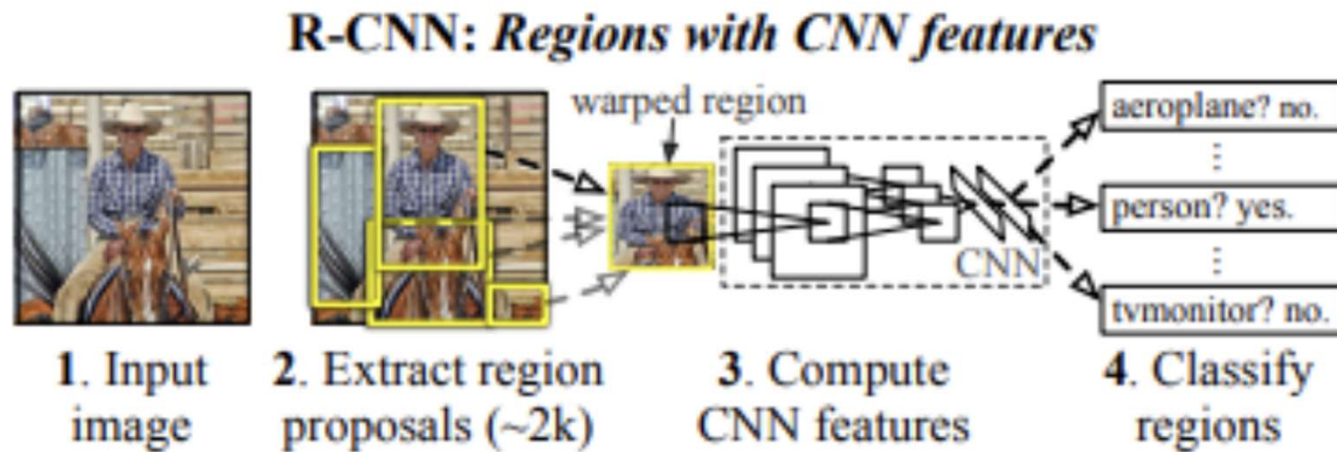

기본

R-CNN[®]

Rich feature hierarchies for accurate OD and SS

허 지 혜 발표



1. takes an input image
2. look around 2000 bottom-up region proposals
3. computes feature for each proposal using a large CNN
4. classifies each region using SVMs

mAP 53.7% on PASCAL VOC 2010

In other word, Our object detection system consists of three models.

1) Generates category-independent region proposals. (독립적인 범주의 지역 제안 생성)

These proposals define the set of candidate detections available to our detector

2) Large convolutional neural network that extracts a fixed-length feature vector from out region.

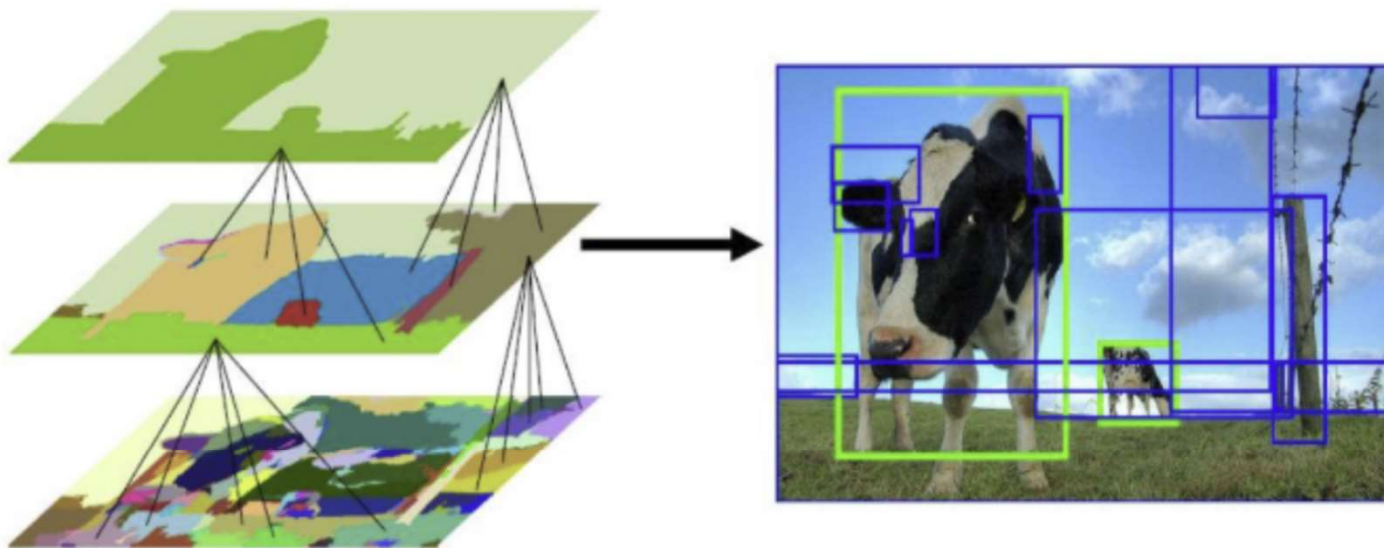
3) A set of class-specific linear SVMs.

Test : PASCAL VOC 2010-12

Detail how their parameters are learned and show results on test dataset.

Region proposals.

We use **selective search** to enable a controlled comparison(비교) with prior detection work.

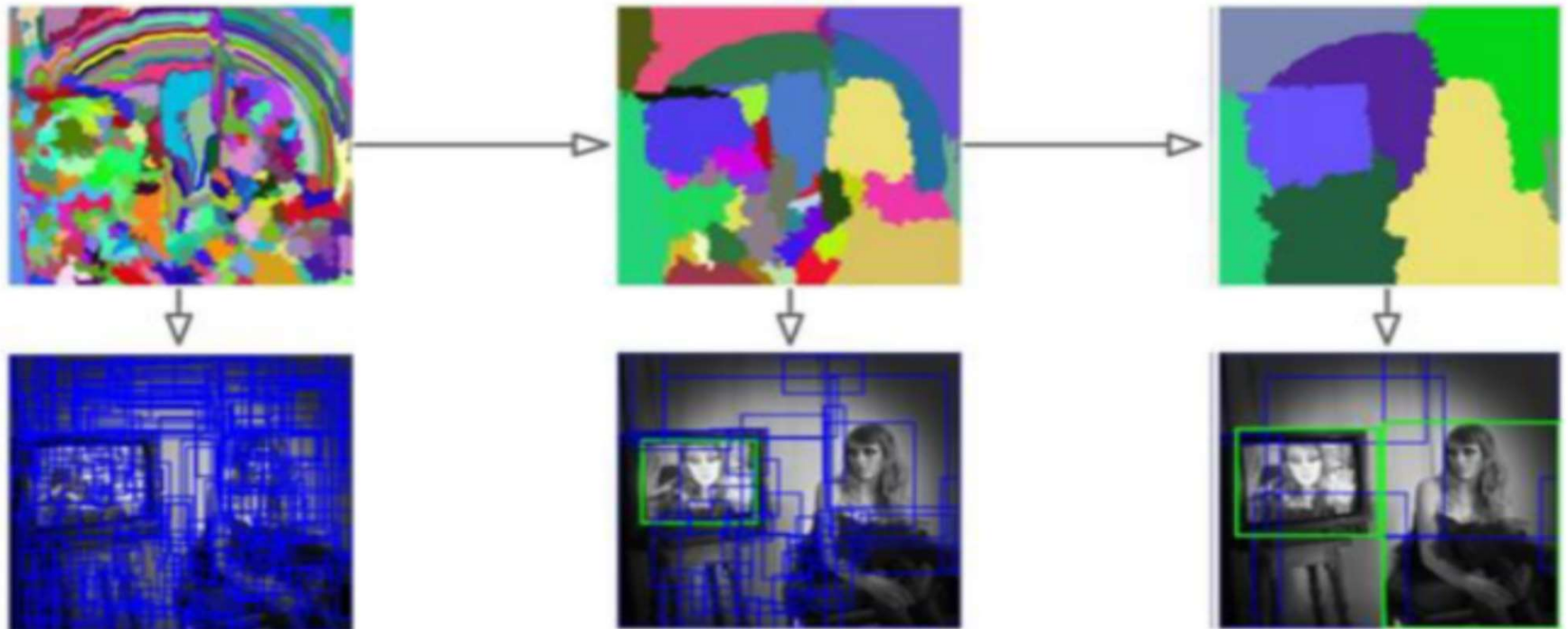


Selective search

- 1) color, size et al
=> small segmented areas
- 2) Bottom-up 방식으로 small segmented areas를 합쳐 big segmented areas를 만든다.
- 3) 위 작업을 반복해 2000개의 region proposal를 생성한다.
(ROI(Regional of Interest)를 기준으로 생성)

Bottom-up segmentation, merging regions at multiple scales

Convert
regions
to boxes



Selective search

< 1. Supervised Pre-Training >

- CNN의 pre-train에 사용한 데이터 : Imagenet
- caffe 코드로 pre-training 수행

< 2. Domain-Specific Fine-Tuning >

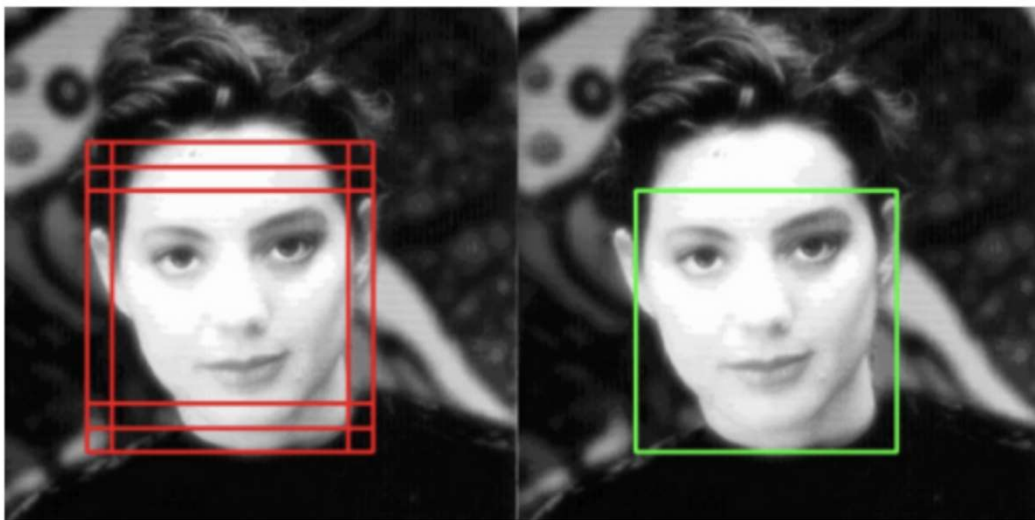
- CNN을 새로운 detection과 wrapped proposal window에 adapt하기 위해, wrapped proposals를 사용해 CNN 파라미터 학습에 SGD 사용
- CNN 아키텍처는 object의 class가 변경됨에 따라 마지막 레이어가 수정된 것 외에는 동일
- IoU 임계값은 0.5사용
- 학습률 0.001

Feature extraction

Make 2000 region proposal -> resize (227,227) RGB image (wrapping all pixel 비율 고려 x)

=> Through five convolutional layers and two fully connected layers for training Imagenet

=> Gain 4096-dimensional feature vector



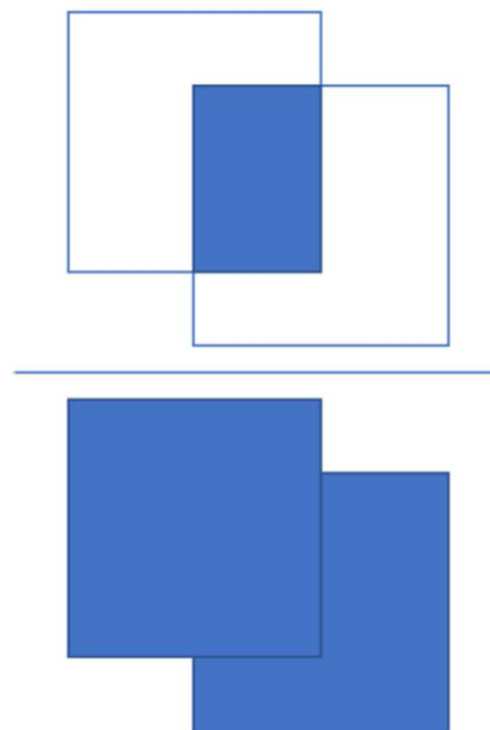
[도표 3] Non Maximum Supperssion

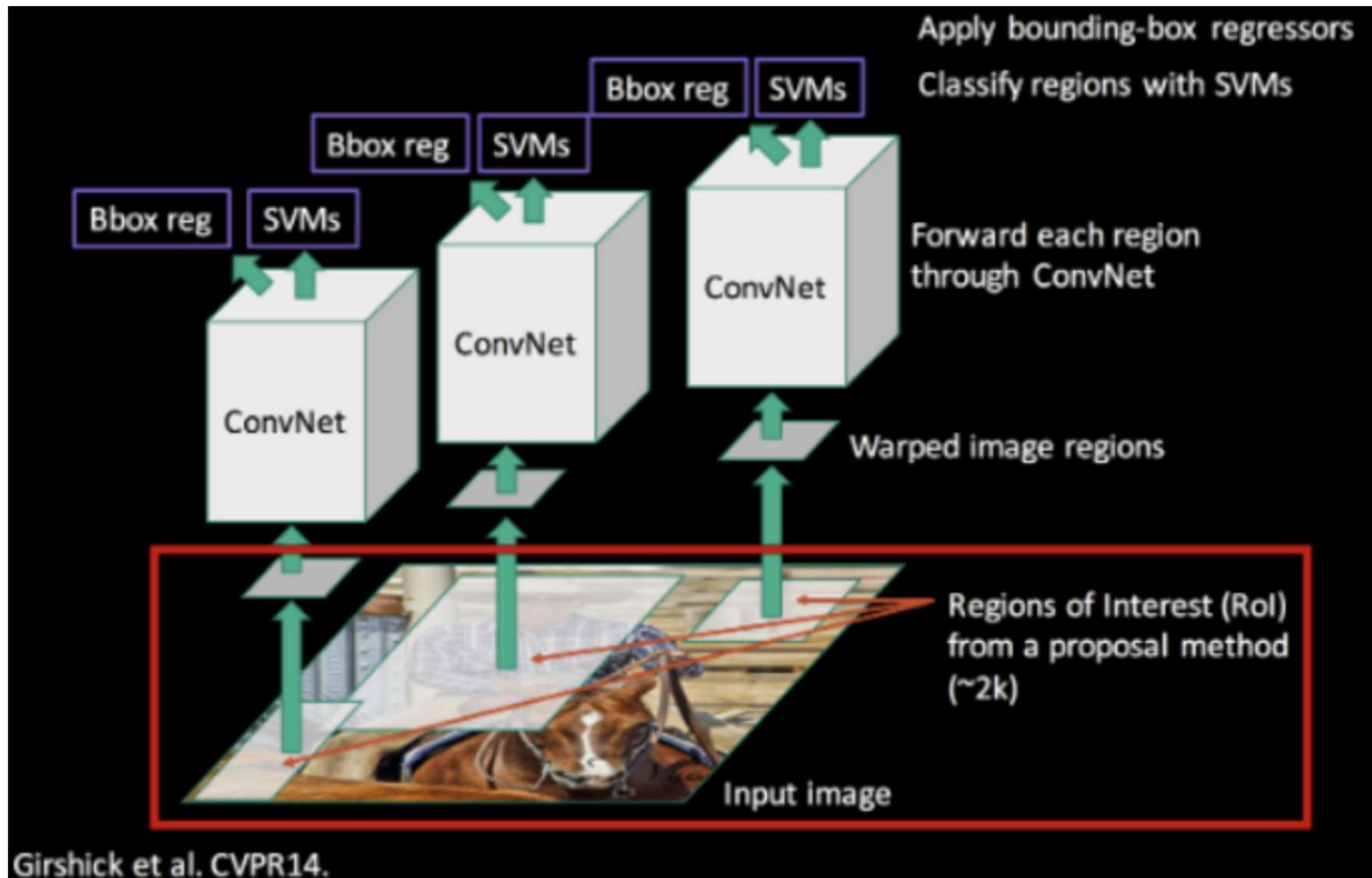
SVM을 통과하여 각각의 박스들은 어떤 물체일 확률값을 가질 수 있다.

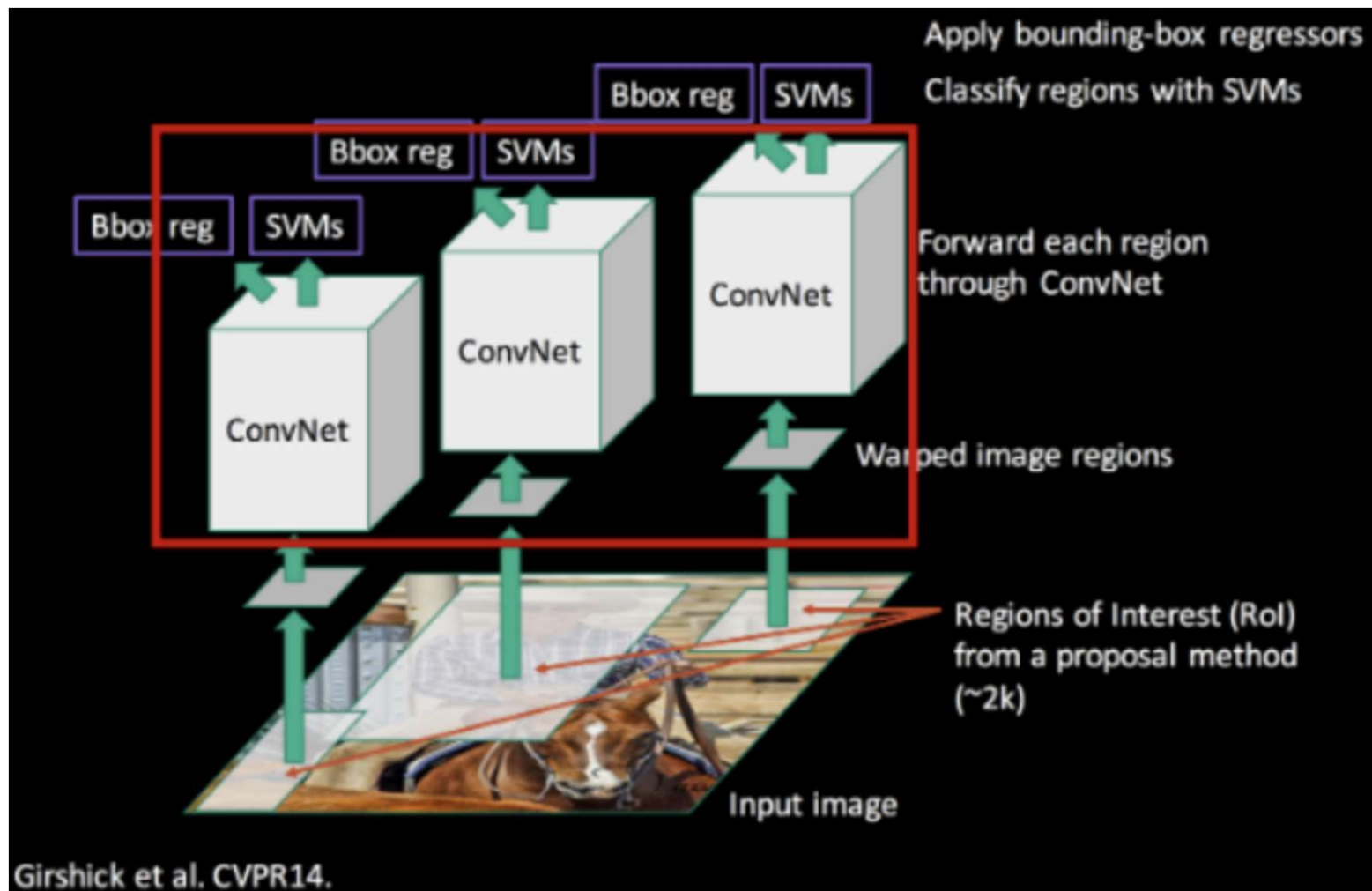
그 확률값 중에 가장 큰 값을 제외하고 나머지 박스들을 삭제해야 한다.

이때 각 박스가 동일한 물체를 나타내는지 판별하기 위해 IoU 값을 0.5보다 큰 값을 뽑아낸다.

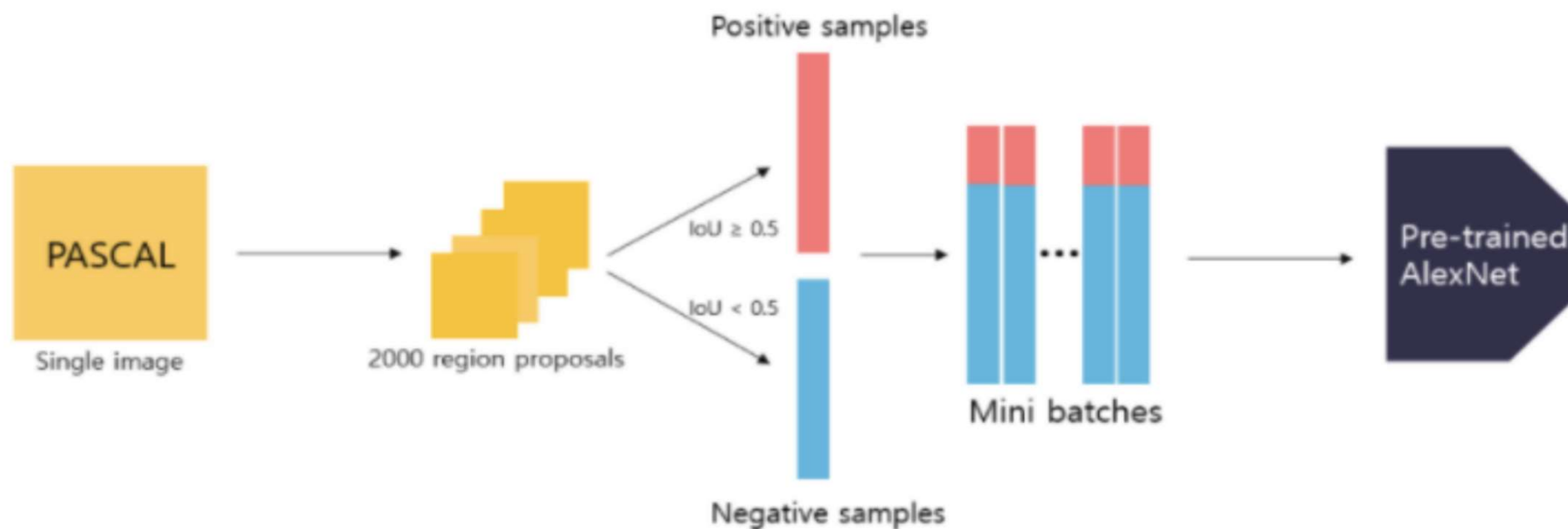
IoU(Intersection over Union)
= Intersection / A + B - Intersection

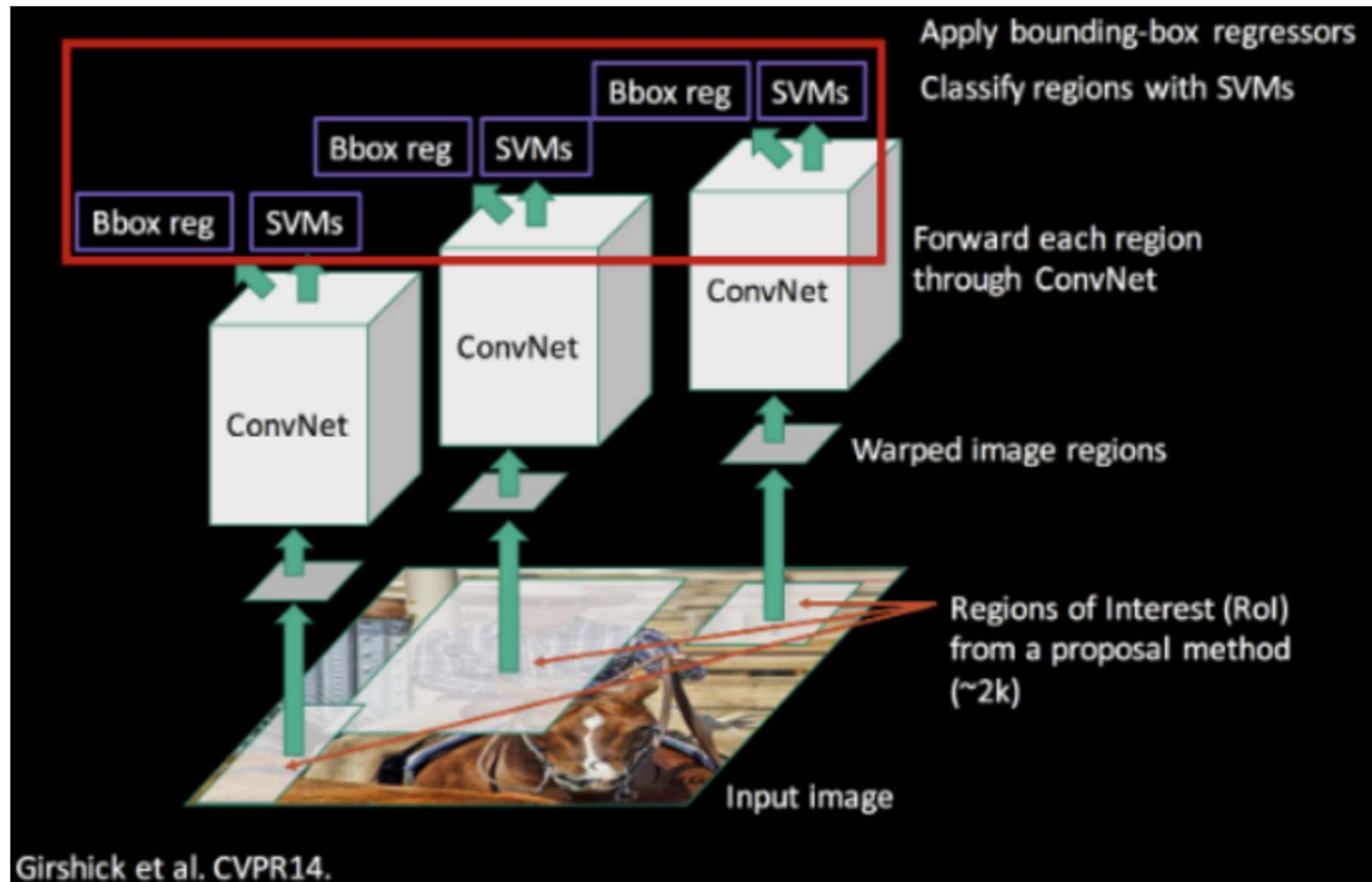


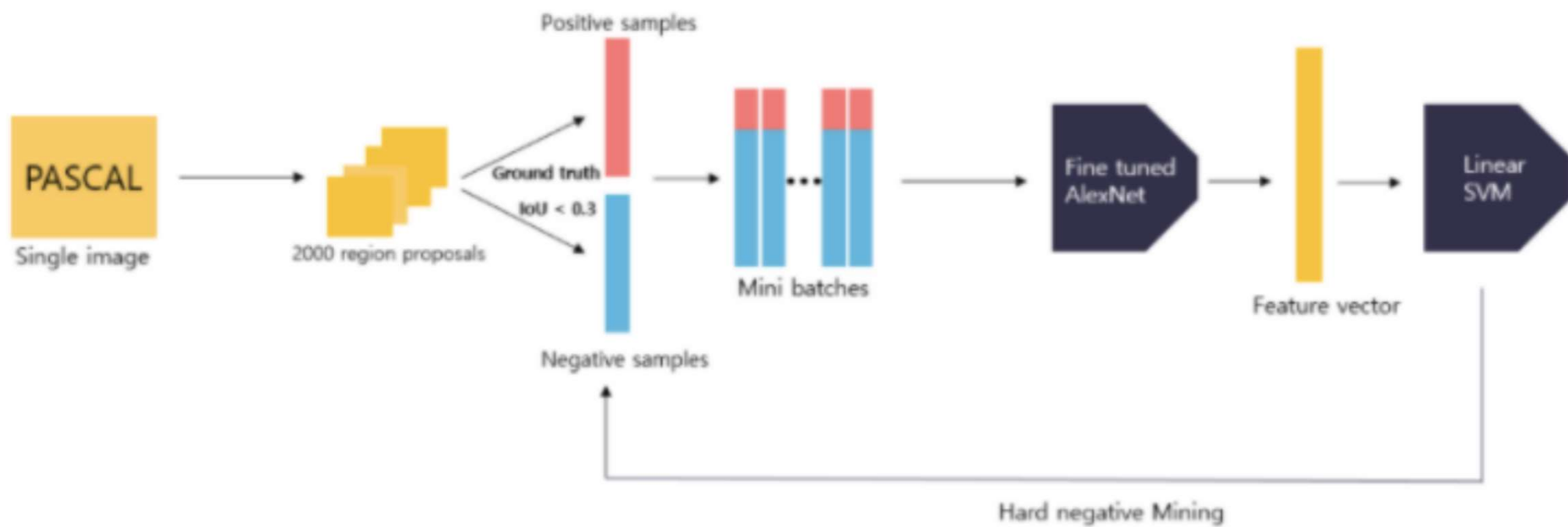




Domain-specific fine-tuning







Training

감사합니다[®]



허 지 혜 발표