

Faster R-CNN:

Towards Real-Time Object Detection with Region Proposal
Networks

Tae-hwi Kim

NIPS, 2015

Preliminaries

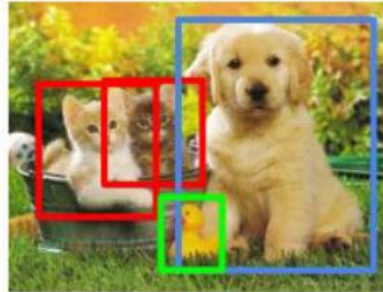
- *Object Detection*

Classification



CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

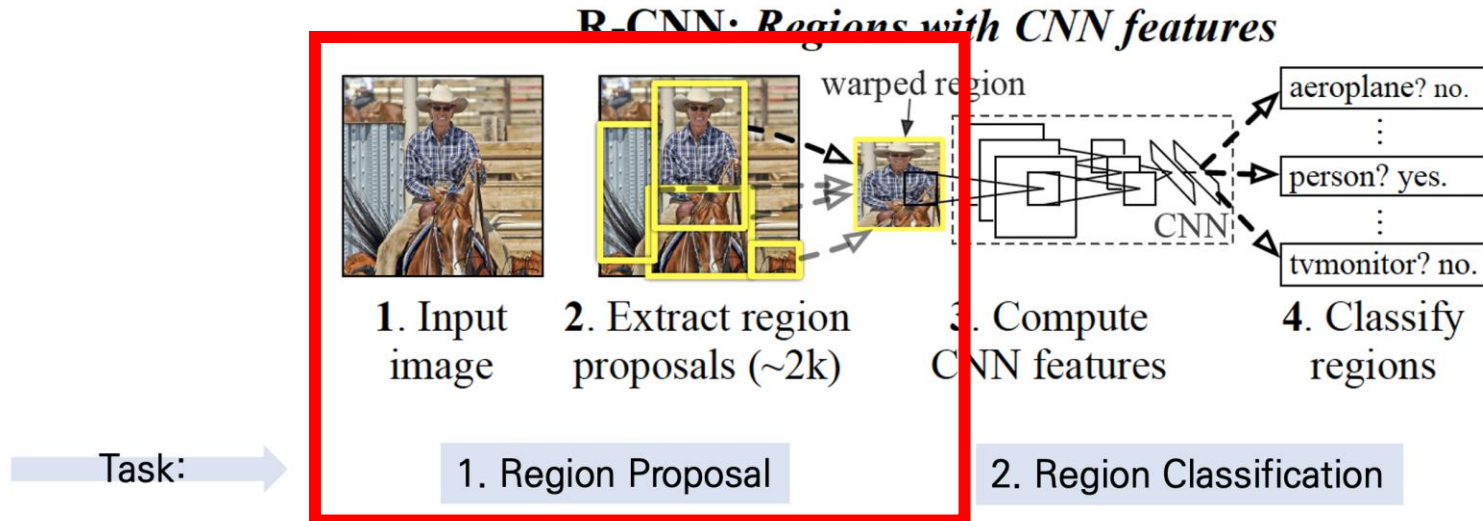
2012년 AlexNet의 시작 이후로 Object Detection(Classification + Localization)이라는 새로운 Task를 해결하기 위한 딥러닝 모델이 필요했음.

> Classification과 Localization을 동시에 수행하는 One-staged, 순차적으로 수행하는 Two-stage 방법으로 나뉘었음.

Preliminaries

- *R-CNN*

2014년 Object Detection을 위해 제시된 대표적인 Two-staged Detector,
CNN을 최초로 Object detection에 최초로 적용.



Preliminaries

- *R-CNN (Region Proposal)*

Data x , Label y : 이미지는 그대로이지만, 레이블로 Bounding box(Vertex & Edge)를 사용하여 학습.

R-CNN은 **Selective Search 알고리즘**을 이용,
> 임의의 Bounding box를 설정하고,
주변의 색(Color), 질감(Texture),
Enclosed 등을 파악해 유사한 픽셀끼리 묶는 알고리즘.

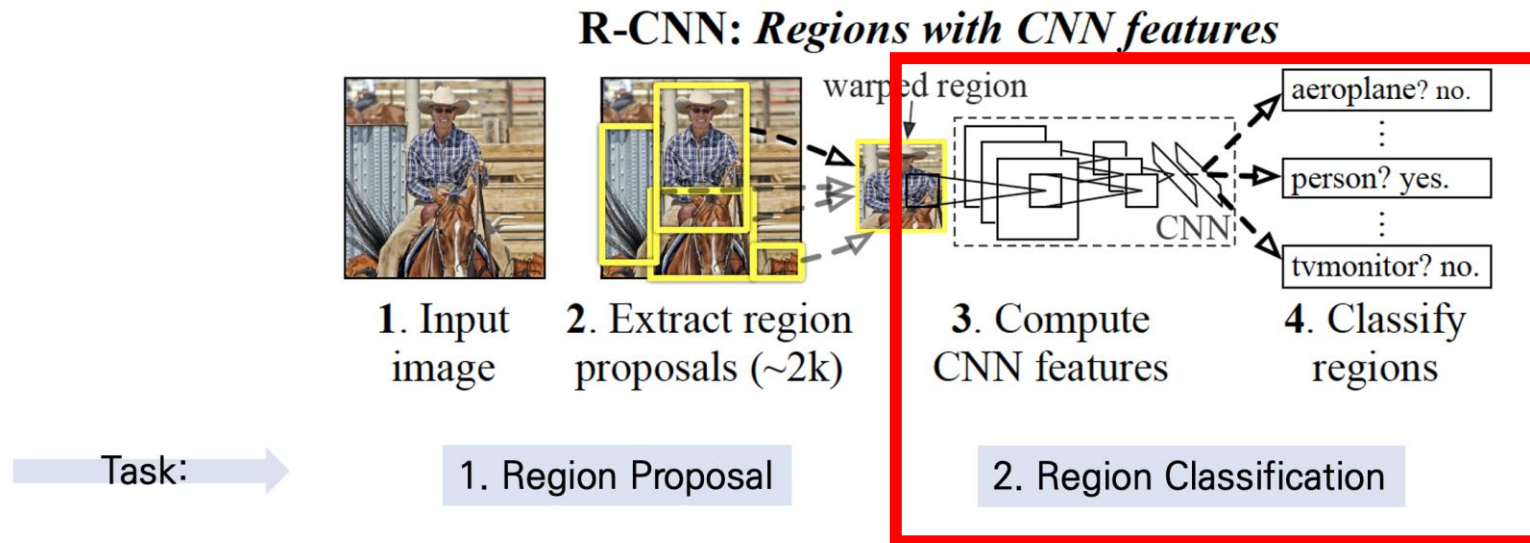
총 2,000개의 Region(feature vector)이
임의로 후보로 뽑히고, 각 후보를 일정한 크기로
조정 후, ImageNet으로 사전 학습된 CNN을 통해
Feature map으로 변환한다.



Preliminaries

- *R-CNN*

이후 Classification을 진행할 때는, SVM 모델을 이용해 분류한다.



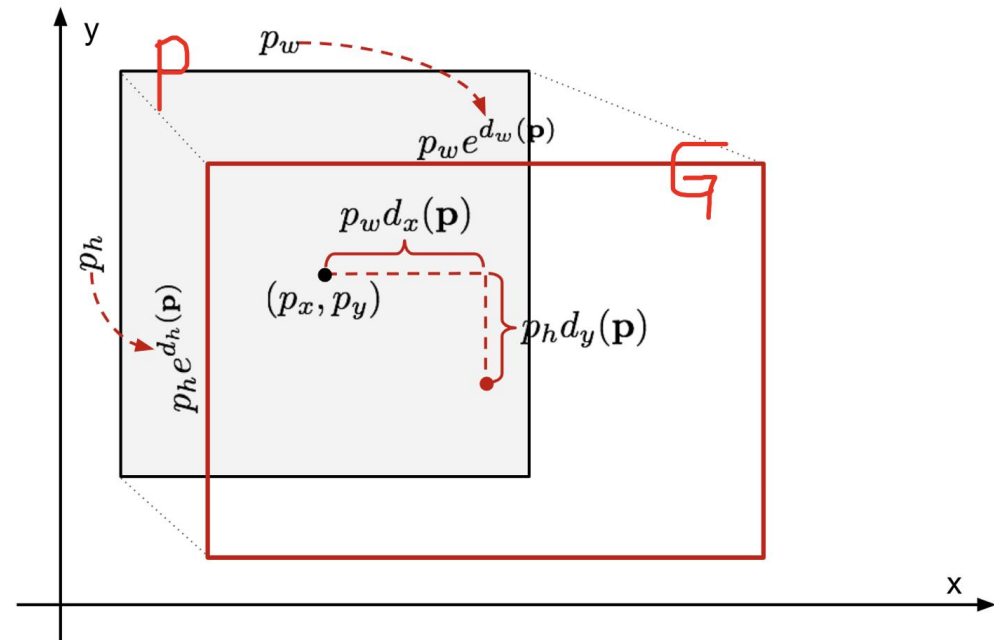
Preliminaries

- *R-CNN (Region Classification)*

VOC2007 데이터셋으로 기준으로 Softmax보다 SVM Classifier의 성능이 더 우수한 것으로 확인되어, R-CNN에서는 SVM을 Classifier로 사용.

Bounding box에 대한 Label과의 오차는
각 박스의 중심(x, y)와 박스의 크기 (width, height)를
통해 구하며, 이를 최소화하는 방향으로 학습.

(Bounding Box Regression)

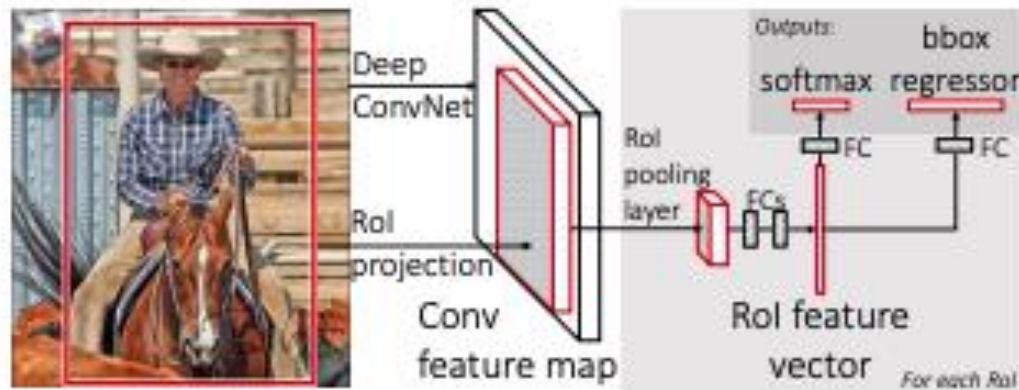


Preliminaries

- *Fast R-CNN*

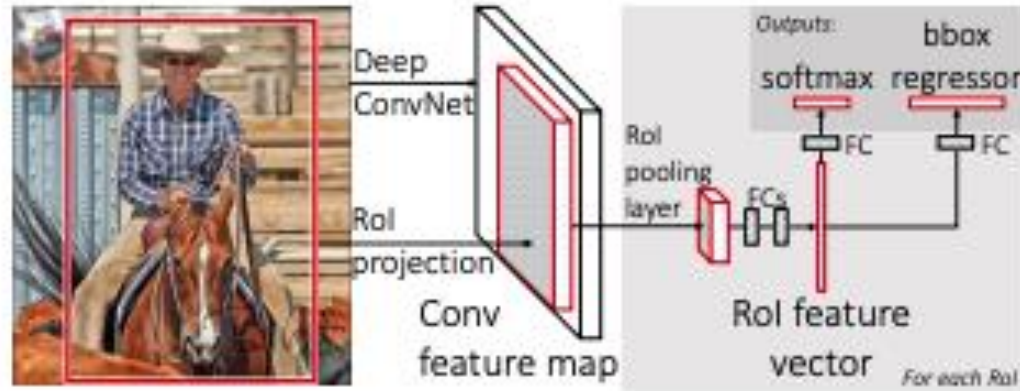
하지만 이 R-CNN은 Region(Input 이미지에 대해 임의로 생성된 Feature vector)을 **CNN에 넣을 때**, 2000개 가량의 Region을 **각각** 넣어야 하기 때문에 실시간으로 사용하기 어려움 (Bottleneck).

> 이미지의 Feature vector에서 후보(Region, RoI)을 뽑지 말고, 하나의 이미지를 Feature map으로 만든 다음에 거기에서 RoI을 생성하자 (*by selective search at R-CNN*)



Preliminaries

- *Fast R-CNN*



Feature map으로 생성된 RoI는 RoI Pooling layer를 거쳐 고정된 크기로 변형되고, 분류를 위해 두 개의 Branch로 나뉘게 된다.

> 하나는 Softmax layer에서 Box 안의 Object에 대한 Classification을, 하나는 R-CNN에서 사용한대로 Bounding Box에 대한 오차를 계산(Regression)한다.

Faster R-CNN

- *Introduction*

Fast R-CNN을 통해 속도, 정확도, 학습 속도가 모두 향상.

하지만 RoI(Region Proposal)에서 여전히 **R-CNN에서의 Selective search 알고리즘이 사용 (CPU 연산이 사용됨)** 되기 때문에 여전히 Bottleneck으로 인한 성능 하락은 해결되지 않음.



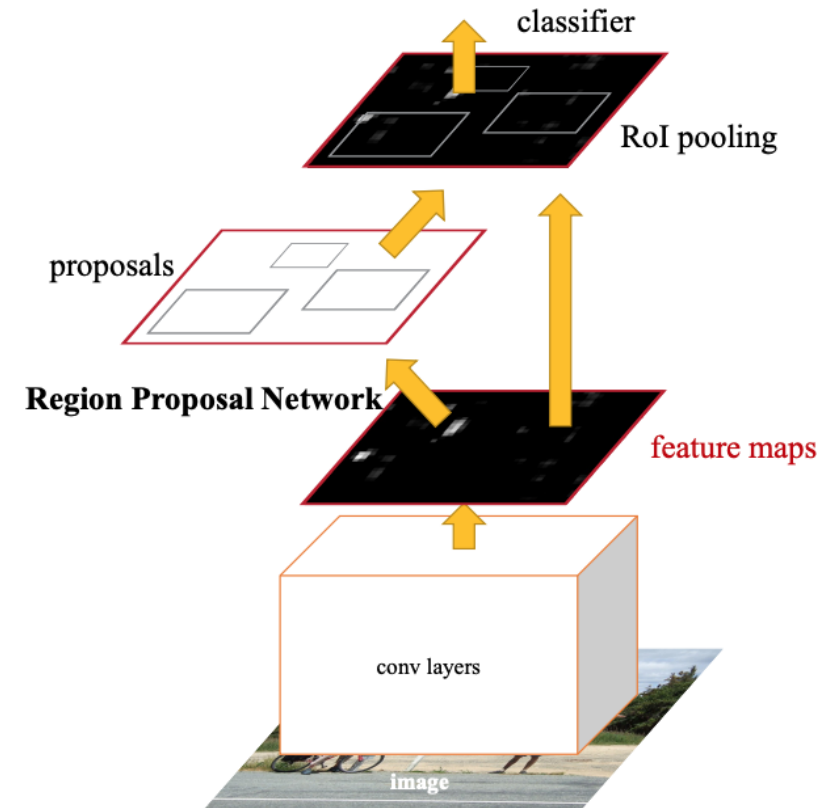
Faster R-CNN

- *Methods*

Selective Search 알고리즘을 대체하기 위한 새로운 방법 제안.

➢ Region Proposal Network (RPN)

: CPU 연산(SS)이 아닌, Conv 연산으로 RoI를 생성하겠다.












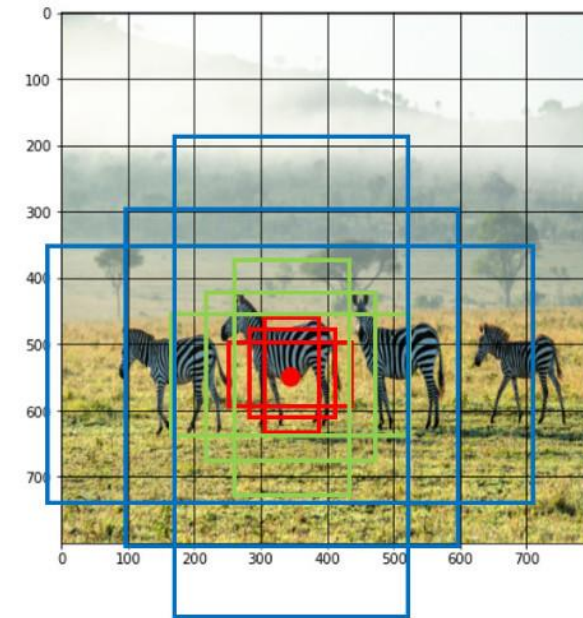
Faster R-CNN

- Methods*

- Region Proposal Network (RPN)

: Feature Map 위에서 Anchor를 두고 (파란 점, 각 픽셀), 이를 기준으로 고정된 크기의 window(k개)를 sliding하여 Anchor Box를 생성한다. (객체의 다양한 크기를 고려)

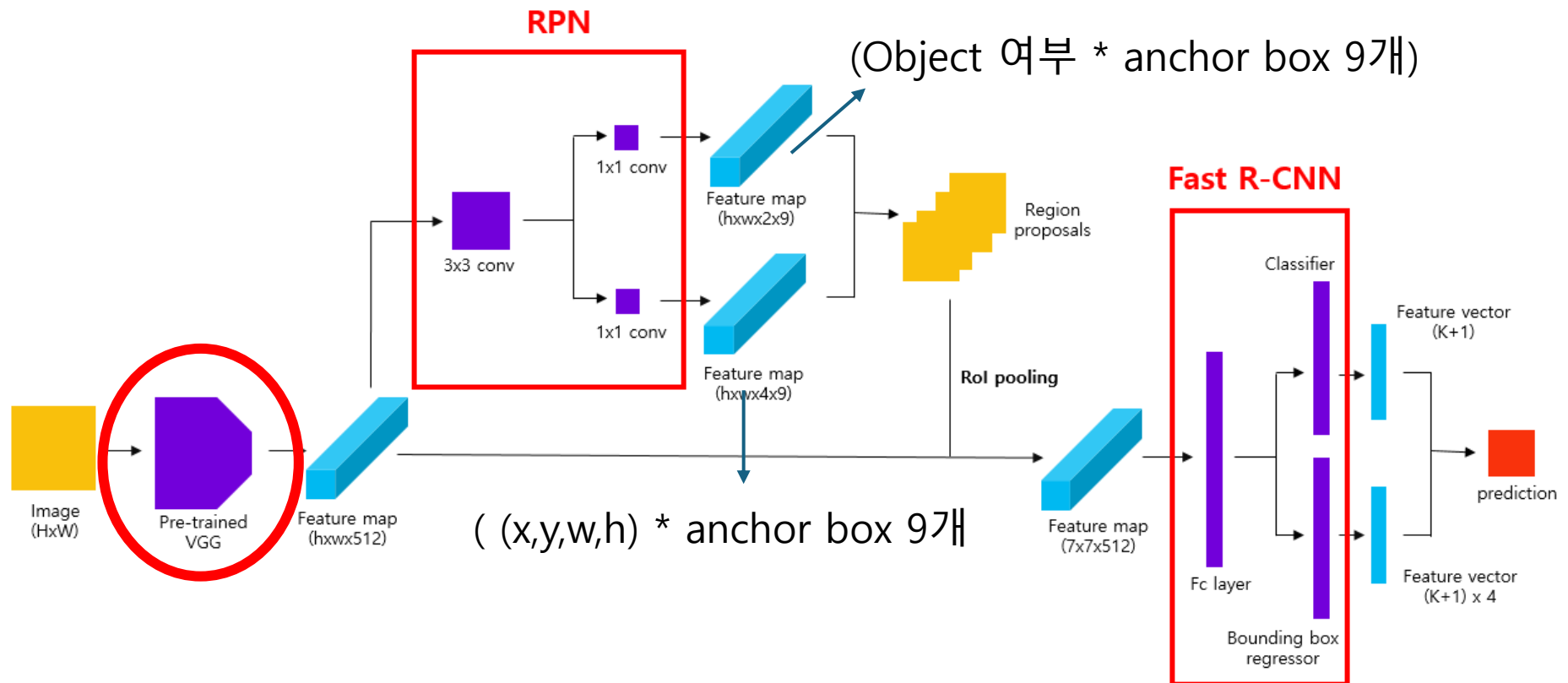
	128	256	512
1:1			
1:2			
2:1			



Faster R-CNN

• Methods

기본적으로 Fast R-CNN의 구조에 Selective Search 부분이 RPN으로 대체된 것으로 볼 수 있음.



Faster R-CNN

- Experimental Results*

Table 3: Detection results on **PASCAL VOC 2007 test set**. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07+12”: union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. [†]: this number was reported in [2]; using the repository provided by this paper, this result is higher (68.1).

method	# proposals	data	mAP (%)
SS	2000	07	66.9 [†]
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	73.2
RPN+VGG, shared	300	COCO+07+12	78.8

Table 4: Detection results on **PASCAL VOC 2012 test set**. The detector is Fast R-CNN and VGG-16. Training data: “07”: VOC 2007 trainval, “07++12”: union set of VOC 2007 trainval+test and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. [†]: <http://host.robots.ox.ac.uk:8080/anonymous/HZJTQA.html>. [‡]: <http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html>. [§]: <http://host.robots.ox.ac.uk:8080/anonymous/XEDH10.html>.

method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared [†]	300	12	67.0
RPN+VGG, shared [‡]	300	07++12	70.4
RPN+VGG, shared [§]	300	COCO+07++12	75.9

Faster R-CNN

- Experimental Results*

11

Table 11: Object detection results (%) on the **MS COCO** dataset. The model is VGG-16.

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

Faster R-CNN

- *Pros*

1. Region Proposal을 Convolutional Layer(**GPU 연산 가능**)를 통해 수행하여 실시간 객체 탐지를 가능하게 하는 새로운 RoI 생성 방법 제안.
2. 기존 Fast R-CNN에 비해 다양한 데이터셋에 대해 우수한 성능.

- *Cons*

1. Pre-trained VGG과 RPN, Detector(R-CNN)로 나뉘는 구조에서, 각 요소를 제대로 학습하기 위한 학습 절차가 다소 복잡하고, 구현이 어려움 (**Multi-stage training**)