Spatial-aware Graph Relation Network for Large-scale Object Detection

CVPR 2019

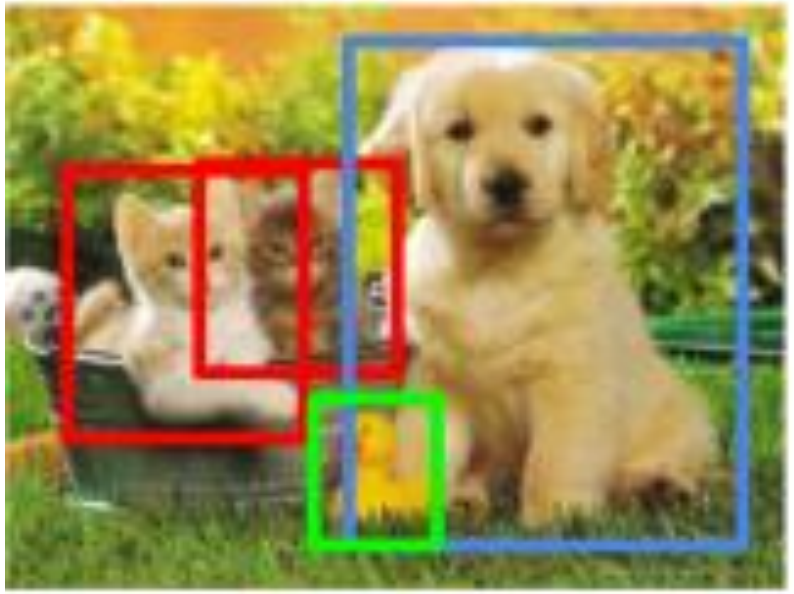
Hang Xu, ChenHan Jiang, Xiaodan Liang, Zhenguo Li

Hoseok Do

Seoul National University

Problem

- Object Detection
  - A computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class
  - Dataset : MS COCO (80 classes), VOC (20 classes)



- Large Scale Object Detection
  - Recognize thousands of objects entangled with complex spatial and semantic relationships
  - Dataset : VisualGenome (**3000** classes), ADE (**455** classes)

Key Idea

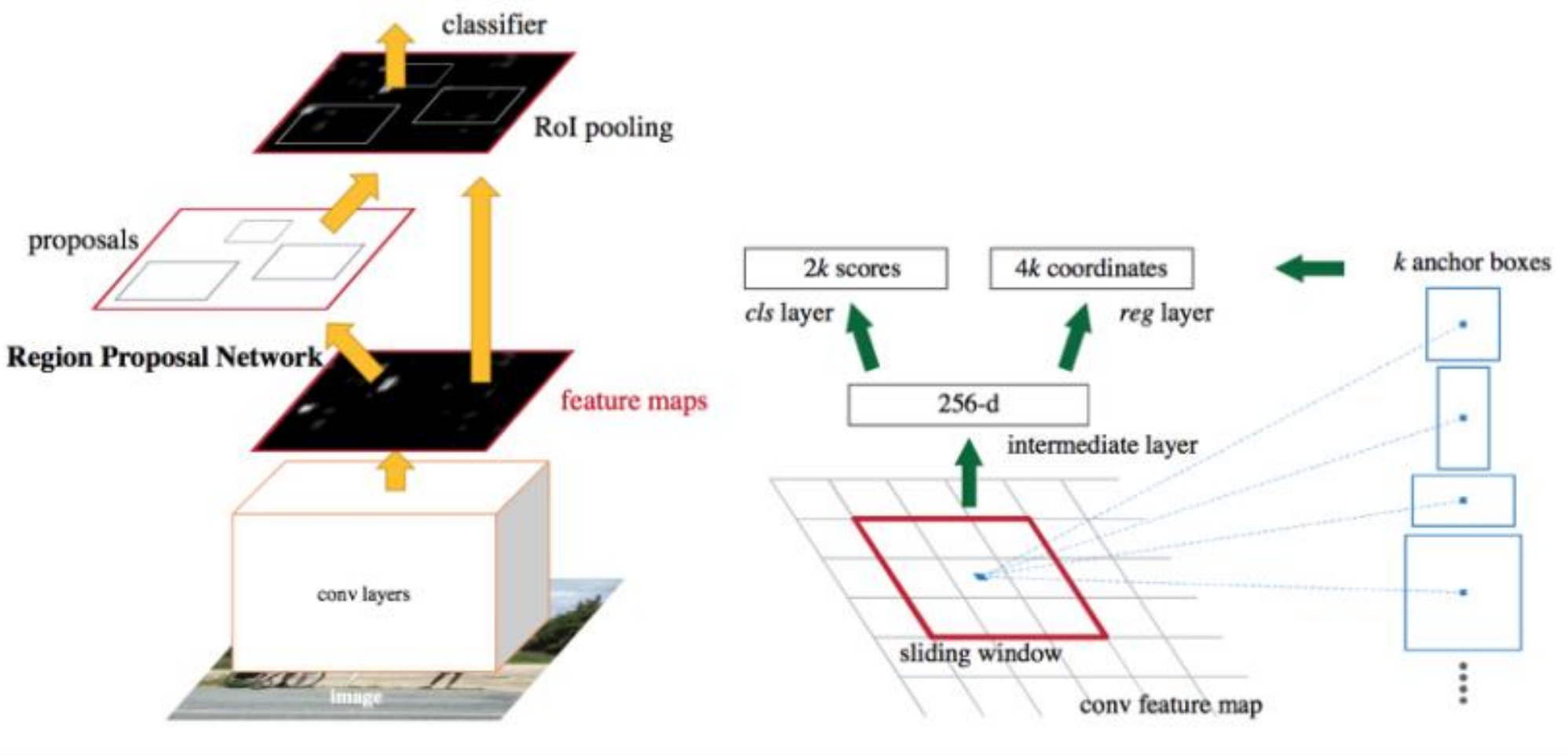
- Using graph to discover and incorporate key semantic and spatial relationships over each object.
- Inject module into existing detection pipelines to boost the performance.

Input Image

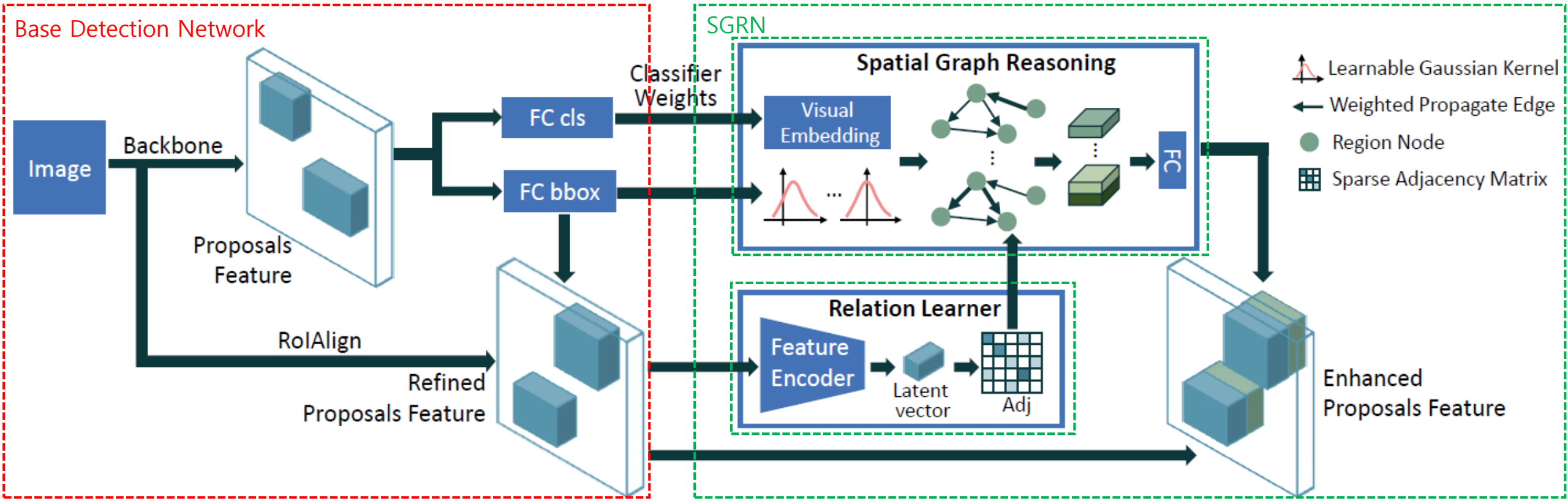
Spatial-aware Sparse Graph

Two Stage Object Detection

- Region Proposal Network + Classifier, Box Regression
- Faster R-CNN



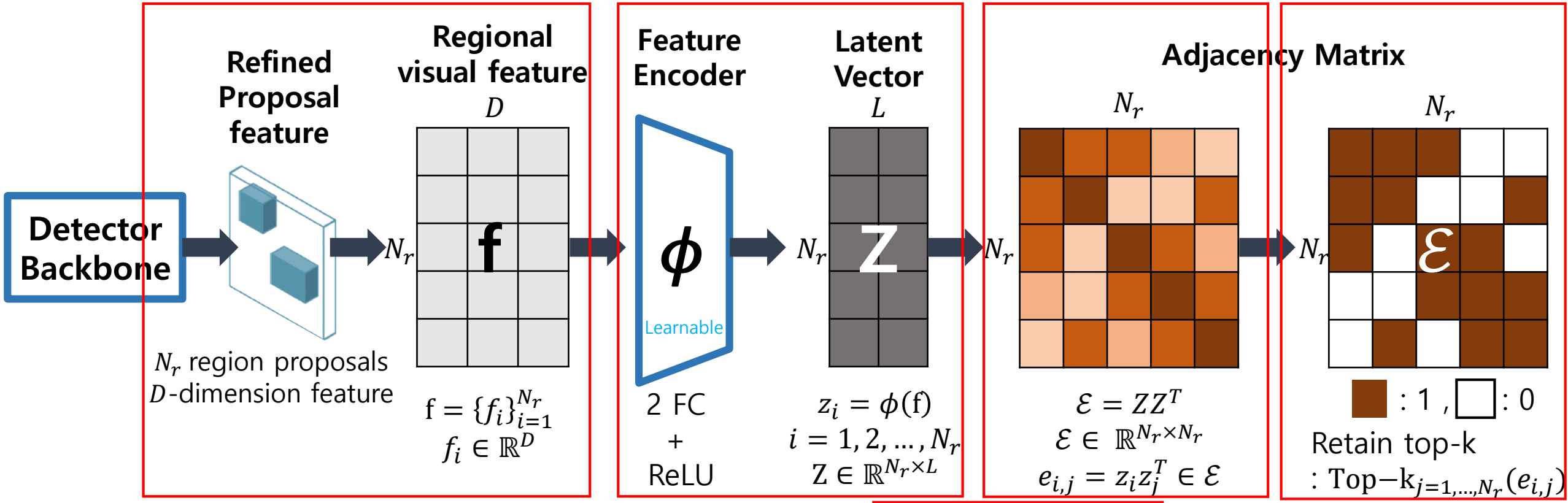
Spatial-aware Graph Relation Network



- Relation Learner
  - Learn a interpretable sparse graph structure
- Spatial Graph Reasoning
  - Graph inference with spatial awareness

Relation Learner Module

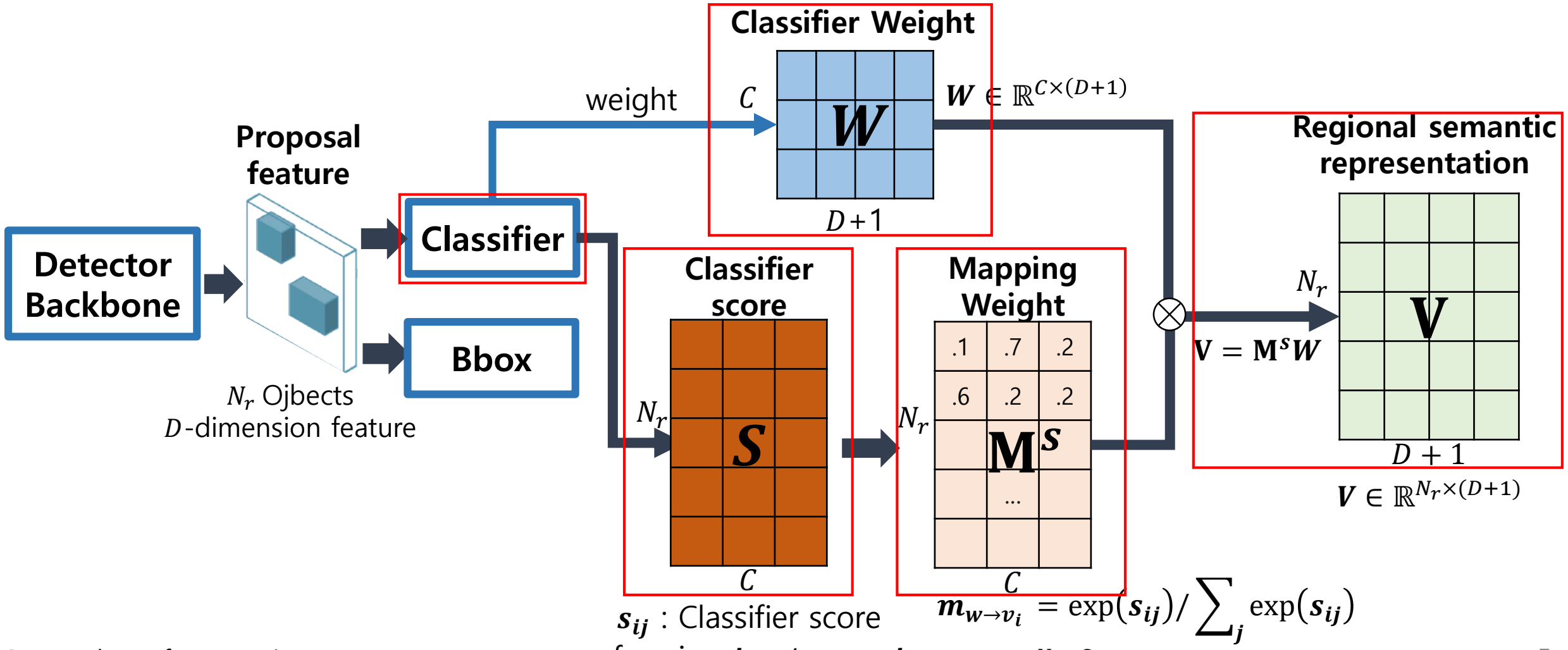
- Learn graph structure from regional visual feature



N_r : number of the region proposals
D : dimension of the regional visual feature
L : dimension of the latent space

Node Feature Embedding of the Regions

- Convert classifier weight to regional semantic representation



C : number of categories

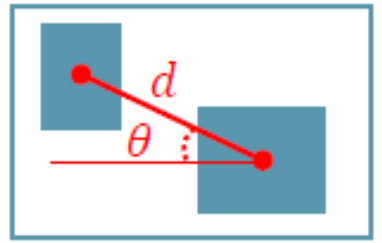
Spatial-aware Graph Reasoning Module

- Node Feature Propagation with Spatial Gaussian
  - Relative position of node j from node i : $u(i,j) = (d, \theta)$
  - k-th Gaussian kernel

    $w_k\big(u(i,j)\big) = \exp\left(-\frac{1}{2}(u(i,j) - \mu_k)^T \Sigma_k^{-1}(u(i,j) - \mu_k)\right)$

    μ_k: 2×1 mean vector, Σ_k: 2×2 covariance matrix
  - Weighted sum feature data from connected nodes

    $f_k'(i) = \sum_{j \in Neighbour(i)} w_k(u(i,j))\, v_j e_{ij}$

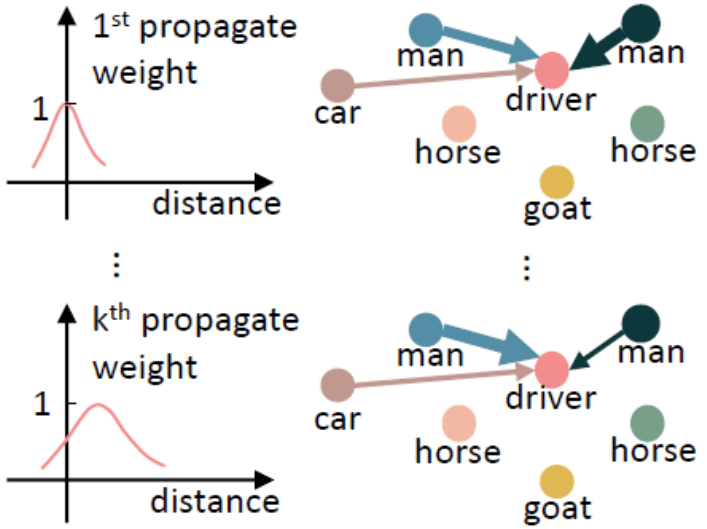


$$\mathrm{d} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$\theta = \arctan\left(\frac{(y_j - y_i)}{(x_j - x_i)}\right)$$

Pseudo-coordinate function

Spatial-aware Sparse Graph

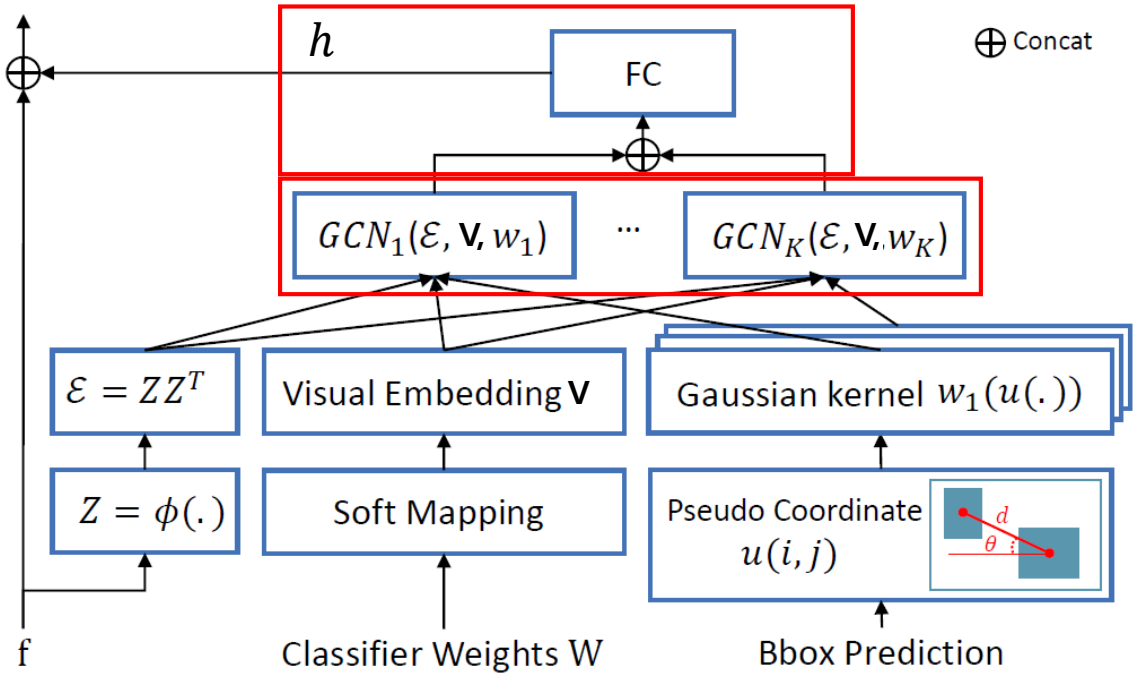


Propagation with Gaussian Kernels

Spatial-aware Graph Reasoning Module

- GCN : $GCN_k(\varepsilon, V, w_k)$
  - ε : Adjacency Matrix from Region Proposal Feature
  - V : Node Representation from Classifier
  - w_k: k-th Gaussian kernel using relative spatial information

- Enhanced Feature : $h_i = L[f'(i)]$
  - Fully Connected Layer : $L \in \mathbb{R}^{E \times (D+1)}$
    E : dimension of the output enhanced feature
  - h_i is concatenated to f_i
    (Improve classification and localization)



Flowchart of the Spatial-aware Graph Reasoning Module

Experiments

%	Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
VG_{1000}	Light-head RCNN[28]	6.2	10.9	6.2	2.8	6.5	9.8	14.6	18.0	18.7	7.2	17.1	25.3
	Cascade RCNN[4]	6.5	12.1	6.1	2.4	6.9	11.2	15.3	19.4	19.5	6.1	19.2	27.5
	HKRM[23]	7.8	13.4	8.1	4.1	8.1	12.7	18.1	22.7	22.7	9.6	20.8	31.4
	Faster-RCNN[45]	5.7	9.9	5.8	2.7	6.9	8.9	13.8	17.0	17.0	6.6	15.8	23.5
	Faster-RCNN w SGRN	$6.8^{+1.1}$	$11.1^{+1.2}$	$7.1^{+1.3}$	$3.3^{+0.6}$	$7.0^{+0.1}$	$10.8^{+1.9}$	$15.3^{+1.5}$	$19.5^{+2.5}$	$19.6^{+2.6}$	$8.3^{+1.7}$	$17.8^{+2.0}$	$26.7^{+3.2}$
	FPN[30]	7.1	12.9	7.3	4.2	7.9	10.7	14.9	19.8	20.0	11.1	19.3	23.6
	FPN w SGRN	$\mathbf{8.1}^{+0.9}$	$\mathbf{13.6}^{+0.7}$	$\mathbf{8.4}^{+1.1}$	$\mathbf{4.4}^{+0.2}$	$\mathbf{8.2}^{+0.3}$	$\mathbf{12.8}^{+2.1}$	$\mathbf{19.5}^{+4.6}$	$\mathbf{26.0}^{+6.2}$	$\mathbf{26.2}^{+6.2}$	$\mathbf{12.4}^{+1.3}$	$\mathbf{23.9}^{+4.6}$	$\mathbf{34.0}^{+10.4}$
VG_{3000}	Light-head RCNN[28]	3.0	5.1	3.2	1.7	4.0	5.8	7.3	9.0	9.0	4.3	10.3	15.4
	Cascade RCNN[4]	3.8	6.5	3.4	1.9	4.8	4.9	7.1	8.5	8.6	4.2	9.9	13.7
	HKRM[23]	4.3	7.2	4.4	2.6	5.5	8.4	10.1	12.2	12.2	5.9	13.0	20.5
	Faster-RCNN[45]	2.6	4.4	2.7	1.7	3.6	4.8	6.2	7.6	7.6	4.3	9.1	12.9
	Faster-RCNN w SGRN	$3.2^{+0.6}$	$5.0^{+0.6}$	$3.4^{+1.3}$	$2.0^{+0.3}$	$4.2^{+0.6}$	$6.5^{+1.7}$	$7.3^{+0.9}$	$9.2^{+1.6}$	$9.2^{+1.6}$	$4.9^{+0.6}$	$11.4^{+1.7}$	$16.2^{+3.3}$
	FPN[30]	3.4	6.1	3.4	2.6	4.8	6.3	6.9	9.1	9.1	6.7	11.5	13.4
	FPN w SGRN	$\mathbf{4.5}^{+1.1}$	$\mathbf{7.4}^{+1.3}$	$\mathbf{4.3}^{+1.0}$	$\mathbf{2.9}^{+0.3}$	$\mathbf{6.0}^{+1.2}$	$\mathbf{8.6}^{+2.3}$	$\mathbf{10.8}^{+3.9}$	$\mathbf{13.7}^{+4.6}$	$\mathbf{13.8}^{+4.7}$	$\mathbf{8.1}^{+1.4}$	$\mathbf{15.1}^{+3.6}$	$\mathbf{21.8}^{+8.4}$
ADE	Light-head RCNN[28]	7.0	11.7	7.3	2.4	5.1	11.2	9.6	13.3	13.4	4.3	10.4	20.4
	Cascade RCNN[4]	9.1	16.8	8.9	3.5	7.1	15.3	12.1	16.4	16.6	6.4	13.8	25.8
	HKRM[23]	10.3	18.0	10.4	4.1	7.8	16.8	13.6	18.3	18.5	7.1	15.5	28.4
	Faster-RCNN[45]	6.9	12.8	6.8	3.1	6.4	12.3	9.3	13.3	13.6	7.9	13.4	20.5
	Faster-RCNN w SGRN	$9.5^{+2.6}$	$15.3^{+2.5}$	$10.1^{+3.3}$	$4.9^{+1.8}$	$8.4^{+2.0}$	$16.0^{+3.7}$	$12.5^{+3.2}$	$17.6^{+4.3}$	$17.7^{+4.1}$	$8.4^{+0.5}$	$16.0^{+2.6}$	$27.3^{+6.8}$
	FPN[30]	10.9	21.0	12.0	7.3	12.1	18.4	13.5	20.3	20.9	13.3	21.9	29.0
	FPN w SGRN	$\mathbf{14.0}^{+3.1}$	$\mathbf{23.1}^{+2.1}$	$\mathbf{14.8}^{+2.8}$	$\mathbf{8.1}^{+0.8}$	$\mathbf{13.7}^{+1.6}$	$\mathbf{21.4}^{+3.0}$	$\mathbf{16.5}^{+3.0}$	$\mathbf{25.5}^{+5.2}$	$\mathbf{26.2}^{+5.3}$	$\mathbf{17.7}^{+4.4}$	$\mathbf{27.5}^{+5.6}$	$\mathbf{35.3}^{+6.3}$

Table 1. Main results of test datasets on VG_{1000}(Visual Genome), VG_{3000} and ADE_{445}. "w SGRN" is the baseline model Faster-RCNN [45] and FPN [30] adding the proposed SGRN method. Note that comparison of HKRM [23] is not fair since their method here used the relation and attribute annotations of Visual Genome.

Conclusion

- Spatial-aware graph relation network (SGRN) for encoding object relation in the detection system

- Modular Design plugged into existing detection pipeline

- GCN Design
  - Graph Structure from feature map
  - Node representation from classifier
  - Propagation weight from spatial relation

Reproducing 실험

- 구현 내용
  - https://github.com/dohoseok/SGRN_torch
  - 공개된 논문 구현 코드가 없어서 Pytorch Faster RCNN 기반으로 SGRN 구현
  - Faster RCNN, Visual Genome 3000 dataset 에 대해서 실험

- 실험 환경
  - Intel i7-9700KF CPU@ 3.60GHz, 32G RAM, NVIDIA GeForce RTX 2080Ti
  - Windows 10, CUDA 10.1, Python 3.6, PyTorch 1.4.0

- 실험 방법
  - Faster RCNN 120만 iteration 학습 (약 12epoch)
  - 위 weight를 pre-weight로 하여 SGRN 모듈 120만회 학습 (약 12epoch)

Reproducing 실험

- 실험 결과

	Faster RCNN	Faster RCNN w SGRN
Average Precision	11.2	11.5

- SGRN 적용하여 성능 향상 확인 (+0.3)
  - 기존 Detector가 충분히 학습된 상태에서 SGRN을 통한 추가 성능 향상 가능.
  - Visual Genome Dataset을 Train과 Test로 나눈 기준과 클래스 선택 등이 달라서 Average Precision 값 자체는 논문과 차이가 있음.
  - 논문에 설명 부족한 부분 존재
    - Kernel 개수인 K 값을 얼마로 하였는지
    - SGRN 학습 시 기존 baseline detector의 weight는 fix하였는지