Paper Review:

Feature Pyramid Networks for Object Detection

Hyunseung Jeon

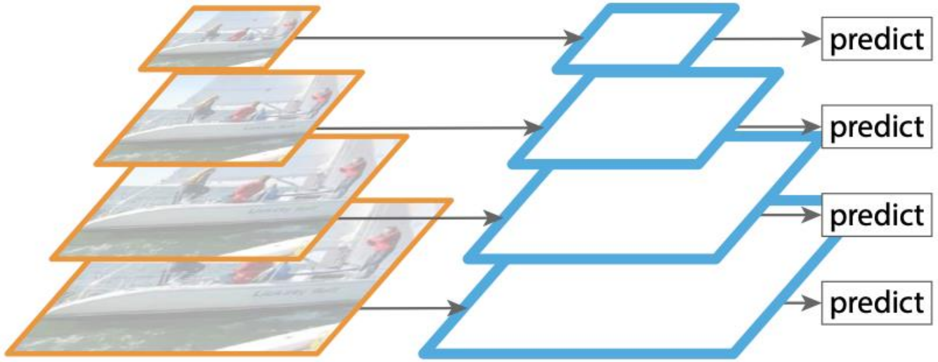
School of CSE, Kyungpook Nat’l Univ.

hyunseung.jeon@knu.ac.kr

Introduction

- Recognizing objects at vastly different scales is a fundamental challenge in computer vision.
- Scale-Invariant matters!
- Especially, detecting small objects...

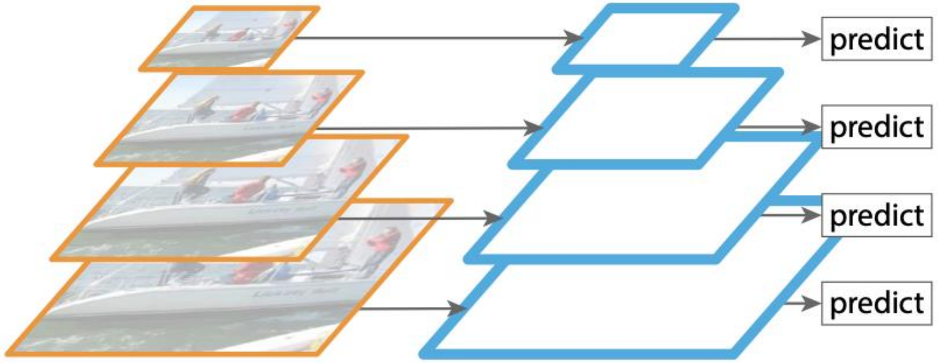
Previous Approaches



(a) Featurized image pyramid

- The principle advantage of featurizing each level of an image pyramid is that **it produces a multi-scale feature representation in which all levels are semantically strong**, including the high-resolution levels.
- Features are computed on each of the image scales independently, **which is slow**.

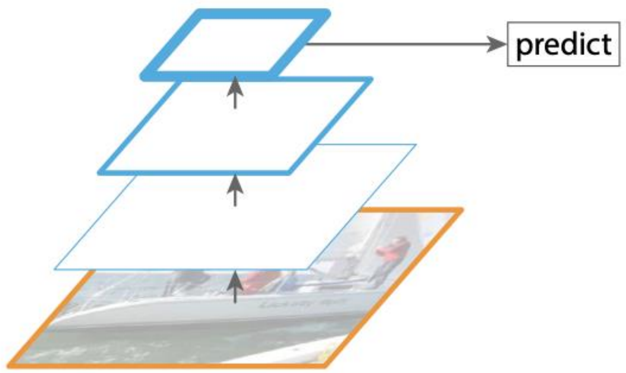
Previous Approaches



(a) Featurized image pyramid

- Inference time increases considerably, making this approach **impractical for real applications**.
- Training deep networks **end-to-end on an image pyramid is infeasible in terms of memory**, and so, if exploited, image pyramids are used only at test time.

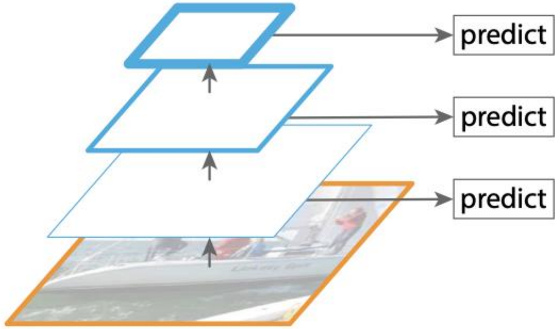
Previous Approaches



(b) Single feature map

- Used in YOLO v1
- ConvNets are also more robust to variance in scale.
- But even with this robustness, pyramids are still needed to get the most accurate results.
- Uses only single scale features → introduces large semantic gaps caused by different depths.

Previous Approaches



(c) Pyramidal feature hierarchy

- Used in SSD
- SSD-style pyramid would **reuse the multi-scale feature maps from different layers** computed in the forward pass and thus come free of cost.
- It **misses the opportunity to reuse the higher-resolution maps** of the feature hierarchy.
- “Only bottom-up pathway”, low performance on small object detection.

Goals

- Our goal is to leverage a ConvNet's pyramidal feature hierarchy, **which has semantics from low to high levels**, and build a feature pyramid with high-level semantics throughout.
- The goal of this paper is to naturally leverage the pyramidal shape of a ConvNet's feature hierarchy while **creating a feature pyramid that has strong semantics at all scales**.

- "creating a feature pyramid" → FPN is feature detector, not a single model that performs object detection.

Feature Pyramid Network

- **Low-resolution** has **semantically strong features**.
- **High-resolution** has **semantically weak features**.
- To achieve this goal, we rely on an architecture that **combines** low-resolution, semantically strong features with high-resolution, semantically weak features **via a top-down pathway and lateral connection**.
- The construction of our pyramid involves **a bottom-up pathway, a top-down pathway**, and **lateral connections**, as introduced in the following.

- Lateral connection = Skip-connection

Feature Pyramid Network

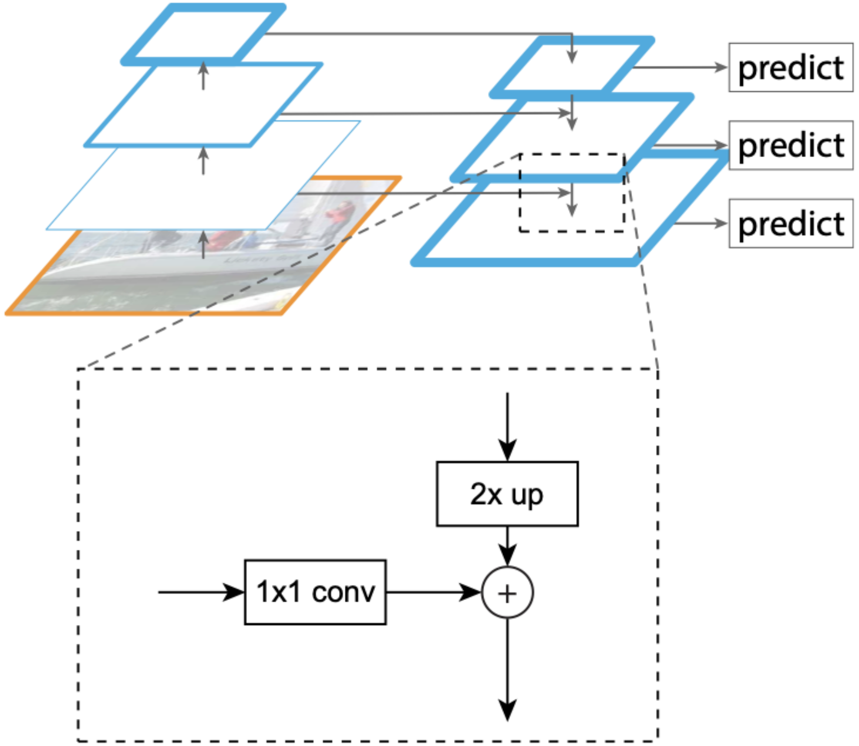
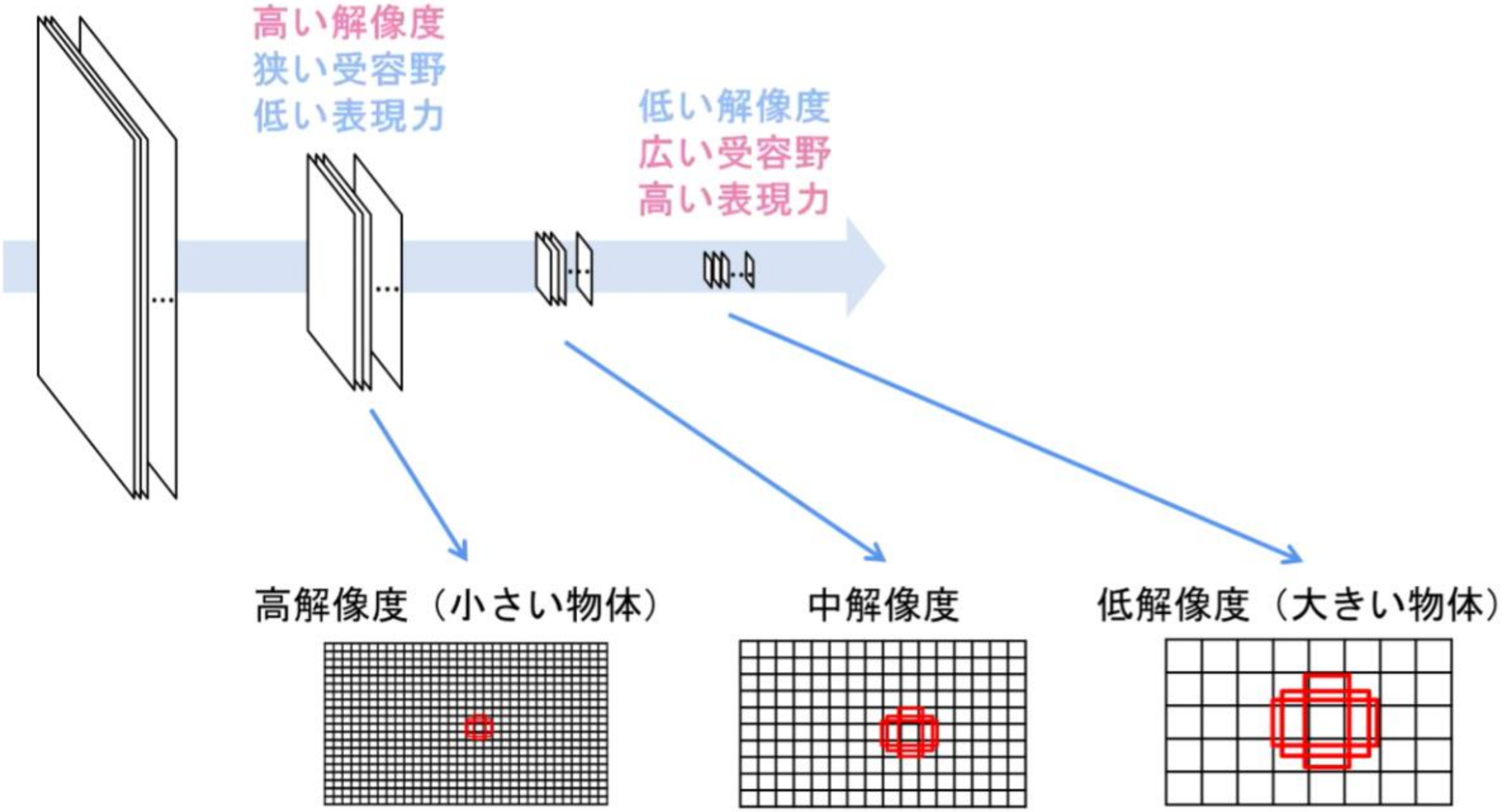


Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition.

Bottom-up pathway

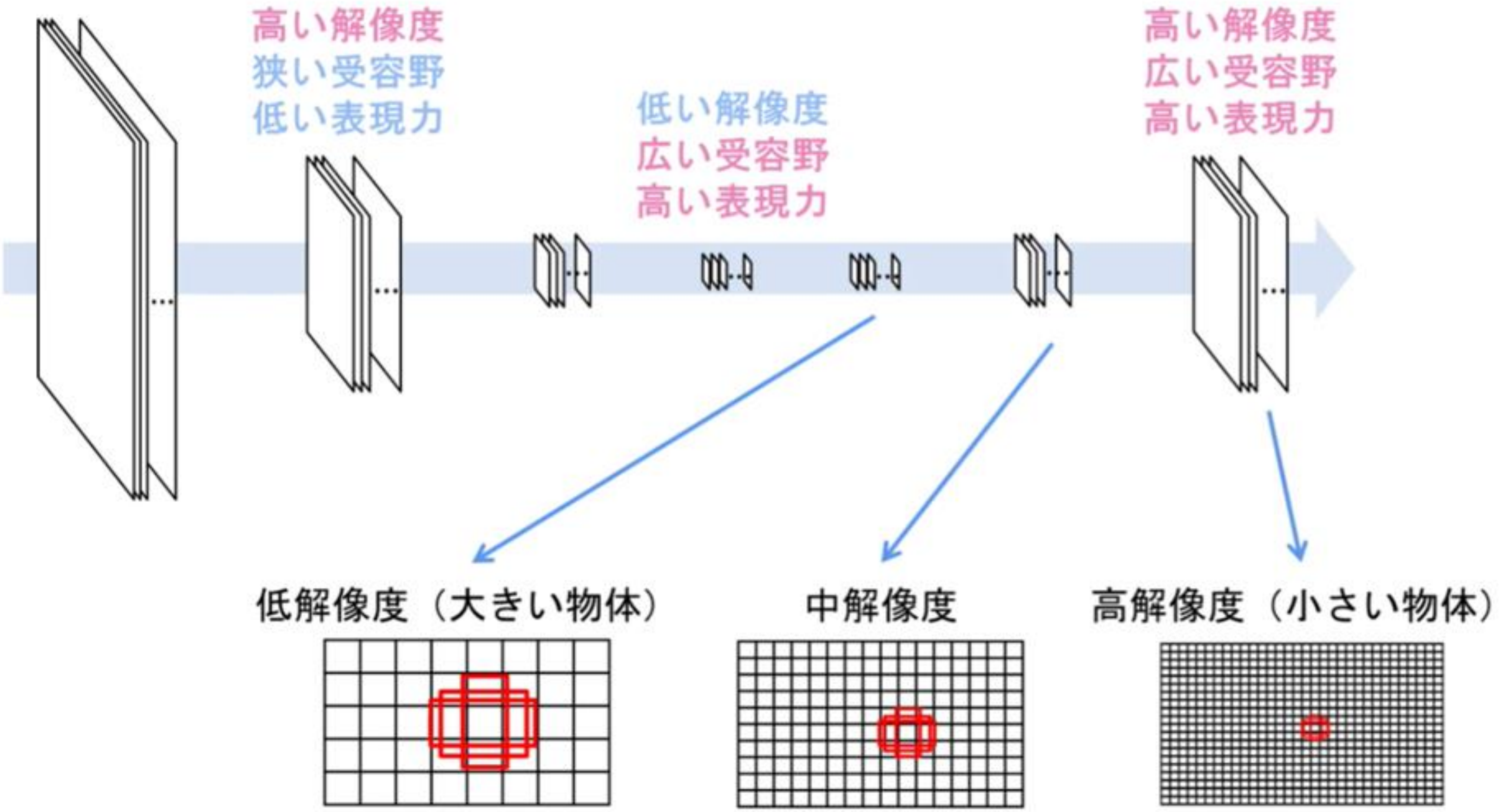


Bottom-up pathway

- The bottom-up pathway is the **feed-forward computation** of the backbone ConvNet, which computes a feature hierarchy consisting of feature maps at several scales with a scaling step of 2.
- There are often many **layers producing output maps of the same size** and we say **these layers are in the same network stage.**

- **We choose the output of the last layer of each stage** as our reference set of feature maps, which we will enrich to create our pyramid.
- This choice is natural since the **deepest layer of each stage should have the strongest features.**

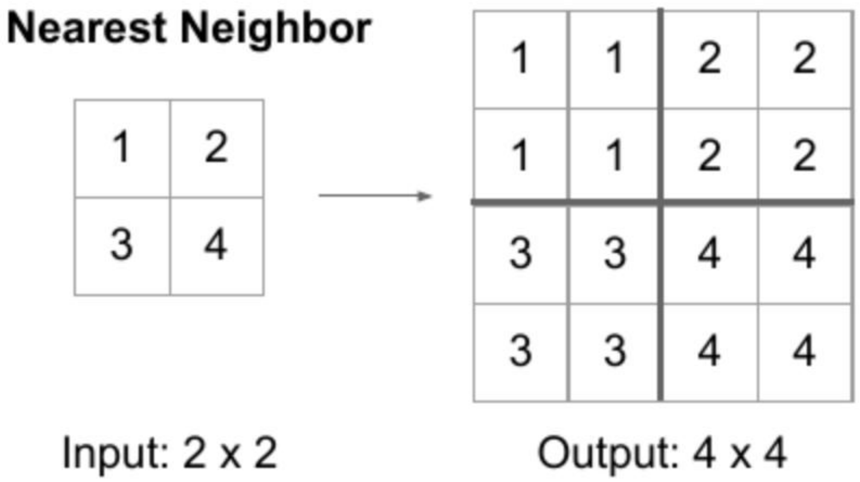
Top-down pathway and lateral connections



Top-down pathway and lateral connections

- The top-down pathway hallucinates higher resolution features by **upsampling spatially coarser**, but **semantically stronger**, feature maps from higher pyramid levels.
- These features are then **enhanced with features from the bottom-up pathway via lateral connections.**
- Each **lateral connection merges feature maps** of the same spatial size from the **bottom-up pathway** and the **top-down pathway**.

Top-down pathway and lateral connections



- We upsample the spatial resolution by a factor of 2 (using **nearest neighbor upsampling** for simplicity).
- The upsampled map is then merged with the corresponding bottom-up map (**which undergoes a 1×1 convolutional layer to reduce channel dimensions**) by **element-wise addition.**

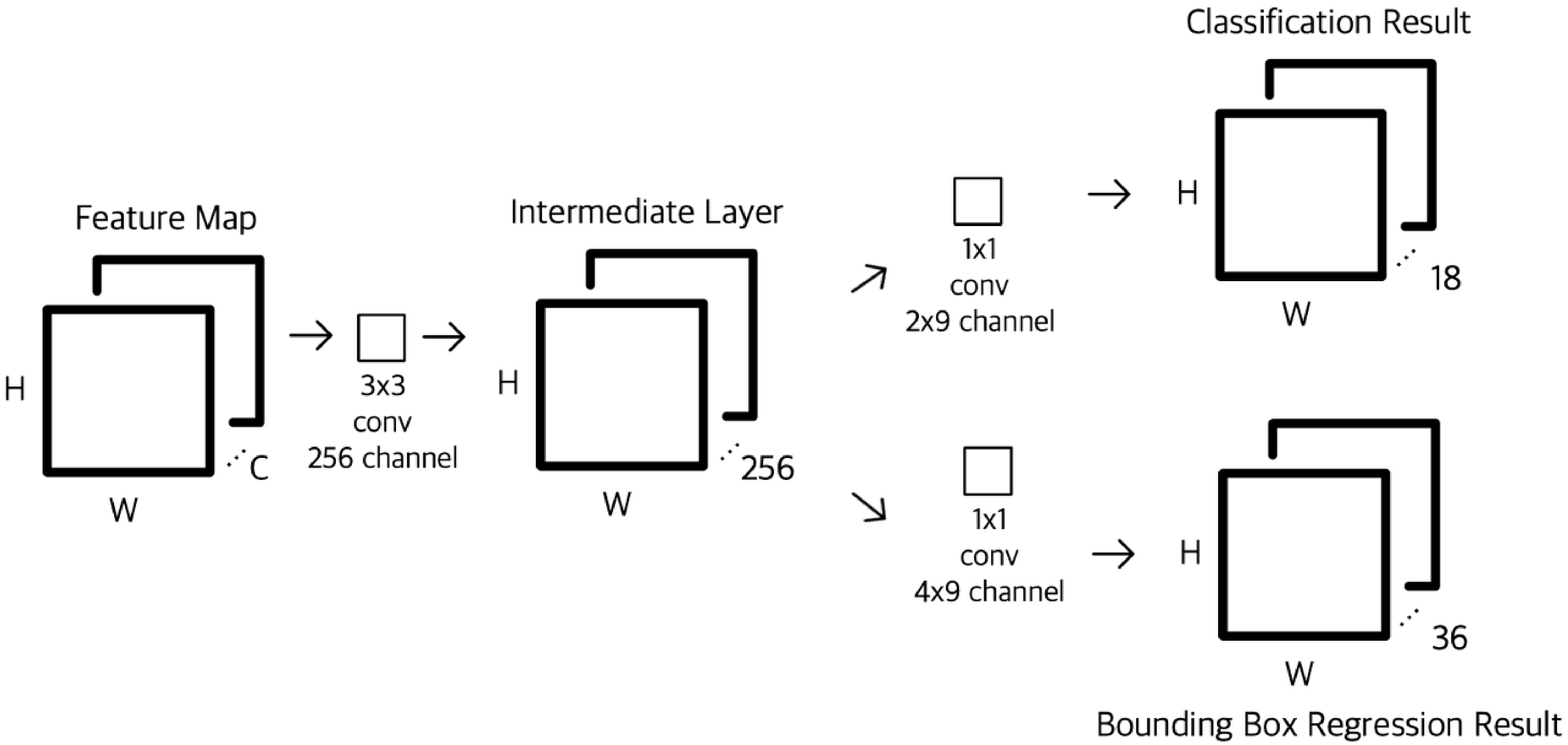
Top-down pathway and lateral connections

- The top-down pathway hallucinates higher resolution features by **upsampling spatially coarser**, but **semantically stronger**, feature maps from higher pyramid levels.
- These features are then **enhanced with features from the bottom-up pathway via lateral connections.**
- Each **lateral connection merges feature maps** of the same spatial size from the **bottom-up pathway** and the **top-down pathway**.

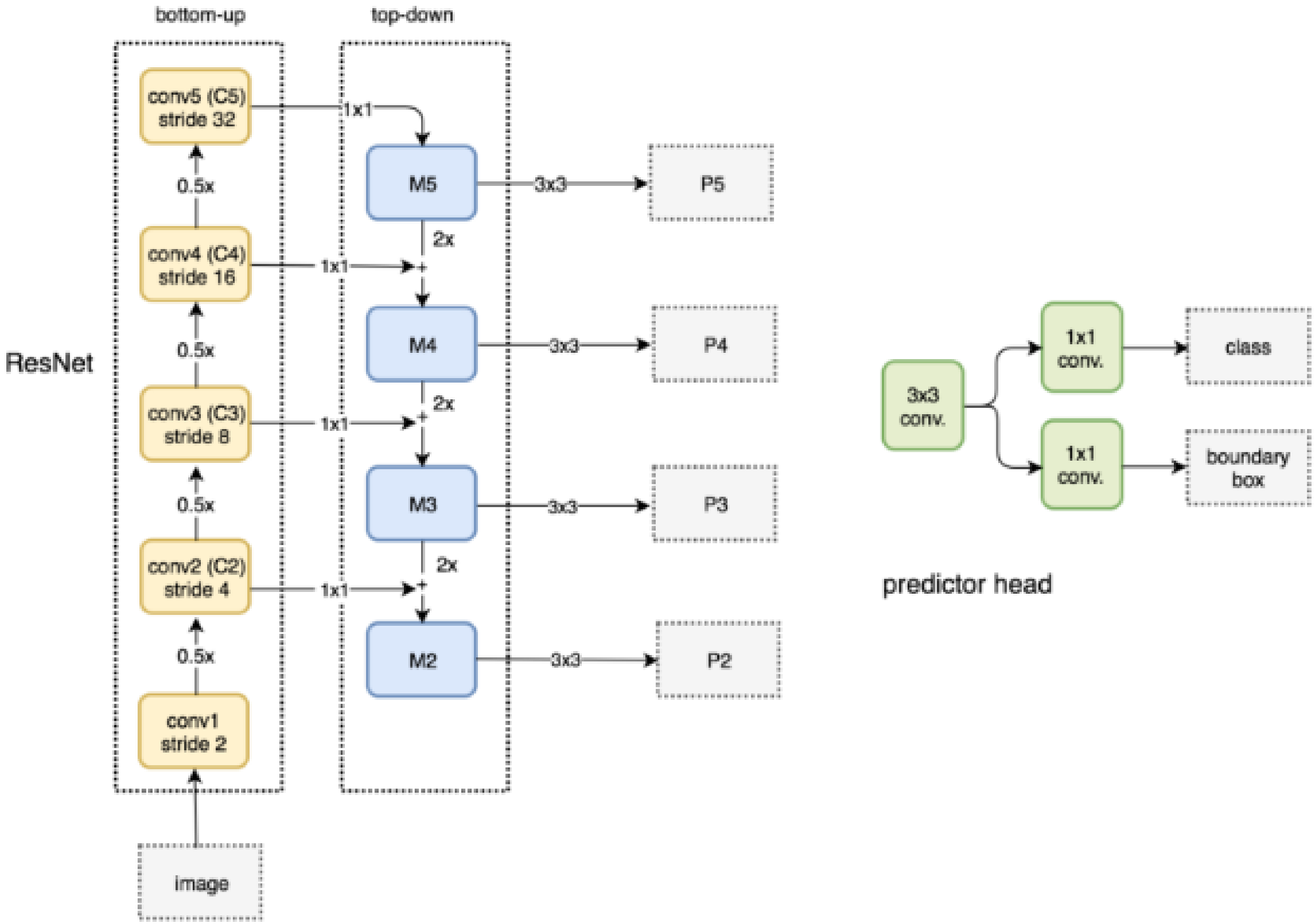
Applications

- Our method is a generic solution for building feature pyramids inside deep ConvNets.
- **RPN** for bounding box proposal generation
- **Fast R-CNN** for object detection

Feature Pyramid Networks for RPN



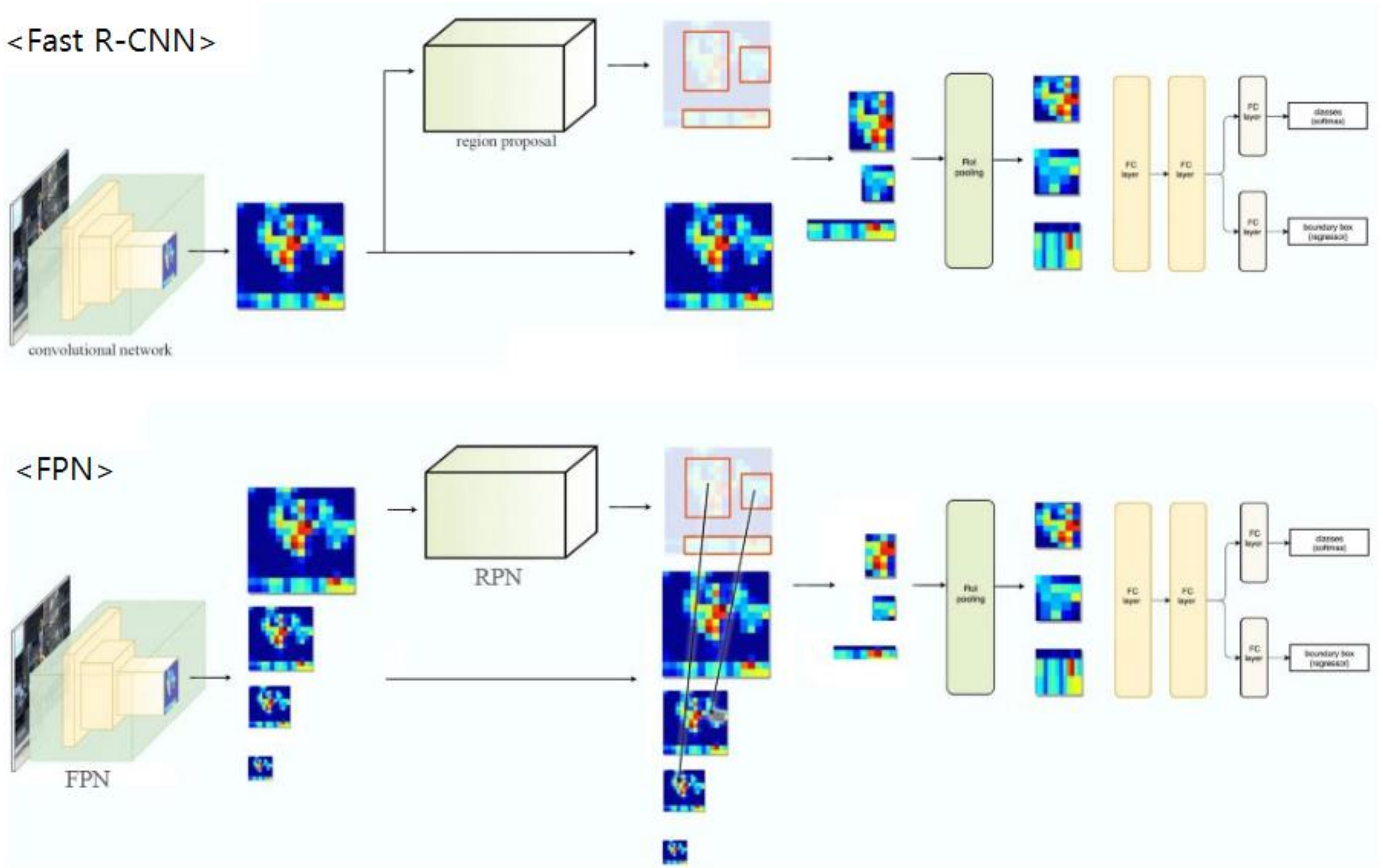
Feature Pyramid Networks for RPN



Feature Pyramid Networks for RPN

- We adapt RPN by **replacing the single-scale feature map with our FPN**.
- This is realized by a 3×3 convolutional layer followed by two sibling 1×1 convolutions for classification and regression, which we refer to as a **network head.**
  - The parameters of the heads are shared across all feature pyramid levels.
- We attach a head of the same design (3×3 conv and two sibling 1×1 convs) to each level on our feature pyramid.

Feature Pyramid Networks for Fast R-CNN



Feature Pyramid Networks for Fast R-CNN

- To use it with our FPN, we need to assign RoIs of different scales to the pyramid levels.
- We view our feature pyramid as if it were produced from an image pyramid. Thus we can adapt the assignment strategy of region-based detectors in the case when they are run on image pyramids.
- We attach predictor heads (in Fast R-CNN the heads are class-specific classifiers and bounding box regressors) to all RoIs of all levels.
    - The heads all share parameters, regardless of their levels.

Experiments on Object Detection

- COCO detection dataset
- Trained on `trainval35k`
- Evaluated on `minival`

Experiments on Object Detection

RPN	feature	# anchors	lateral?	top-down?	AR^{100}	AR^{1k}	AR^{1k}_s	AR^{1k}_m	AR^{1k}_l
(a) baseline on conv4	C_4	47k			36.1	48.3	32.0	58.7	62.2
(b) baseline on conv5	C_5	12k			36.3	44.9	25.3	55.5	64.2
(c) **FPN**	$\{P_k\}$	200k	✓	✓	**44.0**	**56.3**	**44.9**	**63.4**	66.2
Ablation experiments follow:									
(d) bottom-up pyramid	$\{P_k\}$	200k	✓		37.4	49.5	30.5	59.9	**68.0**
(e) top-down pyramid, w/o lateral	$\{P_k\}$	200k		✓	34.5	46.1	26.5	57.4	64.7
(f) only finest level	P_2	750k	✓	✓	38.4	51.3	35.1	59.7	67.6

Table 1. Bounding box proposal results using RPN [29], evaluated on the COCO `minival` set. All models are trained on `trainval35k`. The columns "lateral" and "top-down" denote the presence of lateral and top-down connections, respectively. The column "feature" denotes the feature maps on which the heads are attached. All results are based on ResNet-50 and share the same hyper-parameters.

- Region proposal(Bounding box proposal) with RPN

Experiments on Object Detection

Fast R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(a) baseline on conv4	RPN, $\{P_k\}$	C_4	conv5			54.7	31.9	15.7	36.5	45.5
(b) baseline on conv5	RPN, $\{P_k\}$	C_5	2*fc*			52.9	28.8	11.9	32.4	43.4
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	**45.8**
Ablation experiments follow:										
(d) bottom-up pyramid	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓		44.9	24.9	10.9	24.4	38.5
(e) top-down pyramid, w/o lateral	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*		✓	54.0	31.3	13.3	35.2	45.3
(f) only finest level	RPN, $\{P_k\}$	P_2	2*fc*	✓	✓	56.3	33.4	17.3	37.3	45.6

Table 2. Object detection results using **Fast R-CNN** [11] on *a fixed set of proposals* (RPN, $\{P_k\}$, Table 1(c)), evaluated on the COCO `minival` set. Models are trained on the `trainval35k` set. All results are based on ResNet-50 and share the same hyper-parameters.

- Object Detection with Fast R-CNN

Experiments on Object Detection

Faster R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(*) baseline from He *et al.* [16]†	RPN, C_4	C_4	conv5			47.3	26.3	-	-	-
(a) baseline on conv4	RPN, C_4	C_4	conv5			53.1	31.6	13.2	35.6	**47.1**
(b) baseline on conv5	RPN, C_5	C_5	2*fc*			51.7	28.0	9.6	31.9	43.1
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	45.8

Table 3. Object detection results using **Faster R-CNN** [29] evaluated on the COCO `minival` set. *The backbone network for RPN are consistent with Fast R-CNN.* Models are trained on the `trainval35k` set and use ResNet-50. †Provided by authors of [16].

- Object Detection with Faster R-CNN

References

- https://arxiv.org/abs/1612.03144
- https://www.youtube.com/watch?v=05qlCP-xL9Y