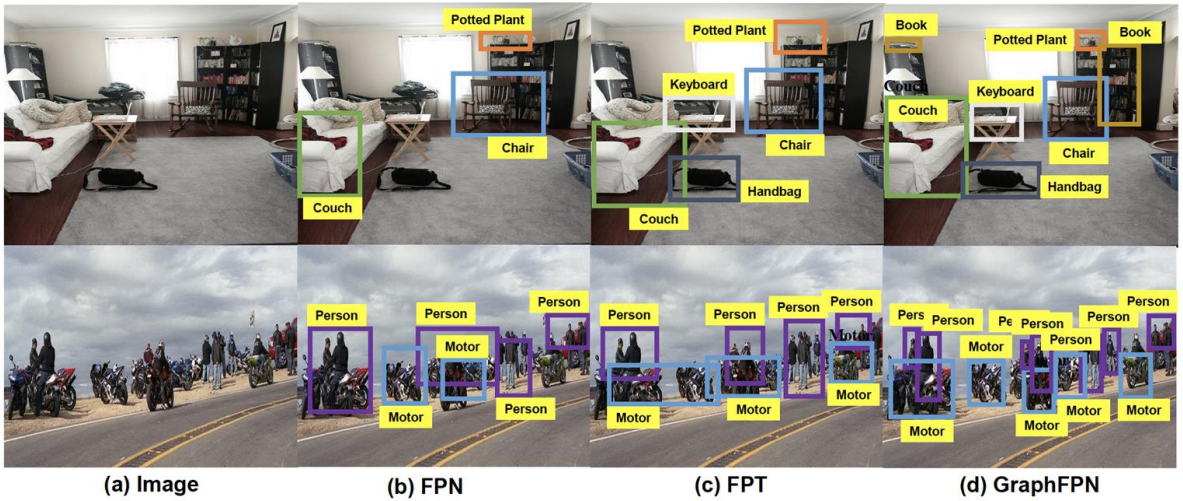
GraphFPN:
Graph Feature Pyramid Network for Object Detection

ICCV 2021

Contents

- Introduction
- Related Work
- Methods
  - Graph Neural Network
  - Graph FPN
  - Feature Mapping
- Experiments



Introduction

Introduction

- Deep convolutional neural networks exploit local connectivity and weights sharing, and have led to a series of breakthroughs in computer vision tasks
  (image recognition, object detection, semantic segmentation …)

- This methods always use a fixed multiscale network topology (i.e. 2D grids of neurons) independent of the intrinsic image structures.
  → fixed network topology may not be optimal for multiscale feature learning.

- Human parse visual scenes into part-whole hierarchies, and model part-whole relationships in different images dynamically

- The highest-level features at the top of the pyramid need to propagate through multiple intermediate scales and interact with the features at these scales before reaching the features at the bottom of the pyramid. During such propagation and interaction, essential feature information may be lost or weakened.

Introduction

- We propose a novel graph feature pyramid network to exploit intrinsic image structures and support simultaneous feature interactions across all scales. This graph feature pyramid network inherits its structure from a superpixel hierarchy of the input image. Contextual and hierarchical layers are designed to promote feature interactions within the same scale and across different scales, respectively.

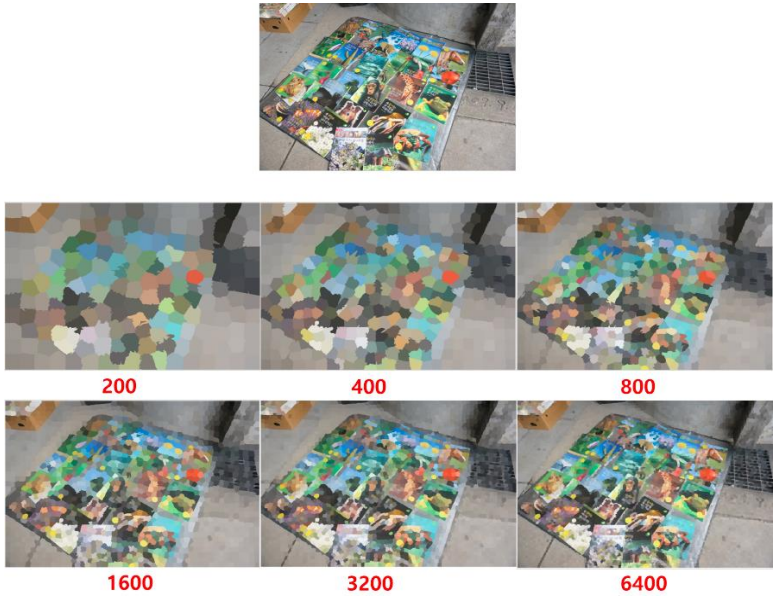
- We further introduce two types of local channel attention mechanisms for graph neural networks by generalizing existing global channel attention mechanisms for convolutional neural networks.

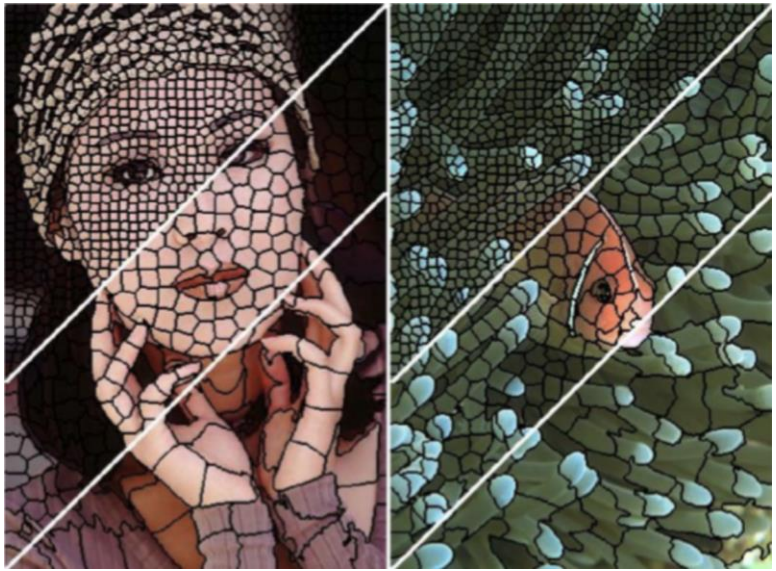
- Extensive experiments on MS-COCO 2017 validation and test datasets demonstrate that our graph feature pyramid network can help achieve clearly better performance than existing state-of the-art object detection methods no matter they are feature pyramid based or not. The reported ablation studies further verify the effectiveness of the proposed network components.

Related Work

Related Work

- Super-Pixel hierarchy for an input image.
  - Super-Pixel이란 지각적으로(perceptually) 의미있는 픽셀들을 모아서 그룹화해준 것
  - Pixel grouping을 통해 이미지에서 일정 class나 object별로 구분지을 수 있음



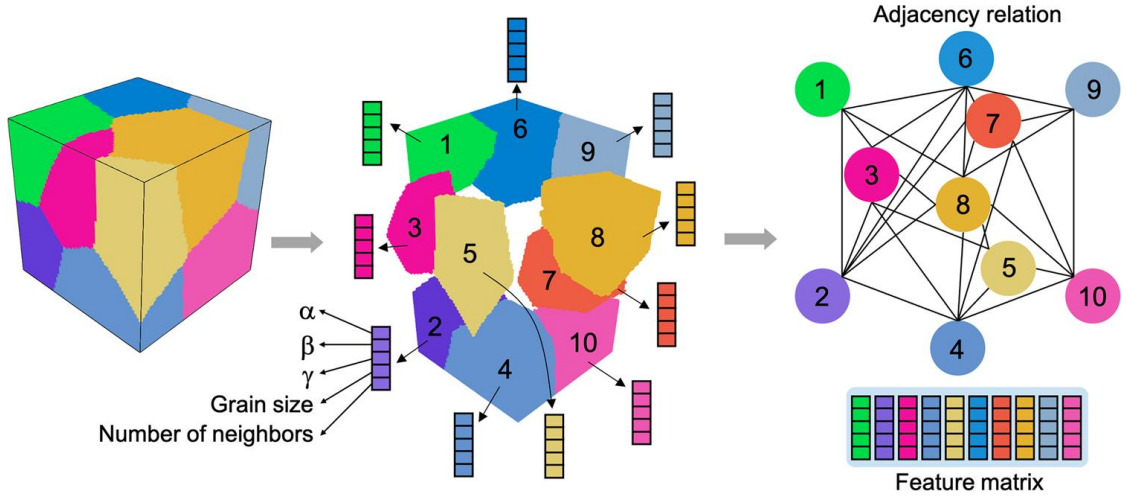


Related Work

- superpixel hierarchy for an input image.
  - 계산 효율성
    pixel을 grouping하는 것은 계산량이 꽤나 들것이라고 생각될 수도 있지만, 사실 이 알고리즘은 이미지의 복잡성을 수십 또는 수백만 픽셀 단위로 줄일 수 있음.
  - 의미 공유
    이미지 안의 pixel은 단일 pixel로서의 의미는 가지지 않지만, superpixel로서의 pixel은 각각 그룹의 공통점을 공유하여 의미를 가지게 됨.
  - 분류
    이미지에서의 객체 간 의미있는 바운더리를 찾아 분류할 수 있음.
  - 표현 효율성
    pixel 단위를 좀더 크게하여 좀 더 큰 의미를 가지는 pixel group을 형성하여 더욱 효율적으로 pixel끼리 상호작용

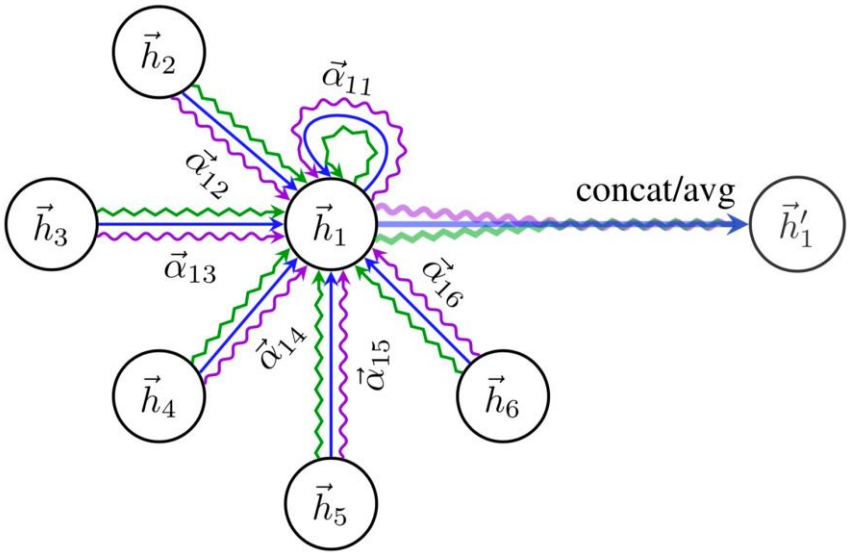
Related Work

- Graph Neural Networks.
  - GNN은 노드 간의 의존성을 유연하게 모델링할 수 있으며, 불규칙한 데이터 구조에 적용할 수 있음.



Related Work

- Graph Attention Network
  - Local self-Attention layer를 활용하여 인접 노드에 대한 가중치를 지정



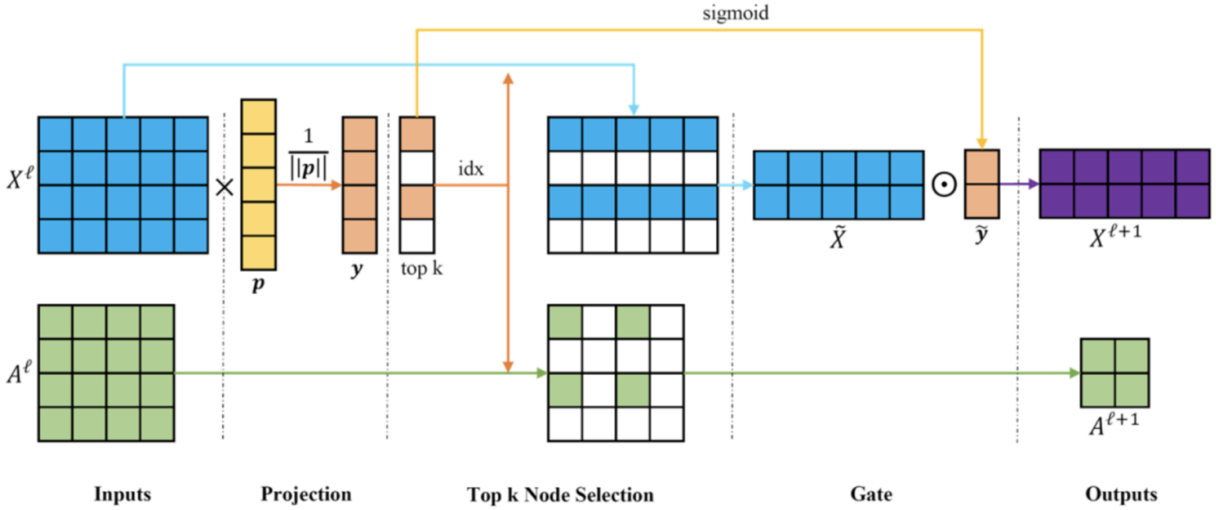
$$\alpha_{ij} = \frac{exp(LeakyRELU(\vec{a}^T[\mathbf{W}\vec{h}_i|\mathbf{W}\vec{h}_j]))}{\Sigma_{k\in N_i} exp(LeakyRELU(\vec{a}^T[\mathbf{W}\vec{h}_i|\mathbf{W}\vec{h}_k]))}$$

$$\vec{h}_i' = \sigma(\Sigma_{j\in N_i}\alpha_{ij}\mathbf{W}\vec{h}_j)$$

Graph Attention Networks: ICLR 2018

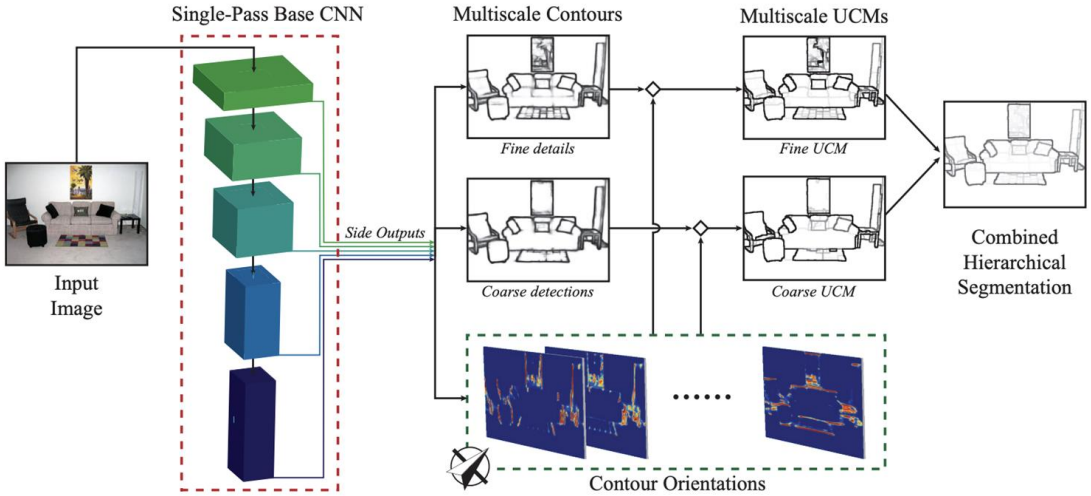
Related Work

- Graph Pooling
  - Trainable Projection Vector를 사용하여 중요한 노드를 선택하여 그래프를 Pooling하고, 이전 정보를 이용하여 풀링 작업 이전의 구조로 그래프를 복원.



Related Work

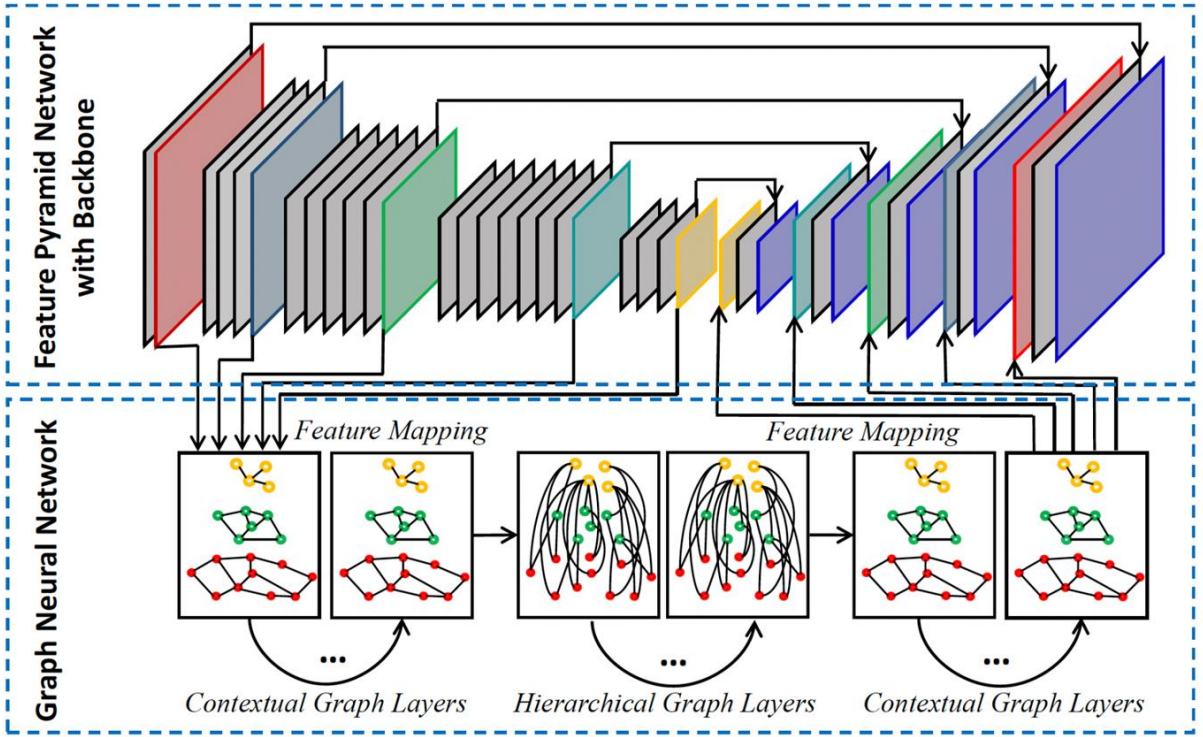
- COB(Convolutional Oriented Boundaries)
  - 입력 이미지가 주어지면 COB의 계층 분할을 사용하여 이미지별 슈퍼픽셀 계층을 구축하고, 그 위에 그래프 특징 피라미드 네트워크를 추가로 구축.



Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks: TPAMI 2018

Related Work

- Graph FPN



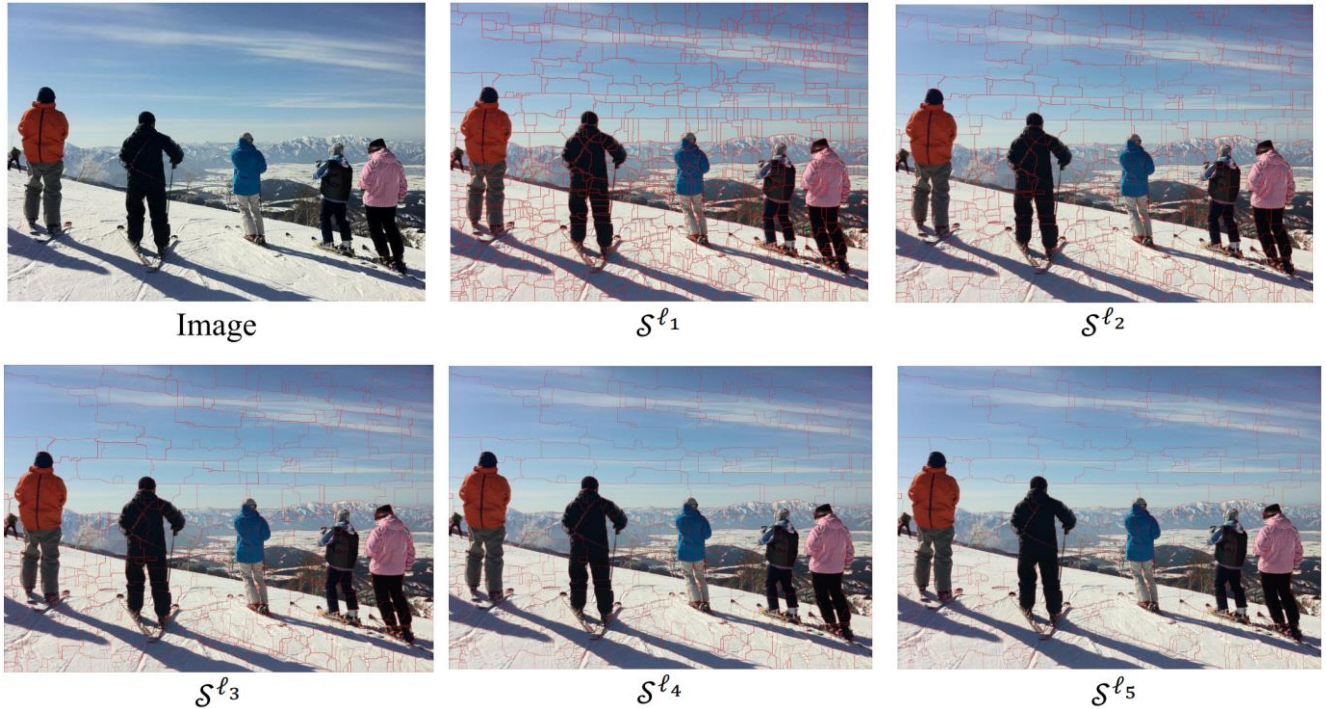
Methods

Methods

- Super-Pixel Hierarchy
  - COB모델을 사용하여 이미지를 Super-Pixel Image Partition $\{S^0, S^1, S^2, \dots, S^L\}$을 만듦.
  - Image Partition에서 sub-partition $S = \{S^{l_1}, S^{l_2}, S^{l_3}, S^{l_4}, S^{l_5}\}$ 을 만듦
  - superpixels이 가장 잘 나타나는 이미지를 S^{l_1}.
  - $S^{l_{i+1}}$은 S^{l_i}에서 Supe-Pixel이 Union된 이미지.
  - $\{l_1, l_2, l_3, l_4, l_5\}$은 CNN의 Pooling에 맞춰 1/4로 맞춤.

Methods

- Super-Pixel Hierarchy



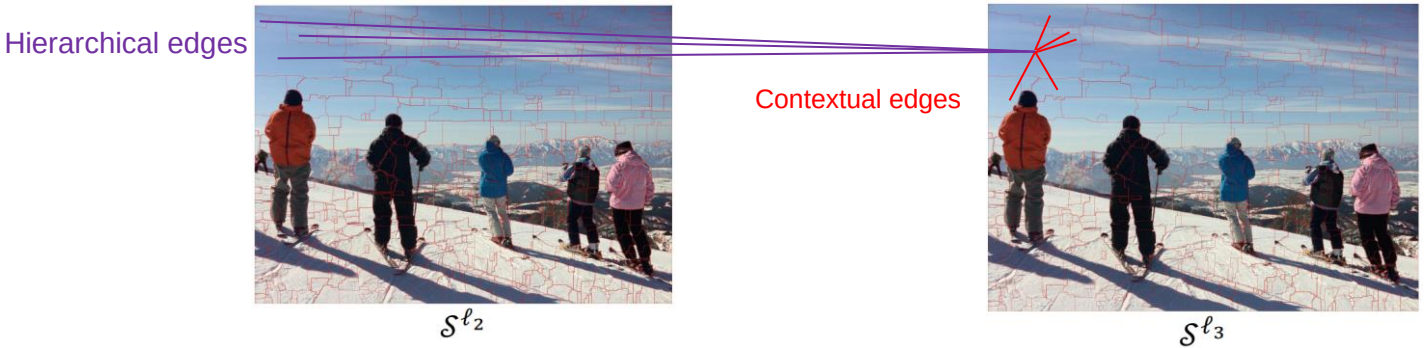
Methods

- Multi-scale Graph Pyramid
  - Super-Pixel hierarchy 구조에 맞춰 $\{\mathcal{G}^1, \mathcal{G}^2, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5\}$ 구조의 Graph Pyramid
  - Super-Pixel hierarchy의 Super-Pixel마다 Graph의 하나의 노드로 선택.
  - Graph pyramid의 level마다 노드의 수는 ¼로 감소.

COB

Methods

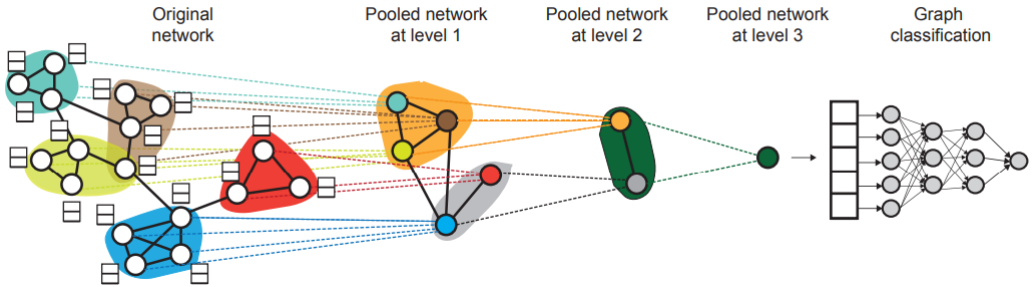
- Multi-scale Graph Pyramid
  - 저자들은 Contextual edges, Hierarchical edges를 정의.
  - Contextual edges → 같은 level의 Graph 에서 이웃하는 Super-Pixel연결.
  - Hierarchical edges → 다른 level의 Super-Pixel 병합(상/하) 관계 연결.



Methods

- Multi-scale Graph Pyramid
  - Hierarchical edges는 모든 level의 그래프에서 Dense하게 연결
    → 높은 계산 Cost를 발생시킴.

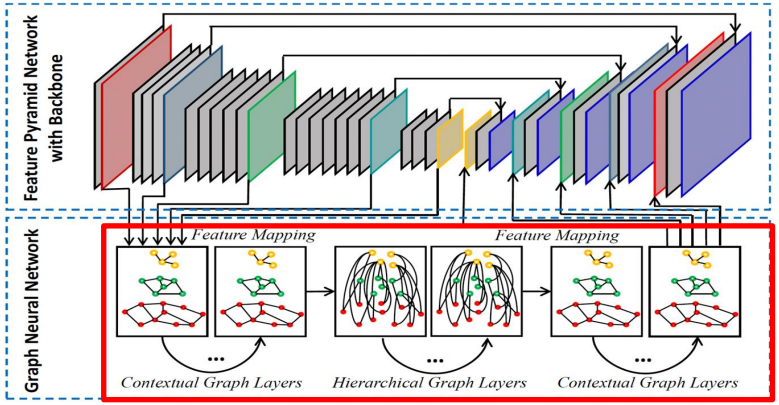
  - 그래프의 노드의 Cosine Similarity를 사용하여 Hierarchical edges를 줄임.



Hierarchical Graph Representation Learning with Differentiable Pooling: NeurIPS 2018

Methods

- Graph Neural Network Layers
  - 저자들은 Contextual Graph Layer, Hierarchical Graph Layer 정의
  - 두 Layer 모두 같은 연산을 사용하지만 edge는 다름.
  - 그래프 FPN은 $L1, L3$ 는 Contextual GNN, $L2$는 Hierarchical GNN을 사용함.



Methods

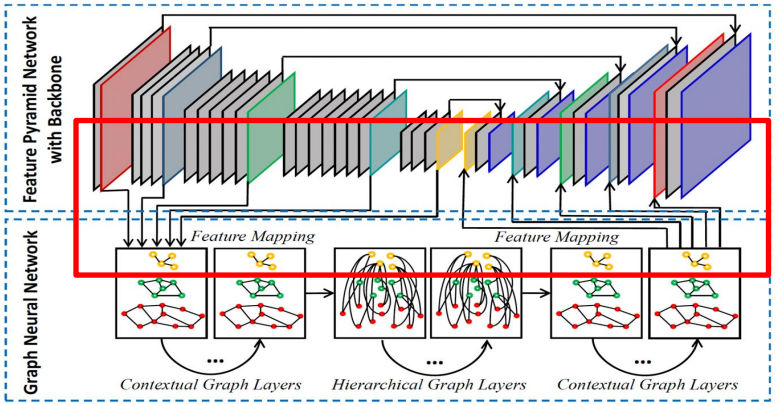
- Graph Neural Network Layers
  - 두 Layer 모두 Graph Attention Module을 적용.

$$\vec{h}_i' = \mathcal{M}\left(\vec{h}_i, \{\vec{h}_j\}_{j \in \mathcal{N}_i}\right)$$
$$\vec{h}_i'' = \sigma(W_1 \vec{a}_i') \odot \vec{h}_i'$$
$$\vec{h}_i''' = \beta X \vec{h}_i'' + \vec{h}_i''$$

- $\mathcal{M} \rightarrow Single\ Head\ Self - Attention$
- $\vec{a}_i' \rightarrow Average\ Vector\ of\ node\ i\ and\ it's\ neighbors$
- $X \rightarrow softmax(A^T A) \in \mathbb{R}^{CxC}$, $A \in \mathbb{R}^{(|\mathcal{N}_i|+1)xC}$, X is Channel Similarity Matrix

Methods

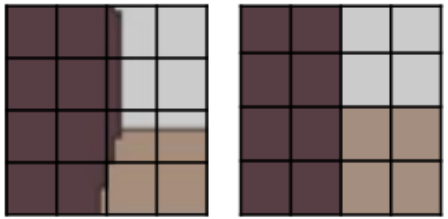
- Feature Mapping between GNN and CNN
  - CNN 과 GNN사이의 Feature Mapping이 필요.
  - $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3, \mathcal{C}^4, \mathcal{C}^5\}$, $\mathcal{C}$ is final feature maps of the five convolutional stage in backbone
  - $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3, \mathcal{P}^4, \mathcal{P}^5\}$, $\mathcal{P}$ is feature maps from graph pyramid



Methods

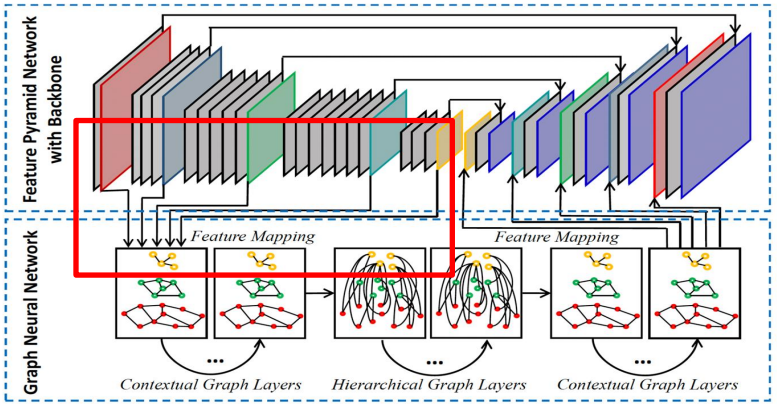
- Feature Mapping between GNN and CNN($\mathcal{C} \rightarrow \mathcal{S}$)
  - Backbone의 feature를 Super-Pixel에 mapping.
  - Grid cell이 Super-Pixel에 겹치면 많이 겹친 cell에 mapping

$$\vec{h}_k^i = \delta(W_2[\Delta_{max}(C_k^i) \Vert \Delta_{min}(C_k^i)]),\ \delta \text{ is ReLU}$$



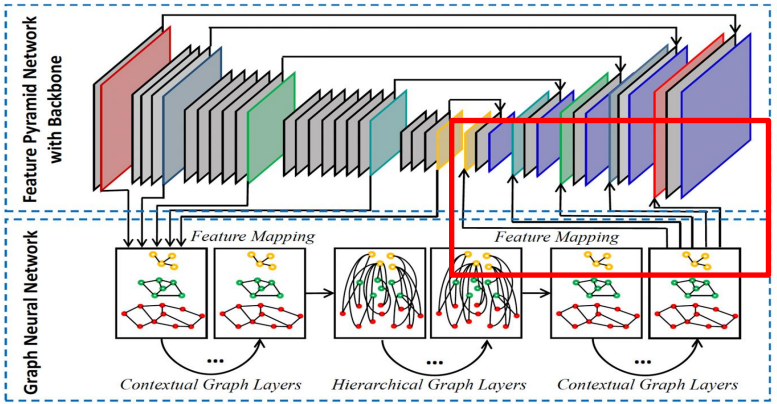
(a) Superpixel Partition (b) Feature Mapping

Figure 2. Mapping between CNN grid cells and superpxiels. Each grid cell is assigned to one superpixel it overlaps most. Each superpixel has a small collection of grid cells assigned it.



Methods

- Feature Mapping between GNN and CNN($\mathcal{S} \rightarrow \mathcal{P}$)
  - Grpah pyramid에서 나온 feature를 각 Super-Pixel 영역에 mapping($\mathcal{P}$).
  - FPN의 feature와 $\mathcal{P}$를 concat하여 $\bar{\mathcal{P}}^i$ 를 만듦.
  - $\bar{\mathcal{P}}^i$에 1x1 convolution 으로 Channel을 맞춰 $\tilde{\mathcal{P}}^i$ 만듦.



Experiments

Experiments

- GraphFPN이 FPN으로 학습되는 원래의 Multi scale feature training을 크게 향상시키며, multi-scale feature interaction과 fusion이 Object Detection에 매우 효과적이라는 것을 나타냄.
- Super-Pixel 계층 구조 위에 구축된 그래프가 이미지의 고유 구조(intrinsic structures)를 capture할 수 있으며, high-level image understanding 작업에 도움이 된다는 것을보여줌.

Method	Training Strategy	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN [40]	Baseline	33.1	53.8	34.6	12.6	35.3	49.5
Faster R-CNN+FPN [29]	Baseline	36.2	59.1	39.0	18.2	39.0	52.4
Faster R-CNN+FPN [29]	MT+AH	37.9	59.6	40.1	19.6	41.0	53.5
PAN [32]	Baseline	37.3	60.4	39.9	18.9	39.7	53.0
PAN [32]	MT+AH	39.0	60.8	41.7	20.2	41.5	54.1
ZigZagNet [28]	Baseline	39.5	–	–	–	–	–
ZigZagNet [28]	MT+AH	40.1	61.2	42.6	21.9	42.4	54.3
Faster R-CNN+FPN+FPT [52]	Baseline	41.6	60.9	44.0	23.4	41.5	53.1
Faster R-CNN+FPN+FPT [52]	AH	41.1	62.0	46.6	24.2	42.1	53.3
Faster R-CNN+FPN+FPT [52]	MT	41.2	62.1	46.0	24.1	41.9	53.2
Faster R-CNN+FPN+FPT [52]	MT+AH	42.6	62.4	46.9	24.9	43.0	**54.5**
Ours	Baseline	42.1	61.3	46.1	23.6	41.1	53.3
Ours	AH	42.7	63.0	47.2	25.6	43.1	53.3
Ours	MT	42.4	62.7	46.9	24.3	43.1	53.6
Ours	MT+AH	**43.7**(↑1.1)	**64.0**(↑1.6)	**48.2**(↑1.3)	**27.2**(↑2.3)	**43.4**(↑0.4)	54.2(↓0.3)

Table 1. Comparison with state-of-the-art feature pyramid based methods on MS-COCO 2017 test-dev [31]. "AH" and "MT" stand for augmented head and multi-scale training strategies [32] respectively. The backbone of all listed methods is ResNet101 [12].

Experiments

Method	Detection Framework	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet + FPN [30]	RetineNet	40.4	60.2	43.2	24.0	44.3	52.2
Faster R-CNN+FPN [29]	Faster R-CNN	42.0	62.5	45.9	25.2	45.6	54.6
DETR [5]	Set Prediction	44.9	64.7	47.7	23.7	49.5	62.3
Deformable DETR [56]	Set Prediction	43.8	62.6	47.7	26.4	47.1	58.0
Sparse R-CNN+FPN [52]	Sparse R-CNN	45.6	64.6	49.5	28.3	48.3	61.6
Ours	Faster R-CNN	**46.7**(↑1.1)	**65.1**(↑0.5)	**50.1**(↑0.6)	**29.2**(↑0.9)	49.1(↓0.8)	61.8(↓0.2)

Table 2. Comparison with other popular object detectors on MS-COCO 2017 val set [31]. The backbone of all listed methods is ResNet101 [12].

Methods	Params	GFLOPs	Test Speed (s)
faster RCNN [40]	34.6 M	172.3	0.139
faster RCNN + FPN [29]	64.1 M	240.6	0.051
faster RCNN + FPN + FPT [52]	88.2 M	346.2	0.146
faster RCNN + FPN + GraphFPN	100.0 M	380.0	0.157
COB + faster RCNN + FPN + GraphFPN	121.0 M	393.1	0.277

Table 3. The number of learnable parameters, the total computational cost, and the average test speed of a few detection models. All experiments are run on an NVidia TITAN 2080Ti GPU.

Experiments

- Contextual Graph Layer, Hierarchical Graph Layer 에 대한 Ablation study

CGL-1	HGL	CGL-2	AP	AP_S	AP_M	AP_L
✓	✓	✓	**39.1**	**22.4**	**38.9**	**56.7**
×	✓	✓	38.2	22.1	38.7	56.1
✓	✓	×	38.7	22.1	38.9	56.6
×	✓	×	36.2	19.2	36.3	54.4
✓	×	×	37.2	22.1	35.1	55.6

Table 4. Ablation study on the contextual and hierarchical layers in GraphFPN. "CGL-1" stands for the first group of contextual layers before the hierarchical layers, "HGL" stands for the hierarchical layers, and "CGL-2" stands for the second group of contextual layers after the hierarchical layers. ✓ and × stand for the existence of a module or not. Detection results are reported on the MS-COCO 2017 val set [31].

Experiments

- Graph Layer 의 수 에 따른 Ablation study

N	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	36.1	56.3	35.4	19.3	37.9	55.4
2	37.2	57.6	38.5	21.2	38.3	55.8
3	**39.1**	**58.3**	**39.4**	**22.4**	**38.9**	**56.7**
4	38.1	57.8	38.9	22.2	38.6	56.3
5	37.1	57.1	38.0	21.9	37.9	55.4

Table 6. Ablation study on the number of layers in GraphFPN. N is the number of layers in each of the three groups of layers. Hence, the total number of layers is 3N. Detection results are reported on the MS-COCO 2017 val set [31].

Experiments

- Attention Module에 대한 Ablation Study

SA	LCA	LSA	AP	AP_S	AP_M	AP_L
✓	✓	✓	**39.1**	**22.4**	**38.9**	**56.7**
×	✓	✓	37.8	21.9	37.4	56.2
✓	×	✓	37.9	21.6	37.3	56.4
✓	✓	×	37.6	21.8	37.7	55.1
✓	×	×	37.1	21.1	36.7	54.1

Table 5. Ablation study on the attention mechanisms. "SA" stands for the spatial attention module, "LCA" stands for the local channel-wise attention module and "LSA" stands for the local channel self-attention module. ✓ and × stand for the existence of a module or not. Detection results are reported on the MS-COCO 2017 val set [31].

Experiments

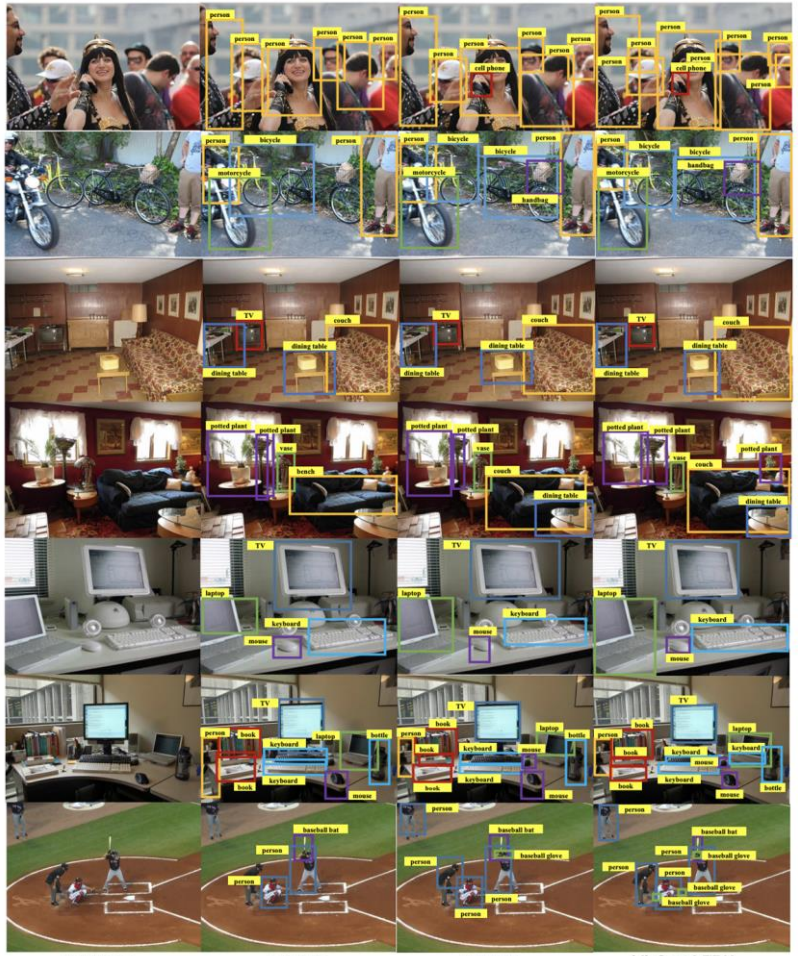


Figure 1. Sample detection results from FPN [30], FPT [54], and our GraphFPN based method. Images are from the MS COCO 2017 validation set [32].

Thank You