-101	512×512	False	15.8	38.8	59.4	41.7	20.5	43.9	53.4
M2Det (Ours)	ResNet-101	512×512	True	-	**43.9**	**64.4**	**48.0**	**29.6**	**49.6**	**54.3**

➢ Compare with State-of-the-art

MS-COCO, test-dev detection results, compare with powerful two-stage detectors

two-stage:										
Faster R-CNN (Ren et al. 2015)	VGG-16	∼1000×600	False	7.0	21.9	42.7	-	-	-	-
OHEM++ (Shrivastava et al. 2016)	VGG-16	∼1000×600	False	7.0	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN (Dai et al. 2016)	ResNet-101	∼1000×600	False	9	29.9	51.9	-	10.8	32.8	45.0
CoupleNet (Zhu et al. 2017)	ResNet-101	∼1000×600	False	8.2	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN w FPN (Lin et al. 2017a)	Res101-FPN	∼1000×600	False	6	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN (Dai et al. 2017)	Inc-Res-v2	∼1000×600	False	-	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN (He et al. 2017)	ResNeXt-101	∼1280×800	False	3.3	39.8	62.3	43.4	22.1	43.2	51.2
Fitness-NMS (Tychsen-Smith and Petersson 2018)	ResNet-101	∼1024×1024	True	5.0	41.8	60.9	44.9	21.5	45.0	57.5
Cascade R-CNN (Cai and Vasconcelos 2018)	Res101-FPN	∼1280×800	False	7.1	42.8	62.1	46.3	23.7	45.5	55.2
SNIP (Singh and Davis 2018)	DPN-98	-	True	-	45.7	67.3	51.1	29.3	48.8	57.1
RetinaNet800 (Lin et al. 2017b)	Res101-FPN	∼1280×800	False	5.0	39.1	59.1	42.3	21.8	42.7	50.2
M2Det (Ours)	VGG-16	800×800	False	11.8	41.0	59.7	45.0	22.1	46.5	53.8
M2Det (Ours)	VGG-16	800×800	True	-	**44.2**	**64.6**	**49.3**	**29.2**	**47.9**	**55.1**

➢ Compare with State-of-the-art



Method	mAP	time
[A] YOLOv3 – 608	33.0	51
[B] SSD – 321	28.0	61
[B*] SSD – 321	28.2	22
[C] DSSD – 321	28.0	85
[D] R-FCN	29.9	85
[E] SSD – 513	31.2	125
[E*] SSD – 513	31.0	37
[F] DSSD – 513	33.2	156
[G] FPN FRCN	36.2	172
[H*] CornerNet	40.5	228
RetinaNet	39.1	198
[*]RefineDet	36.7	110
M2Det	**41.0**	**84.7**

(*) Tested on our machine for fair comparison

Inference speed comparison

Environment: NVIDIA Titan X, CUDA9.2, cuDNN 7.1.4, Pytorch0.4.0

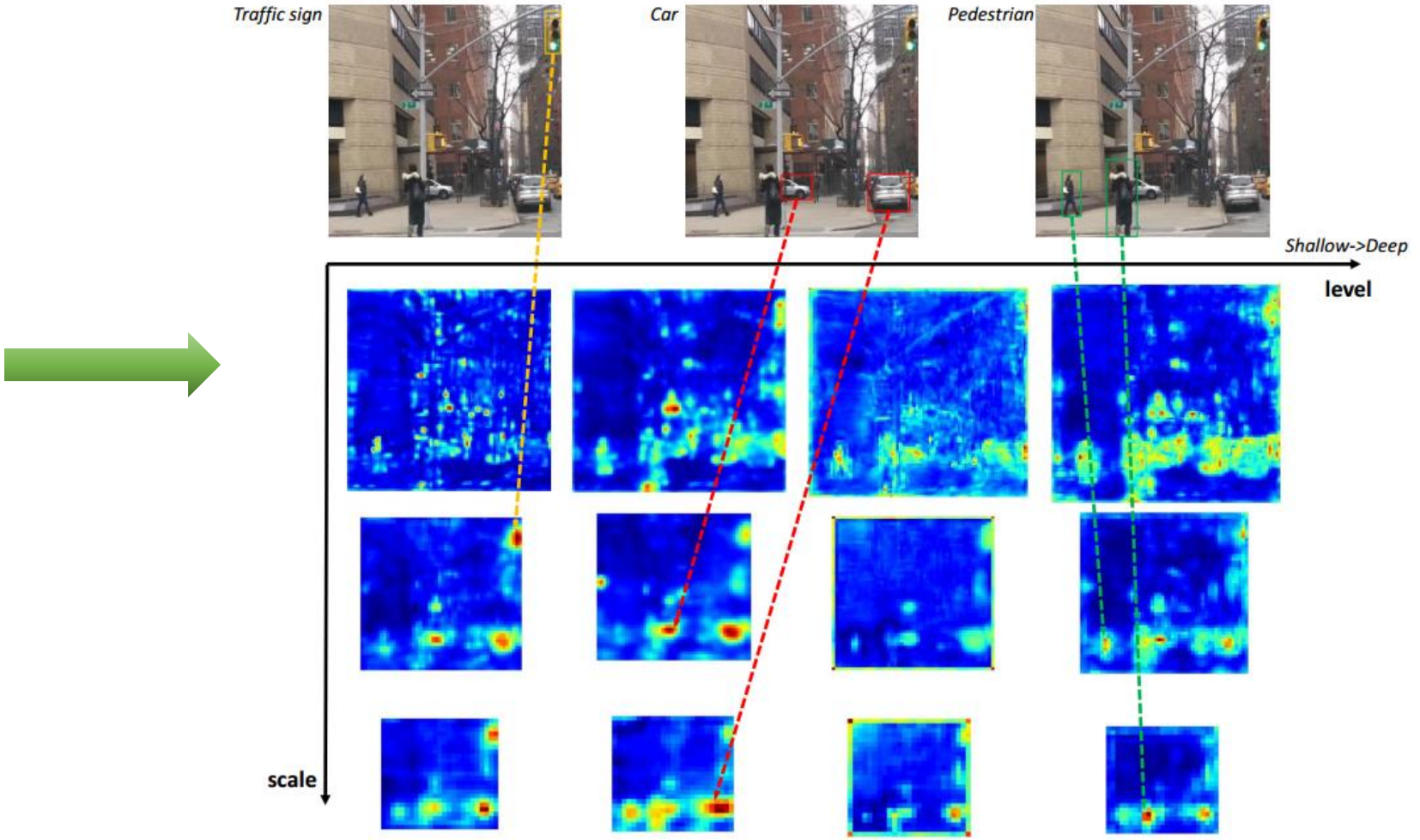
Compute method:
(cnntime_total + nmstime_total)/1000

For fair comparison, we reproduce the results of SSD321-ResNet101, SSD513-ResNet101, CornerNet and RefineDet on our machine.

Content

- Introduction
- Proposed method
- Experiments
- Summary

➢ Discussion: What is Multi-scale Multi-level Features



➢ Discussion: Why M2Det?

- The MLFPN can deepen the network, so that the gap between localization tasks and pre-training classification task can be dwindled
- The Multi-level pyramid can handle complex appearance variation across the object instances with equivalent scale.
- The results of VGG based M2Det largely get large improvements compared with SSD, so M2Det can remedy the deficiency of weak backbones. This also benefit to pretrain-free situations.

Question&Answer?