Faster R - CNN

KUBIG - IMAGE STUDY

14기 - 산업경영공학부 - 남이량

01 논문 소개

02 논문 분석

03 요약 및 정리

01 논문 소개

분석 논문

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

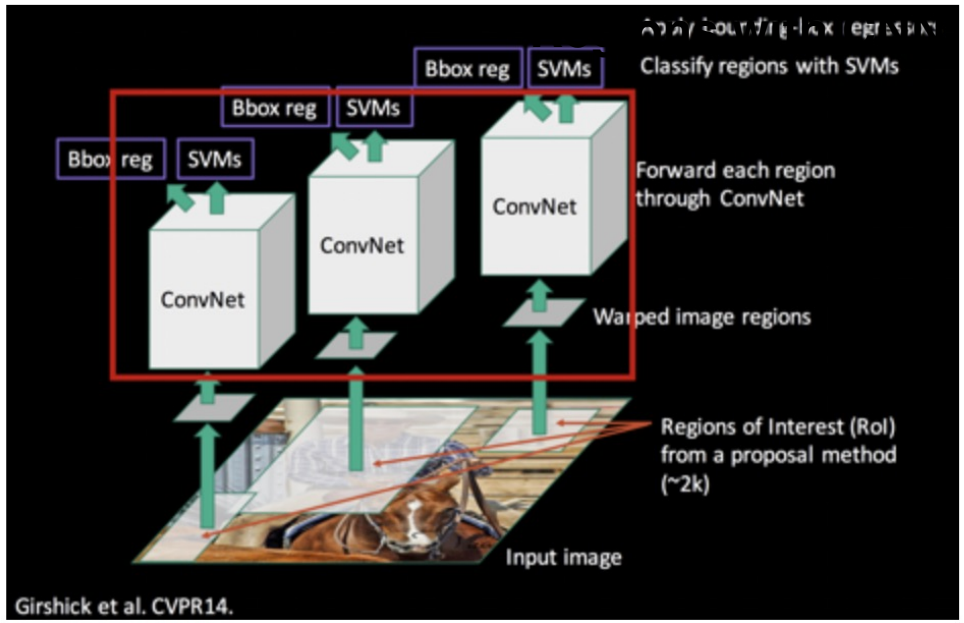
기존의 Fast R CNN에 있어 selective search에 의해 발생하는 연산 속도의 문제를 해결하기 위해
RPN (Region Proposal Network)을 도입한 방법을 소개하는 논문이다.

02 논문 분석

1. What is R - CNN : Regions with CNN

Object Detection에 있어서 처음으로 딥러닝을 적용한 방법

Object Detection = Multi Object에 대해 Classification & Localization task을 수행한다.



1. 물체의 영역을 찾는 Region Proposal

2. 1.결과에서 고정된 크기의 Feature Vector를
   warping/crop 하여 Pre- Trained 된 CNN의 input으로 사용한다.

3. CNN의 feature map을 활용하여 SVM에 대한 분류

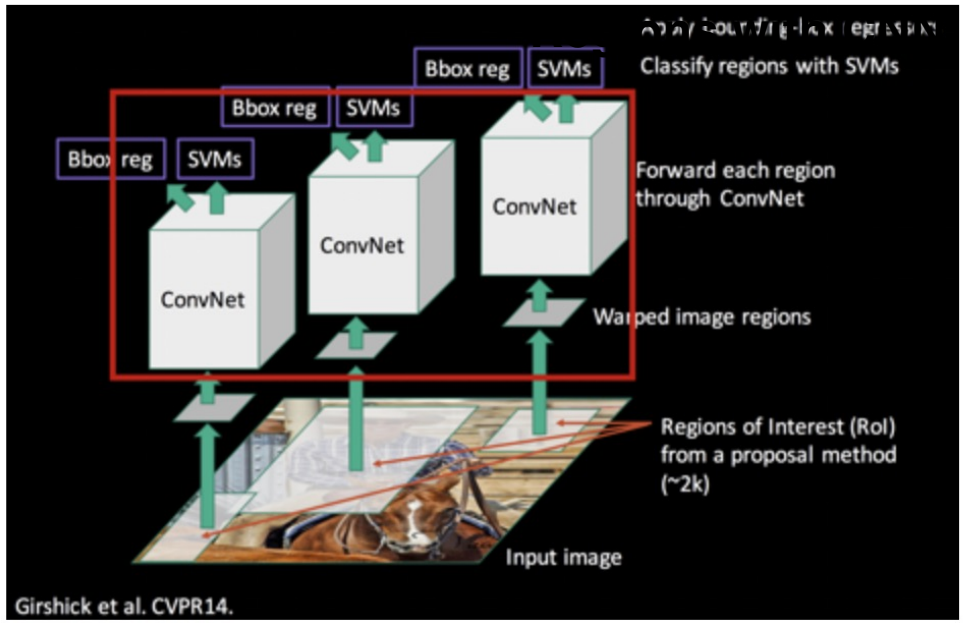
4. Regressor를 통한 Bounding box regression을 진행한다.

02 논문 분석

1. What is R - CNN : Regions with CNN

Object Detection에 있어서 처음으로 딥러닝을 적용한 방법

Object Detection = Multi Object에 대해 Classification & Localization task을 수행한다.



1. 물체의 영역을 찾는 Region Proposal

2. 1.결과에서 고정된 크기의 Feature Vector를
   warping/crop 하여 Pre- Trained 된 CNN의 input으로 사용한다.

3. CNN의 feature map을 활용하여 SVM에 대한 분류

4. Regressor를 통한 Bounding box regression을 진행한다.

이 때 Region Proposal을 찾는 selective search가 CPU를 사용!!
- 너무 오래 걸린다!!

Nam IRyang

2. R - CNN -> Fast R - CNN
What is Fast R - CNN

R - CNN

1. ROI 마다 CNN 연산을 해서 속도가 느림

2. Multi-stage Pipelines으로써 모델을 한번에 학습을 못 시킴

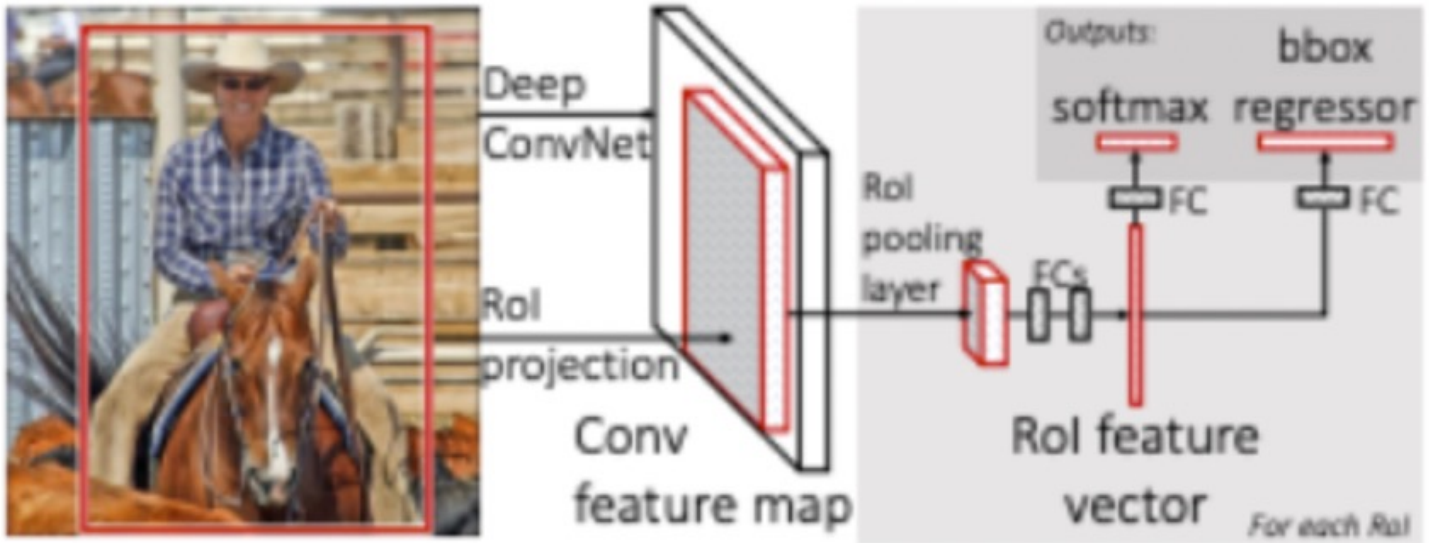
Fast R - CNN

1. ROI - Pooling

2. CNN 특성 추출부터 classification, bounding box regression까지 한번에 학습 시킴

2. R - CNN -> Fast R - CNN
What is Fast R - CNN

IMAGE -> ROI Pooling -> Fixed sized feature vector -> FC layer -> learning classification & Bounding Box



Fast R-CNN

2. R - CNN -> Fast R - CNN
What is Fast R - CNN

1) ROI Pooling

feature map에서 region proposals에 해당하는 관심 영역(Region of Interest)을
지정한 크기의 grid로 나눈 후 max pooling을 수행하는 방법입니다. - > 고정된 크기의 feature map을 출력하게 된다.

1) 먼저 원본 이미지를 CNN 모델에 통과시켜 feature map을 얻습니다.

2) 그리고 동시에 원본 이미지에 대하여 Selective search 알고리즘을 적용하여 region proposals를 얻습니다.

3) 이제 feature map에서 각 region proposals에 해당하는 영역을 추출합니다. 이 과정은 RoI Projection을 통해 가능합니다.
작아진 feature map에 대해 region proposals를 sub sampling ratio에 맞게 변경시켜준다.

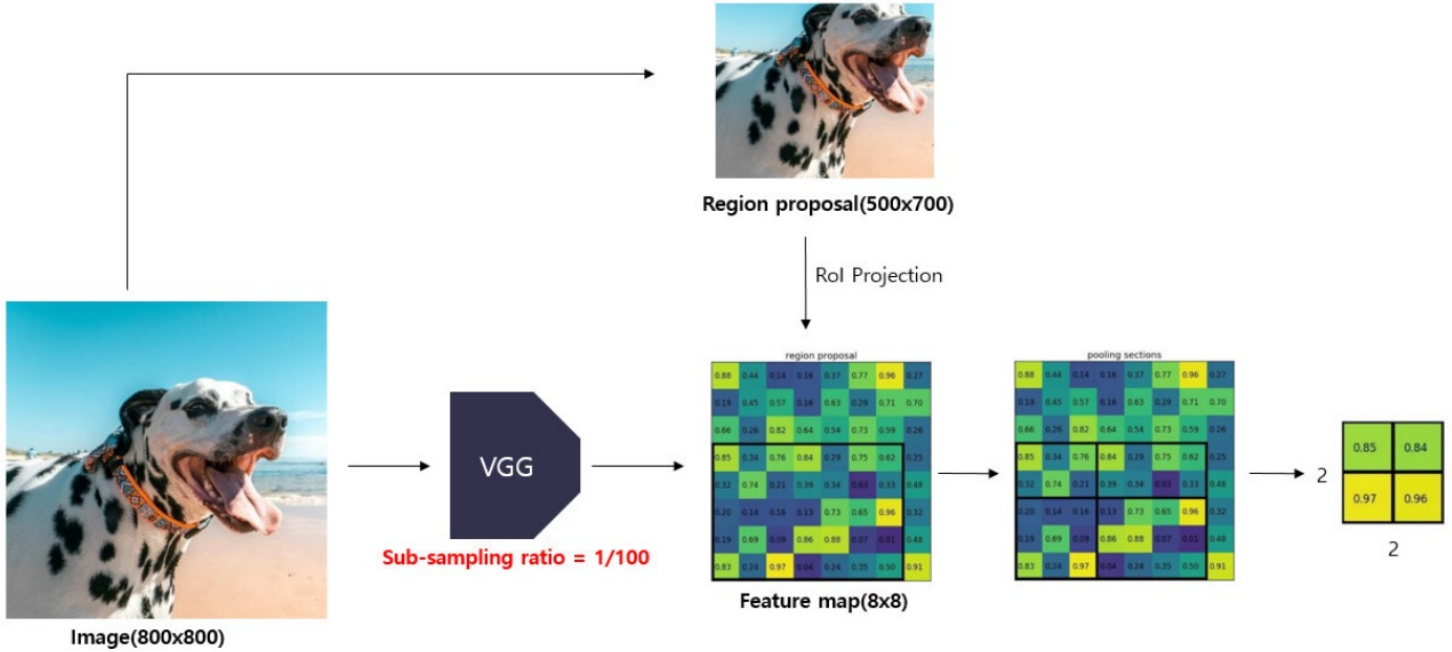
4) 추출한 RoI feature map을 지정한 sub-window의 크기에 맞게 grid로 나눠줍니다.

5) grid의 각 셀에 대하여 max pooling을 수행하여 고정된 크기의 feature map을 얻습니다.

2. R - CNN -> Fast R - CNN
What is Fast R - CNN

1) ROI Pooling

feature map에서 region proposals에 해당하는 관심 영역(Region of Interest)을
지정한 크기의 grid로 나눈 후 max pooling을 수행하는 방법입니다. - > 고정된 크기의 feature map을 출력하게 된다.



RoI pooling

2. R - CNN -> Fast R - CNN
What is Fast R - CNN

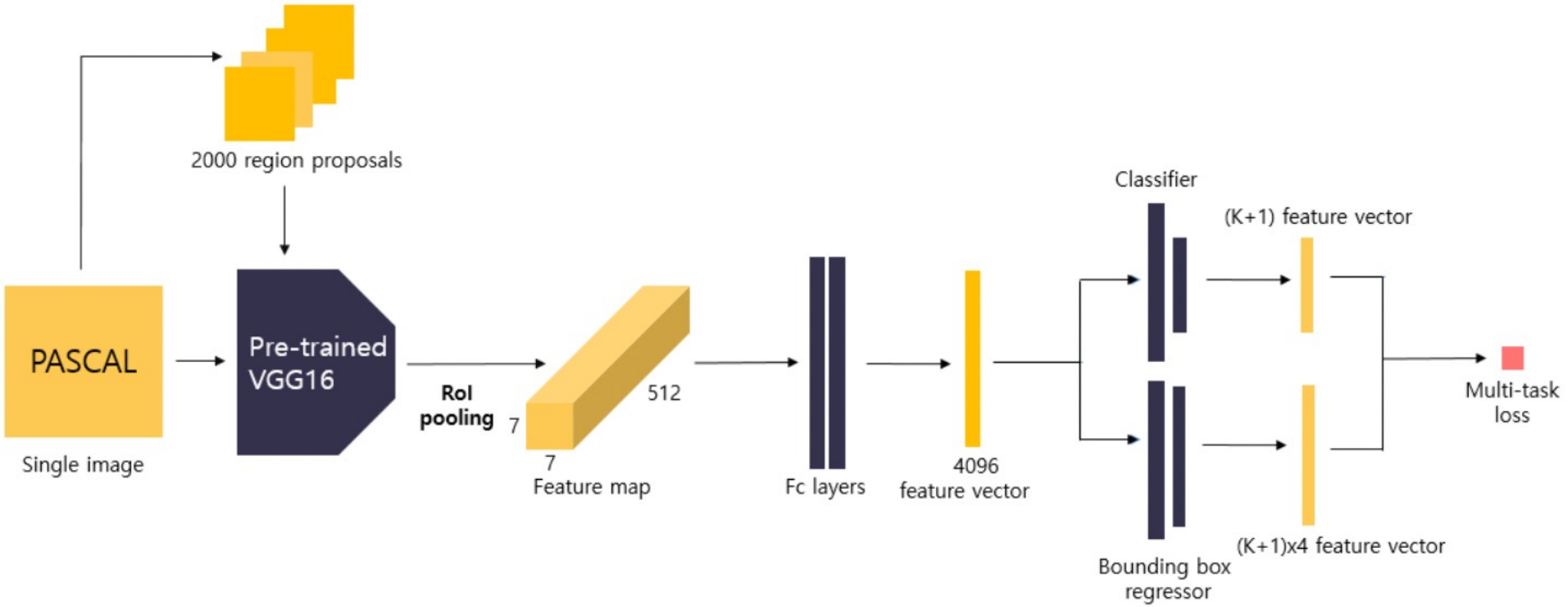
3) Hierarchical Sampling

- for feature sharing
- R – CNN에서 region proposal의 연산이 따로 따로 이루어졌던 점에 의해 발생하는 과도한 연산 시간의 문제를 해결하고자 논문에서 사용

4) Truncated SVD

- Fast – R CNN에서 ROI를 처리할 때 fc – layer에서 시간이 너무 오래 걸리는 문제를 해결하기 위해 사용
- Detection 시간이 30프로 이상 감소했다고 함

2. R - CNN -> Fast R - CNN
What is Fast R - CNN



Training Fast R-CNN

3. R - CNN -> Fast R - CNN -> Faster R - CNN
What is Faster R - CNN

Fast R – CNN에는 여전히 selective search 알고리즘으로 region proposal을 추출한다
이는 CPU에서 행해지는 연산이기 때문에 오랜 시간이 걸린다.

또한 Fast R – CNN은 End to End learning이 안된다.
* End to End : 입력에서 출력까지 하나의 네트워크로 한번에 처리됨을 의미

이러한 두 문제를 해결하기 위해 Faster R – CNN이 고안되었다.

Selective Search in CPU -> Region Proposal Network(RPN) -> Anchor Box(다양한 크기의 bounding Box)

3. R - CNN -> Fast R - CNN -> Faster R - CNN
What is Faster R - CNN

1) Anchor Box

Selective Search : 일정 간격의 grid = Bounding Box -> Encoding -> feature map : Dense Sampling

-> 이러면 Bounding Box의 크기가 고정되어 있게 되고 이러면 다양한 크기의 객체를 인식하지 못한다.

Anchor Box : 지정한 위치에 들어가는 사전에 정의한 다양한 크기의 bounding boxes

각 그리드의 중심을 기준으로 Anchor box를 생성한다.

원본 이미지에서 sub sampling ratio를 기준으로 anchor box를 생성하는 anchor를 정한다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN

What is Faster R - CNN

1) Anchor Box

Width = w, Height= h

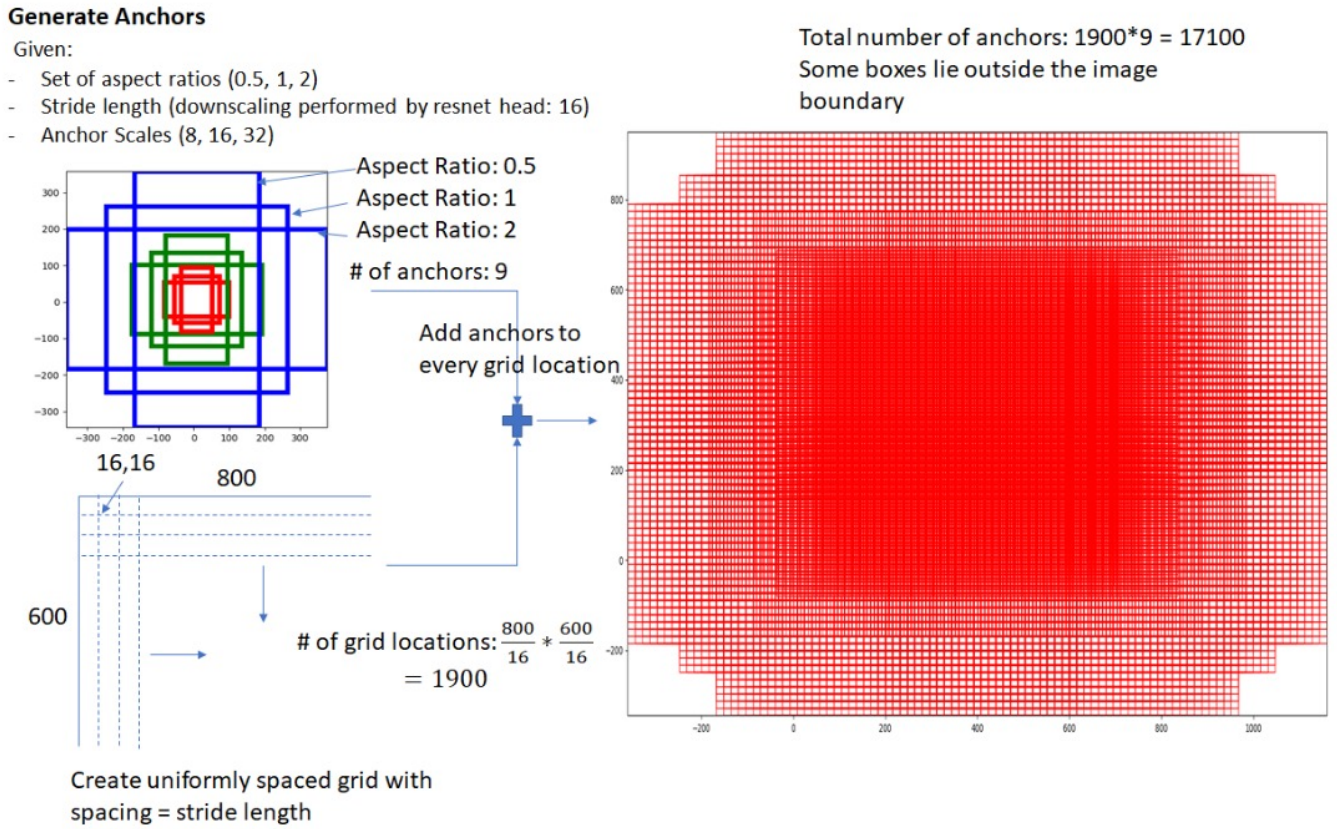
$$w \times h = s^2$$

$$w = \frac{1}{2} \times h$$

$$\frac{1}{2} \times h^2 = s^2$$

$$h = \sqrt{2s^2}$$

$$w = \frac{\sqrt{2s^2}}{2}$$



Aspect ratio = w : h

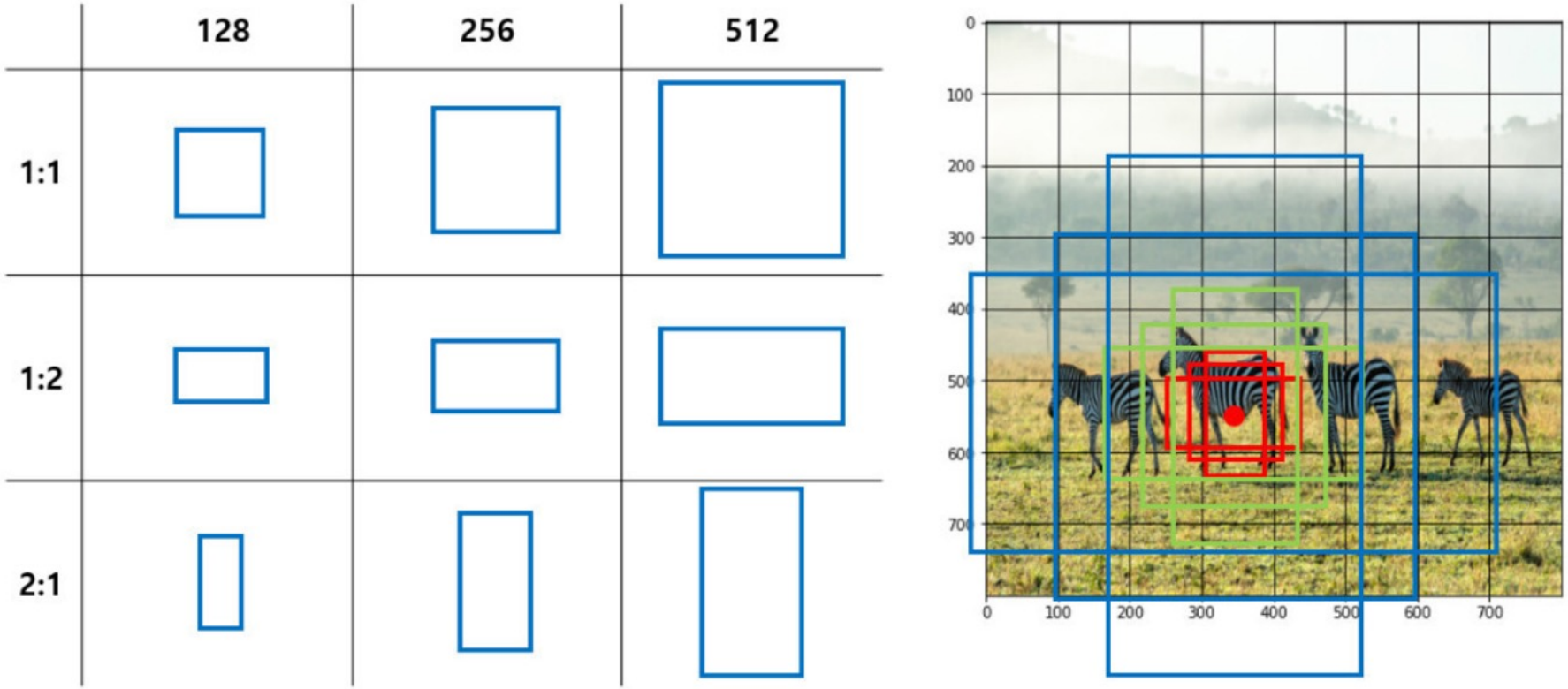
Aspect ratio에 따라 Anchor Box의 형태를 결정한다.

Nam IRyang

3. R - CNN -> Fast R - CNN -> Faster R - CNN

What is Faster R - CNN

1) Anchor Box

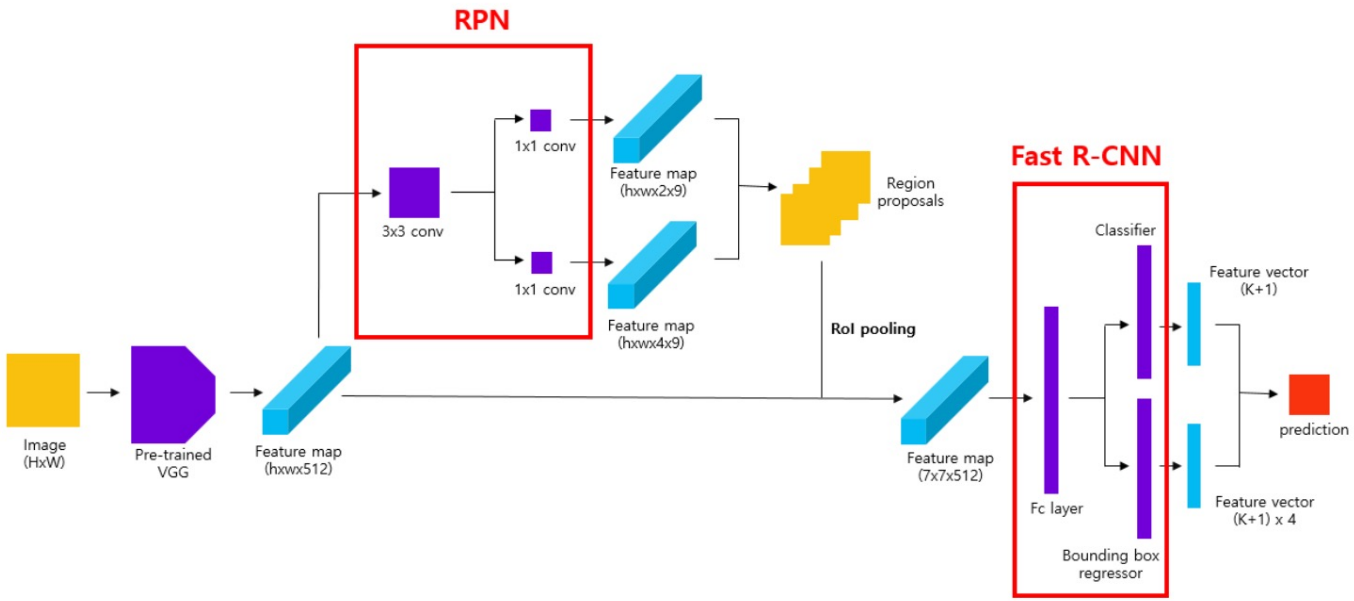


Anchor boxes

3. R - CNN -> Fast R - CNN -> Faster R - CNN
What is Faster R - CNN

2) Region Proposal Network(RPN)

원본 이미지에서 만들어진 수많은 Anchor Box에 대해서 생성된 region proposal에 대해서 class score를 매기고 Bounding box coefficient를 출력하는 기능을 수행한다.



Region proposal Network

3. R - CNN -> Fast R - CNN -> Faster R - CNN
What is Faster R - CNN

2) Region Proposal Network(RPN)

1) 원본 이미지를 pre - trained된 VGG 모델에 입력하여 feature map을 얻는다.

2) 이렇게 얻은 feature map에 대하여 3*3 conv 연산을 수행한다.
이때 padding을 추가하여 feature map의 크기가 바뀌지 않도록 한다.

3) class score를 매기기 위해서 feature map에 대하여 1x1 conv 연산을 적용합니다.
RPN에서는 후보 영역이 어떤 class에 해당하는지까지 구체적인 분류를 하지 않고
객체가 포함되어 있는지 여부만을 분류합니다.
Feature map의 channel 수는 2(object 여부) x 9(anchor box 9개)가 됩니다.

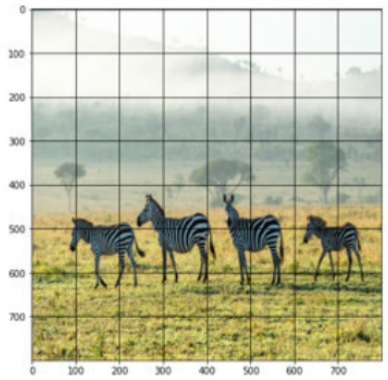
4) bounding box regressor를 얻기 위해 feature map에 대하여 1x1 conv 연산을 적용합니다.
이 때 출력하는 feature map의 channel 수가 4(bounding box regressor)x9(anchor box 9개)가 되도록 설정합니다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN

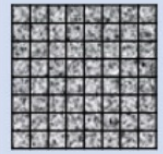
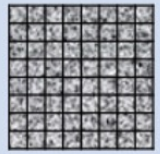
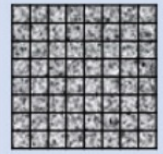
What is Faster R - CNN

2) Region Proposal Network(RPN)

RPN 결과



	object O	object X
Anchor type1		
...	...	...
Anchor type 9		

	t_x	t_y	t_w	t_h
Anchor type 1				
...	...	...	...	...
Anchor type 9				

RPN result

3. R - CNN -> Fast R - CNN -> Faster R - CNN

What is Faster R - CNN

3) Multi - Task Loss

RPN과 Fast R-CNN을 학습시키기 위해 Multi-task loss를 사용합니다.
하지만 RPN에서는 객체의 존재 여부만을 분류하는 반면, Fast R-CNN에서는 배경을 포함한 class를 분류한다는 점에서 차이가 있습니다.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

- i : mini-batch 내의 anchor의 index
- p_i : anchor i에 객체가 포함되어 있을 예측 확률
- p_i^* : anchor가 양성일 경우 1, 음성일 경우 0을 나타내는 index parameter
- t_i : 예측 bounding box의 파라미터화된 좌표(coefficient)
- t_i^* : ground truth box의 파라미터화된 좌표
- L_{cls} : Loss loss
- L_{reg} : Smooth L1 loss
- N_{cls} : mini-batch의 크기(논문에서는 256으로 지정)
- N_{reg} : anchor 위치의 수
- λ : balancing parameter(default=10)

3. R - CNN -> Fast R - CNN -> Faster R - CNN

What is Faster R - CNN

3) Multi - Task Loss

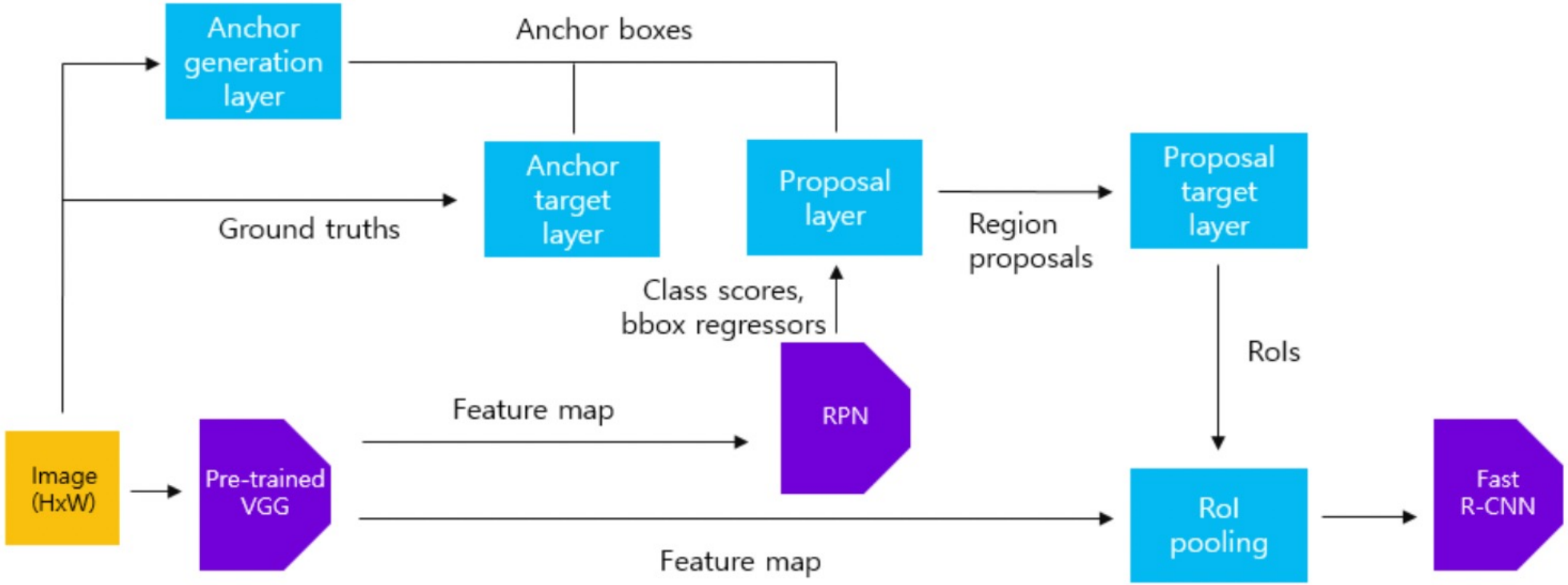
RPN과 Fast R-CNN을 학습시키기 위해 Multi-task loss를 사용합니다.
하지만 RPN에서는 객체의 존재 여부만을 분류하는 반면, Fast R-CNN에서는 배경을 포함한 class를 분류한다는 점에서 차이가 있습니다.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

이후 class score에 따라 상위 N개의 region proposals만을 추출하고, Non maximum suppression을 적용하여 최적의 region proposals만을 Fast R-CNN에 전달하게 된다.

- i : mini-batch 내의 anchor의 index
- p_i : anchor i에 객체가 포함되어 있을 예측 확률
- p_i^* : anchor가 양성일 경우 1, 음성일 경우 0을 나타내는 index parameter
- t_i : 예측 bounding box의 파라미터화된 좌표(coefficient)
- t_i^* : ground truth box의 파라미터화된 좌표
- L_{cls} : Loss loss
- L_{reg} : Smooth L1 loss
- N_{cls} : mini-batch의 크기(논문에서는 256으로 지정)
- N_{reg} : anchor 위치의 수
- λ : balancing parameter(default=10)

3. R - CNN -> Fast R - CNN -> Faster R - CNN
What is Faster R - CNN



Training Faster R-CNN

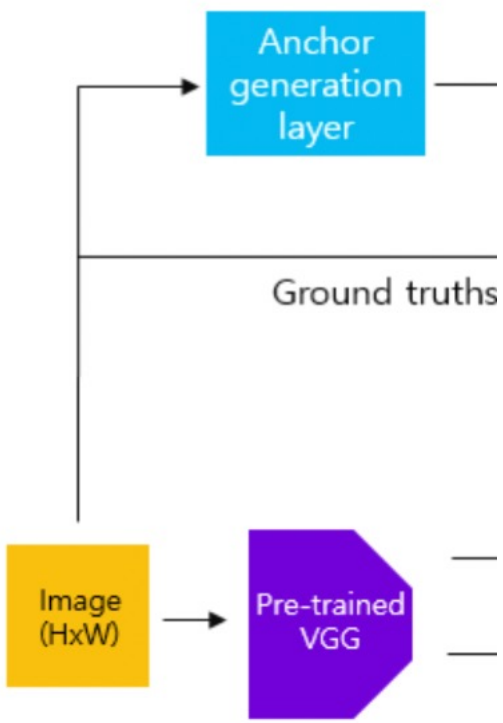
3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

1) Feature extraction by pre - trained VGG 16

Pre - trained 된 VGG16에 원본 이미지(800*800*3)을 넣으면 50*50*12 크기의 feature map을 얻는다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

2) Generate Anchors by Anchor generation layer



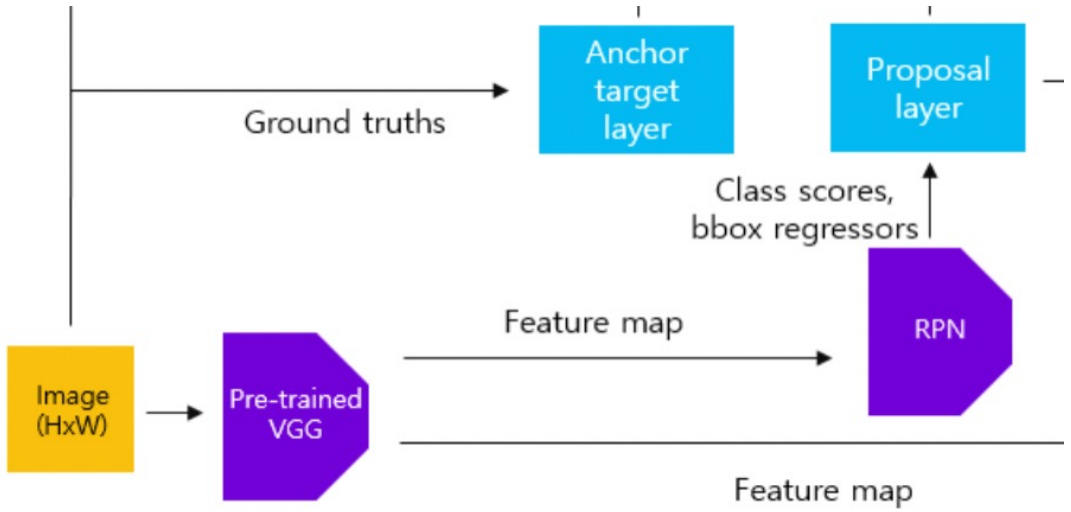
Region proposal을 추출하기 앞서 원본 이미지에 대해서 Anchor box를 생성하는 과정이다

원본 이미지의 크기에 sub-sampling ratio를 곱한 만큼의 grid cell이 생성 되며, 이를 기준으로 각 grid cell마다 9개의 anchor box를 생성합니다.

해당 논문에서는 50*50의 grid cell이 만들어지고 한 grid 마다 9개의 anchor box가 그려져 총 22500개의 anchor box가 만들어진다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

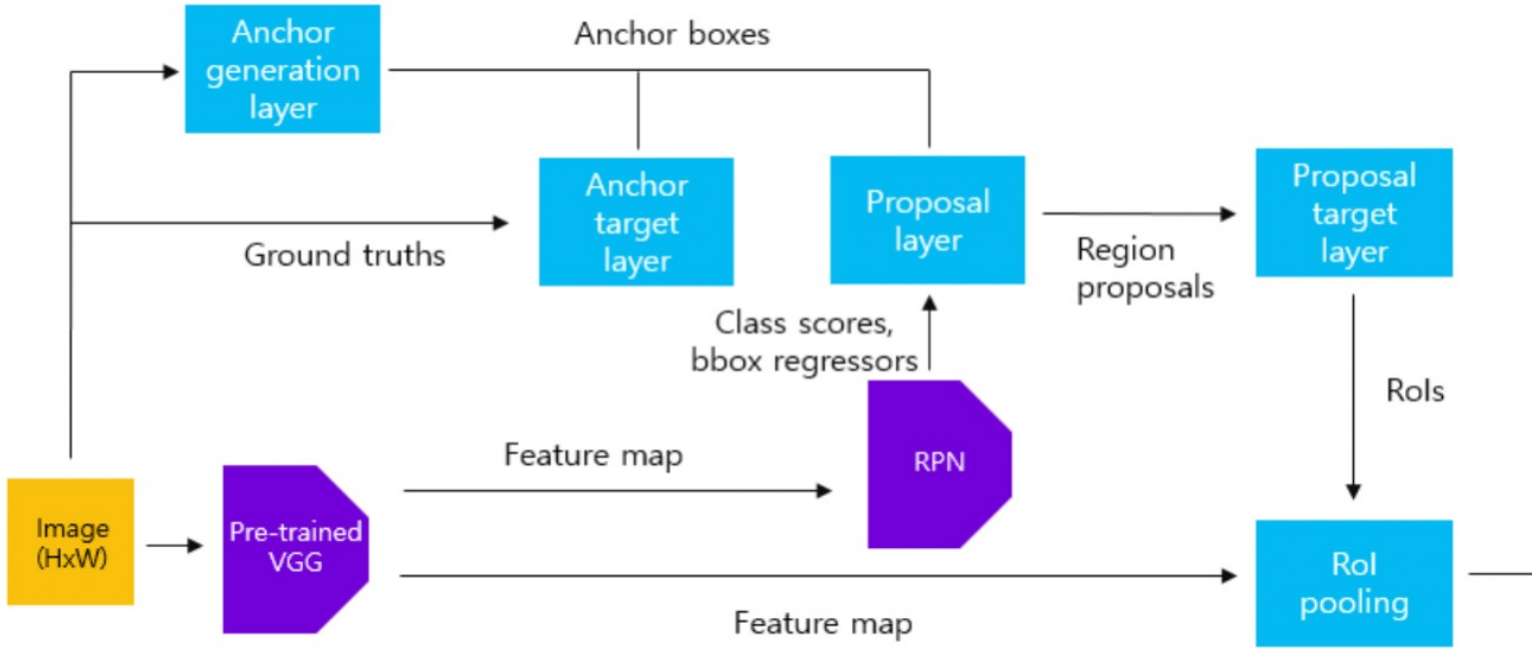
3) Class scores and Bounding Box regressor by RPN



VGG16으로부터의 feature map을 RPN에 통과시켜 anchor에 대한 class score, bounding box regressor를 반환하는 역할을 한다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

4) Region proposal 추출



2)에서 구한 anchor 과 3)에서 구한 결과에 대해서Non maximum suppression를 적용하여 부적절한 객체를 제거한다. 그 이후 class score 상위 n개를 추출한 후 이후 regression coefficients를 anchor box에 적용하여 anchor box가 객체의 위치를 더 잘 detect하도록 조정한다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

5) Select anchors for training RPN by Anchor target layer

RPN에서 사용할 anchor box를 선택하는 과정이다.

이미지를 벗어나지 않은 anchor boxes에 대해서 객체의 유무에 따라 positive / negative / foreground로 분류한다.
이 때 ground truth box와 가장 큰 IoU 값을 가지는 경우, ground truth box와의 IoU 값이 0.7 이상인 경우에
해당하는 box를 positive sample로 선정하고 0.3 이하는 negative로 분류한다. 이러한 과정을 통해 RPN에 학습시킬
데이터 셋을 구성한다.

6) Select anchors for training Fast R - CNN by Proposal target layer

앞서 4)에서 산출한 region proposals 중에서 Fast R – CNN 모델을 학습하기에 적당한 것들을 추출하는 과정이다.

먼저 region proposals와 ground truth box와의 IoU를 계산하여 0.5 이상일 경우 positive,
0.1~0.5 사이일 경우 negative sample로 label됩니다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

7) Max pooling by ROI pooling

앞서 6)에서 구한 RP들에 대해서 Fast R – CNN에서 진행한 ROI pooling을 진행한다.

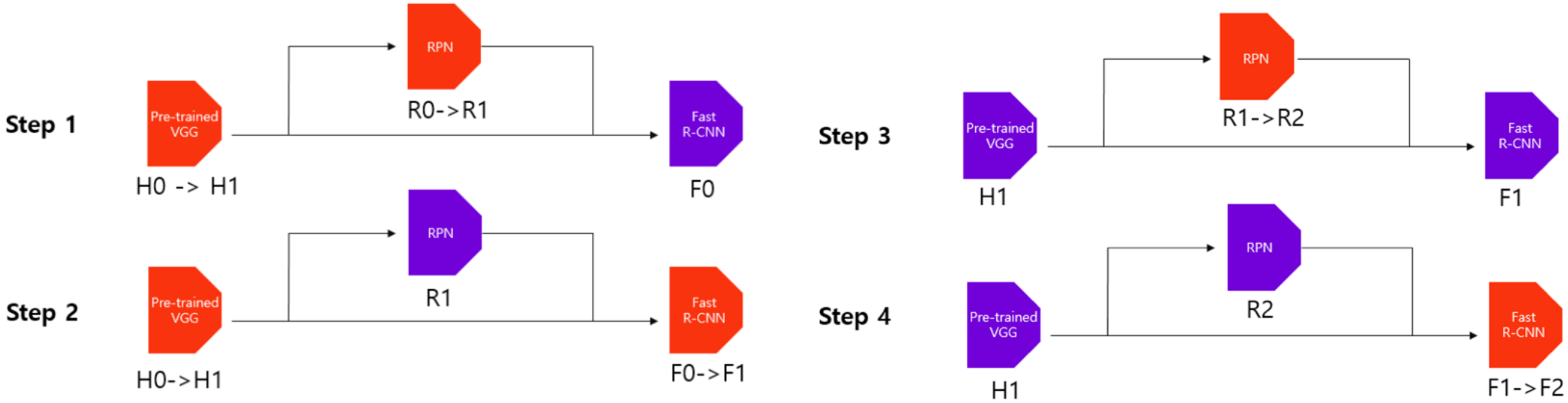
8) Train Fast R - CNN by Multi - Task Loss

마지막으로 Multi – Task Loss를 기반으로 Fast R – CNN을 학습시킨다.

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

* Alternating Training

Faster R-CNN 모델을 학습시키기 위해 RPN과 Fast R-CNN을 번갈아가며 학습시키는 Alternating Training 방법을 사용한다

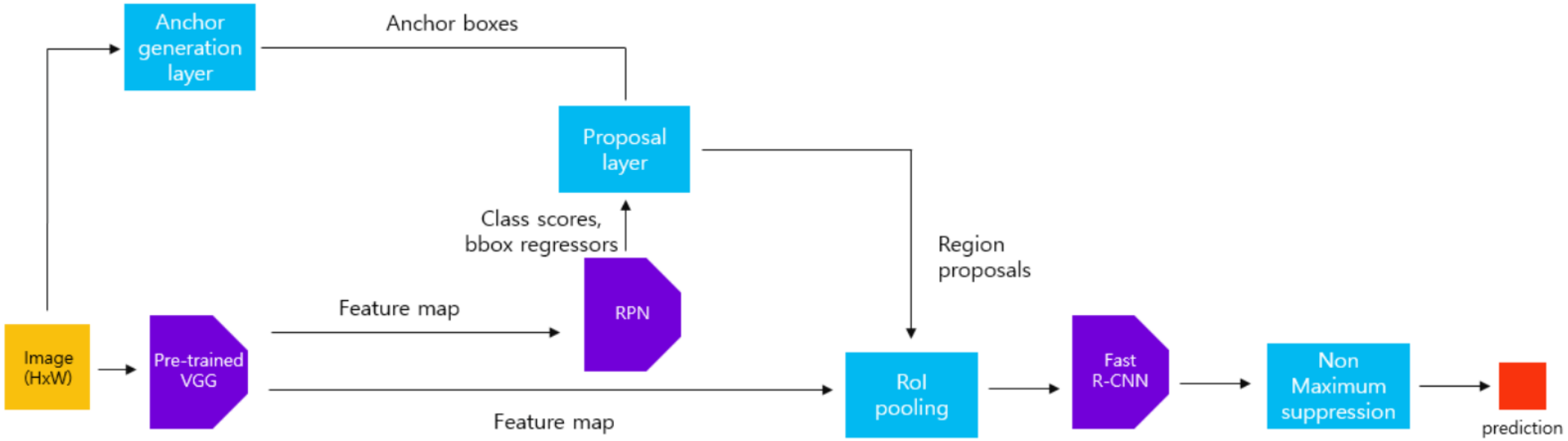


Aleternating Training

3. R - CNN -> Fast R - CNN -> Faster R - CNN
Process of Faster R - CNN

* Detection

Detection에서는 Anchor generation layer, Proposal target layer를 사용하지않고 Proposal layer에서 선택한 RP에 대해서 Fast R - CNN을 학습시킨 후에 NMS를 진행한 후 결과값을 산출한다.



Faster R-CNN detection

03 요약 정리

R - CNN -> Fast R - CNN -> Faster R - CNN

ROI pooling

RPN

Share Feature extraction's feature
-> End to End structure

시간과 정확도에 대한 향상이 유의미했다.