BIGDATA & AI ANALYTICS
EXPERT COMPANY

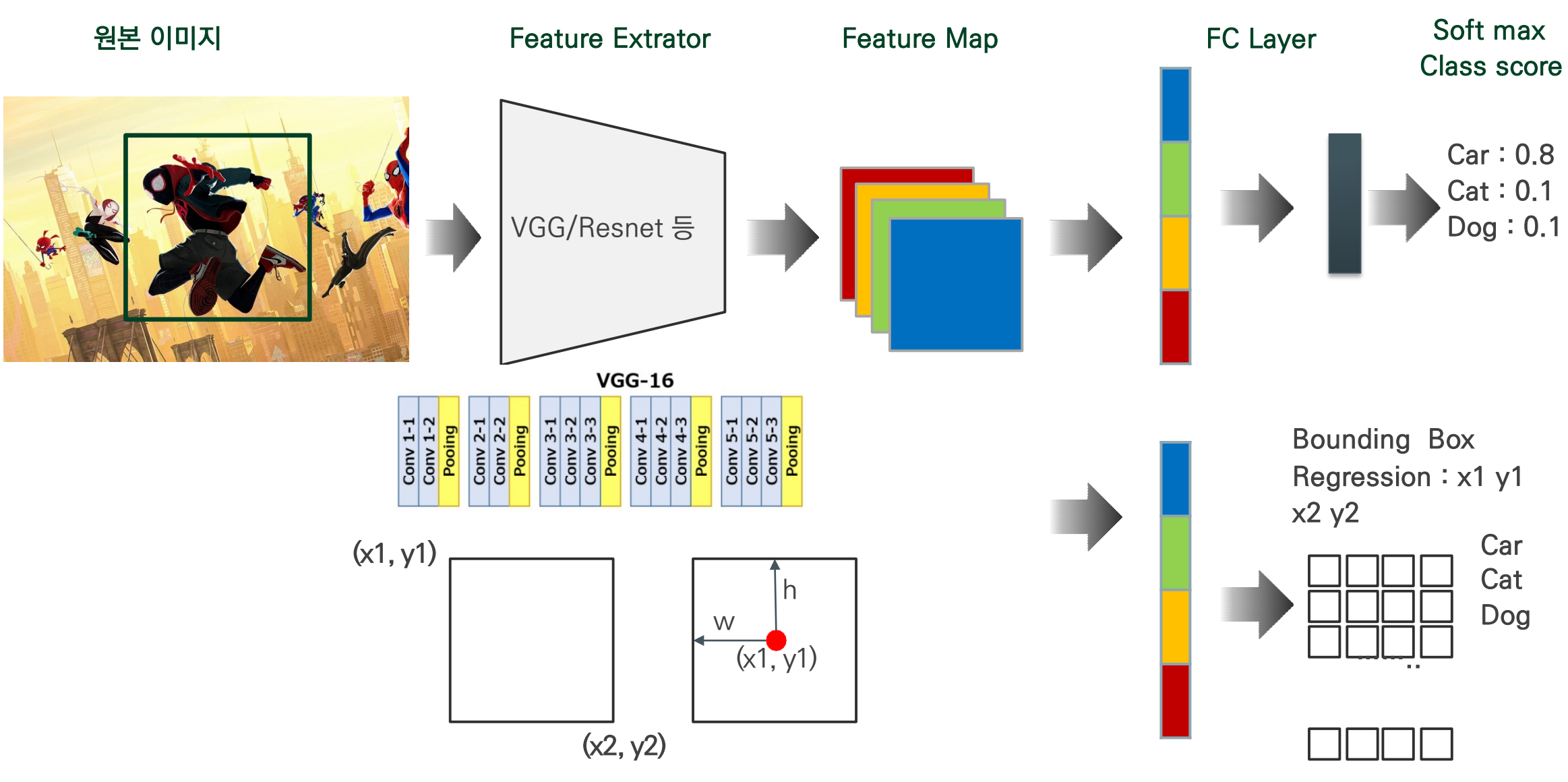
Two-Stage Detection
Faster RCNN

kubwa®

BIGDATA & AI ANALYTICS
EXPERT COMPANY

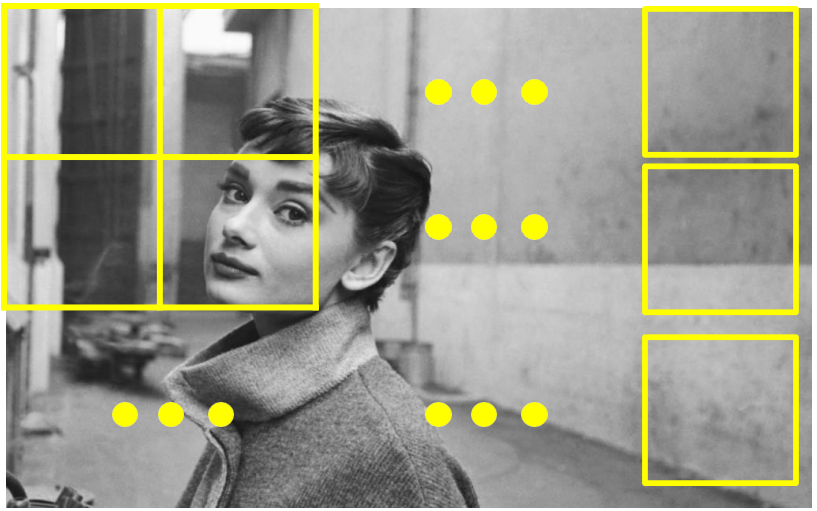
RCNN 개요

Object Detection – bbox Regression



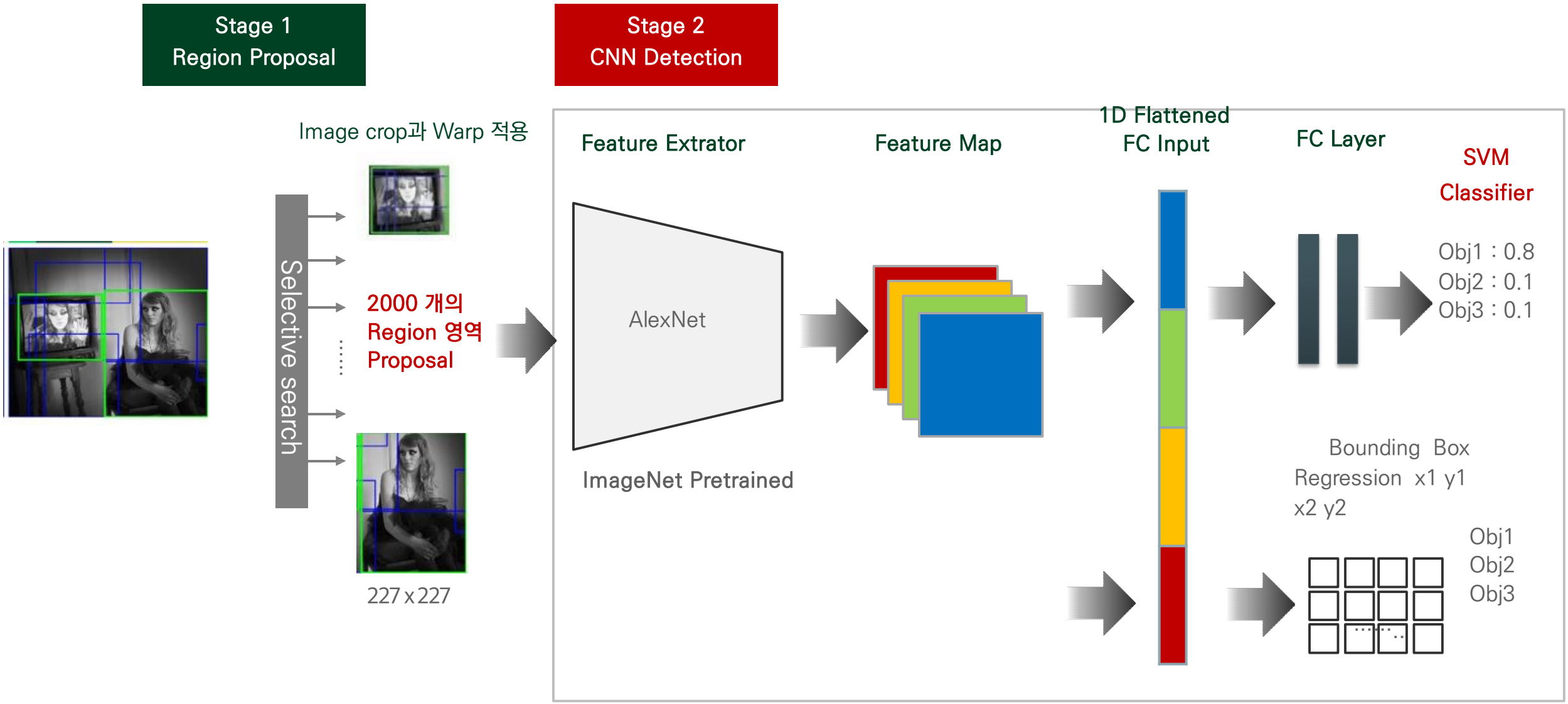
Sliding Window, Region Proposal 방식

Sliding Window 방식

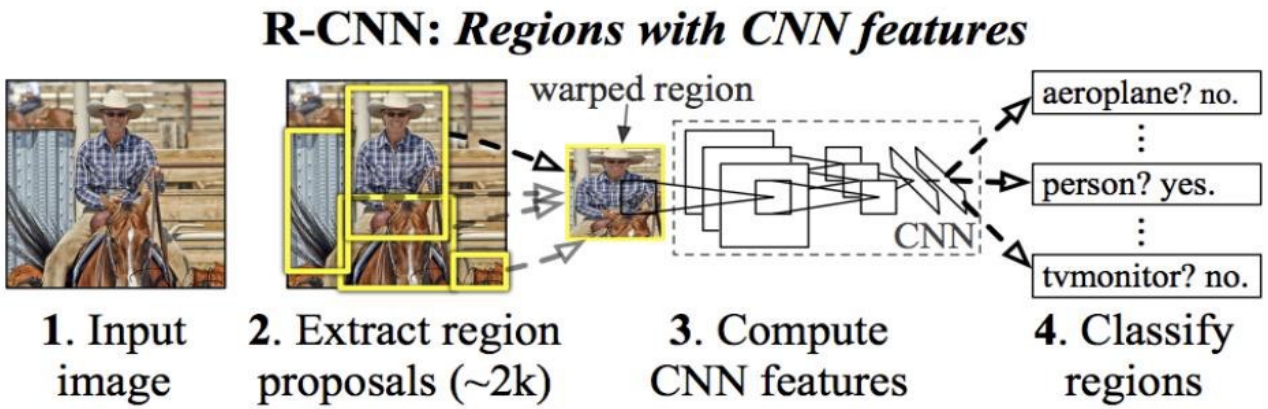


Region Proposal 방식

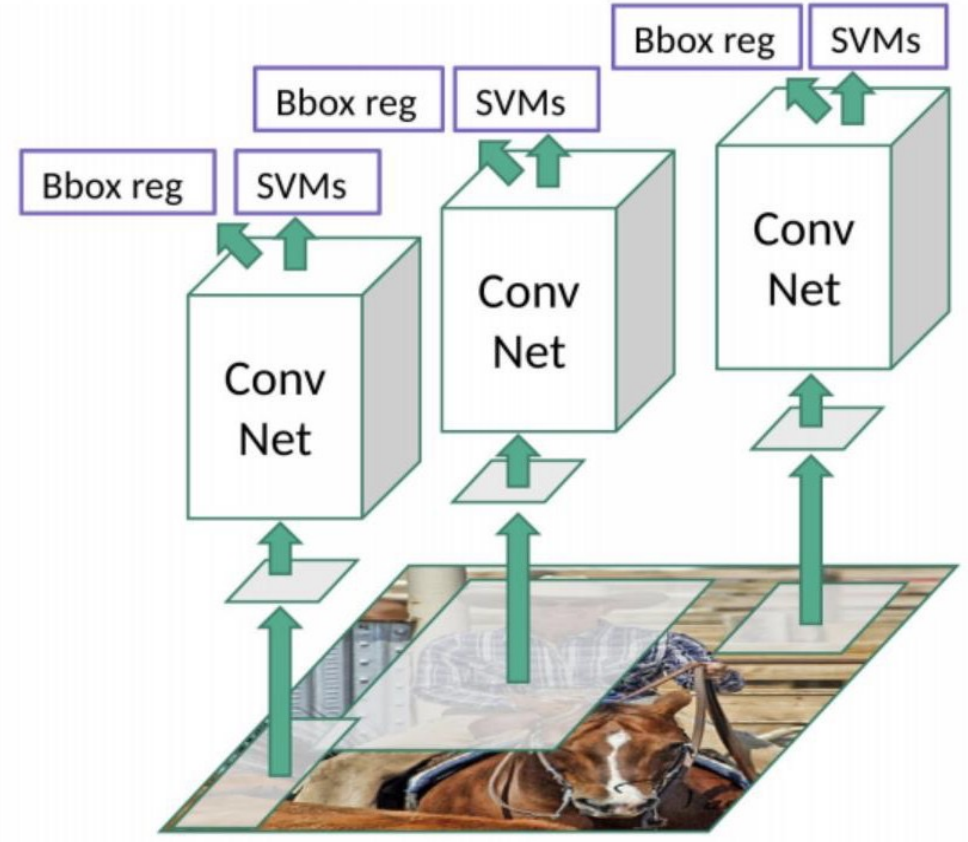
RCNN – Region Proposal 기반



R-CNN 개요



R-CNN



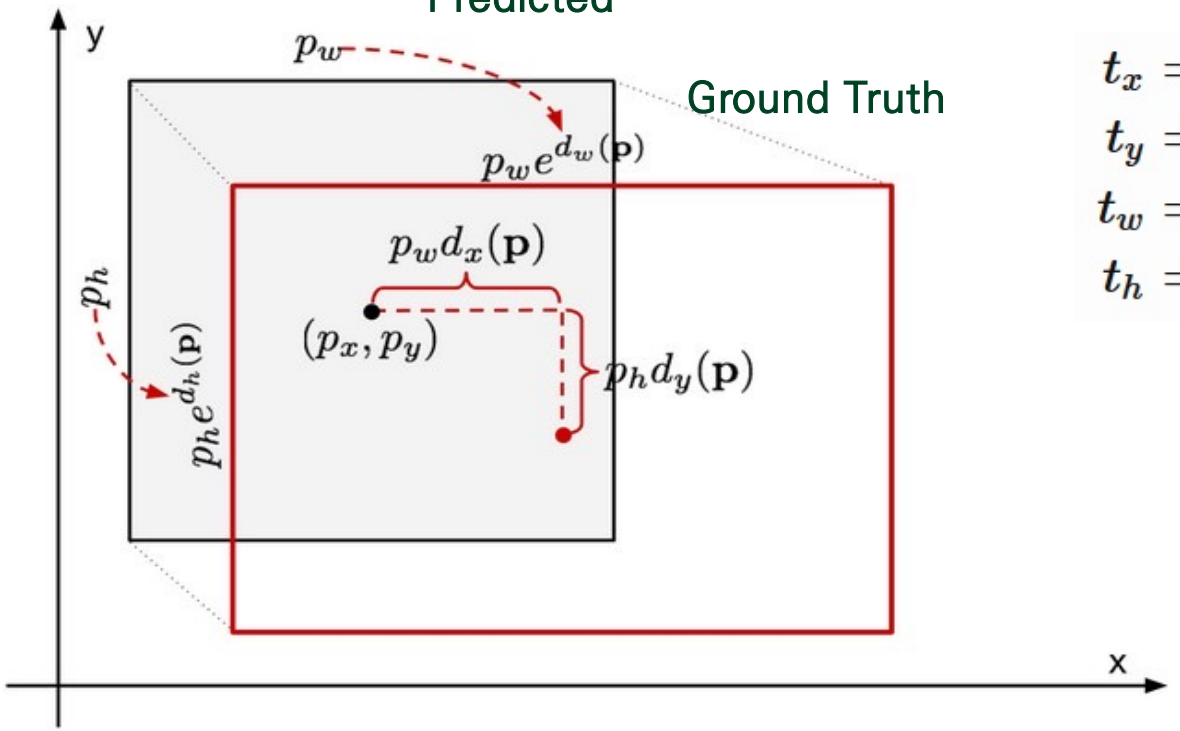
R-CNN

Bounding Box Regression

수정 예측 값

$$\hat{g}_x = p_w d_x(\mathbf{p}) + p_x$$
$$\hat{g}_y = p_h d_y(\mathbf{p}) + p_y$$
$$\hat{g}_w = p_w \exp(d_w(\mathbf{p}))$$
$$\hat{g}_h = p_h \exp(d_h(\mathbf{p}))$$

Predicted



Target

$$t_x = (g_x - p_x)/p_w$$
$$t_y = (g_y - p_y)/p_h$$
$$t_w = \log(g_w/p_w)$$
$$t_h = \log(g_h/p_h)$$

손실 함수

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$$

R-CNN 성능 비교

kubwa

PASCAL VOC 2010 기준

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

RCNN의 장단점

kubwa

장점

- 당시에 높은 Detection 정확도
- 동시대의 다른 알고리즘 대비 매우 높은 Detection 정확도

단점

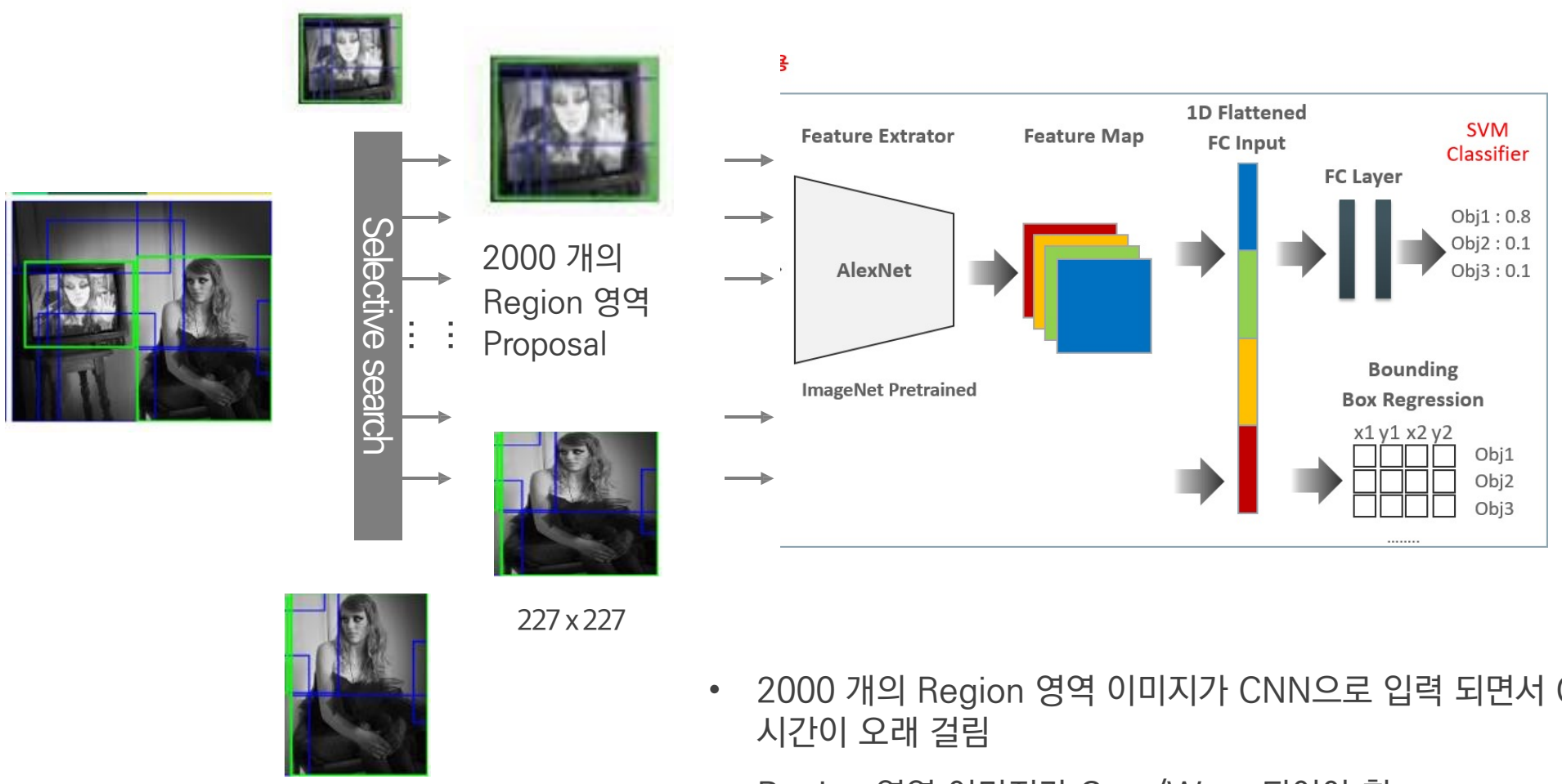
- 1장의 이미지를 Object Detection 하는데 약 50초 소요
- 너무 느린 Detection 시간과 복잡한 아키텍처 및 학습 프로세스
- 하나의 이미지에서 selective search를 수행하여 2000개의 region 영역을 도출
- 2000개씩 생성된 region 이미지를 CNN Feature map 생성
- 복잡한 구성 요소들: Selective search, CNN Feature Extractor, SVM과 Bounding box regressor로 구성

BIGDATA & AI ANALYTICS
EXPERT COMPANY

SPP Net
(Spatial Pyramid Pooling)

RCNN 네트워크의 문제점

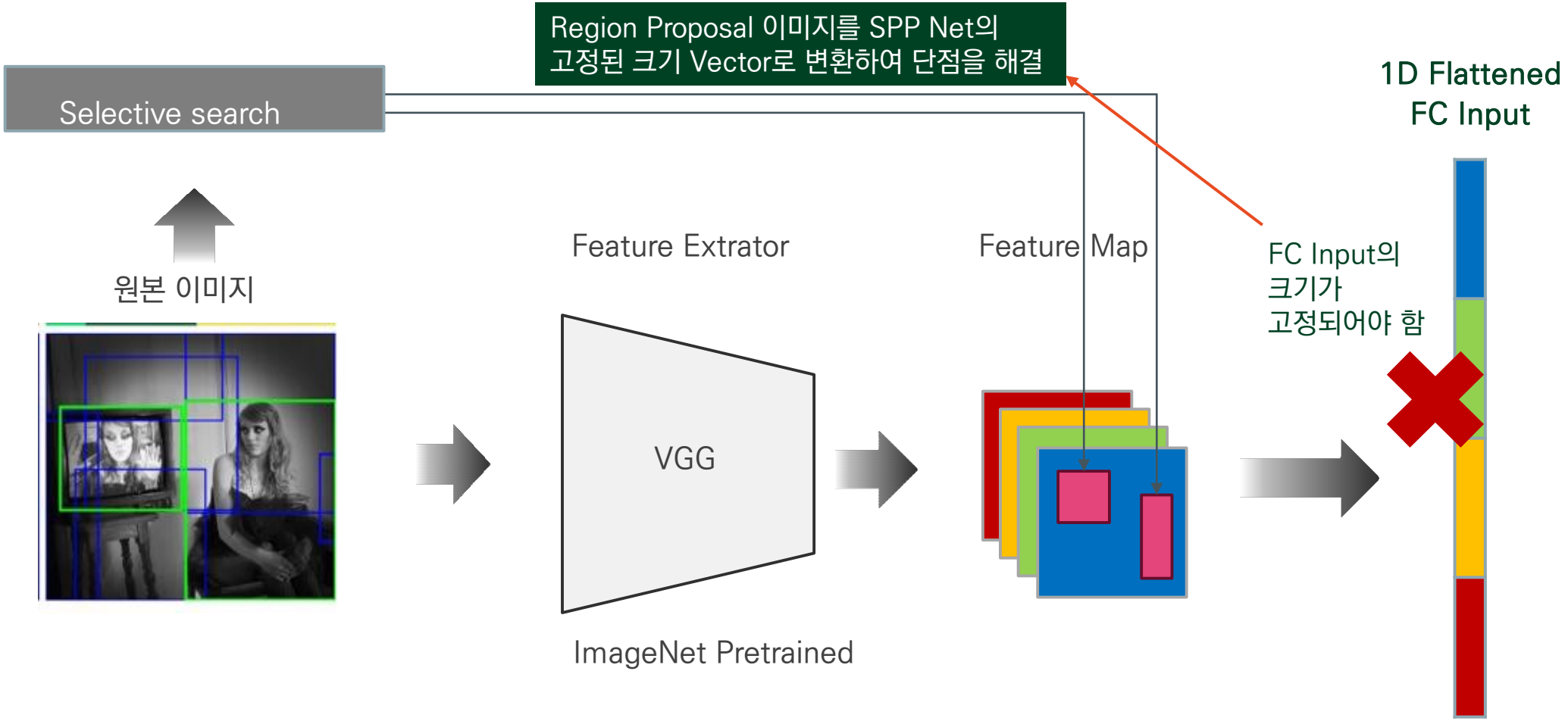
Image crop과 Warp 적용



- 2000 개의 Region 영역 이미지가 CNN으로 입력 되면서 Object Detection 수행 시간이 오래 걸림
- Region 영역 이미지가 Crop/Warp 되어야 함.

RCNN 속도 지연 개선

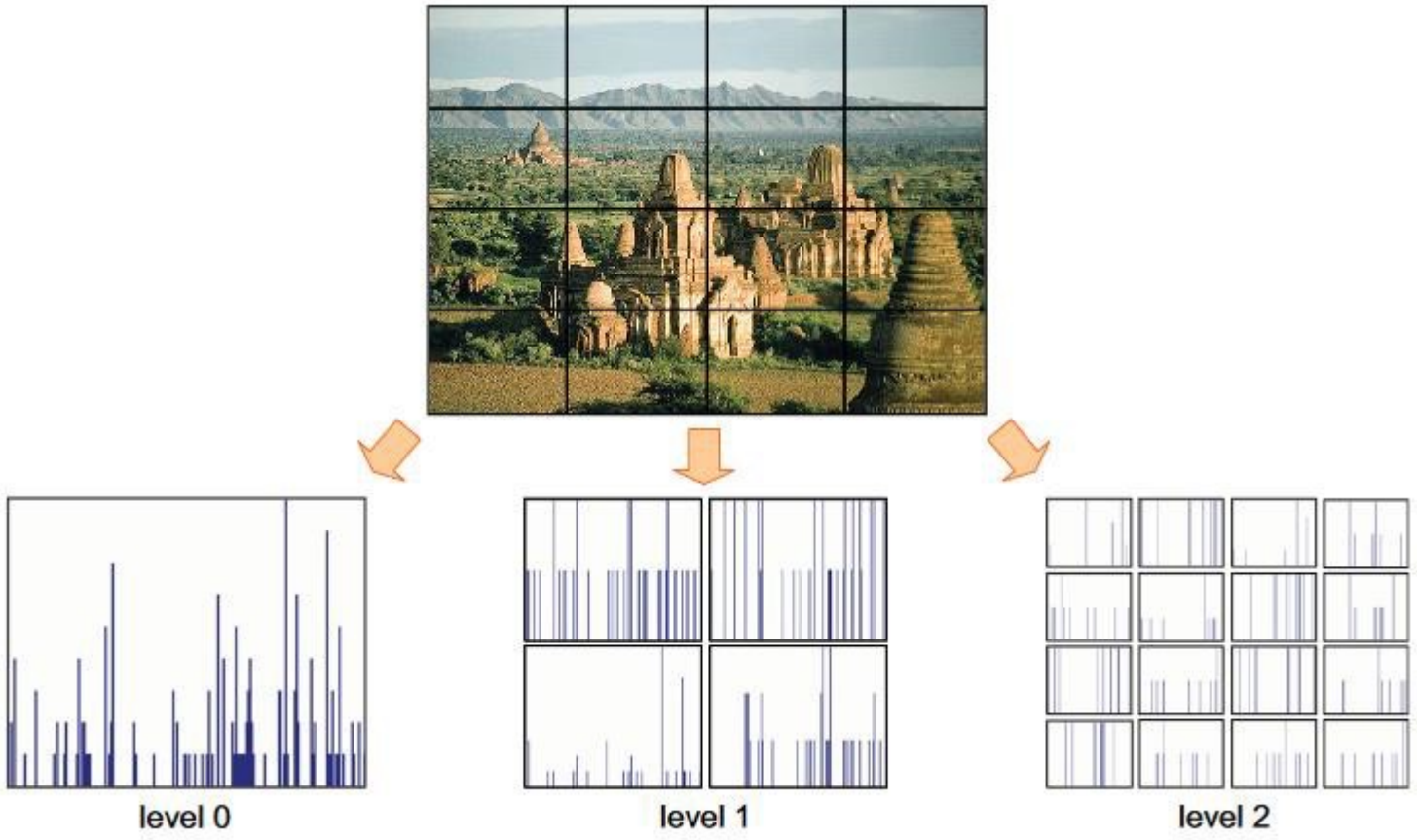
- 2000개의 Region Proposal 이미지를 CNN 으로 Feature Extraction 하지 않고 원본 이미지만 CNN으로 Feature Map 생성
- 원본 이미지의 Selective search 로 추천된 영역의 이미지만 Feature Map으로 매핑하여 별도 추출



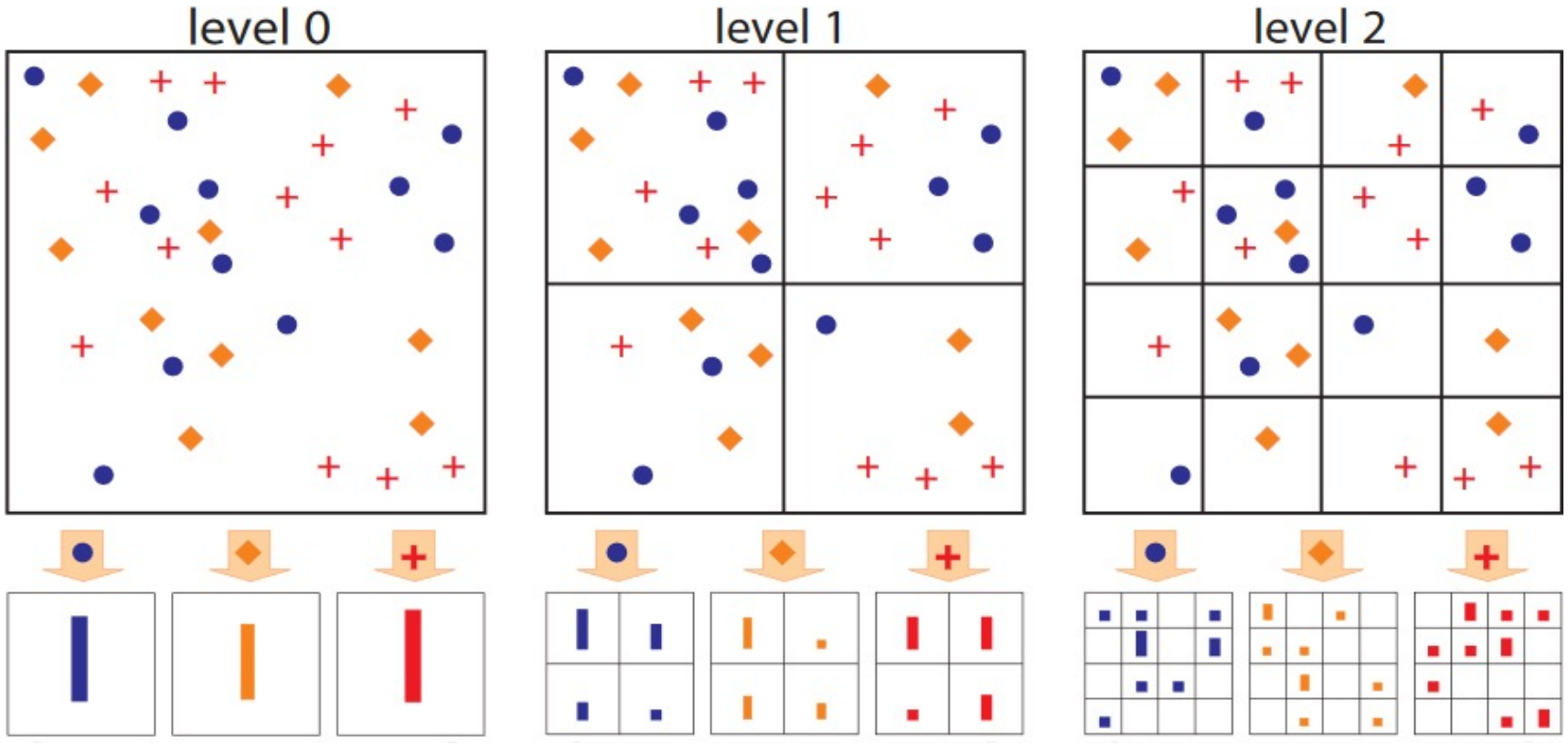
SPP(Spatial pyramid Pooling)

- SPP는 CNN상에서 Image classification에서 서로 다른 이미지의 크기를 고정된 크기로 변환하는 매칭 기법

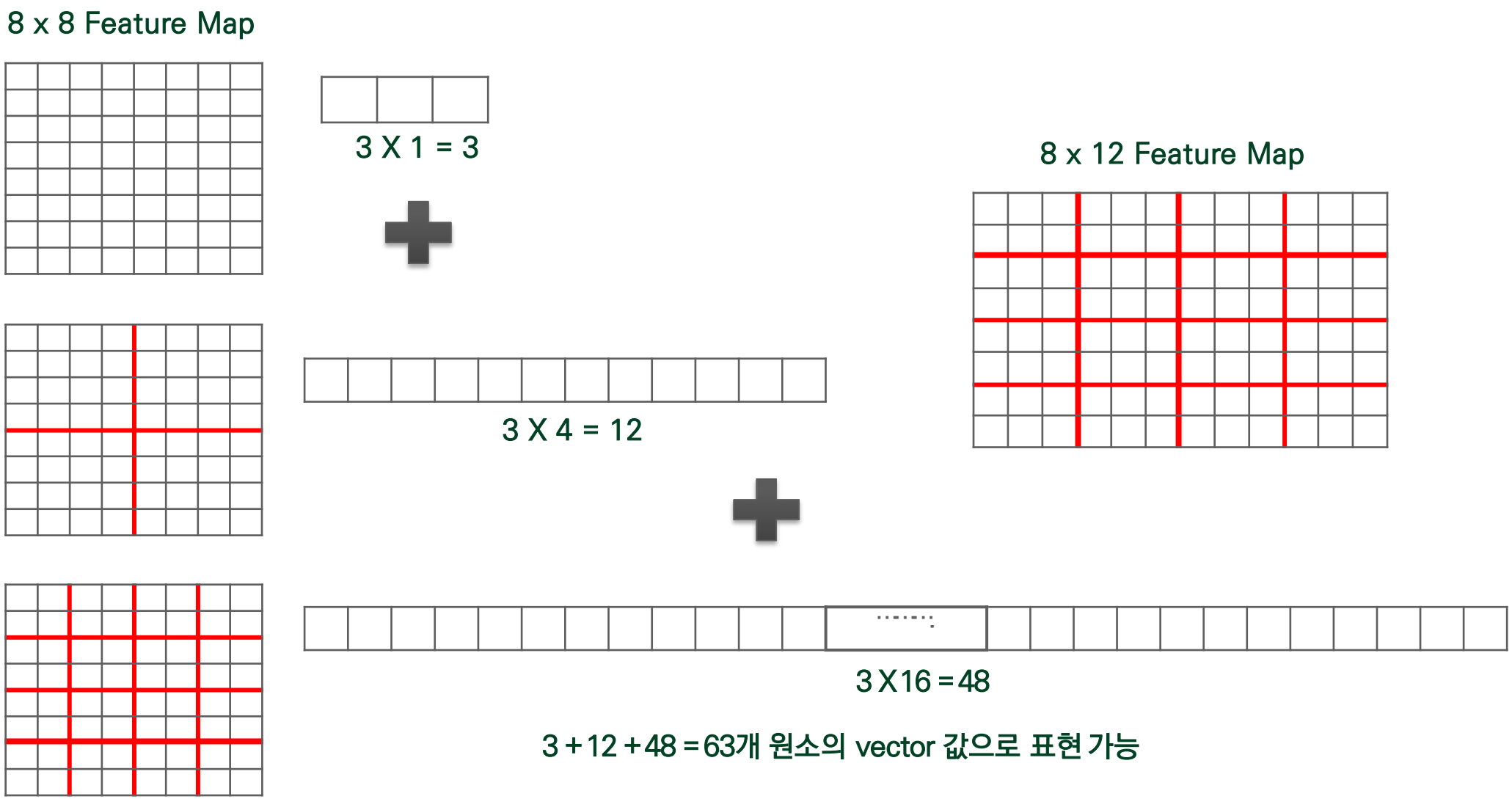
SPM(Spatial Pyramid Matching) 개요



SPM(Spatial Pyramid Matching) 개요



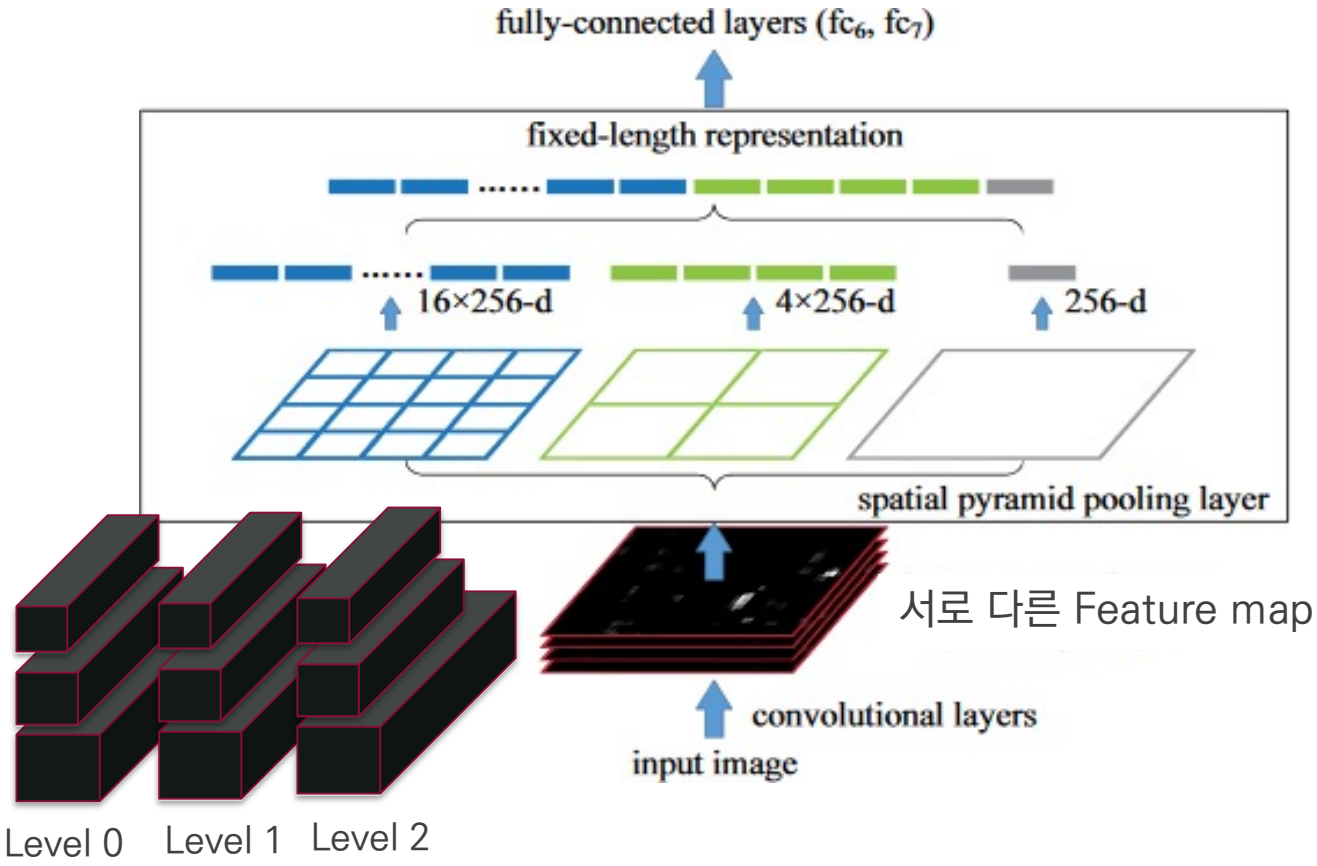
SPM을 Feature Map을 균일한 Vector 크기로 표현



3 + 12 + 48 = 63개 원소의 vector 값으로 표현 가능

SPP-Net 이미지 Classification

이미지 Classification에서의 SPP-Net 구조

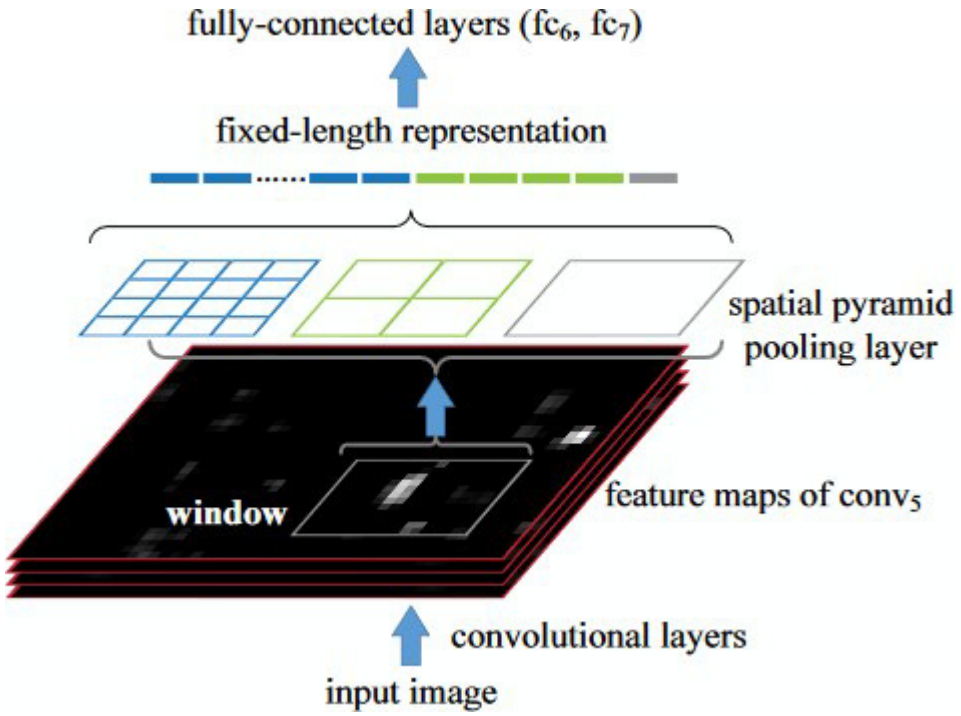


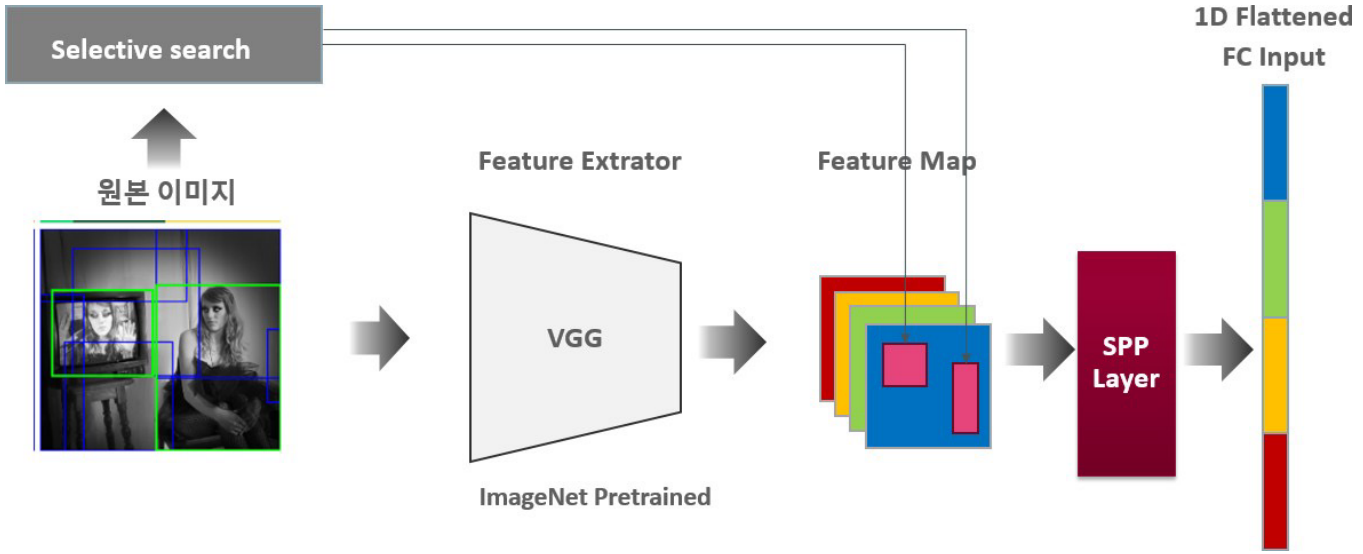
서로 다른 Feature map 사이즈

2014 ILSVRC Classification 결과

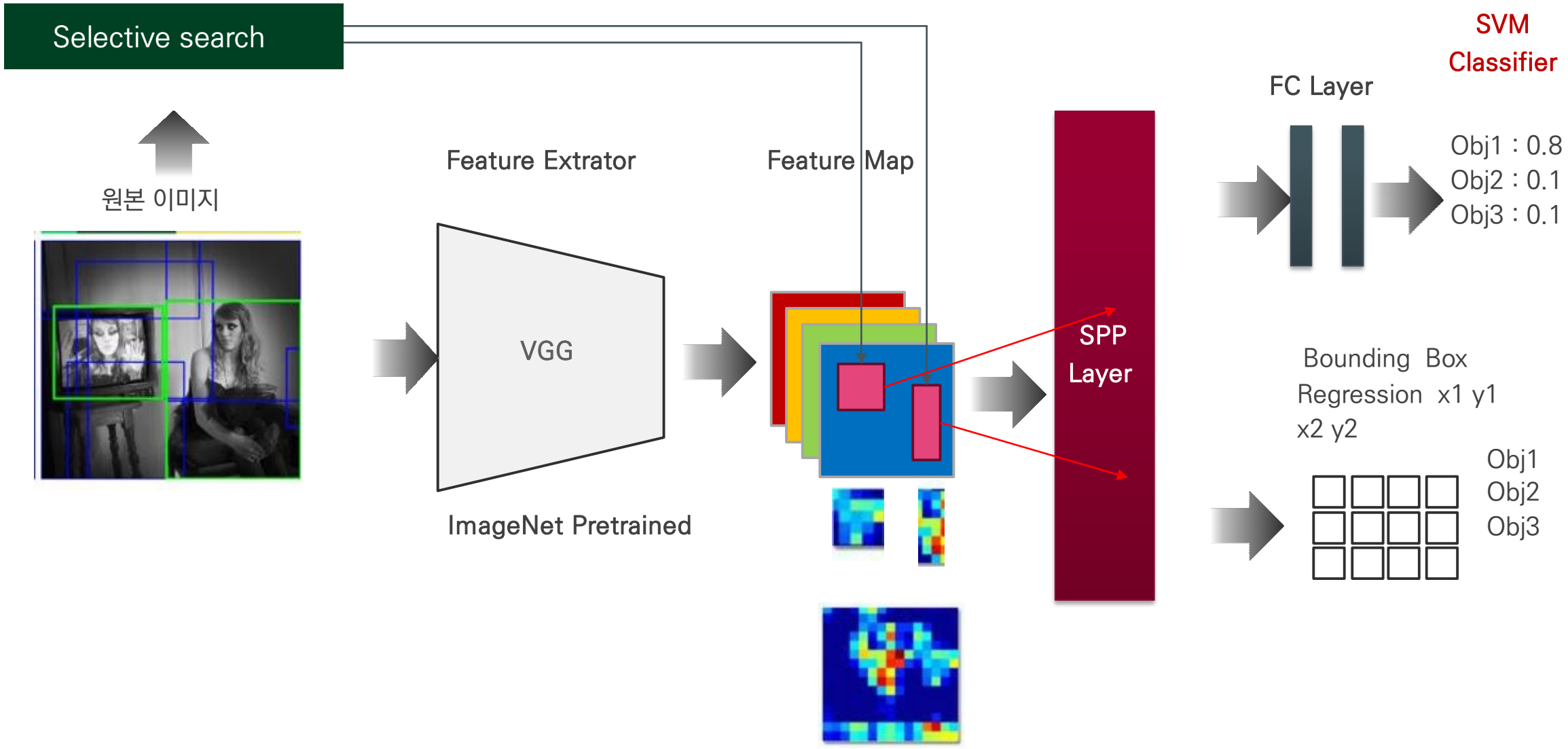
rank	team	top-5 test
1	GoogLeNet [32]	**6.66**
2	VGG [33]	7.32
3	ours	8.06
4	Howard	8.11
5	DeeperVision	9.50
6	NUS-BST	9.79
7	TTIC_ECP	10.22

SPP-Net Object Detection



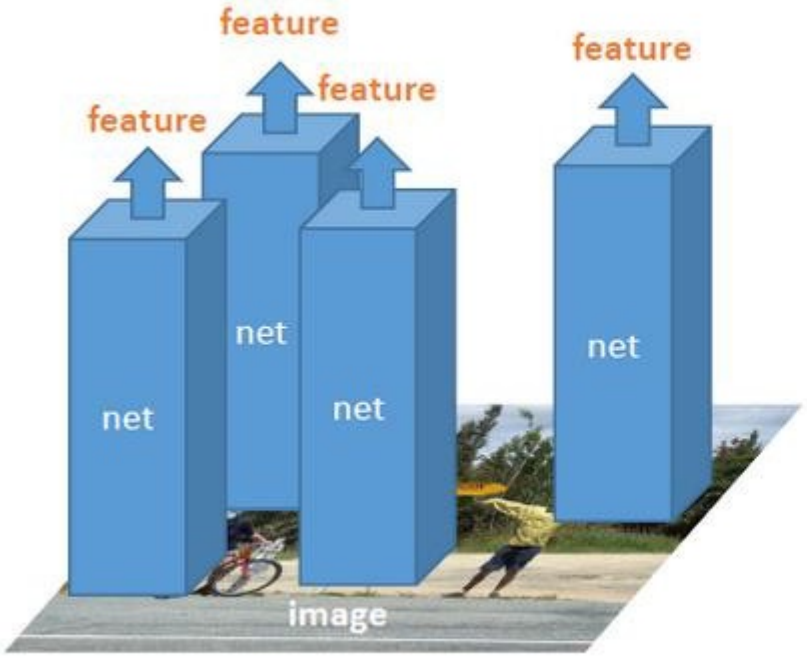


SPP-Net 구조


Selective search
원본 이미지
Feature Extrator
VGG
ImageNet Pretrained
Feature Map
SPP
Layer
FC Layer
SVM
Classifier
Obj1 : 0.8
Obj2 : 0.1
Obj3 : 0.1
Bounding Box
Regression x1 y1
x2 y2
Obj1
Obj2
Obj3

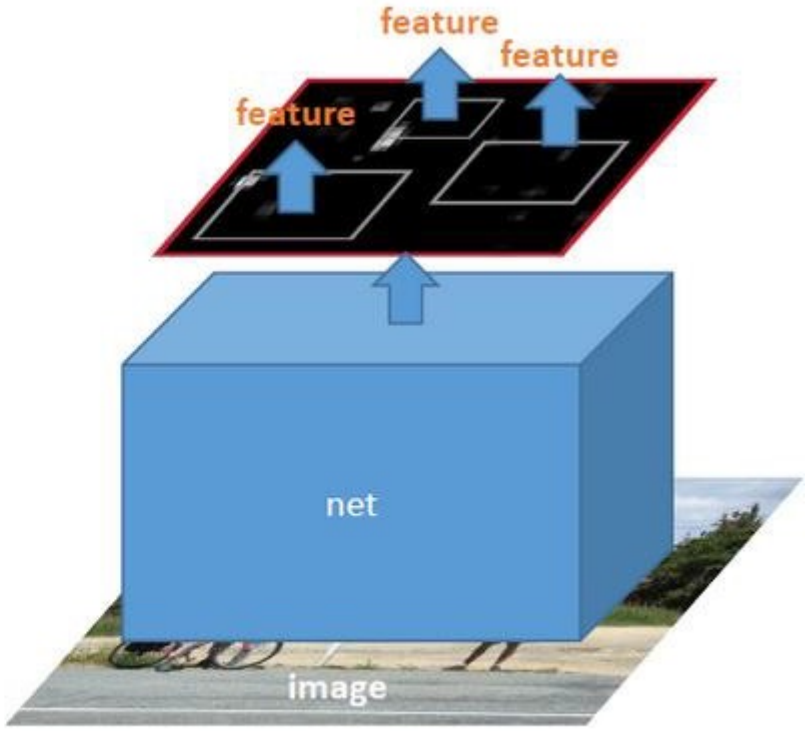
R-CNN과 SPP-Net 비교

이미지 한 개에 2000번 CNN을 통과



R-CNN

이미지 한 개는 한번만 CNN 통과



SPP-net

kubwa

SPP-Net 적용 후 성능 결과

PASCAL VOC 2007 데이터 세트 적용 결과

method	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
DPM [23]	33.7	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5
SS [20]	33.8	43.5	46.5	10.4	12.0	9.3	49.4	53.7	39.4	12.5	36.9	42.2	26.4	47.0	52.4	23.5	12.1	29.9	36.3	42.2	48.8
Regionlet [39]	41.7	54.2	52.0	20.3	24.0	20.1	55.5	68.7	42.6	19.2	44.2	49.1	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3
DetNet [40]	30.5	29.2	35.2	19.4	16.7	3.7	53.2	50.2	27.2	10.2	34.8	30.2	28.2	46.6	41.7	26.2	10.3	32.8	26.8	39.8	47.0
RCNN ftfc$_7$ (A5)	54.2	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7
RCNN ftfc$_7$ (ZF5)	55.1	64.8	68.4	47.0	39.5	30.9	59.8	70.5	65.3	33.5	62.5	50.3	59.5	61.6	67.9	54.1	33.4	57.3	52.9	60.2	62.9
SPP ftfc$_7$ (ZF5)	55.2	65.5	65.9	51.7	38.4	32.7	62.6	68.6	69.7	33.1	66.6	53.1	58.2	63.6	68.8	50.4	27.4	53.7	48.2	61.7	64.7
RCNN bb (A5)	58.5	68.1	72.8	56.8	**43.0**	36.8	**66.3**	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8
RCNN bb (ZF5)	**59.2**	68.4	**74.0**	54.0	40.9	35.2	64.1	**74.4**	69.8	**35.5**	66.9	53.8	**64.2**	69.9	69.6	**58.9**	**36.8**	**63.4**	**56.0**	62.8	64.9
SPP bb (ZF5)	**59.2**	**68.6**	69.7	**57.1**	41.2	**40.5**	**66.3**	71.3	**72.5**	34.4	**67.3**	**61.7**	63.1	**71.0**	**69.8**	57.6	29.7	59.0	50.2	**65.2**	**68.0**

ILSVRC 2014 Obejct Detection 결과

rank	team	mAP
1	NUS	**37.21**
2	ours	35.11
3	UvA	32.02
-	(our single-model)	(31.84)
4	Southeast-CASIA	30.47
5	1-HKUST	28.86
6	CASIA_CRIPAC_2	28.61

PASCAL VOC 2007 데이터 세트 기반 수행 시간

	SPP-net 1-scale	SPP-net 5-scale	RCNN
mAP	58.0	**59.2**	58.5
GPU time / img	**0.14s**	0.38s	9s
speed-up	**64x**	**24x**	-

kubwa®

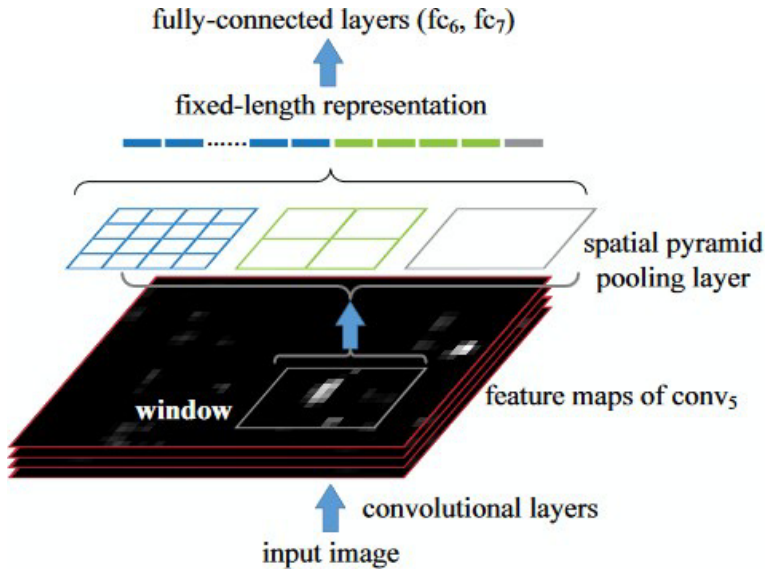
BIGDATA & AI ANALYTICS
EXPERT COMPANY

Fast RCNN

Fast RCNN 주요 특징

SPP Layer를 ROI Pooling Layer로

SPP → ROI Pooling

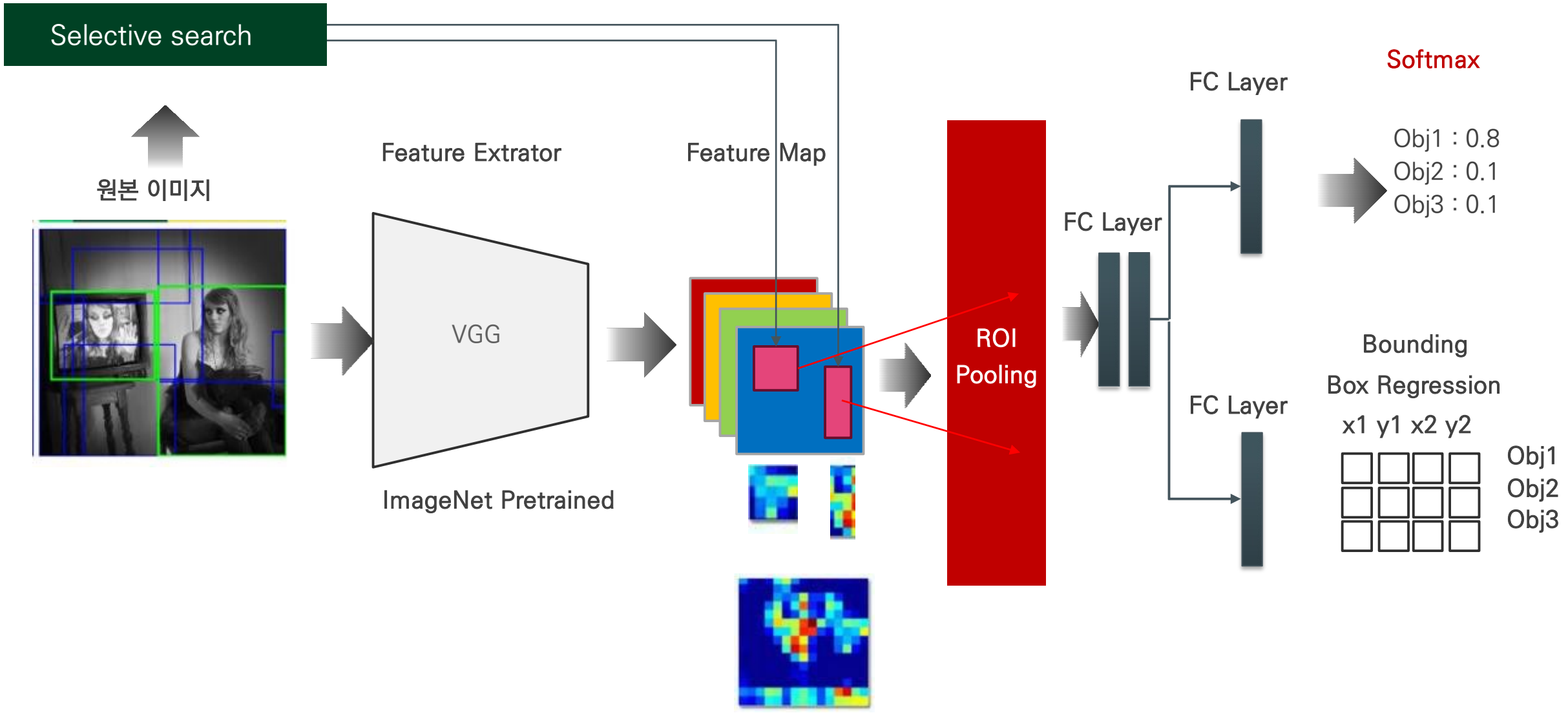


End-to-End Network Learning (ROI Proposal은 제외)

- SVM을 Softmax 로 변환
- Multi-task loss 함수로 Classification과 Regression을

  함께 최적화

Fast RCNN 구조


Selective search
원본 이미지
Feature Extrator
VGG
ImageNet Pretrained
Feature Map
ROI
Pooling
FC Layer
FC Layer
FC Layer
Softmax
Obj1 : 0.8
Obj2 : 0.1
Obj3 : 0.1
Bounding
Box Regression
x1 y1 x2 y2
Obj1
Obj2
Obj3

Multi-task loss 함수

Classification과 Regression Loss를 함께 반영한 Loss 함수

Multi-task Loss 함수:

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v),$$

Classification Loss　　Regression Loss

Regression Loss 함수:

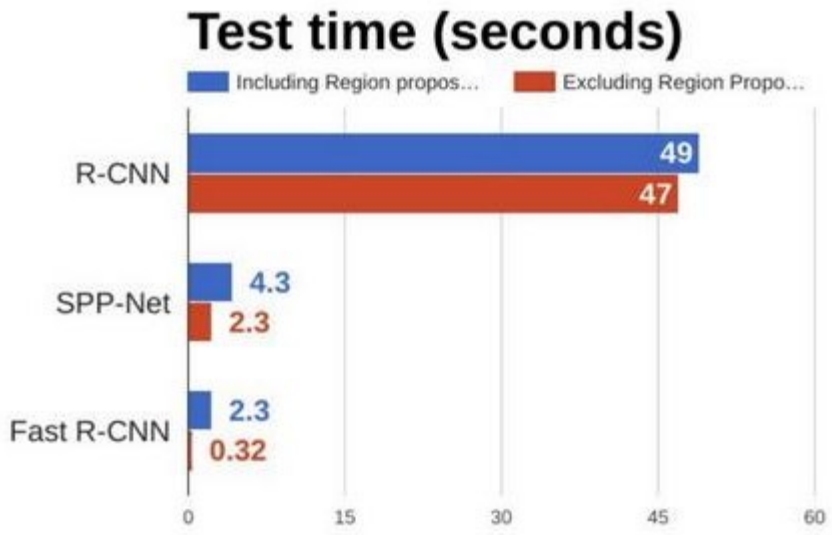
$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i), \qquad \text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Fast RCNN 성능 결과

PASCAL VOC 2012 데이터 세트 적용 결과

method	train set	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	mAP
BabyLearning	Prop.	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6	63.2
NUS_NIN_c2000	Unk.	80.2	73.8	61.9	43.7	**43.0**	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	**38.6**	**68.3**	58.0	68.7	63.3	63.8
R-CNN BB [10]	12	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3	62.4
FRCN [ours]	12	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7	65.7
FRCN [ours]	07++12	**82.3**	**78.4**	**70.8**	**52.3**	38.7	**77.8**	**71.6**	**89.3**	**44.2**	**73.0**	**55.0**	**87.5**	**80.5**	**80.8**	**72.0**	35.1	**68.3**	**65.7**	**80.4**	**64.2**	**68.4**

수행 시간 비교



kubwa®

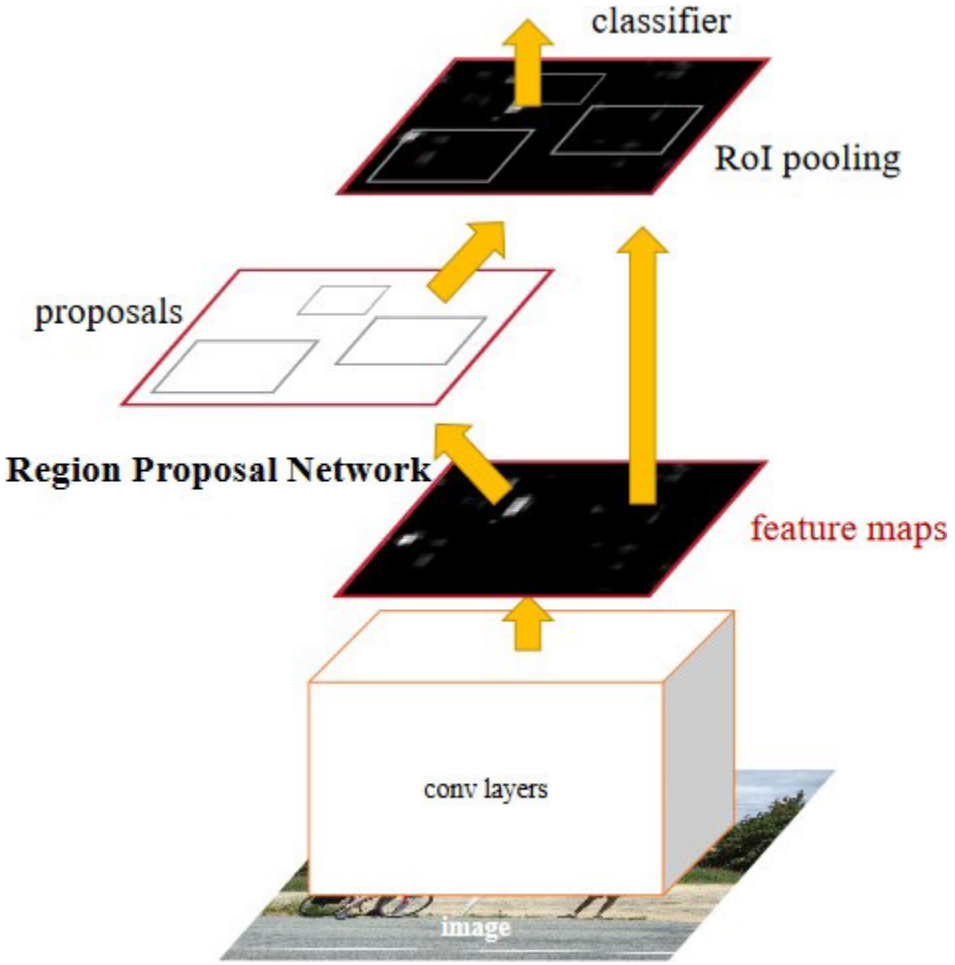
BIGDATA & AI ANALYTICS
EXPERT COMPANY

Faster RCNN

Faster RCNN 개요

kubwa

Faster RCNN = RPN + Fast RCNN



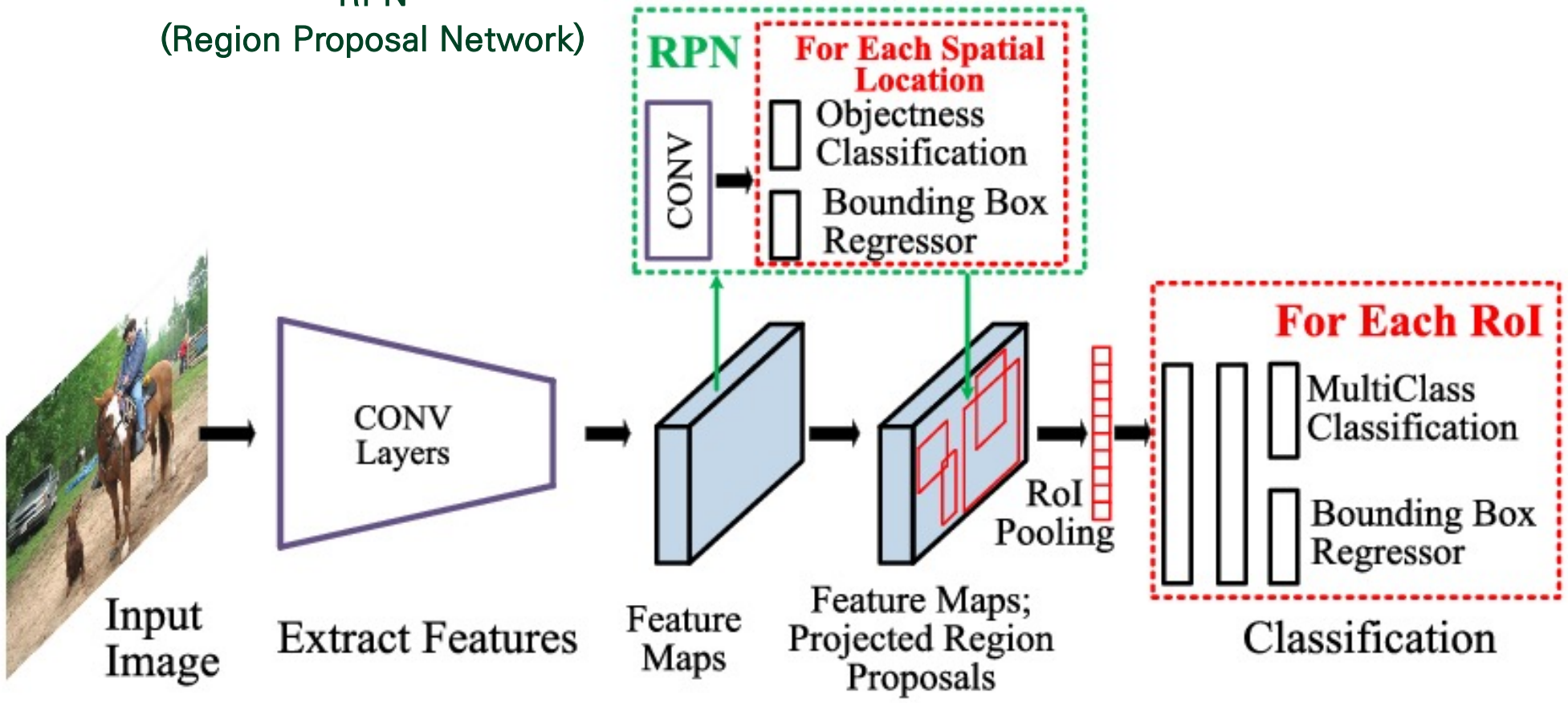
Faster RCNN

Faster R-CNN 구조

Selective Search를 Neural Network 구조로 변경

- GPU 사용으로 빠른 학습/Inference
- End to End Network 학습

RPN
(Region Proposal Network)



Region Proposal Network 구현 이슈

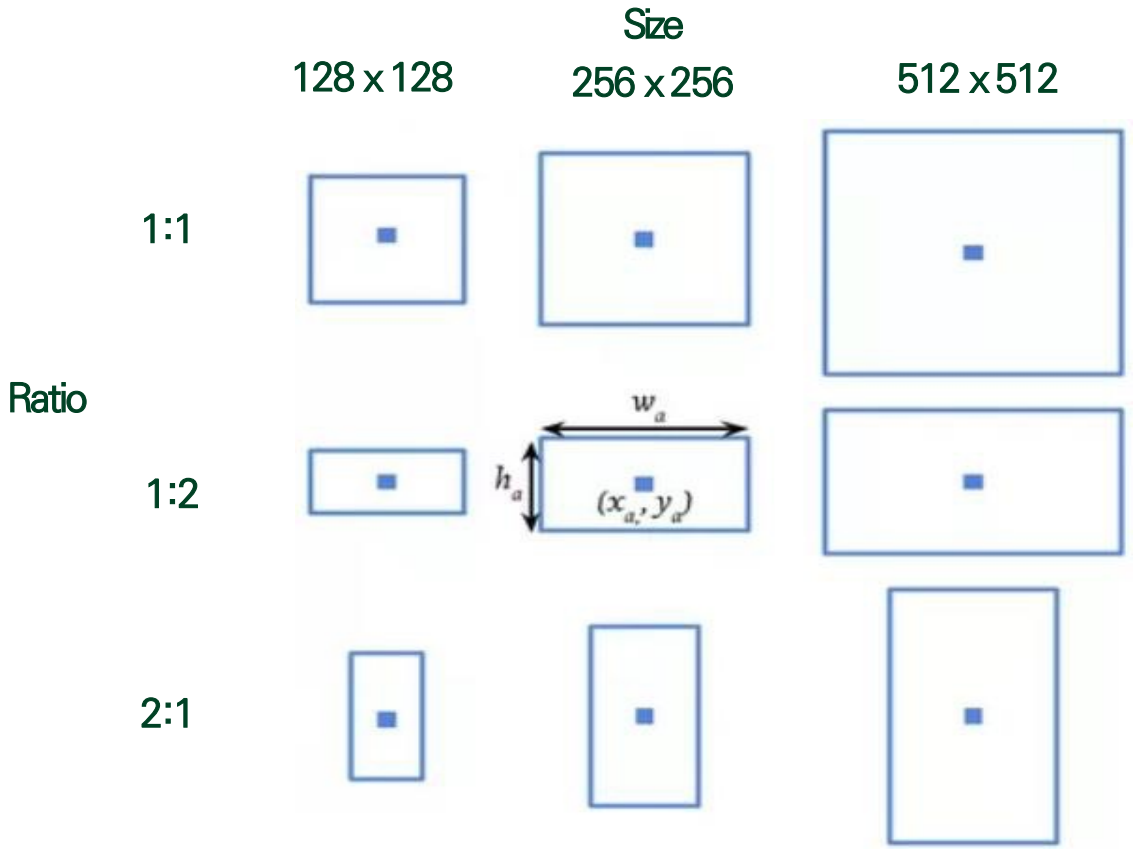
Selective Search를 대체하기 위한 Region Proposal Network 구현 이슈

- 데이터로 주어질 피처는 pixel 값, Target은 Ground Truth Bounding Box인데 이를 이용해 어떻게 Selective Search 수준의 Region Proposal을 할 수 있을 것인가?

(Reference) Anchor Box

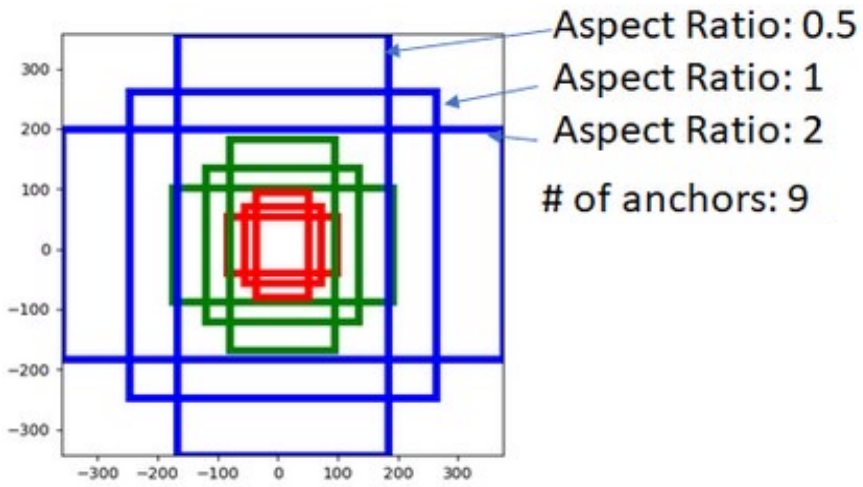
Object 가 있는지 없는지의 후보 Box

Anchor Box

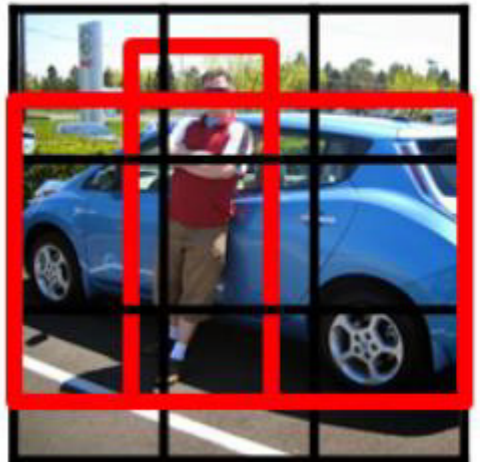


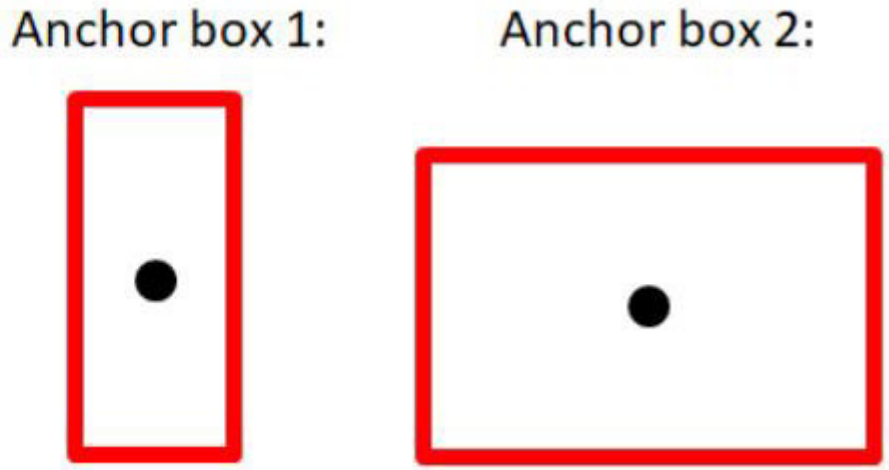
총 9 개의 Anchor box, 3개의 서로 다른 크기,

3개의 서로 다른 ratio로 구성



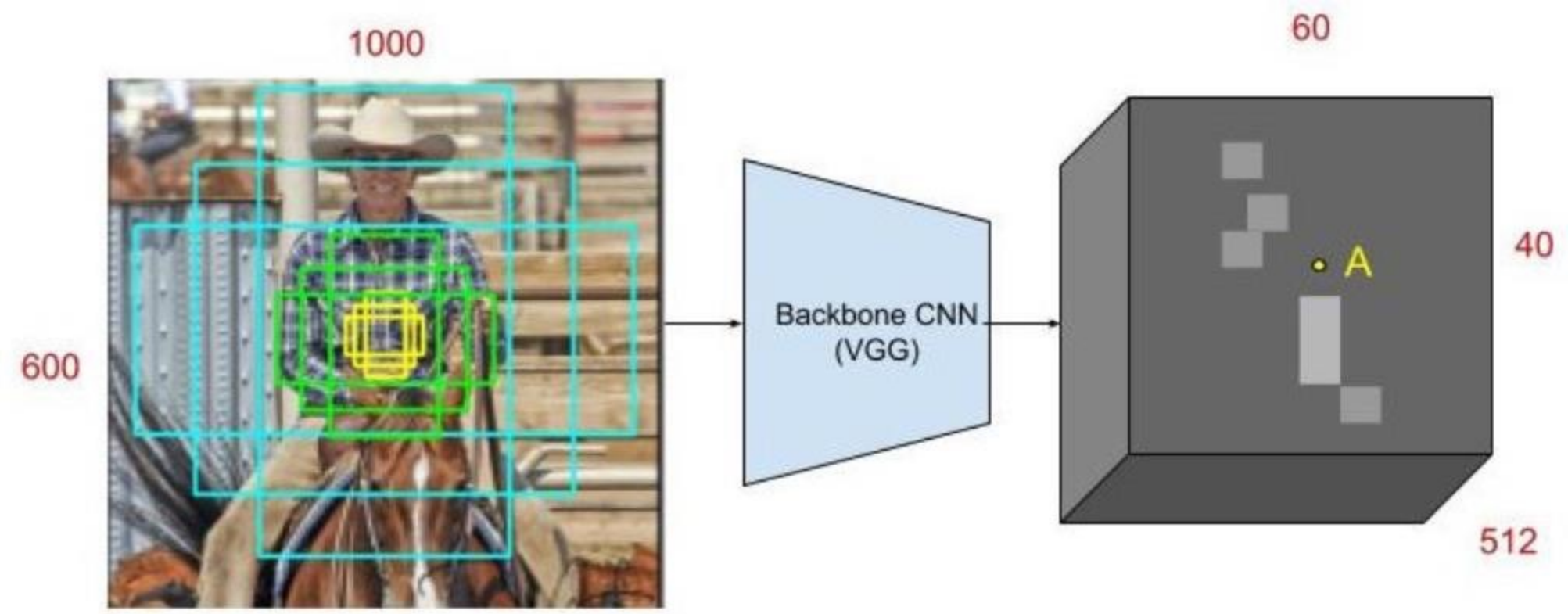
Anchor Box





Anchor Box

- 이미지와 Feature Map에서 Anchor Box 매핑



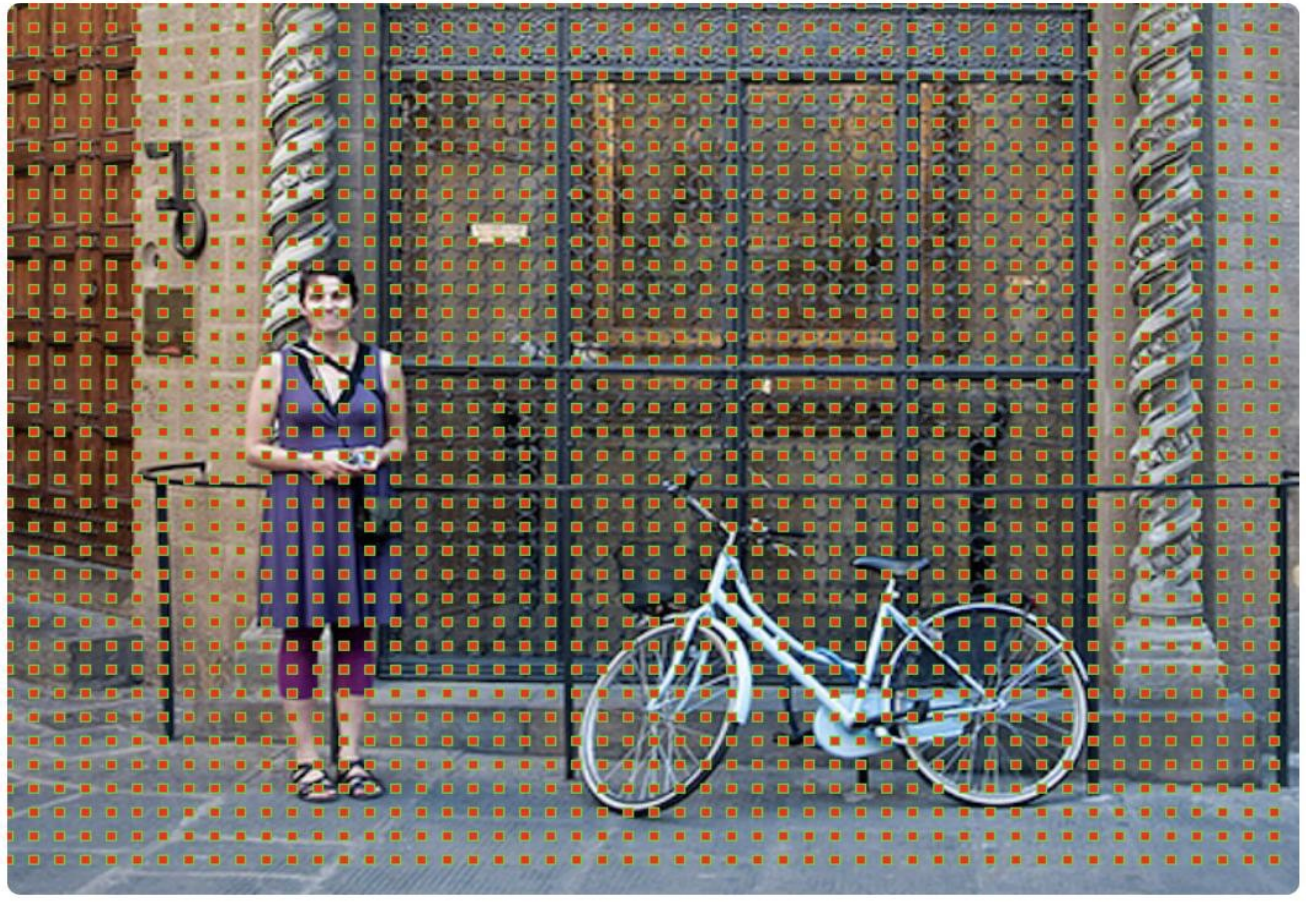
Anchor Box

kubwa

- 이미지와 Feature Map에서 Anchor Box 매핑

800

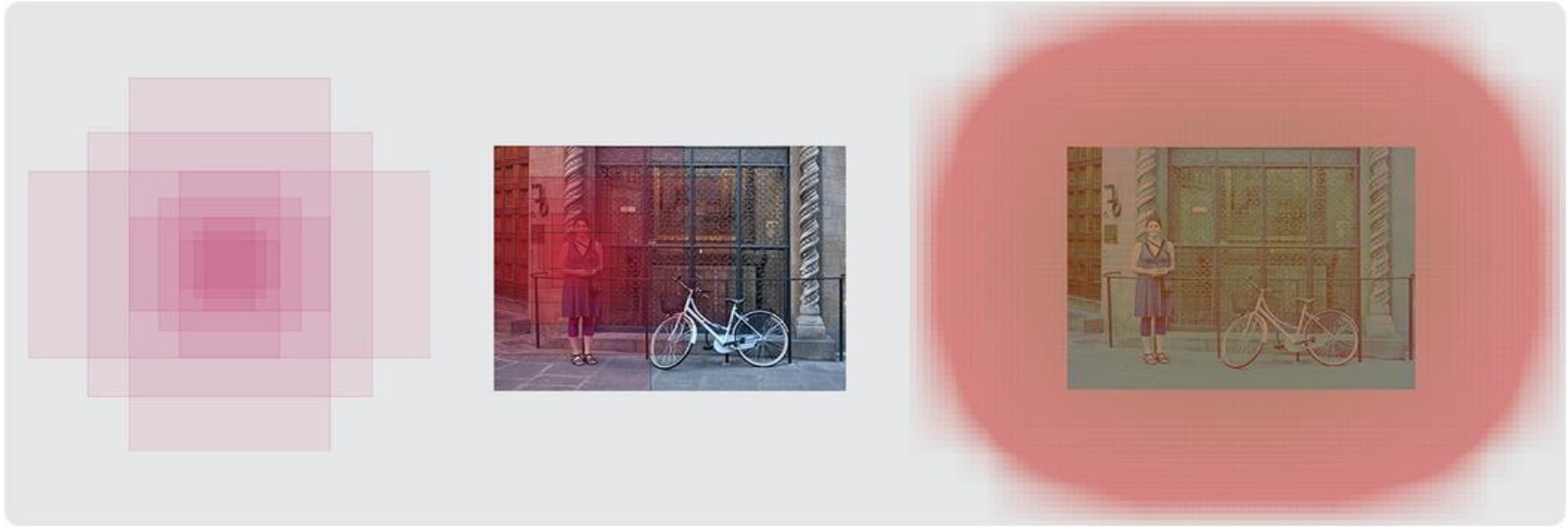
600



Anchor Box

kubwa

- 이미지와 Feature Map에서 Anchor Box 매핑

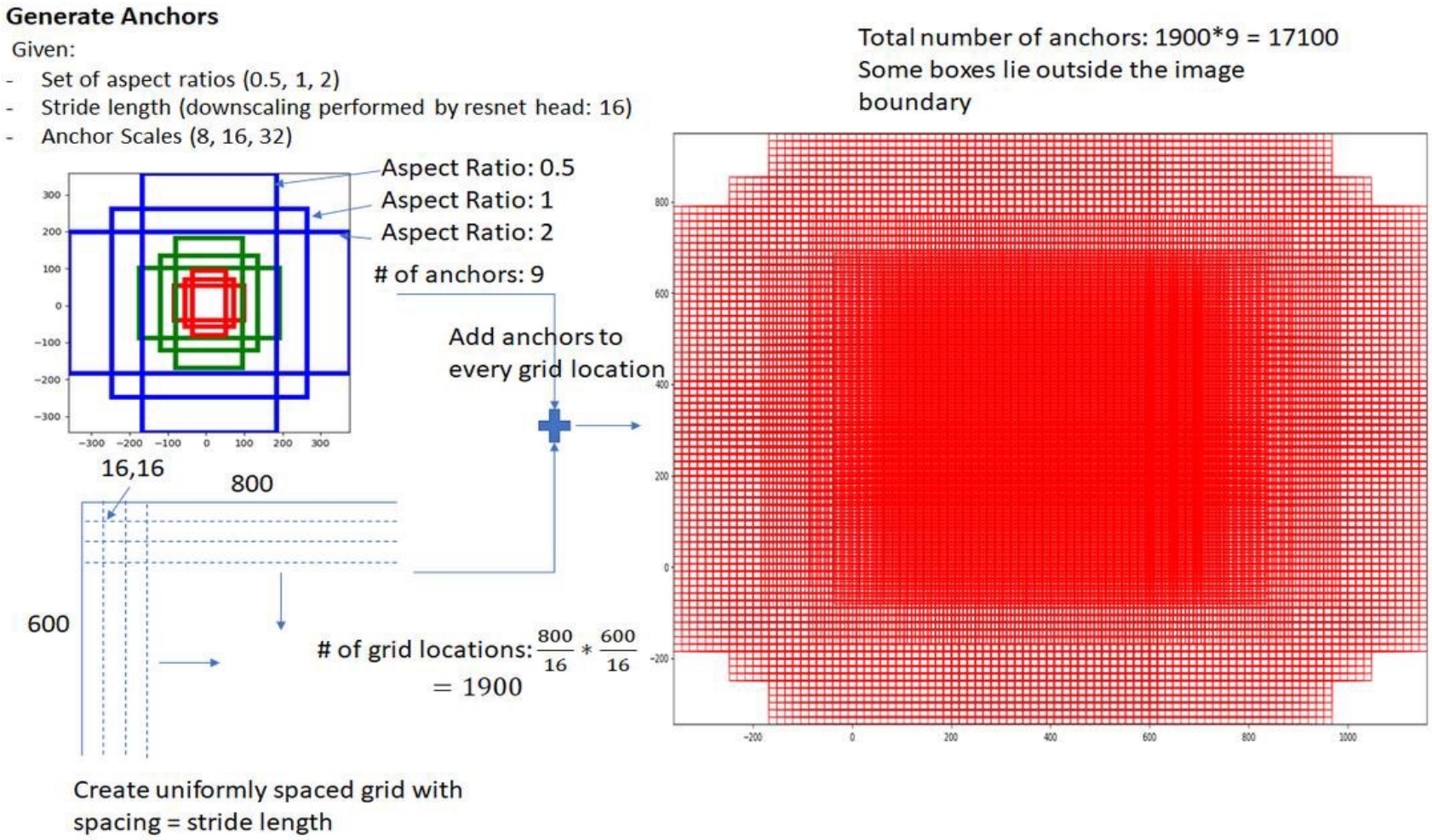


Left: Anchors, Center: Anchor for a single point, Right: All anchors

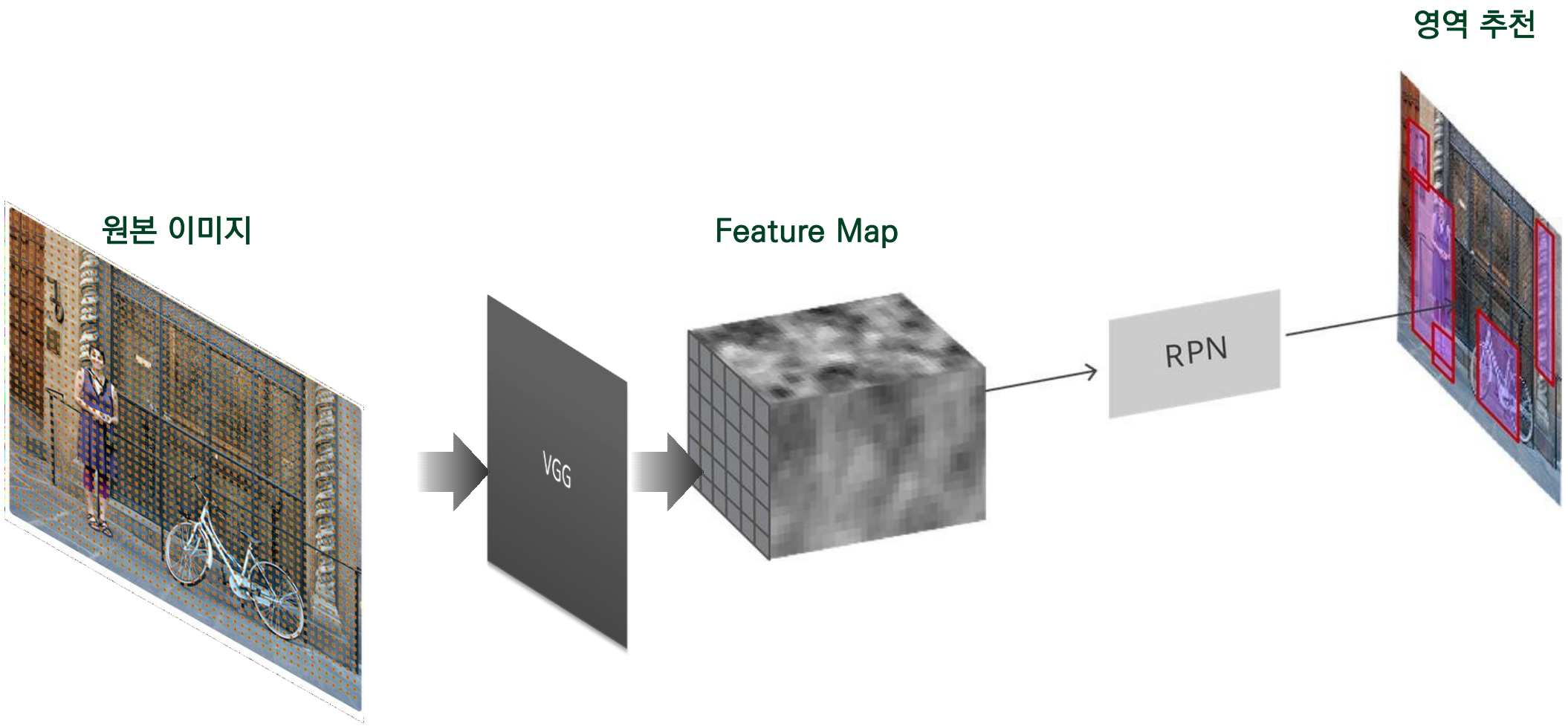
Ref. https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

Anchor Box

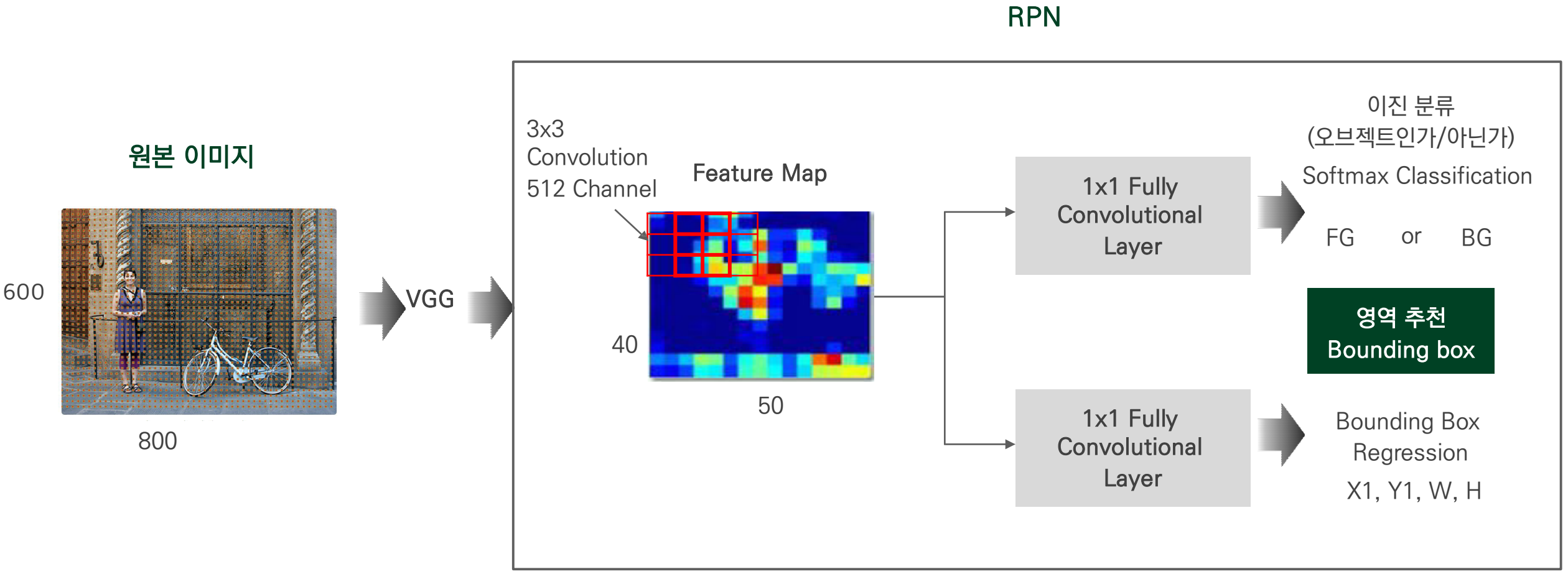
- 이미지와 Feature Map에서 Anchor Box 매핑



Region Proposal Network(RPN)

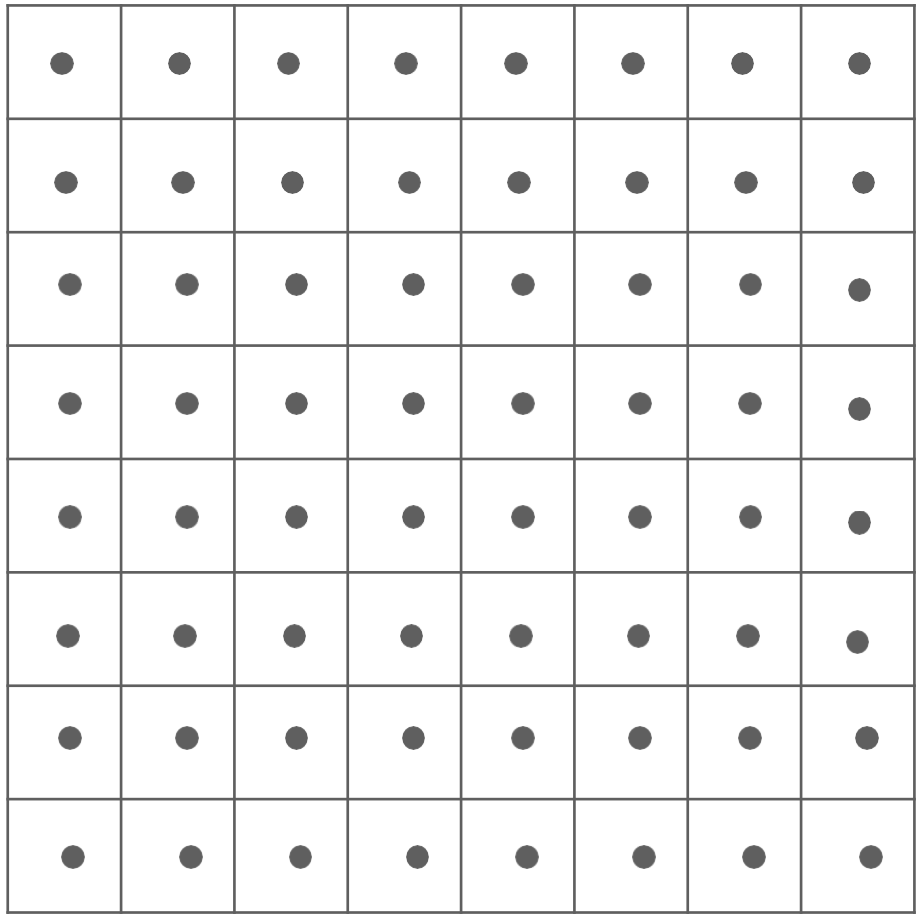


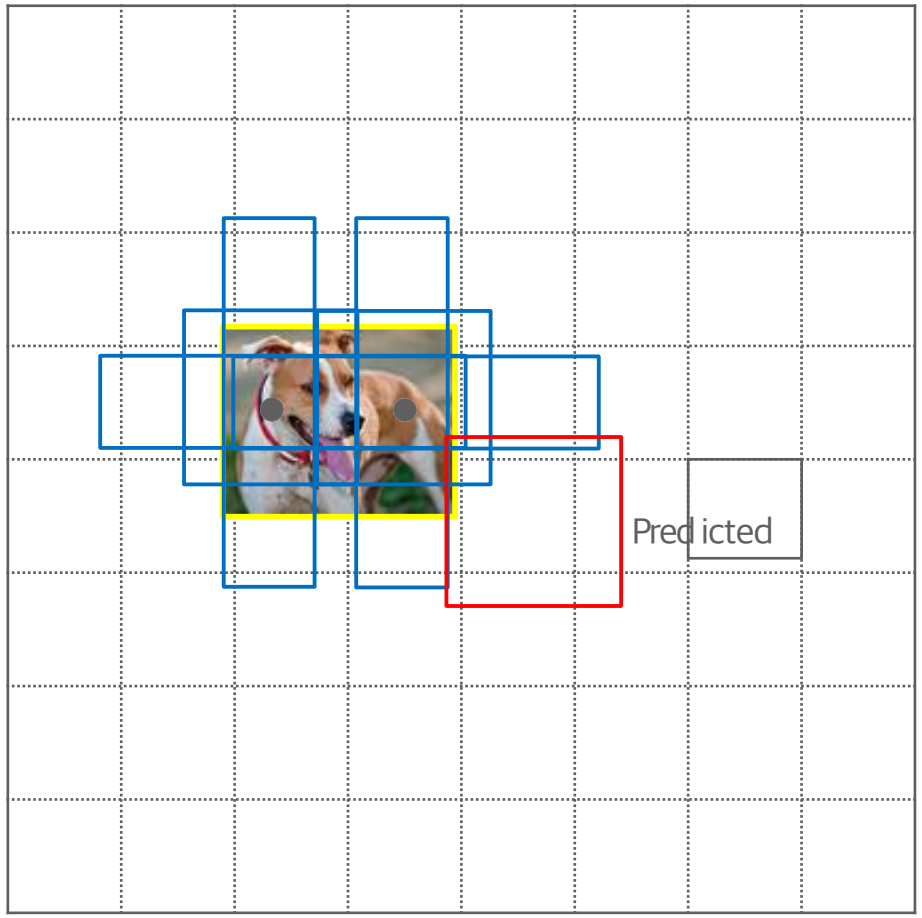
RPN 구조


RPN
원본 이미지
600
800
VGG
3x3
Convolution
512 Channel
Feature Map
40
50
1x1 Fully
Convolutional
Layer
이진 분류
(오브젝트인가/아닌가)
Softmax Classification
FG or BG
영역 추천
Bounding box
1x1 Fully
Convolutional
Layer
Bounding Box
Regression
X1, Y1, W, H

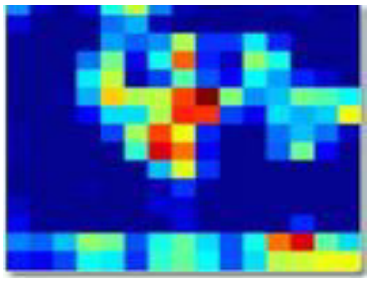
Anchor box 와 Predicted Anchor box

kubwa





RPN 출력



3 x 3 Conv
512 Channel

1x1 Conv
2 x 9 Output Channel

이진 분류 (오브젝트인가/아닌가) Softmax Classification

FG or BG

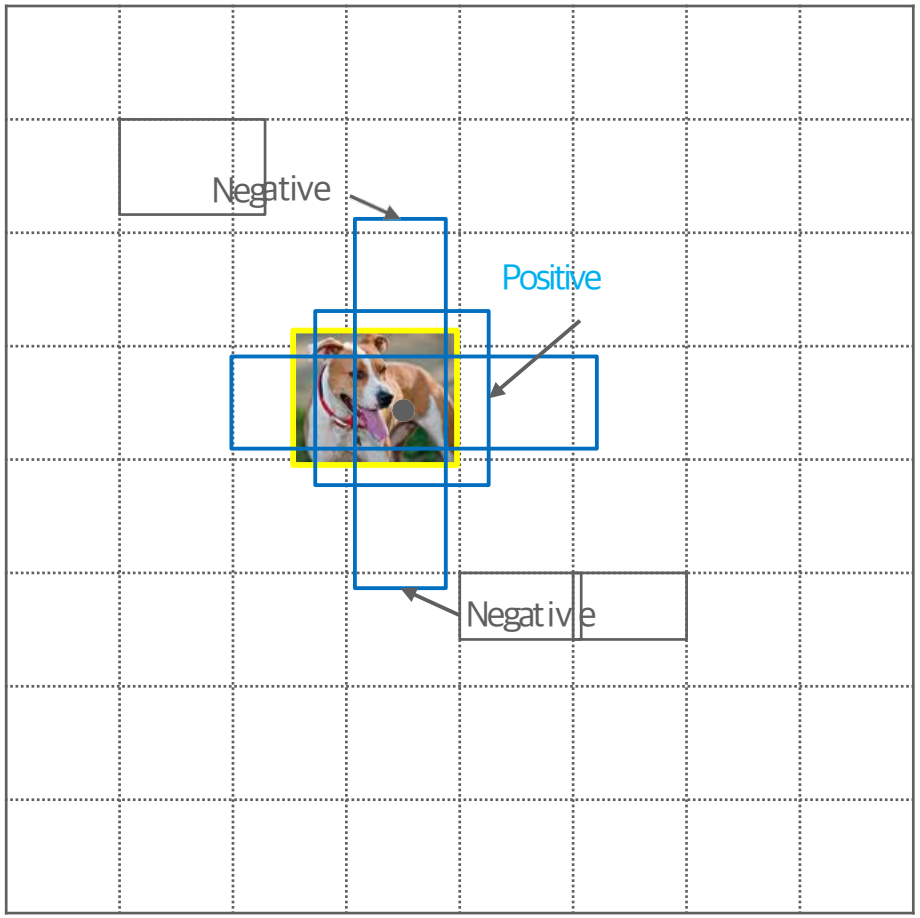
1x1 Conv
4 x 9 Output Channel

Bounding Box Regression

X1, Y1, W, H

영역 추천
Bounding box

Positive/Negative Anchor Box

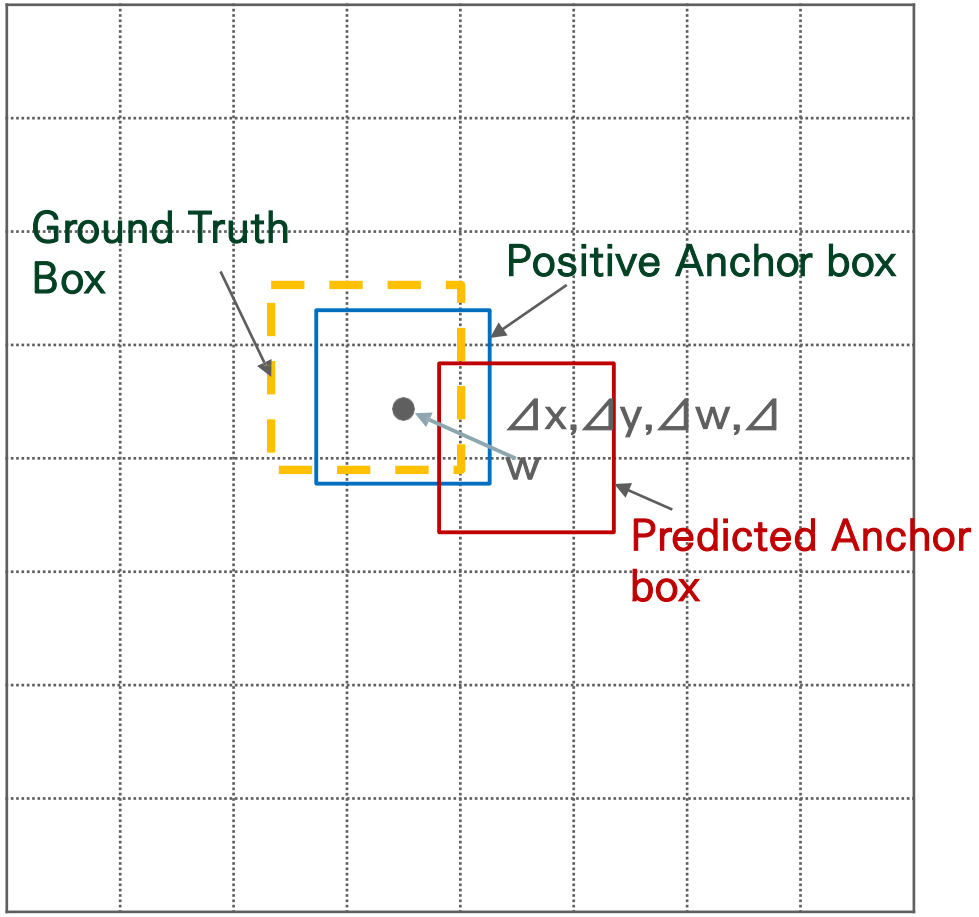


Ground Truth BB 겹치는 IOU 값에 따라 Anchor Box를 Positive Anchor Box, Negative Anchor box 로 분류

- IOU가 0.7 이상이면 Positive
- IOU가 가장 높은 Anochor는 Positive
- IOU가 0.3보다 낮으면 Negative

Bounding Box Regression with Anchor Box

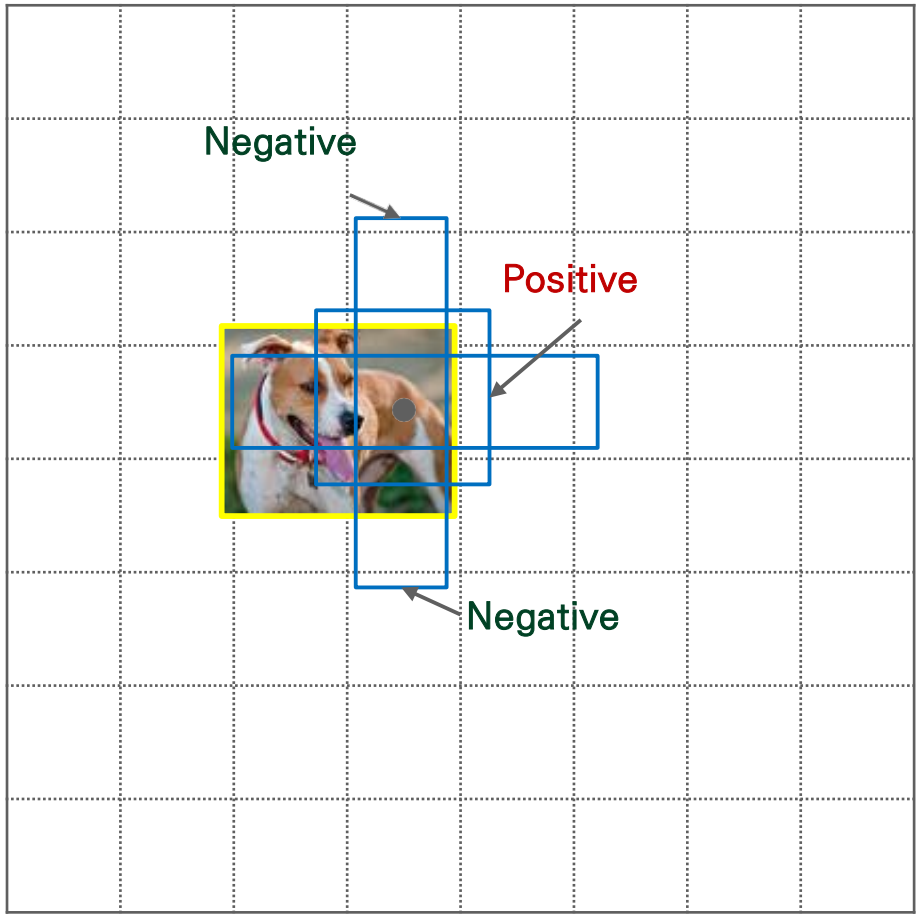
예측 Anchor box는 Positive Anchor box와의 좌표 값 차이를 최소화 할 수 있는
Bounding Box Regression 수행



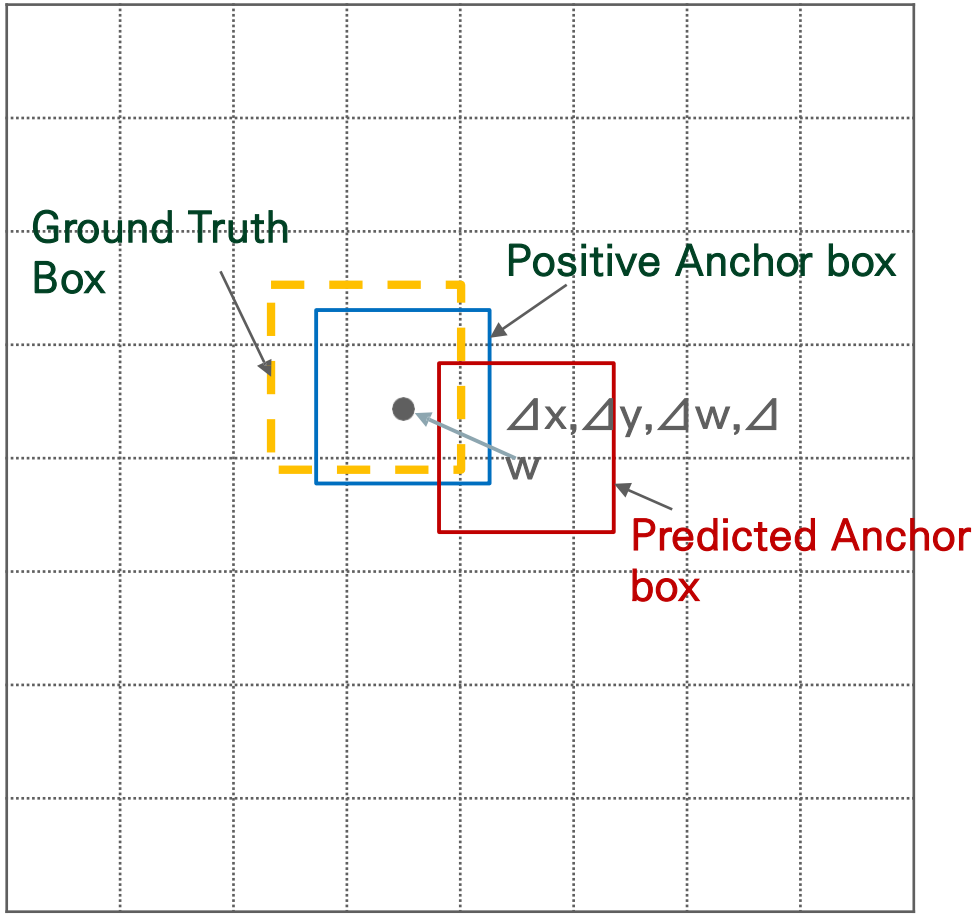
RPN Classification, Bounding Box Regression

kubwa

Classification



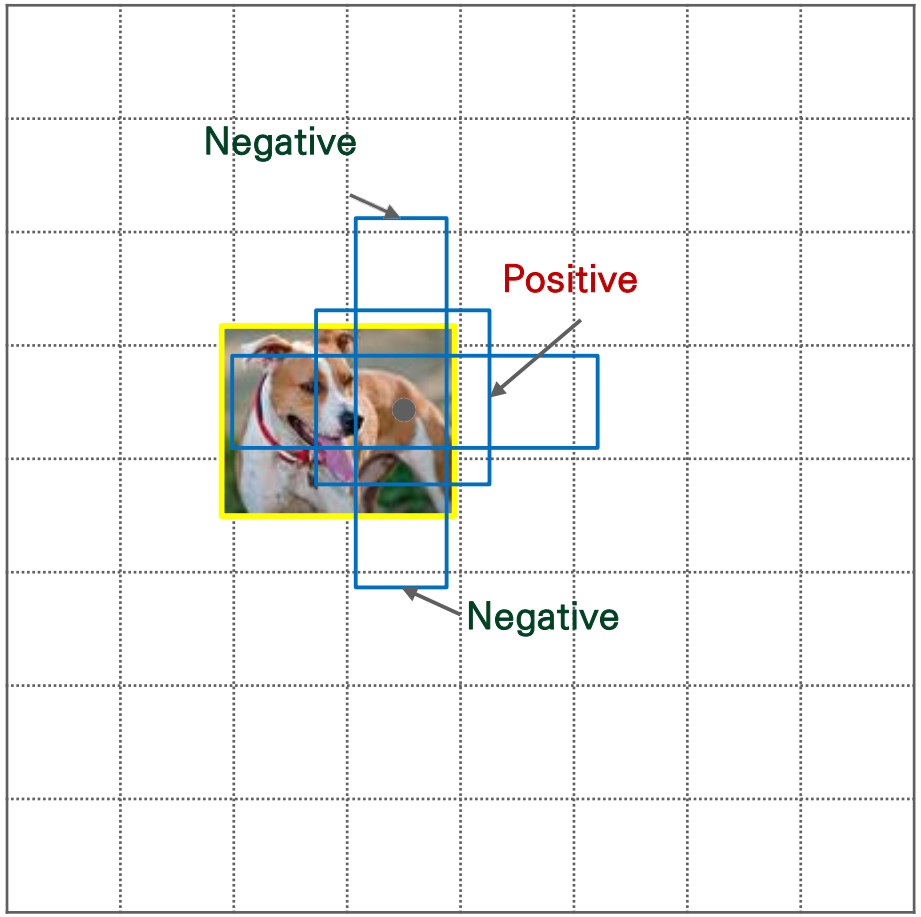
Bounding Box Regression



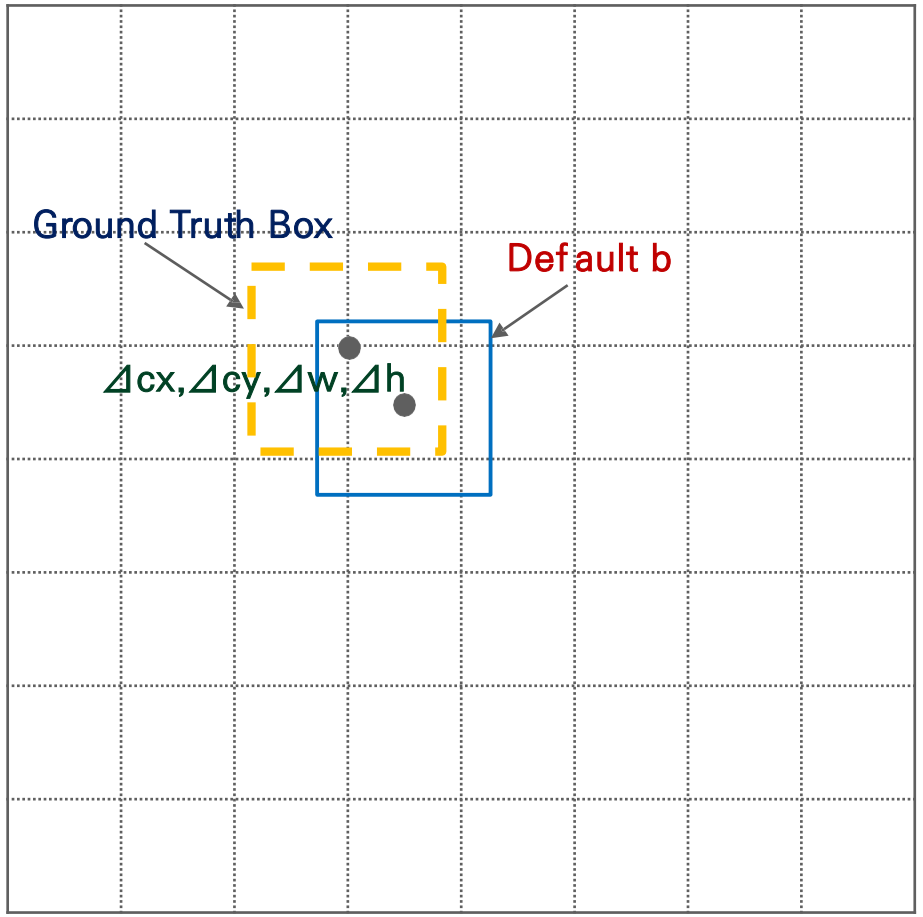
RPN Classification, Bounding Box Regression

kubwa

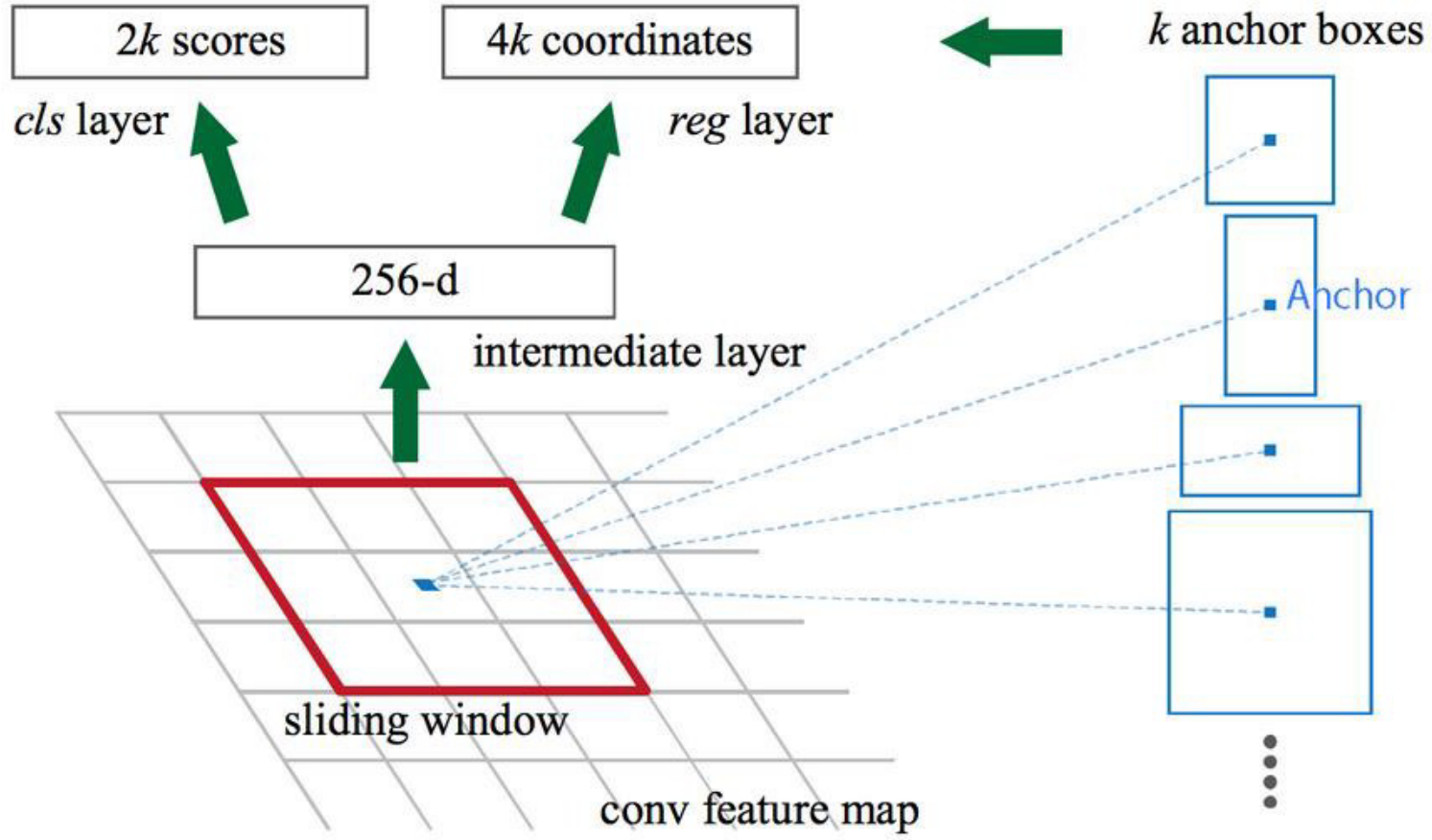
Classification



Bounding Box Regression



Anchor box에 따른 RPN Output



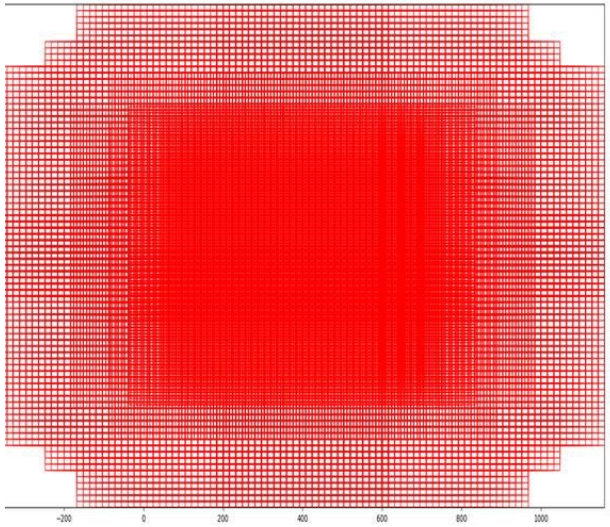
RPN의 Loss Function

$$\mathcal{L}(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}}\sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

Symbol	Explanation
p_i	Anchor i가 오브젝트일 예측 확률
p_i^*	Anchor i의 Ground truth Object 여부(Positive 1, Negative 0)
t_i	Anchor i에 대한 예측 좌표 4개(x, y, w, h)
t_i^*	Anchor i에 실제 좌표 4개(x, y, w, h)
N_{cls}	미니 배치에 따른 정규화 값(256)
N_{box}	박스 개수 정규화 값 (최대 2400)
λ	밸런싱 값: 10

RPN의 학습

Anchor box들



Mini Batch

256 anchors
(128 positive, 128 negative)

.....

.....

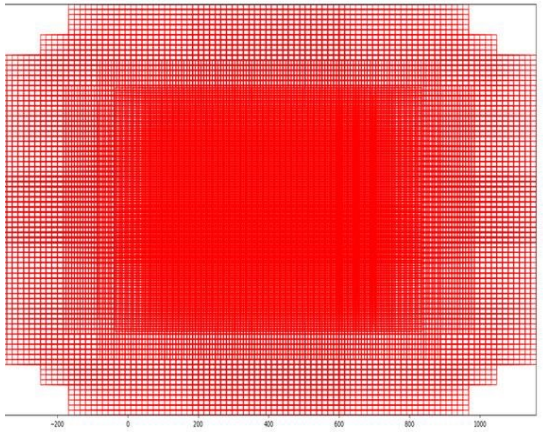
256 anchors
(128 positive, 128 negative)

Training

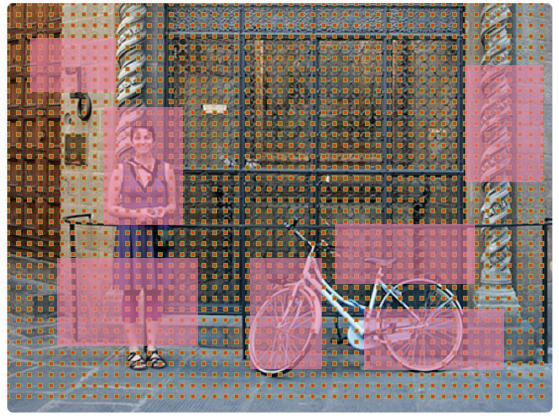
이진 분류 (Object/not Object) Softmax Classification

예측 Region Proposal box의 Objectness Score

- Objectness Score: 예측 Box가 Object 일 확률(Softmax 값) * Ground Truth bounding box와의 IOU값
- Objectness Score가 높은 순으로 Regsion Proposal Box를 추출



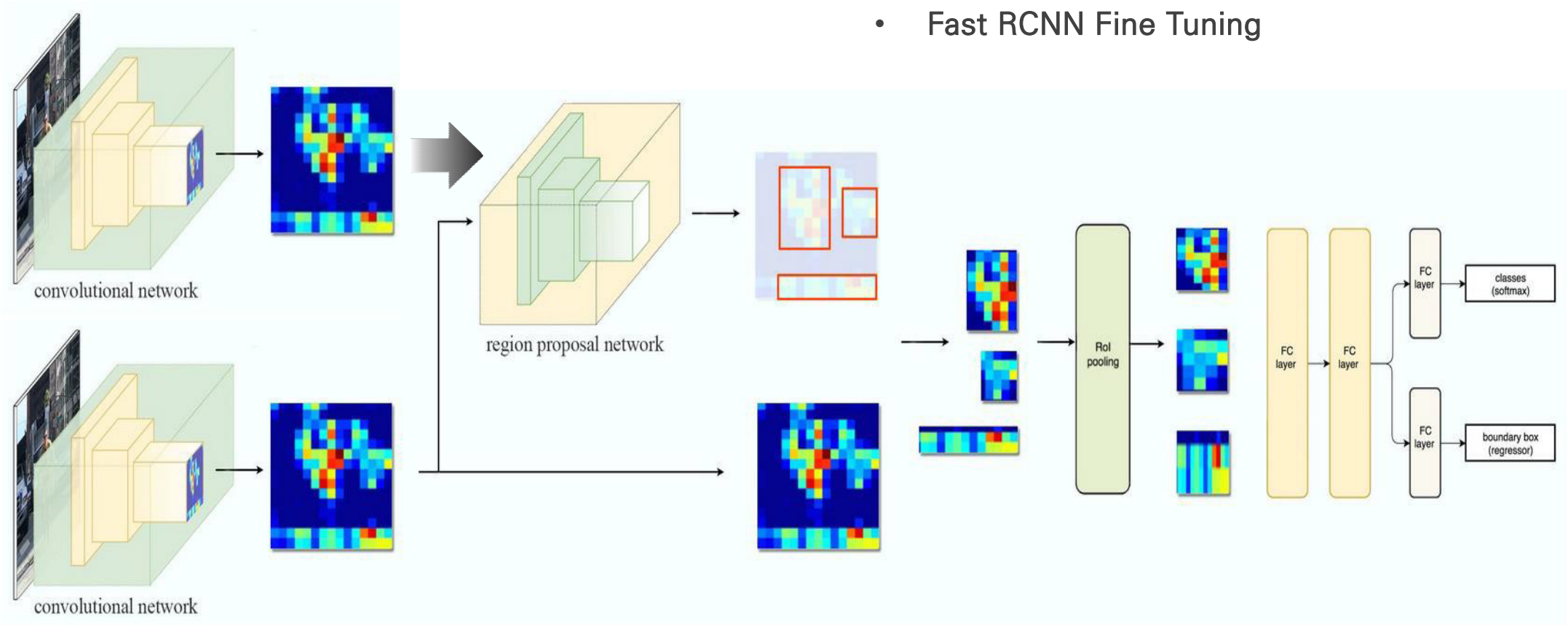
N〈2000
……
N〈300
…..
N〈50



Faster RCNN 학습 순서

kubwa

- RPN을 먼저 학습
- Fast RCNN Classification/Regression 학습
- RPN Fine Tuning
- Fast RCNN Fine Tuning



Faster RCNN 성능

PASCL VOC 데이터 세트에서 Detection 성능

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	**65.7**	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
RPN	300	COCO+07++12	**75.9**	**87.4**	**83.6**	**76.8**	**62.9**	**59.6**	**81.9**	**82.0**	**91.3**	**54.9**	**82.6**	**59.0**	**89.0**	**85.5**	**84.7**	**84.1**	**52.2**	**78.9**	65.5	**85.4**	**70.2**

Coco 데이터 세트에서 Detection 성능

method	proposals	training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	**42.7**	**21.9**

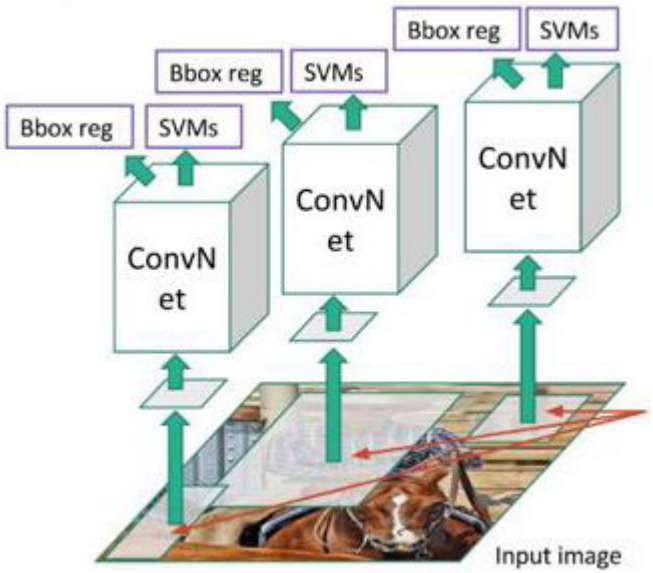
수행 시간 성능

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

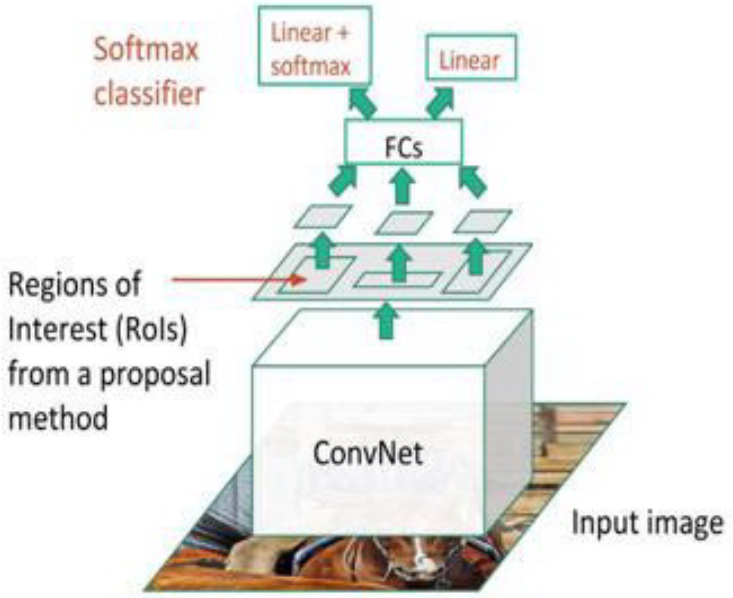
RCNN 계열 요약

kubwa

RCNN



Fast RCNN



Faster RCNN