DIVE INTO CODE

The title of paper

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

https://arxiv.org/pdf/1811.04533.pdf

One sentence described about the paper

Summary

By Noritsugu Yamada

2019/02/6

DIVE INTO CODE

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

Qijie Zhao1 , Tao Sheng1 ,Yongtao Wang1∗ , Zhi Tang1 , Ying Chen2 , Ling Cai2 and Haibin Ling3 1 Institute of Computer Science and Technology, Peking University, Beijing, P.R. China 2AI Labs, DAMO Academy, Alibaba Group 3Computer and Information Sciences Department, Temple University

1, Summary
2, What is ~ …??
3, Experiments, conclusion, and discussion

https://arxiv.org/pdf/1811.04533.pdf

Method of improving the quality of synthetic images

Summary

By Noritsugu Yamada
2019/02/6

Summary

DIVE INTO CODE

Conclusion:

多段階特徴ピラミッドネットワーク（MLFPN）と呼ばれる新しい方法を
提案して、SSDのアーキテクチャに統合し、M2Detすることで
one-stageの検出器ではstate-of-the-artの検出性能を出した!
従来の特徴ピラミッドを2つの点で改良した!

What is this thesis for?

物体検出のbackborn（最初のCNN）に
注目して新しい構造の特徴ピラミッドを構築した

How to verified whether it is valid?

有名な様々な検出器と精度mAPと速度FPNを
比較して評価

Where is an important point compared to previous researches?

従来の検出器はただbackbornの上にのせてるだけ
MLFPNは高次な特徴を得るとともに
マルチスケールの処理が素晴らしくなった

Is there discussions?

特になし

Where are the key points of technology and method?

MLFPNはFFM,TUM,SFAMという3つの構造を持つ

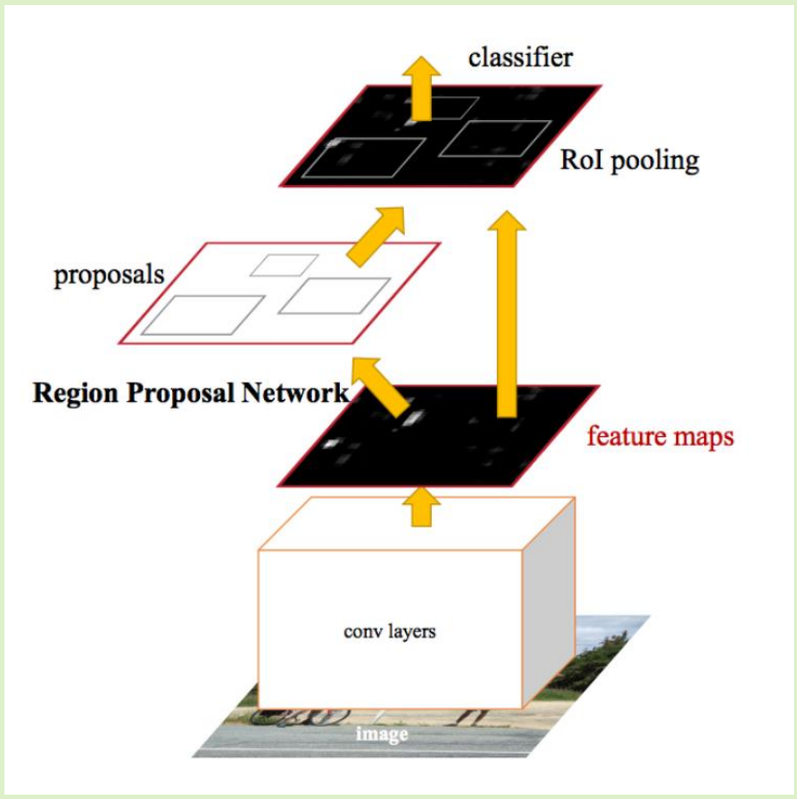
Which reserches should I read next?

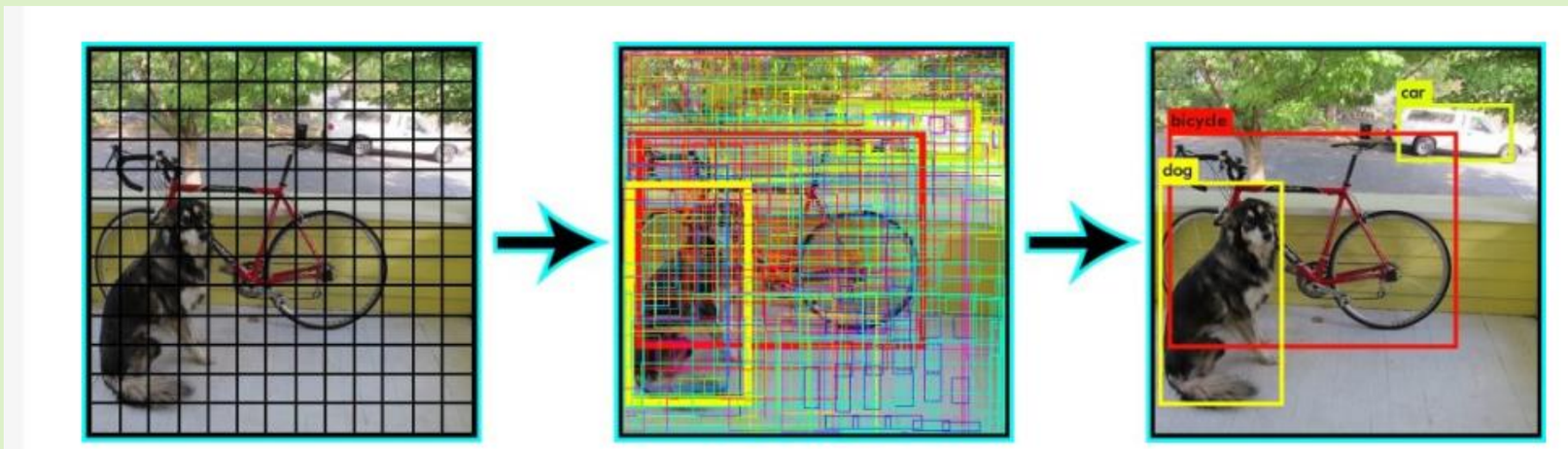
SSD: single shot multibox detector. In ECCV 2016, 21–37.

Table1の全ての検出器

What is ~~…??:

One-stage と Two-stage



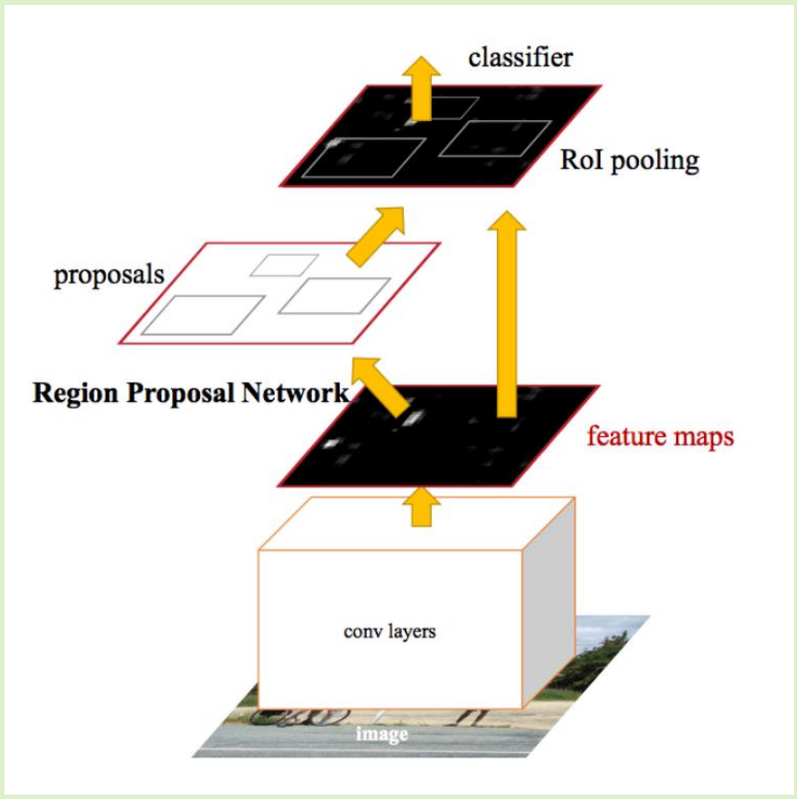


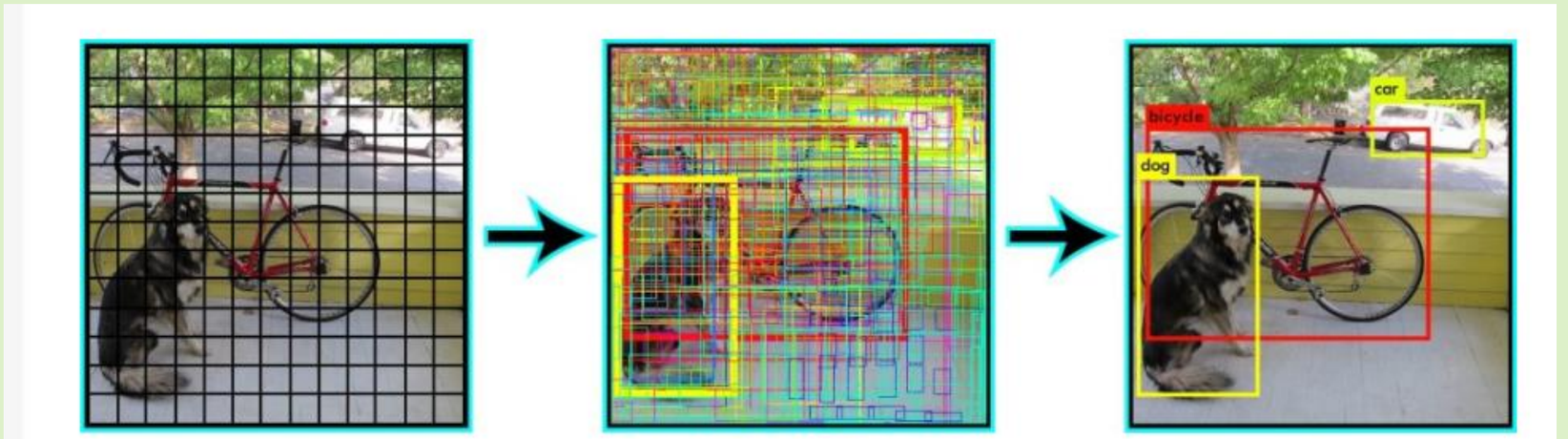
Two-stage（Faster R-CNN等）
bounding-box → クラス分類

One-stage （YOLO, SDD等）
画像をグリッドに分け直接bounding-boxとクラスを分ける
Two-stageより高速だが精度が劣るとされている

What is ~~…??:

One-stage と Two-stage



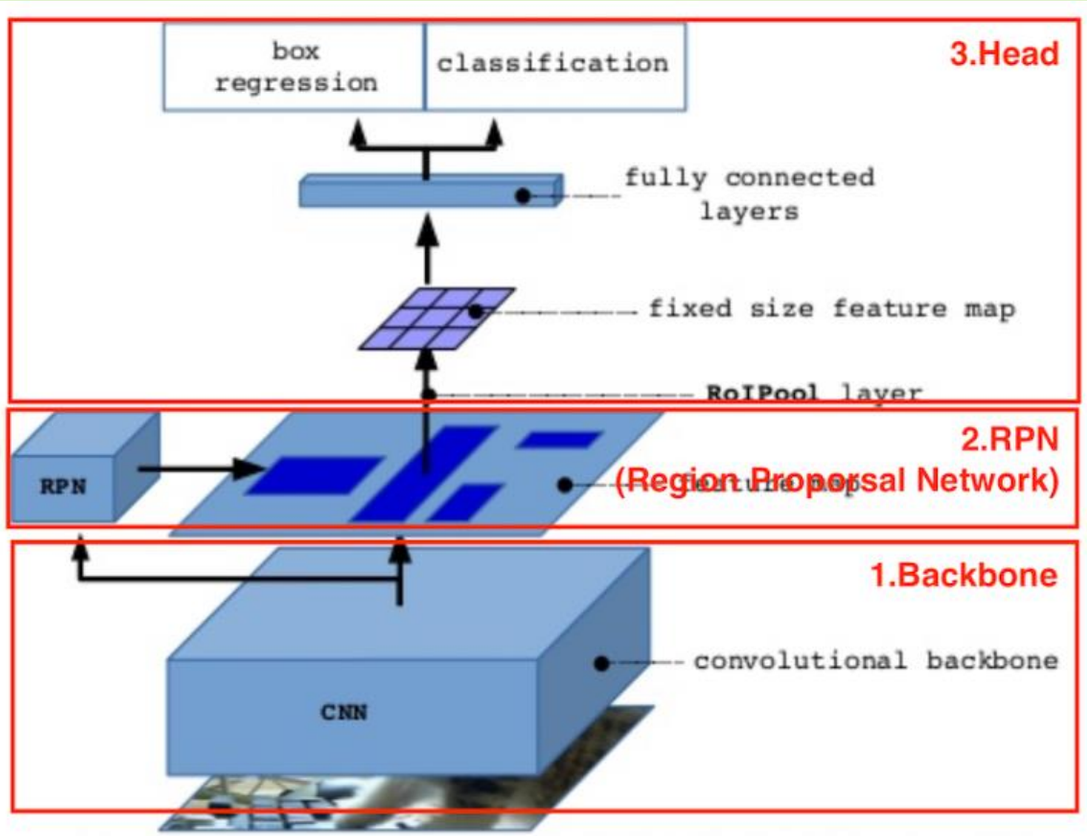


Two-stage（Faster R-CNN等）
bounding-box → クラス分類

One-stage （YOLO, SDD等）
画像をグリッドに分け直接bounding-boxとクラスを分ける
Two-stageより高速だが精度が劣るとされている

What is ~~…??:

問題提起 one-stage, two-stage共通



背景
インスタンス間のスケールが変動すると
上手く検出できないスケール問題があった

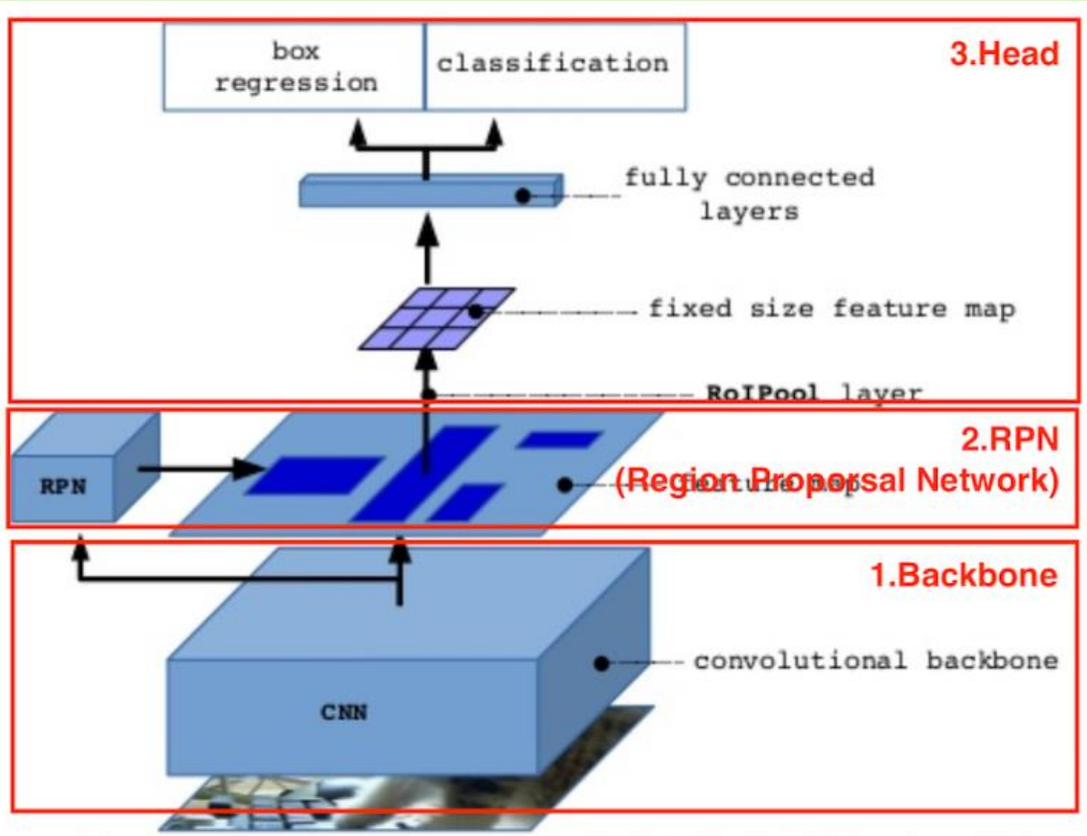
特徴ピラミッドを使うオブジェクト検出器
（Mask-RCNN等）は成果を上げていた

問題提起
従来の特徴ピラミッドを用いる検出器はオブジェクト分類タスクのために設計されている
バックボーン(VGG16やResNet等の入力画像の特量を抽出する役割)
のマルチスケールピラミッドアーキテクチャに従って特徴ピラミッドを単に構築するだけ
（分類器の上に乗せてるだけ）

→いくつかの制限が生じている

DIVE INTO CODE

問題提起



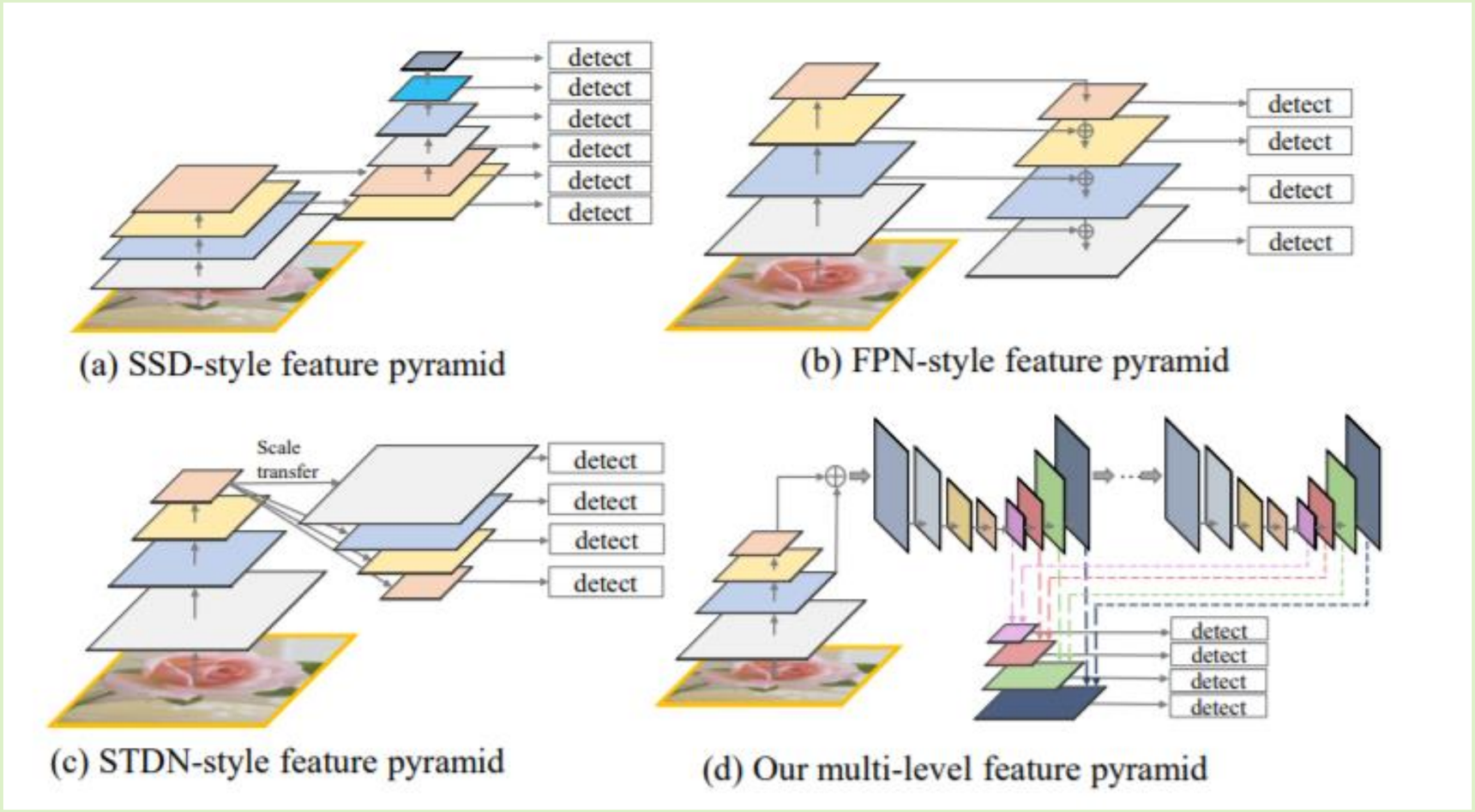
問題提起 制限とは…

1. ピラミッド内の特徴マップは、オブジェクト検出タスクにとって十分に代表的な特徴ではない,
→オブジェクト分類タスクのための分類器で抽出しているので検出タスクとしては不十分

2. ピラミッド内の各フィーチャマップはバックボーンのシングルレベル（浅い層）のレイヤから構築されている
→複雑な特徴を捉える深層の特徴表現が反映されていない

What is ~~…??:

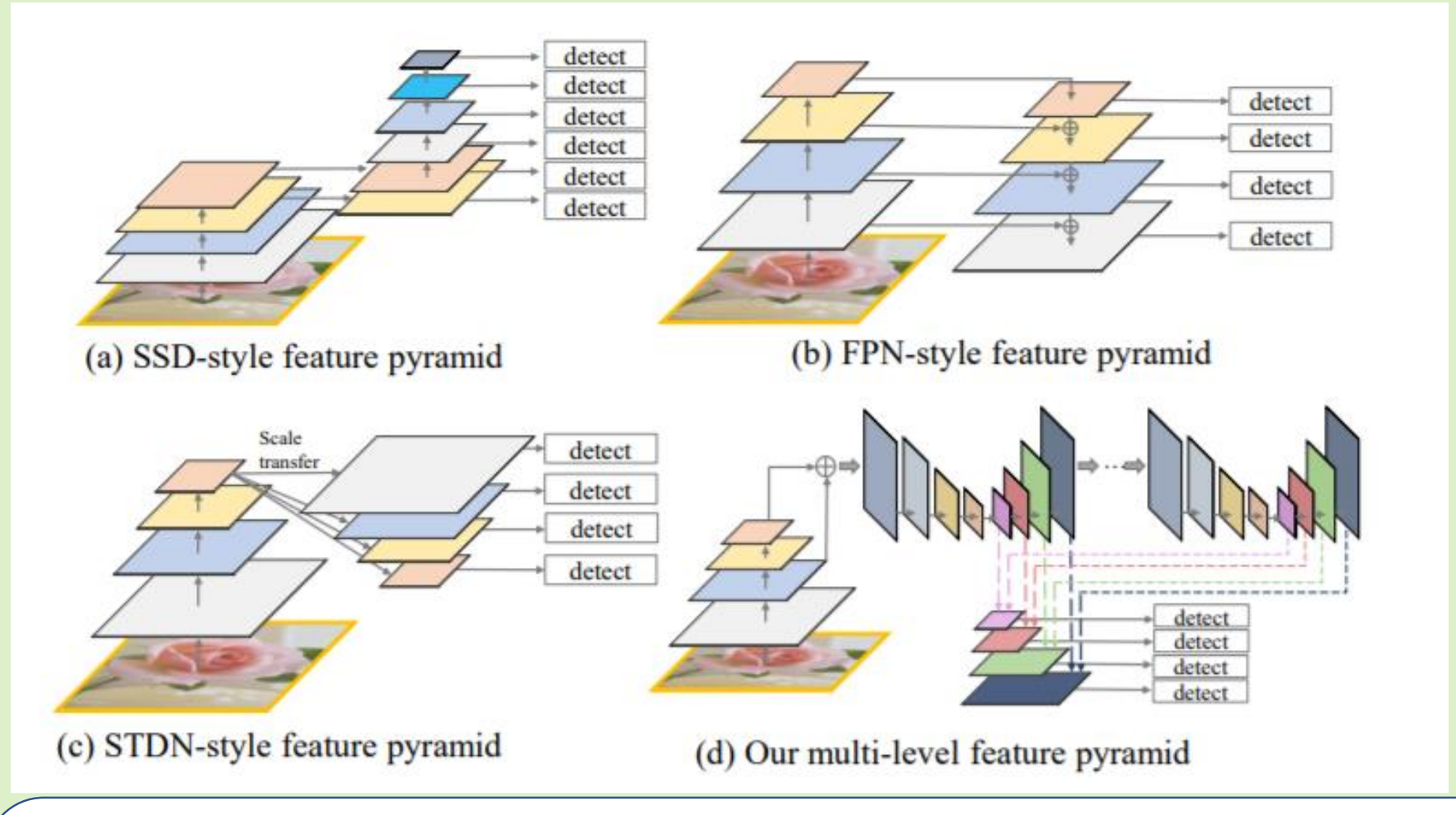
Introductions



ＳＳＤ（Ｌｉｕら、２０１６）は、特徴ピラミッドを構築するために、バックボーンの２つの層（ＶＧＧ１６）およびストライド２畳み込みによって得られる４つの追加の層を直接かつ独立して使用

What is ~~…??:

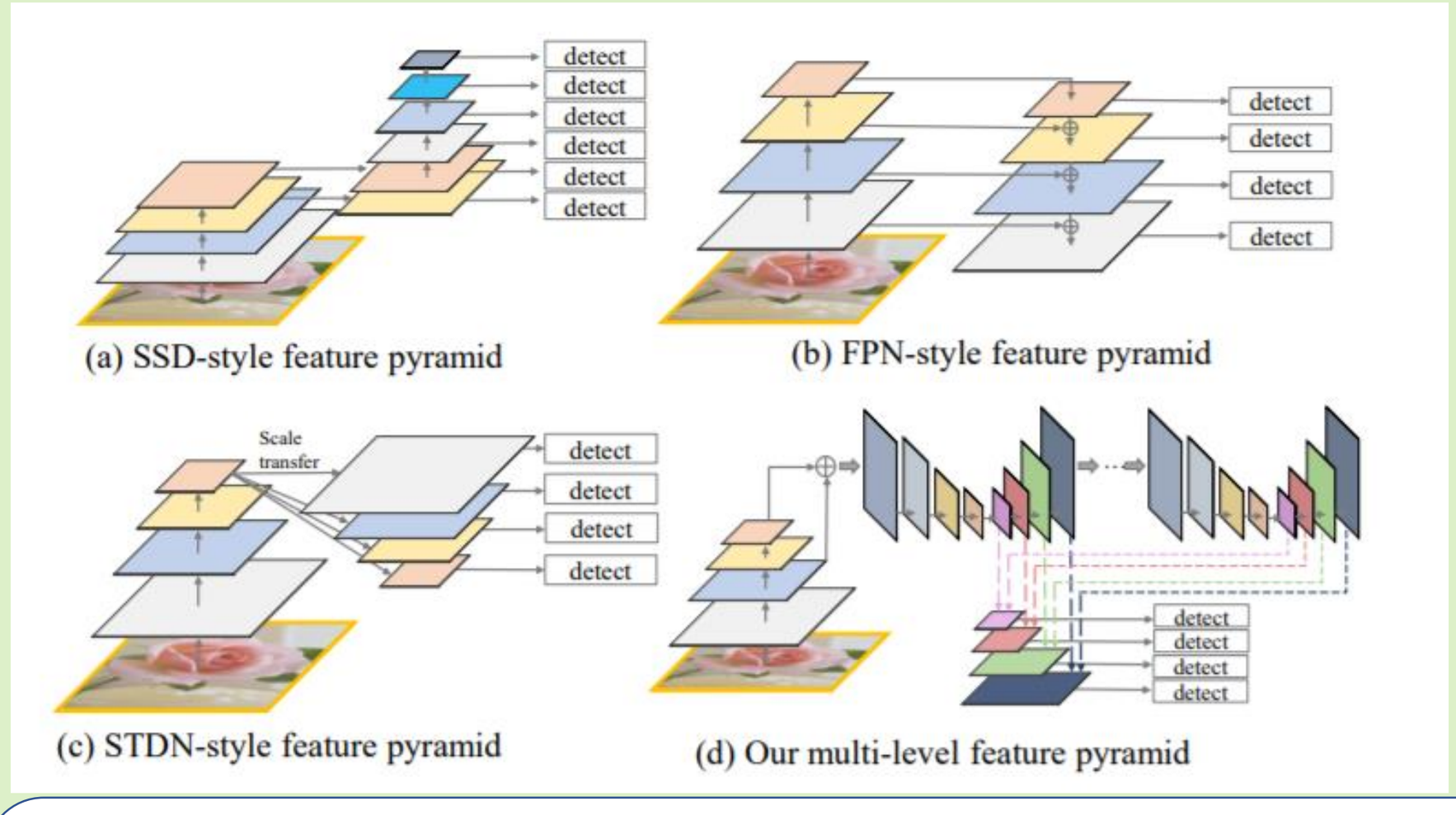
Introductions



FPN（Lin et al.2017a）は、深い層と浅い層をトップダウン方式で融合することによって特徴ピラミッドを構築

What is ~~…??:

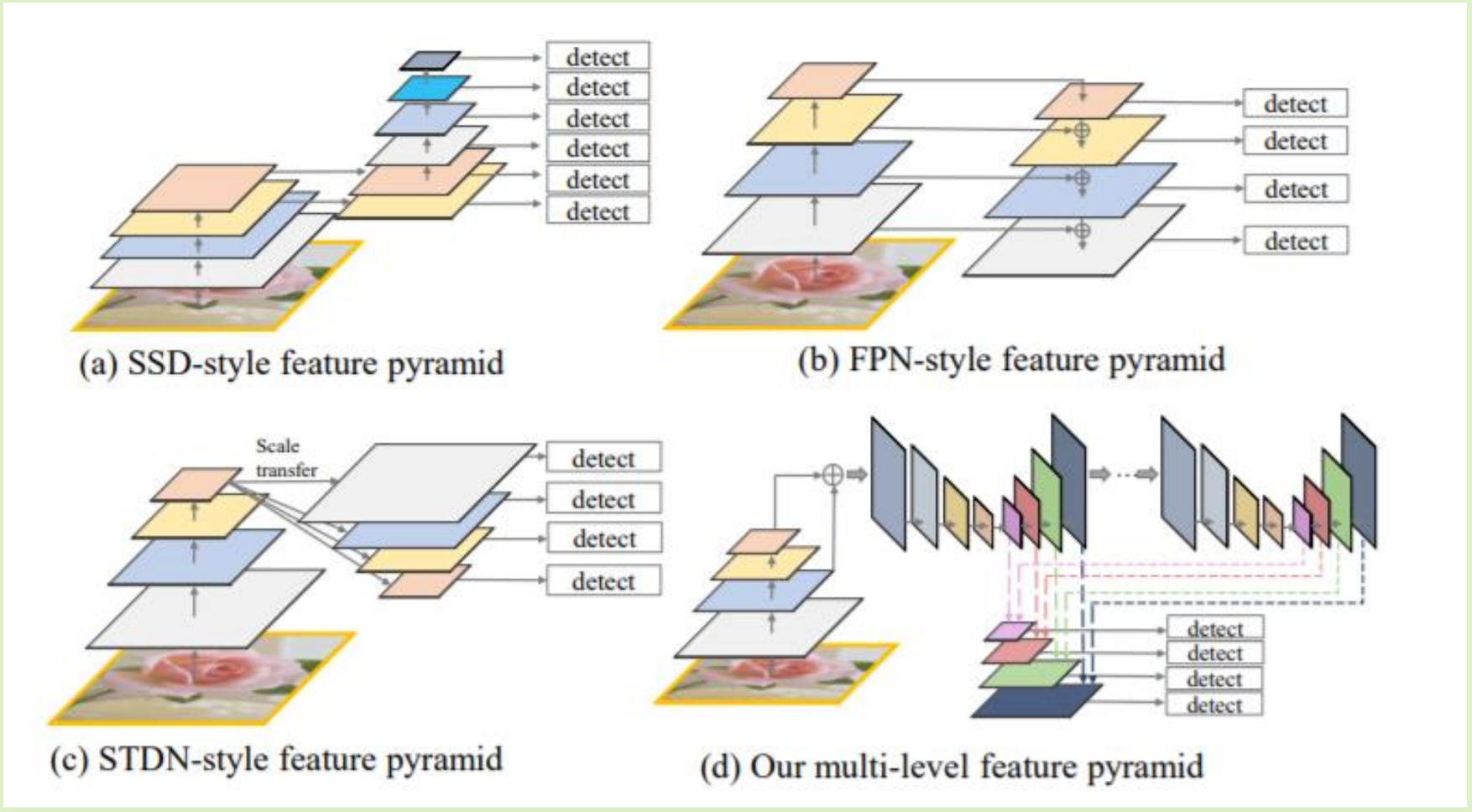
Introductions



STDN（Zhou et al.2018）は、DenseNetの最後の密ブロック（Huang et al.2017）のみを使用して、プールとスケール転送操作によって特徴ピラミッドを構築

What is ~~…??:

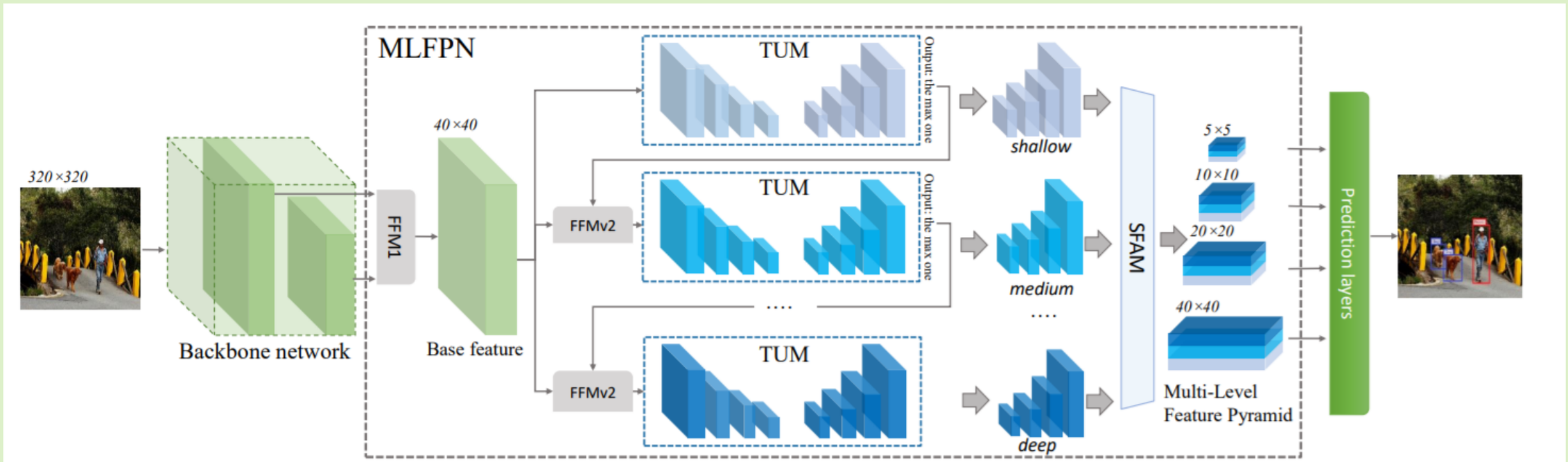
Introductions



M2Detは、バックボーン（VGG or ResNet）とマルチレベル特徴ピラミッドネットワーク（MLFPN）を利用して入力画像から特徴を抽出

What is ~~…??:

MLFPN

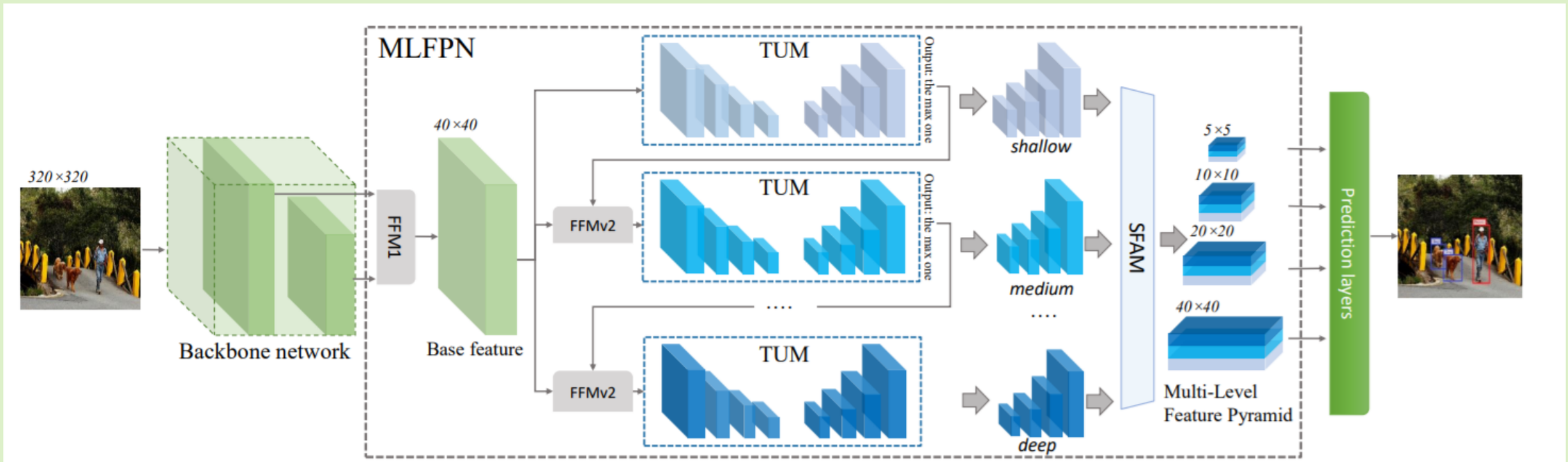


M2Detは、バックボーンとマルチレベル特徴ピラミッドネットワーク（MLFPN）を利用して入力画像から特徴を抽出

FFM(Feature Fusion Modules)v1はバックボーンの機能マップを融合してBase featureを生成

What is ~~…??:

MLFPN

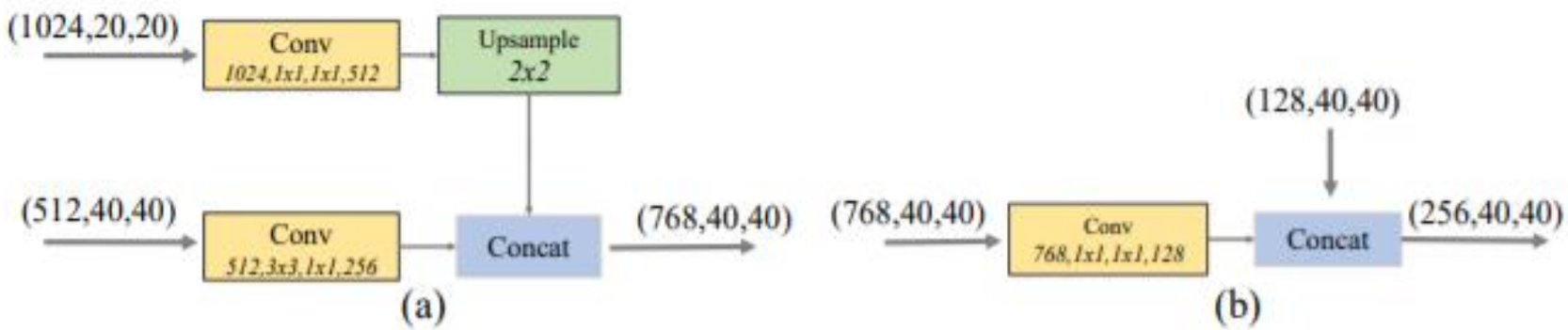


各TUMはマルチスケールフィーチャのグループを生成し、次に交互ジョイントTUM(Thinned U-shape Modules)とFFMv2はマルチレベルマルチスケールフィーチャを抽出

SFAMは機能を複数レベルの機能ピラミッドに集約(実際には、6つのスケールと8つのレベル)

What is ~~…??:

FFM



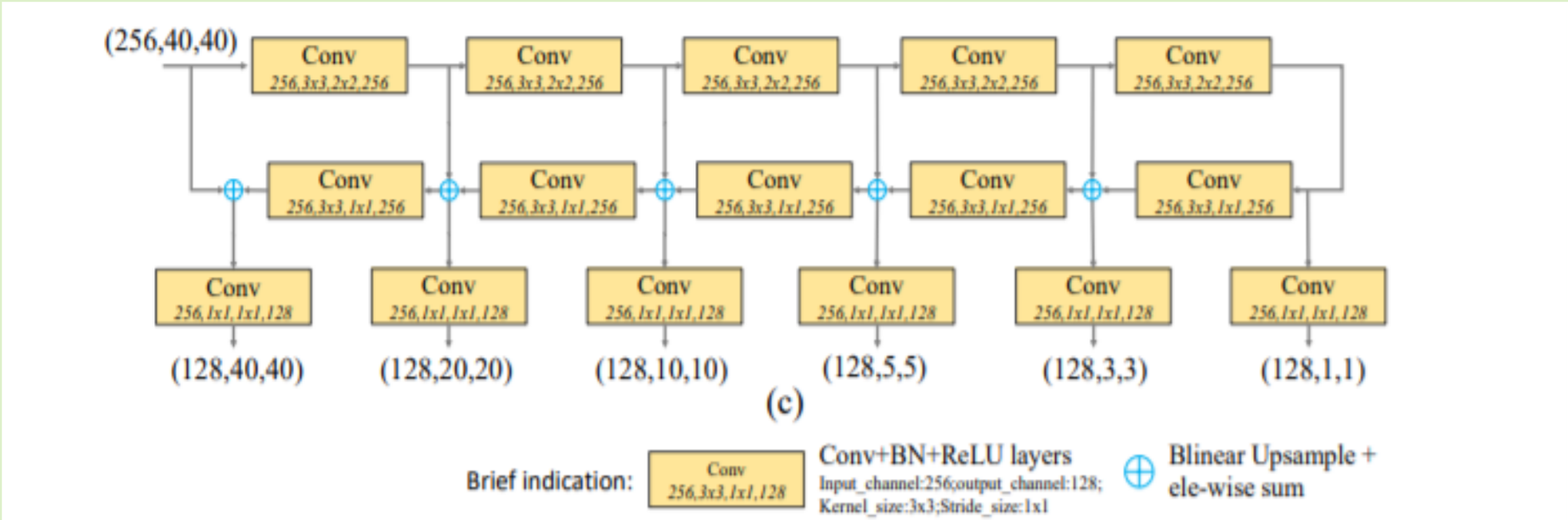
(a) FFMv1,
(b) FFMv2,

FFMはM2Detのさまざまなレベルの特徴を融合するモジュール
入力特徴のチャネルを圧縮する１×１畳み込み（次元削減）をし、特徴マップを集約するために連結演算を使用

FFMv1はバックボーン内のスケールが異なる2つのフィーチャーマップを入力として使用するため、連結操作の前に深い特徴を同じスケールに再スケーリングするためにアップサンプルする
FFMv2は前のTUMのBaseFeature とTUM(sallow)の最大出力フィーチャーマップ（同じ縮尺のもの）を入力として受け取り、次のTUMのための融合特徴を生成

What is ~~…??:

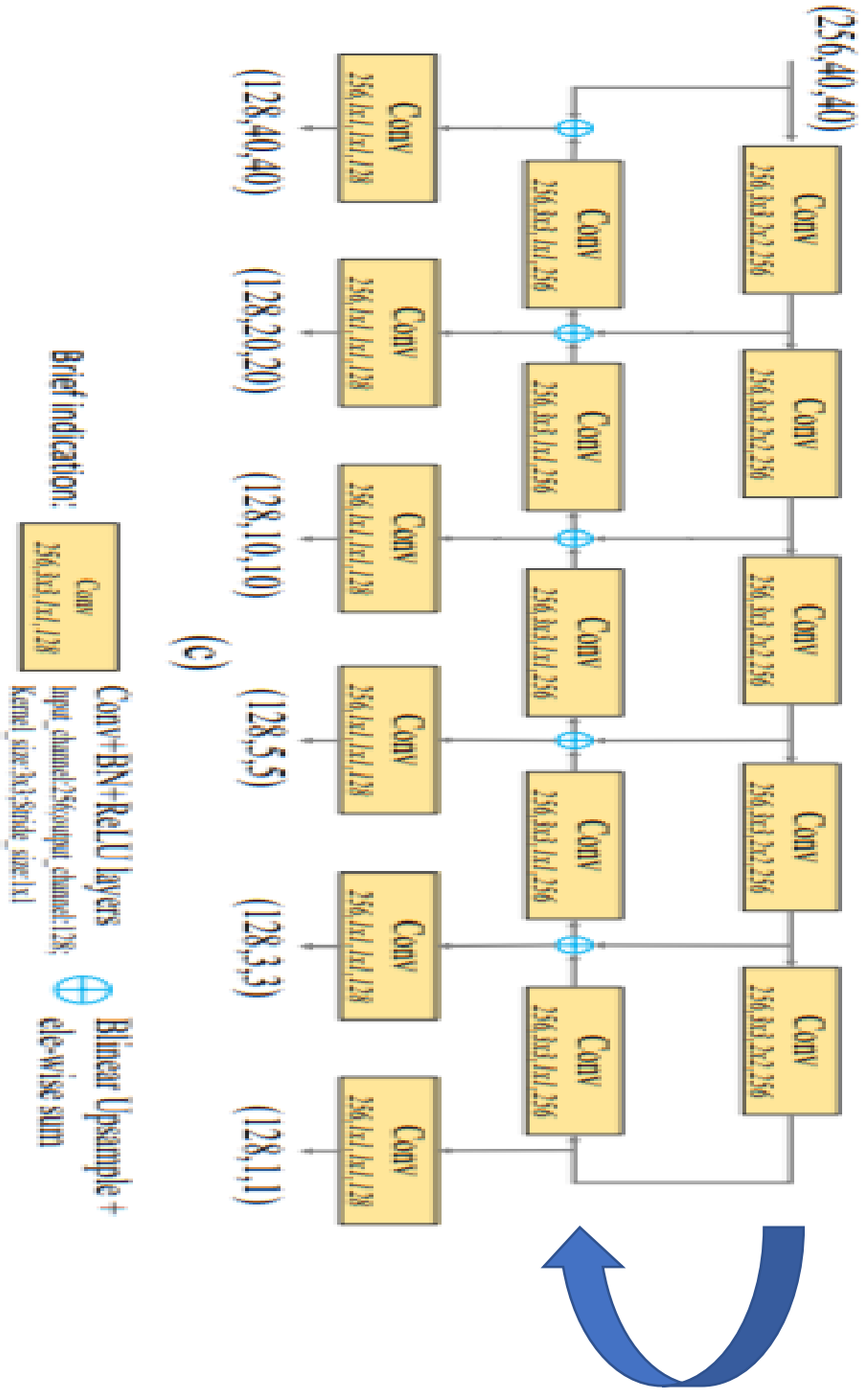
TUM

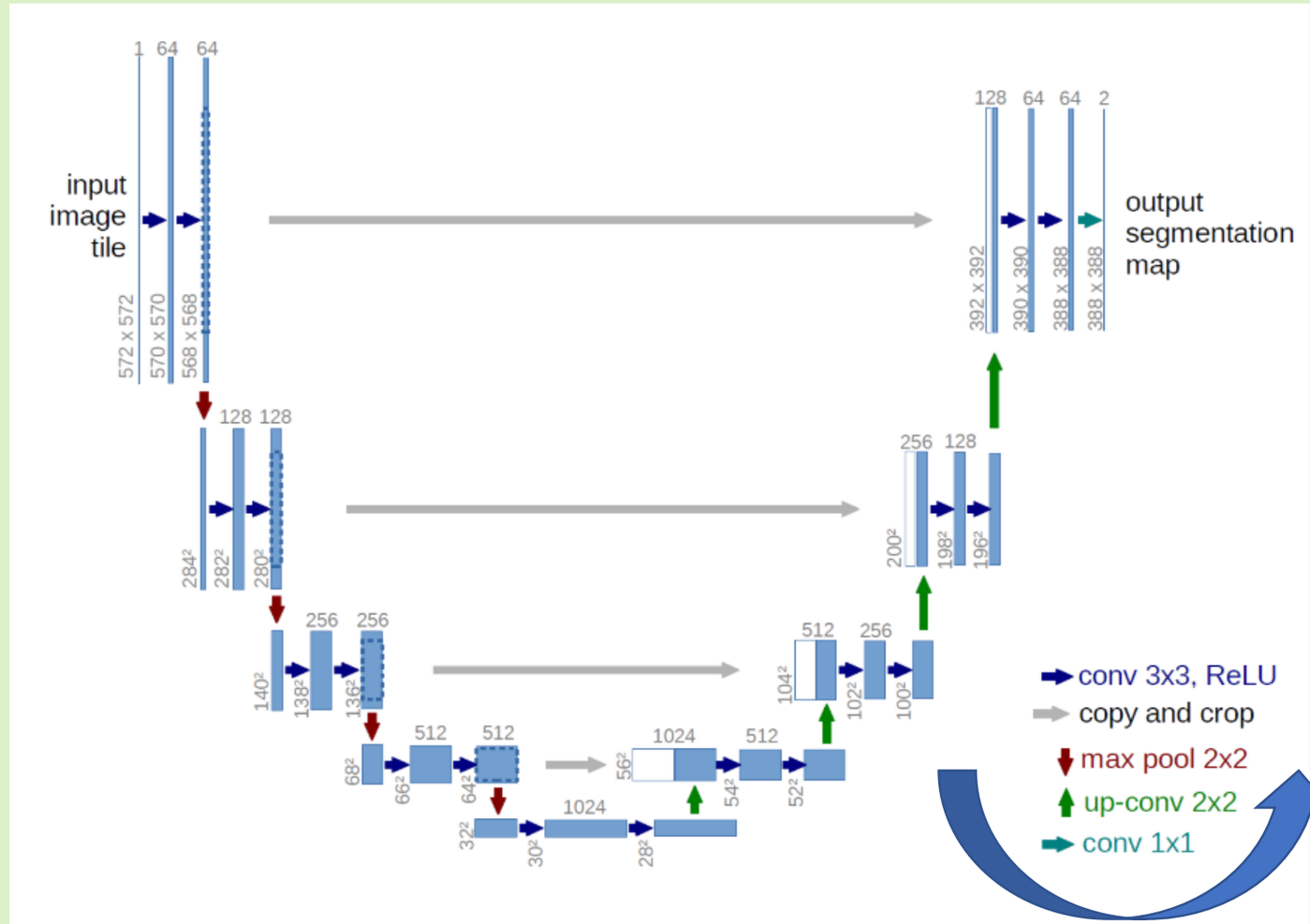


TUM
エンコーダはストライド2の一連の3×3畳み込みレイヤ
学習能力を高め、特徴の滑らかさを保つためにデコーダブランチでのアップサンプルおよび要素ごとの和演算の後に1x1の畳み込みレイヤを行う
TUMのデコーダ内のすべての出力は、現在のレベルのマルチスケール特徴

What is ~~…??:

TUM



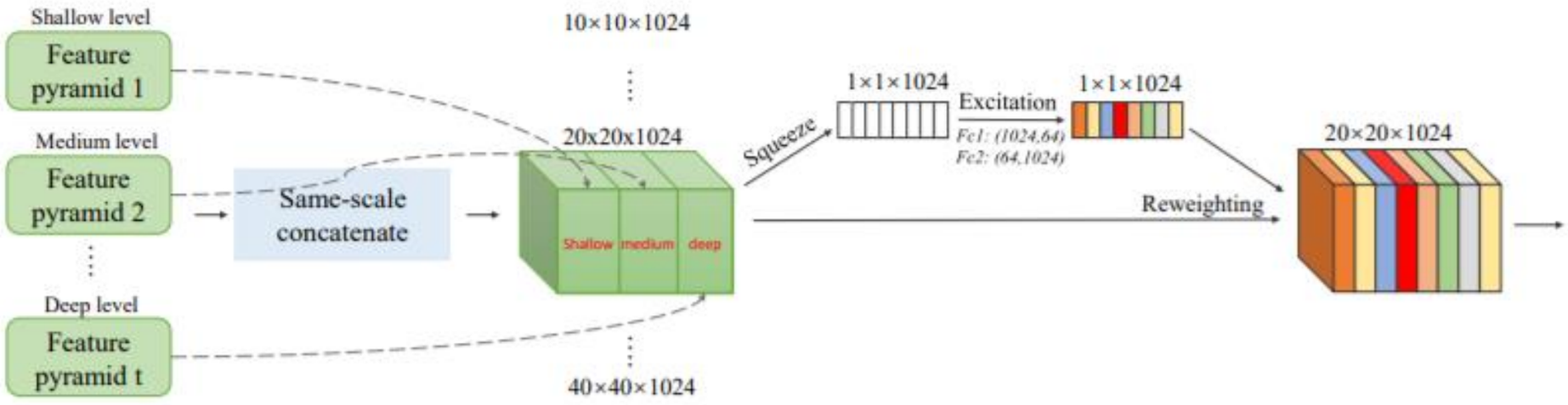


TUMの図を縦にすると右のU-netのような
構造とわかる
目的がU-netと違ってマルチスケールへの変換

DIVE INTO CODE

What is ~~…??:

SFAM



$$Xi = Concat(x\,1\,i\,, x\,2\,i\,, ..., x\,L\,i\,) \in RWi \times Hi \times C$$

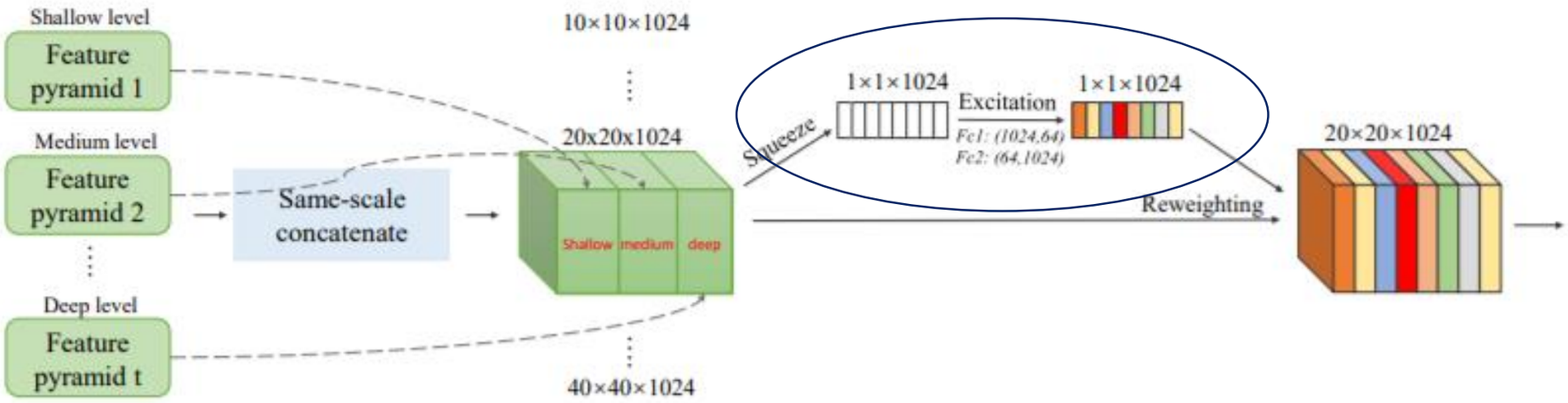
SFAM
第１段階は、チャネル寸法に沿って等価スケールの特徴を一緒に連結する
ＴＵＭによって生成されたマルチレベルマルチスケール特徴をマルチレベル特徴ピラミッドに集約する

第2段階では、チャンネルごとに注目するモジュールを導入して、特徴が最も依存しているチャンネルに焦点を当てるSEブロックに続いて、グローバルアベレージプーリングを使用する

DIVE INTO CODE

What is ~~…??:

SFAM



$$Xi = Concat(x\,1\,i\,, x\,2\,i\,, ..., x\,L\,i\,) \in RWi \times Hi \times C$$

SE
チャネル間の相互依存性を明示的にモデル化することによってチャネルごとの特徴応答を適応的に再較正する「Squeeze and-Excitation」（SE）ブロックアーキテクチャ
SE block (Hu, Shen, and Sun 2017), https://arxiv.org/pdf/1709.01507.pdf

What is ~ …??:

検証

Method	Backbone	Input size	MultiScale	FPS	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L
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
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**
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**
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**

M2det 8 TUMを使用し、各TUMに256チャネル

Multi_scaleのVGGバックボーンを持つM2Det-320が38.9のAPより大きいサイズの検出器を凌駕するよ

Single scaleではResNetを使うことでMask-RCNNに匹敵する38.8のAPでスピードは余裕で勝ってる15.8FPS

Multiscaleではすべてのone_stage検出器を超える44.2 AP

さらにState-of-the-artな検出器よりパラメータ少ない

Method	Backbone	Input size	MultiScale	FPS	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L
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
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**
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**
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**

What is ~ …??:

検証

+ 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**

Table 2: Ablation study of M2Det. The detection results are evaluated on `minival` set

M2Detは複数のサブコンポーネントで構成
最終的なパフォーマンスに対するそれぞれの有効性を検証
ベースラインは、320×320の入力サイズとVGG-16縮小バックボーンを使用した

What is ~ …??:

検証

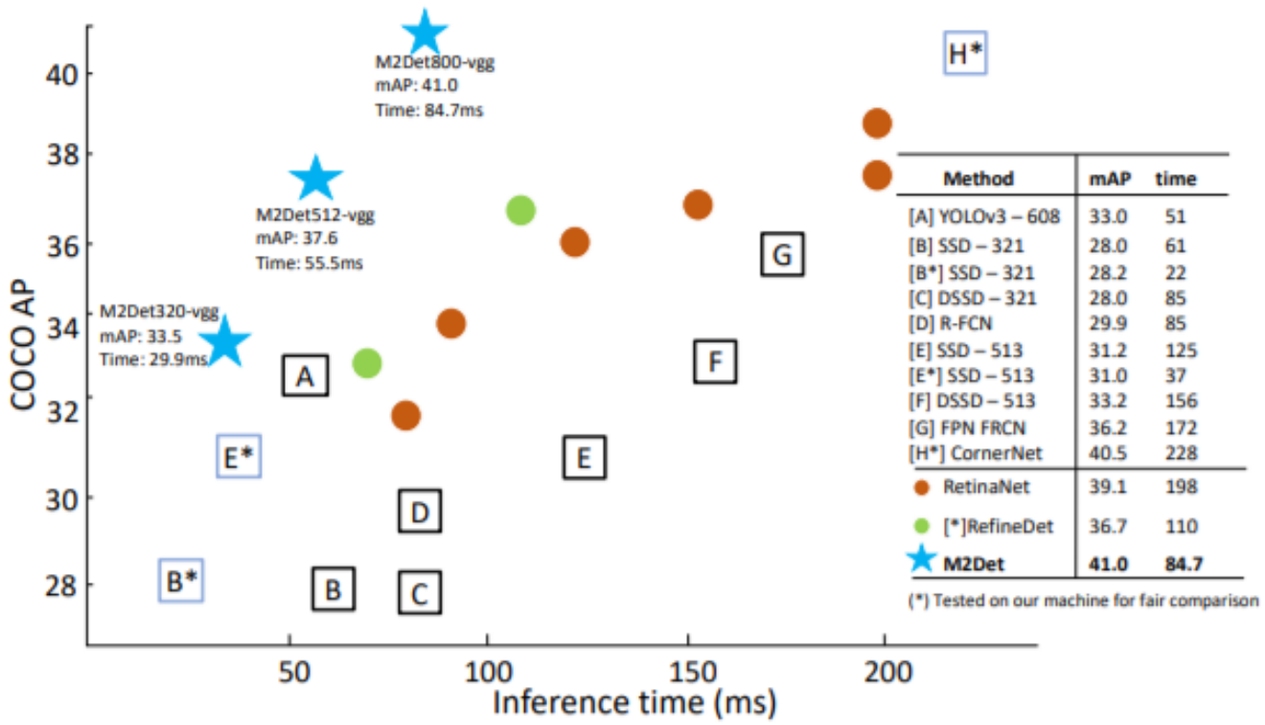
TUMs	Channels	Params(M)	AP	AP_{50}	AP_{75}
2	256	40.1	30.5	50.5	32.0
2	512	106.5	32.1	51.8	34.0
4	128	34.2	29.8	49.7	31.2
4	256	60.2	31.8	51.4	33.0
4	512	192.2	33.4	52.6	34.2
8	128	47.5	31.8	50.6	33.6
8	256	98.9	33.2	52.2	35.2
8	512	368.8	34.0	52.9	36.4
16	128	73.9	32.5	51.7	34.4
16	256	176.8	33.6	52.6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

Table 3: Different configurations of MLFPN in M2Det. The backbone is VGG and input image is 320×320.

チャネルを固定（256）→ＴＵＭの数↑　検出精度↑
TUMの数を固定　→チャネル数↑　検出精度↑

What is ~ …??:

検証



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**

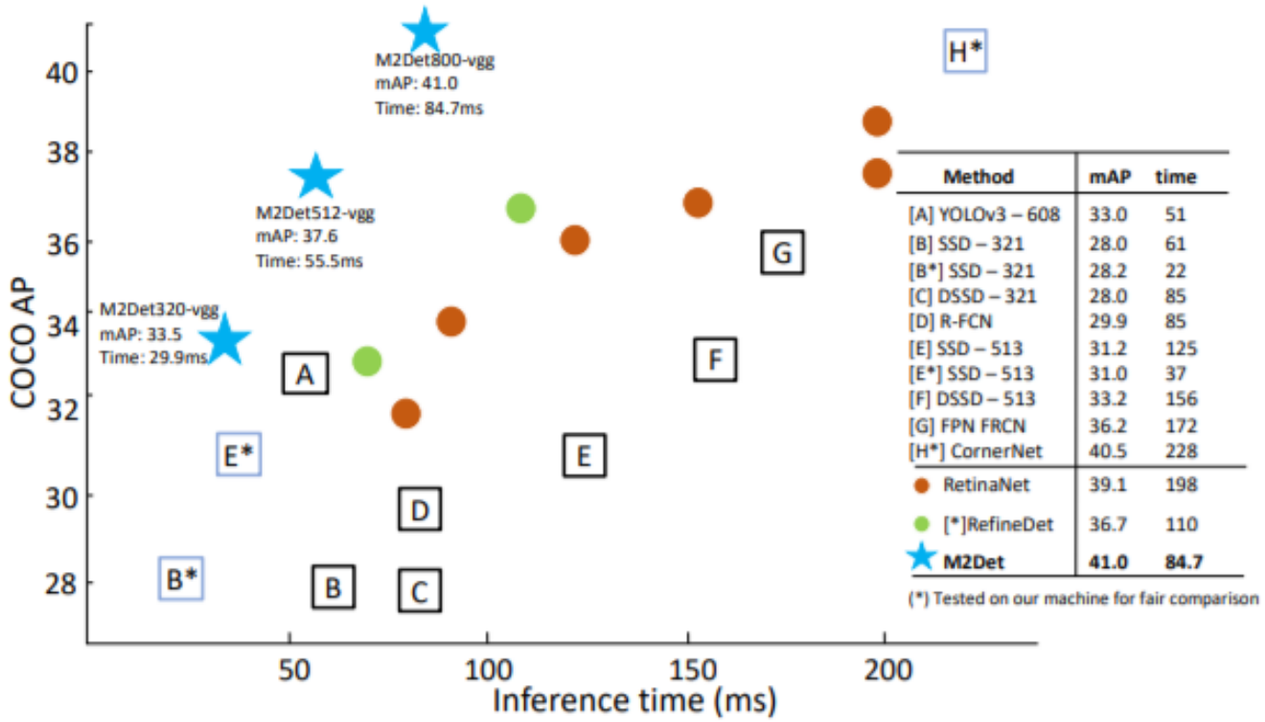
(*) Tested on our machine for fair comparison

Figure 5: Speed (ms) vs. accuracy (mAP) on COCO *test-dev*.

VGG16-をM2Detに縮小して組み立て、
入力サイズ320×320の高速バージョンM2Det、
512×512入力サイズの標準バージョンM2Det、
および800×800入力サイズの最も正確なバージョンM2Det

What is ~ …??:

検証



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

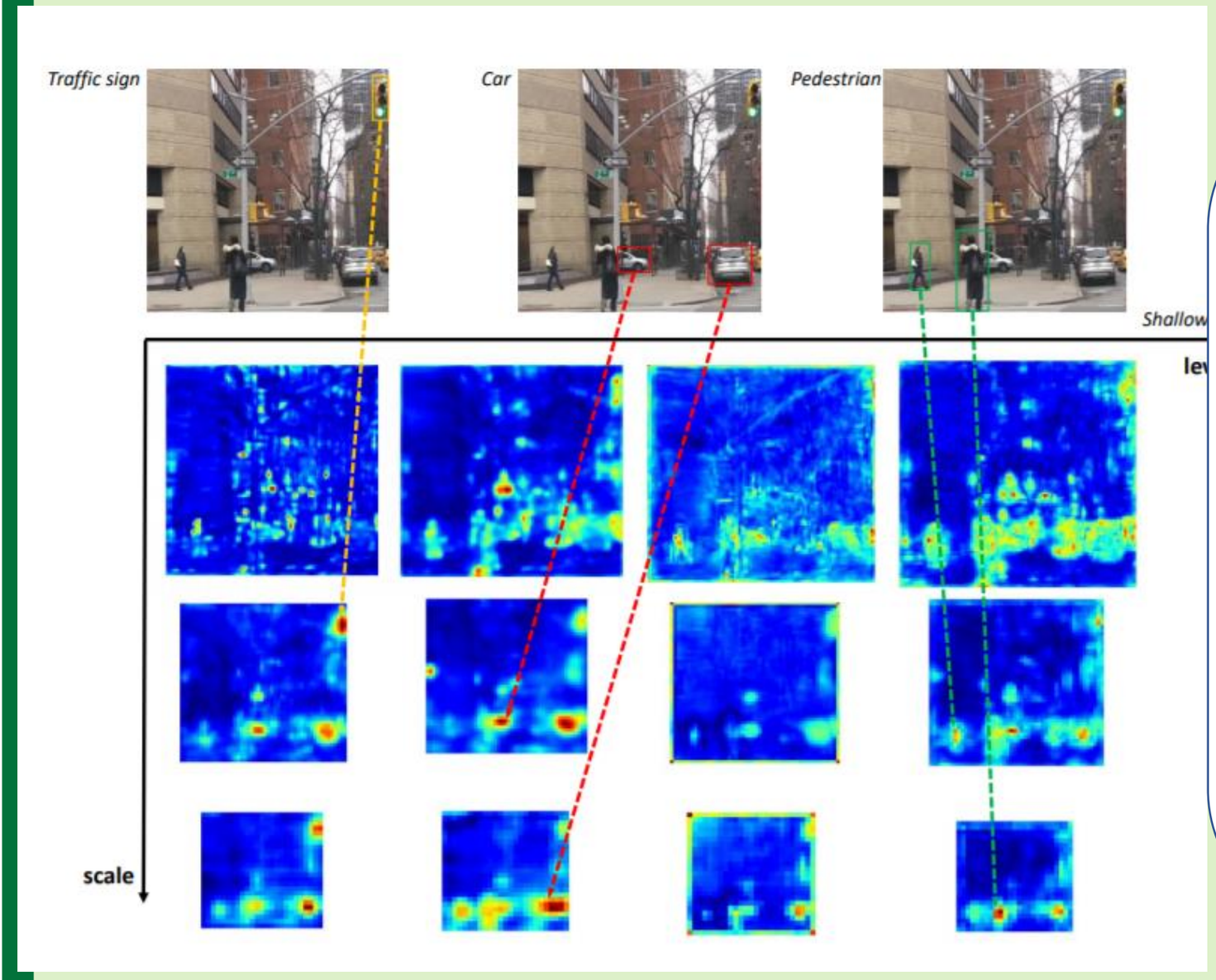
Figure 5: Speed (ms) vs. accuracy (mAP) on COCO *test-dev*.

非常に優れた速度精度曲線

VGG16-をM2Detに縮小して組み立て、
入力サイズ320×320の高速バージョンM2Det、
512×512入力サイズの標準バージョンM2Det、
および800×800入力サイズの最も正確なバージョンM2Det

What is ~~…??:

Visual



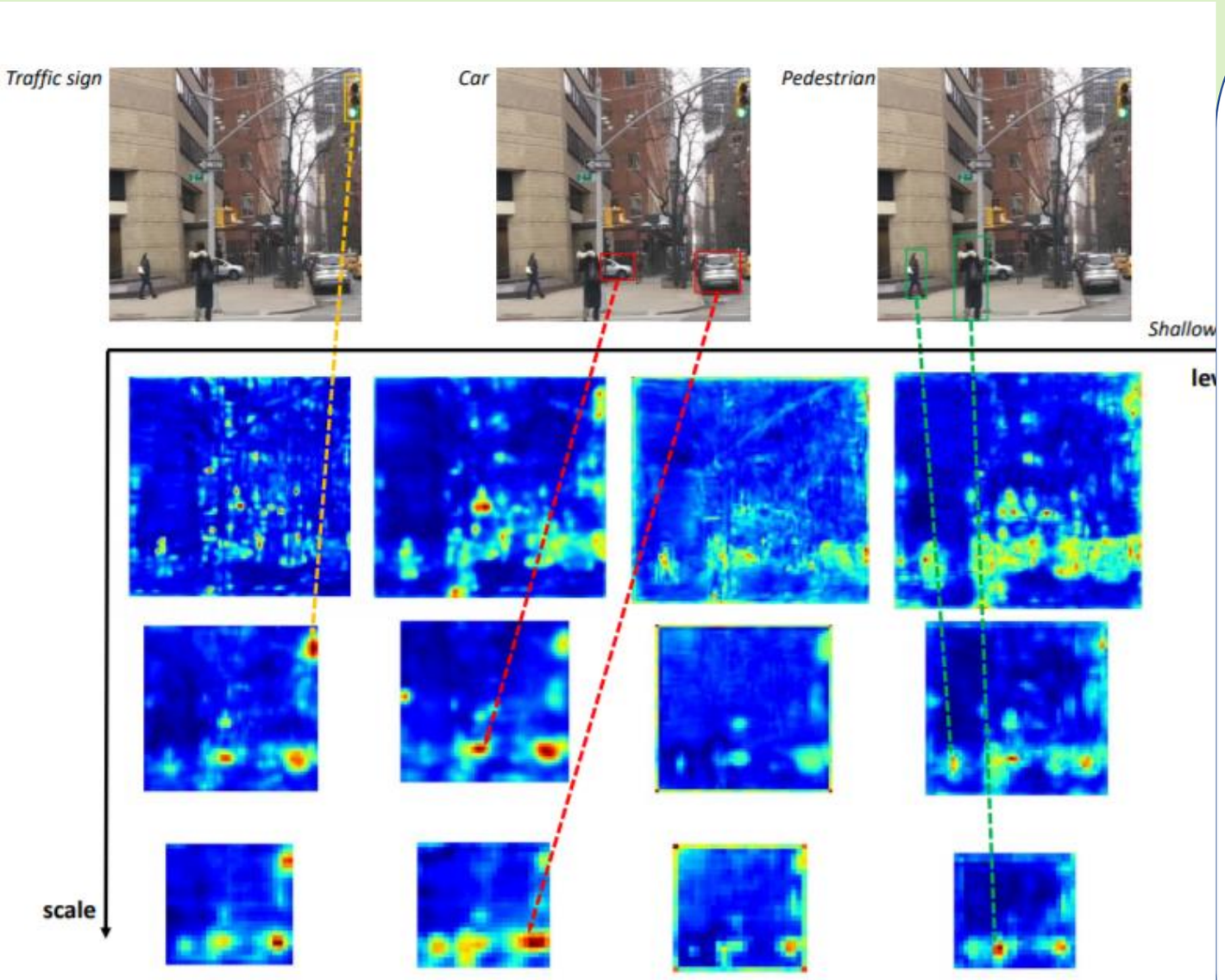
異なる縮尺と大きな外観変動を持つ物体を検出するための効果的な特徴を学習できることを検証

縮尺とレベル次元に沿って分類Conv レイヤの活性化値を視覚化

2人の人物、2台の車、信号機
人物と車は大きさが異なる

What is ~~…??:

Visual



スケール
信号機、小さい方の人と小さい方の車は画像と同じ縮尺の特徴図で最も強い活性化値
大きい方の人と大きい方の車は拡大した特徴図で最も強い活性化値

Deep
人、車および信号機は、それぞれ最高レベル、中レベル、最低レベルの特徴マップにおいて最も強い活性化値

マルチレベル機能を使用する必要性!!

DIVE INTO CODE

Conclusion　まとめ１

本研究では、多段階特徴ピラミッドネットワーク（MLFPN）と呼ばれる新しい方法を提案して、異なるスケールの物体を検出するための効果的特徴ピラミッドを構築

第１に、バックボーンによって抽出されたマルチレベルフィーチャ（すなわち、複数のレイヤ）は、ベースフィーチャとしてＦＦＭｖ1によって融合

第２に、Base featuresは、互に結合されたＴＵＭおよびＦＦＭｖ２のブロックに供給され、マルチレベルマルチスケール特徴（各ＴＵＭのデコーダ層）を抽出

最後に、同じスケール（サイズ）を有する抽出されたマルチレベルマルチスケール特徴は、ＳＦＡＭによるオブジェクト検出のための特徴ピラミッドを構築するために集約

DIVE INTO CODE

Conclusion　まとめ２

なぜ優れているか（2つの改善点）

BackbornからBasefeatureを取り出してTUMを使用したことでバックボーンのみの層よりはるかに深い層で構成されるのでオブジェクト検出により高次な特徴を得た

ＳＦＡＭによって生成されたマルチレベル特徴ピラミッドの各特徴マップは、複数のレベルからのデコーダ層でスケールごとに、オブジェクトを検出するためのマルチレベル機能を使用する。これは、オブジェクトインスタンス間の外観の複雑さの変化を処理するのに適する

Summary

DIVE INTO CODE

Conclusion:

多段階特徴ピラミッドネットワーク（MLFPN）と呼ばれる新しい方法を
提案して、SSDのアーキテクチャに統合し、M2Detすることで
one-stageの検出器ではstate-of-the-artの検出性能を出した!
従来の特徴ピラミッドを2つの点で改良した!

What is this thesis for?

物体検出のbackborn（最初のCNN）に
注目して新しい構造の特徴ピラミッドを構築した

How to verified whether it is valid?

有名な様々な検出器と精度mAPと速度FPNを
比較して評価

Where is an important point compared to previous researches?

従来の検出器はただbackbornの上にのせてるだけ
MLFPNは高次な特徴を得るとともに
マルチスケールの処理が素晴らしくなった

Is there discussions?

特になし

Where are the key points of technology and method?

MLFPNはFFM,TUM,SFAMという3つの構造を持つ

Which reserches should I read next?

SSD: single shot multibox detector. In ECCV 2016, 21–37.

Table1の全ての検出器