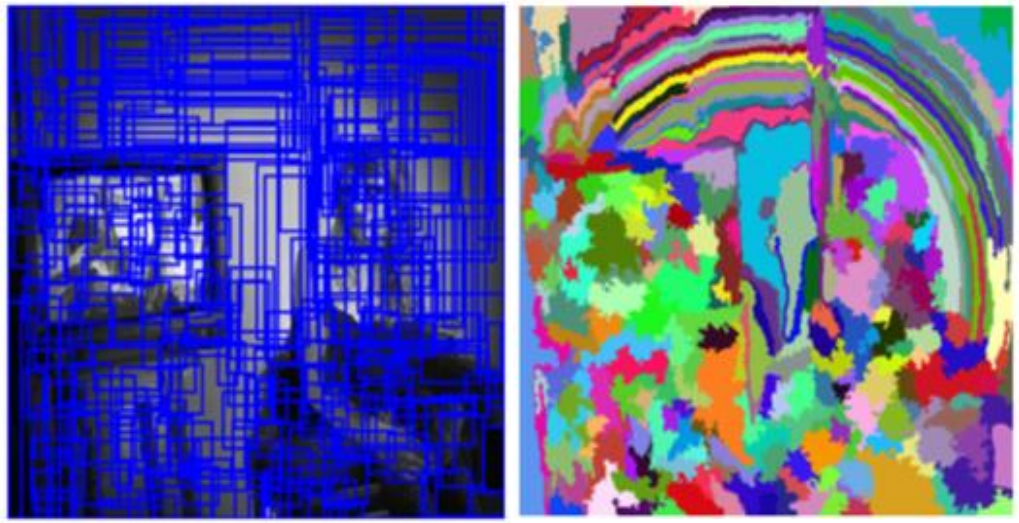
R-CNN

: Rich feature hierarchies for accurate object detection and semantic segmentation

Region Proposal – Selective Search

1. 작은 크기의 초기 영역을 설정



Efficient Graph-Based Image Segmentation

- 카테고리와 무관하게 객체가 있을 가능성이 높은 영역을 추출 (Category-independent region proposals)
- 색감, 질감, 영역 크기 등을 이용해 non-objective segmentation
- 이를 통해 많은 Small segmented areas를 얻는다.

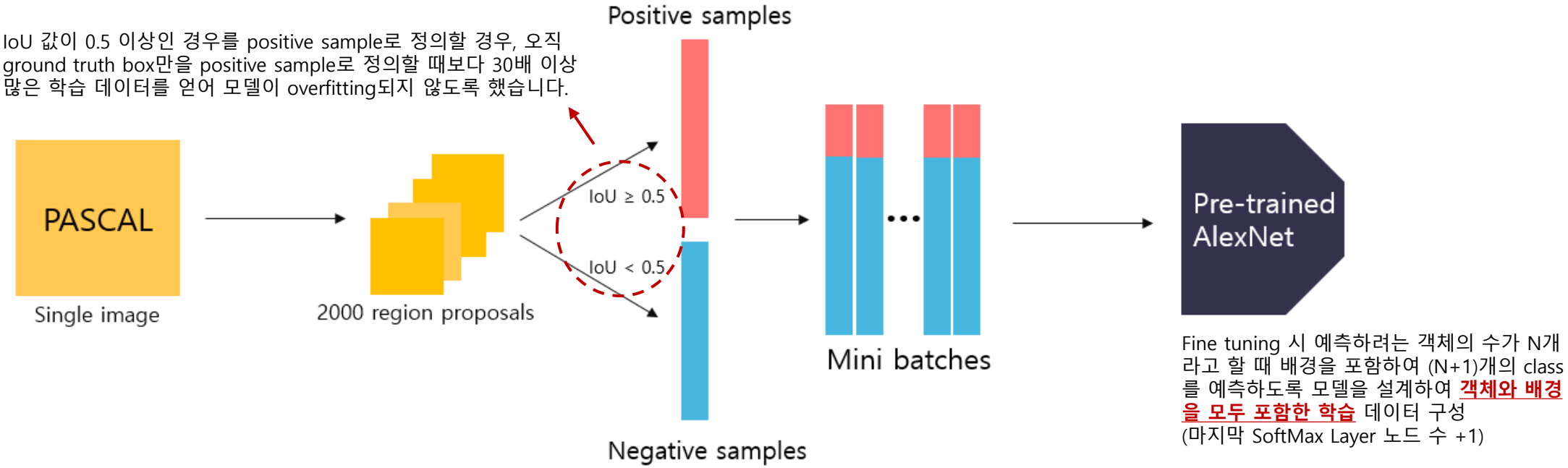
2. 작은 영역을 큰 영역으로 병합

Hierachical Grouping Algorithm

- Bottom Up 방식으로 Small segmented areas를 Big segmented areas로 합칩니다.
- 이 단계를 반복하여 2000개의 Region proposals 생성

Domain-specific fine-tuning

- IoU 값이 0.5 이상인 경우를 positive sample로 정의할 경우, 오직 ground truth box만을 positive sample로 정의할 때보다 30배 이상 많은 학습 데이터를 얻어 모델이 overfitting되지 않도록 했습니다.



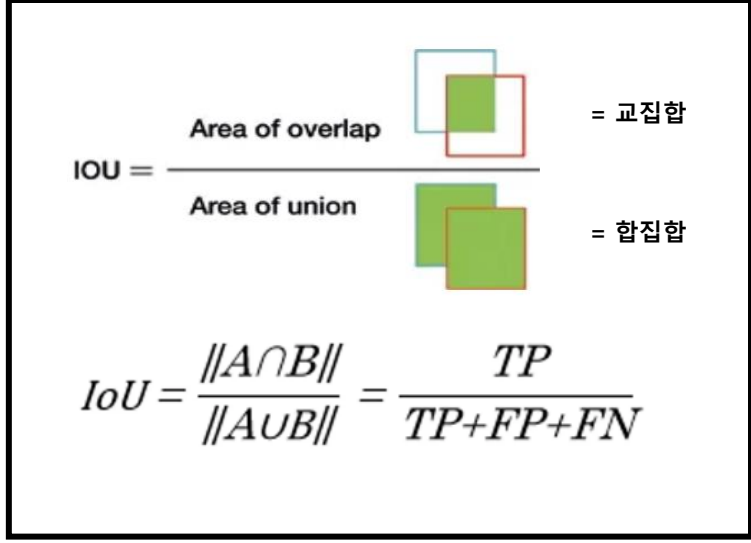
Fine tuning 시 예측하려는 객체의 수가 N개라고 할 때 배경을 포함하여 (N+1)개의 class를 예측하도록 모델을 설계하여 **객체와 배경을 모두 포함한 학습** 데이터 구성
(마지막 SoftMax Layer 노드 수 +1)

1. PASCAL 데이터셋을 **selective search** 하여 2000장의 region proposals를 얻습니다.
2. 각 region proposals의 bounding box와 ground truth box와의 **IoU** 값을 구합니다.
3. IoU 값이 0.5 이상인 경우 positive sample(=객체)로, 0.5 미만인 경우 negative sample(=배경)로 저장합니다.
4. Positive sample = 32, negative sample = 96 로 **mini batch(128)**을 구성합니다.
5. Mini batch를 pre-trained된 **AlexNet**에 입력하여 학습을 진행합니다.

→ AlexNet의 목적을 Classification에서 Localization으로 변경

- ImageNet이 Pre-traine된 AlexNet은 Classification에 적합하기 때문에 Localization에 적합한 모델로 만들기 위해 IoU(위치 기반) 객체, 배경을 분리 한 데이터를 추가 학습하여 Fine tuned AlexNet을 만들어 Training linear SVM using fine tuned AlexNet 단계에서 4096-dimensional feature vector를 만들때 활용

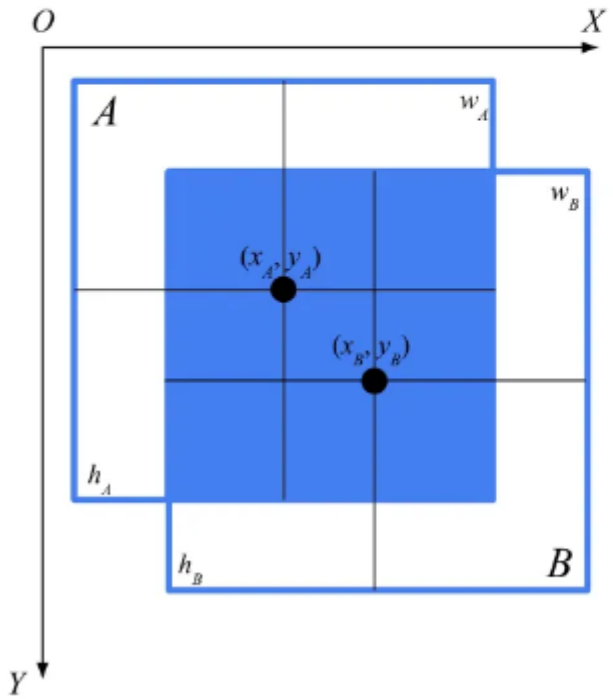
IoU (Intersection over Union) -> 예측한 Bounding Box와 Ground Truth가 일치하는 정도를 0과 1사이의 값으로 나타낸 것.



$$IoU = \frac{\|A \cap B\|}{\|A \cup B\|} = \frac{TP}{TP+FP+FN}$$

IoU 계산 방법

- x : 영역 중심의 x좌표
- y : 영역 중심의 y좌표
- w : 영역의 폭
- h : 영역의 높이



1 2개의 영역이 아래와 같이 주어짐.

$$A(x_A, y_A, w_A, h_A)$$
$$B(x_B, y_B, w_B, h_B)$$

2 교집합된 부분의 가로, 세로 길이 구함.

조건 : dxdy > 0

$$dx = \min(x_{A,max}, x_{B,max}) - \max(x_{A,min}, x_{B,min})$$
$$dy = \min(y_{A,max}, y_{B,max}) - \max(y_{A,min}, y_{B,min})$$

3 IoU 구하는 식으로 정리하면

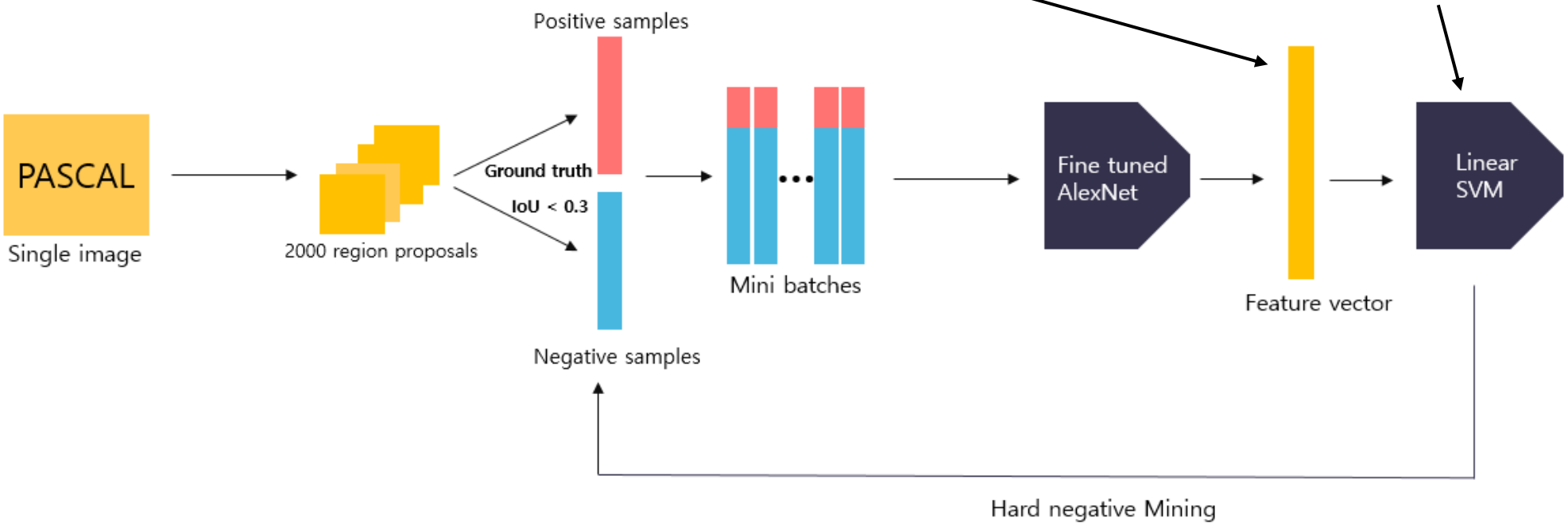
$$A \cup B = A + B - A \cap B$$
$$= w_A h_A + w_B h_B - dxdy$$

$$IoU = \frac{A \cap B}{A \cup B}$$
$$= \frac{dxdy}{w_A h_A + w_B h_B - dxdy}$$

Training linear SVM using fine tuned AlexNet

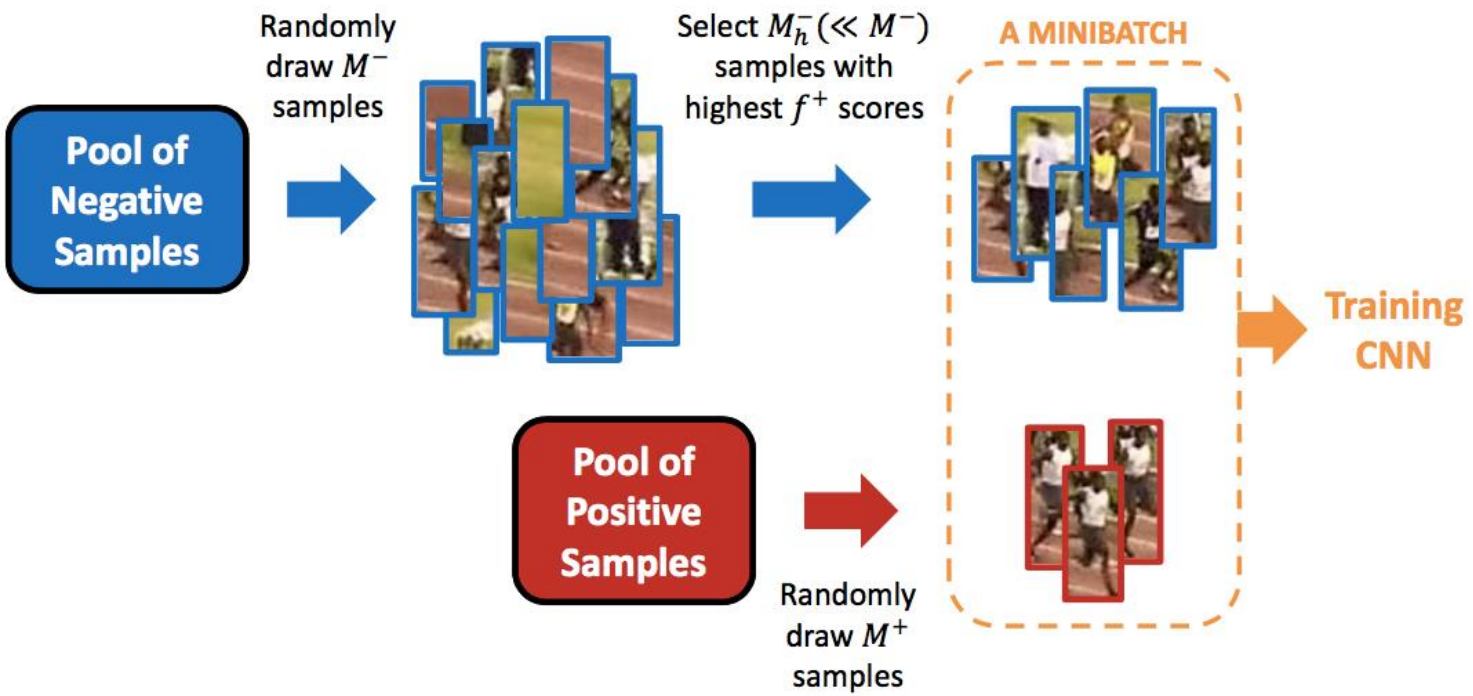
- Linear SVM 모델은 2000x4096 feature vector를 입력 받아 class를 예측하고(object or background) confidence score를 반환
- N개의 class를 예측한다고 할 때, 배경을 포함한 (N+1)개의 독립적인 linear SVM 모델을 학습해야 합니다.
- R-CNN에서는 단순히 N-way softmax layer를 통해 분류를 진행하지 않고, SVM를 이용해 분류하는 이유는 성능이 더 좋다고 합니다.

- Selective search를 통해 얻은 특정 부분의 이미지를 Fine tuned된 AlexNet을 통해 요약한 4096-dimensional feature vector



1. 2000장의 region proposals에서 fine-tuning때와는 다르게 ground truth box만을 positive sample, **IoU** 값이 0.3보다 작은 것은 negative sample. (이 때 0.3은 Grid search를 통해 찾은 값이다.)
2. 이후 fine-tuning 때와 마찬가지로 positive sample 32개 + negative sample 96개 = 128개의 **mini batch**를 만든다.
3. **Fine tuned AlexNet**에 입력하여 **feature vector**를 추출하고 이를 **linear SVM**에 입력하여 학습합니다.

Hard Negative Mining



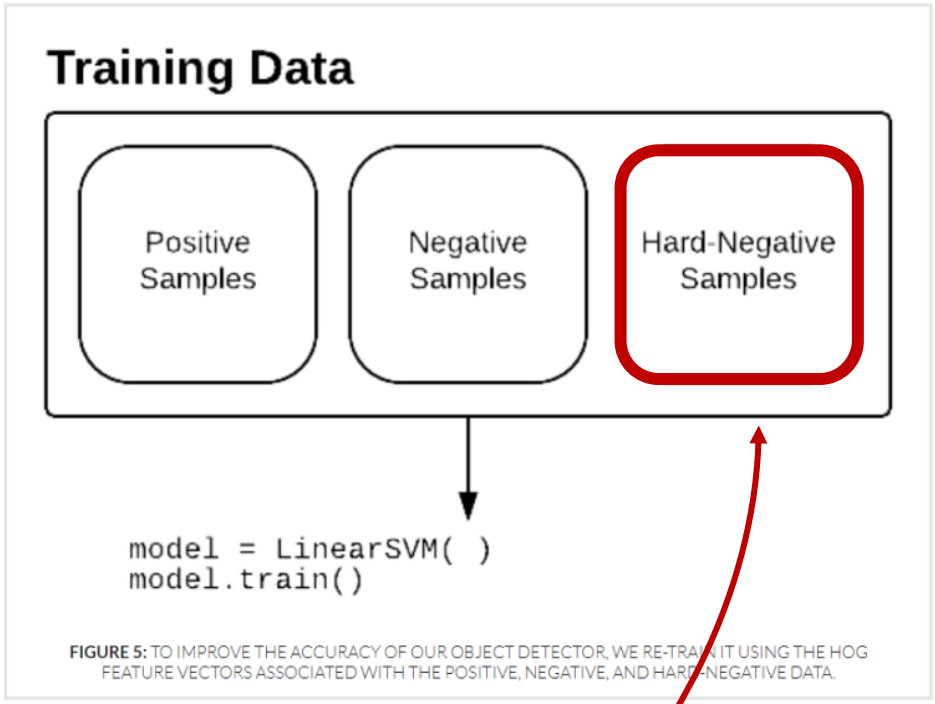


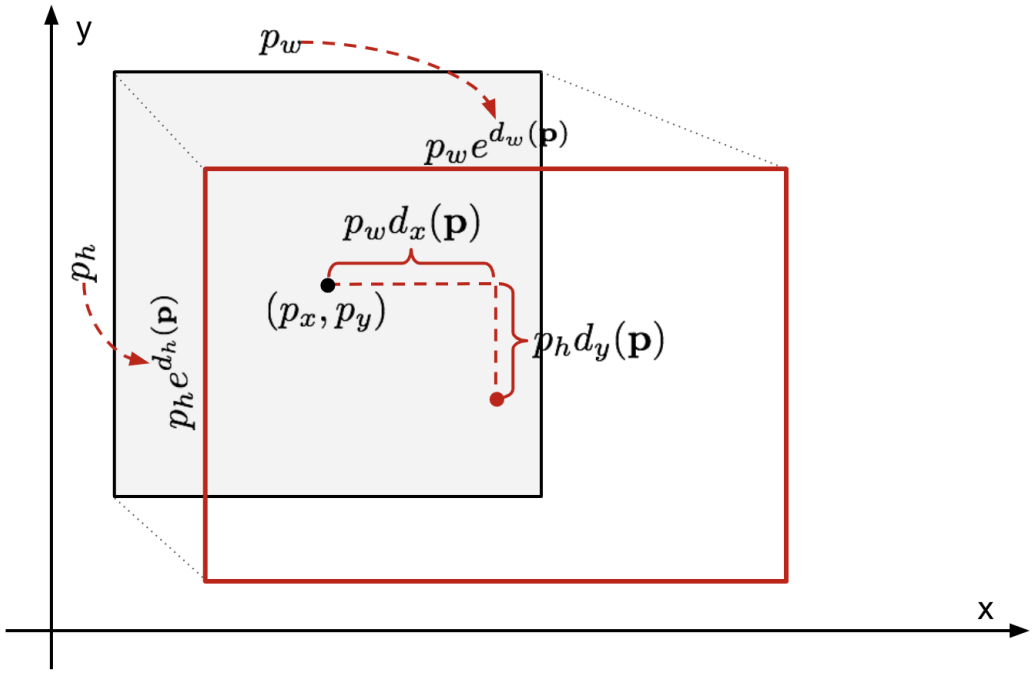
FIGURE 5: TO IMPROVE THE ACCURACY OF OUR OBJECT DETECTOR, WE RE-TRAIN IT USING THE HOG FEATURE VECTORS ASSOCIATED WITH THE POSITIVE, NEGATIVE, AND HARD NEGATIVE DATA.

- Positive sample : Image에서 사람을 탐지하는 경우 실제 사람인 경우
- Negative sample : 그 외의 배경
- True negative sample : 모델이 Bounding box를 배경이라고 예측하고 실제로 배경인 경우
- **False positive sample** : 모델이 Bounding box를 사람이라고 예측했지만, 실제로 배경인 경우

→ Linear SVMs 학습시킬 때 False positive sample들을 추가 학습

- 클래스 불균형 때문에 Positive sample보다 negative sample이 더 많아 모델은 false positive 오류를 주로 범하게 되기에 이와 같은 방법으로 모델이 false positive sample을 더 잘 맞출 수 있게 했습니다.

Detailed Localization by Bounding Box Regressor



$$d_{x(P)} = t_x$$

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

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

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

위의 그림에서 회색 box는 Selective search 알고리즘에 의해 예측된 bounding box이며, 빨간 테두리 box는 ground truth box입니다.
Bounding box regressor는 예측한 bounding box의 좌표 p=(px,py,pw,ph)p=(px,py,pw,ph)(center X, center Y, width, height)가 주어졌을 때, ground truth box의 좌표 g=(gx,gy,gw,gh)g=(gx,gy,gw,gh)로 변환되도록 하는 **Scale invariant Transformation**을 학습합니다.

$\hat{g}_i$: 예측한 bounding box p가 주어졌을 때, Bounding box regressor 모델이 변환한 결과

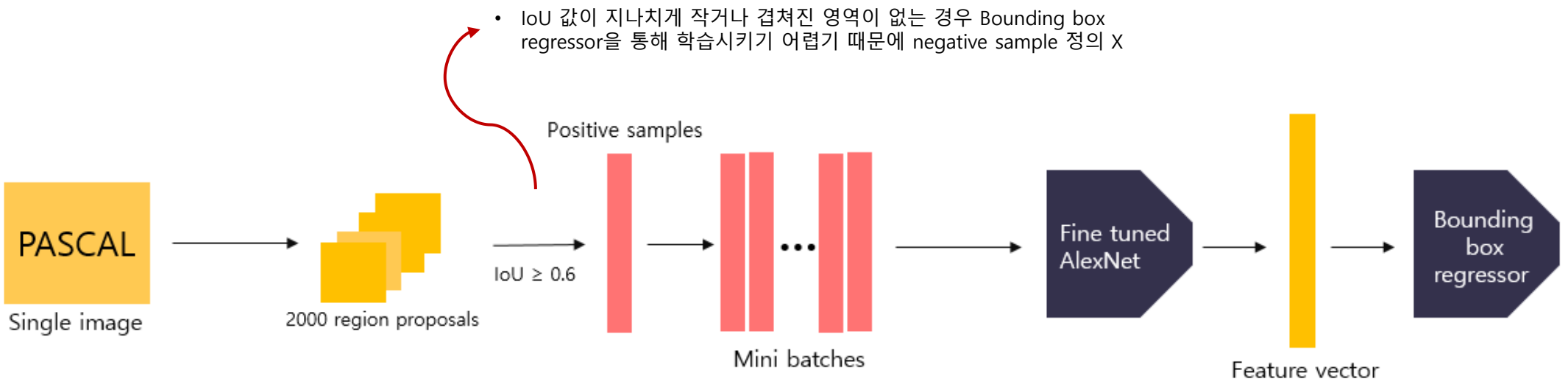
t_i : Bounding box regressor 모델이 학습하고자 하는 목표(target)

$d_i(P)$: Bounding box regressor 모델의 학습 대상

L_{reg} : Bounding box regressor 모델의 loss function으로 **SSE(Sum of Squared Error)**. $\lambda = 1000$

→ 즉, Bounding box regressor 모델은 $d_{i(P)}$가 t_i가 되도록 L_{reg}를 통해 학습시킵니다.

Training Bounding box regressor using fine tuned AlexNet

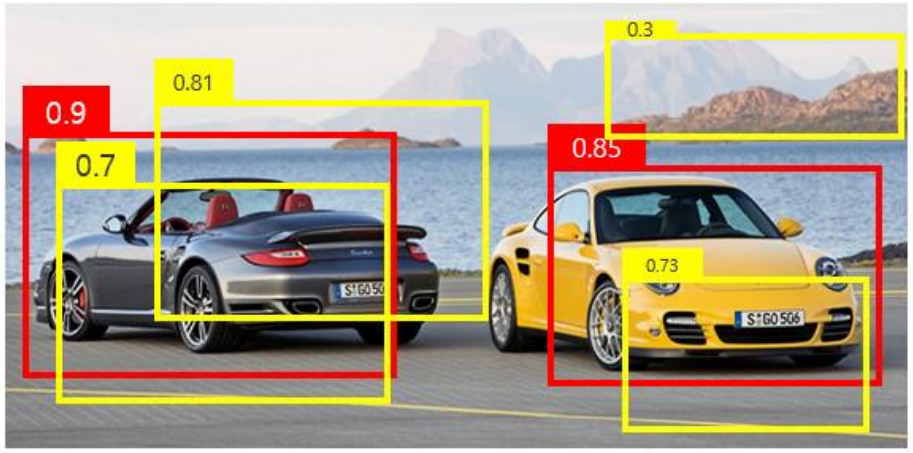


1. PASCAL 데이터셋에 **Selective search** 알고리즘을 적용하여 얻은 region proposals를 학습 데이터로 사용
2. Negative sample 정의하지 않고 **IoU** 값이 0.6 이상인 sample을 positive sample로 정의
3. Positive sample을 fine tuned된 AlexNet에 입력하여 얻은 feature vector를 **Bounding box regressor**에 입력하여 학습

→ Bounding box regressor는 feature vector를 입력 받아
조정된 bounding box 좌표값 (output unit=4)을 반환합니다.

- Bounding box = box(x, y, w, h)

Non maximum Suppression -> Bounding box 중에서 비슷한 위치에 있는 box를 제거하고 가장 적합한 box를 선택하는 Non maximum suppression 알고리즘 적용

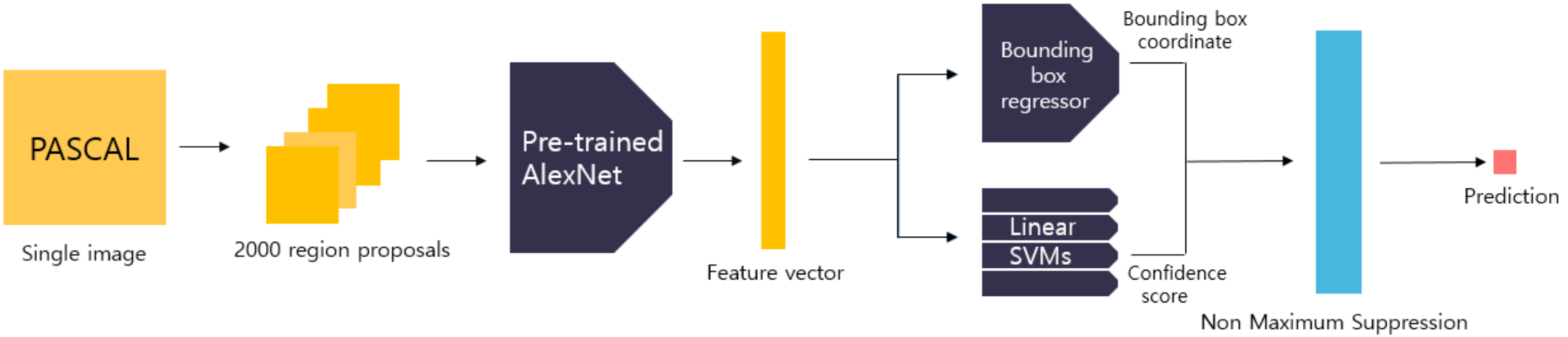


Before Non Maximum Suppression

After Non Maximum Suppression

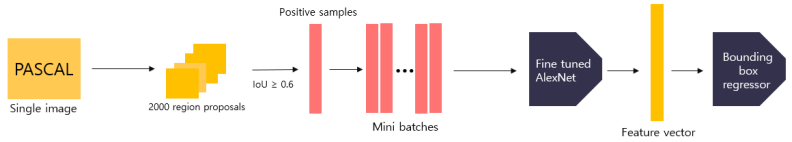
1. Bounding box별로 지정한 **confidence score threshold 이하의 box를 제거**
2. 남은 bounding box를 **confidence score에 따라 내림차순**으로 정렬
3. Confidence score가 높은 순의 bounding box 부터 다른 box와의 IoU 값을 조사하여 **IoU threshold 이상인 box를 모두 제거**
4. 남아있는 box 선택

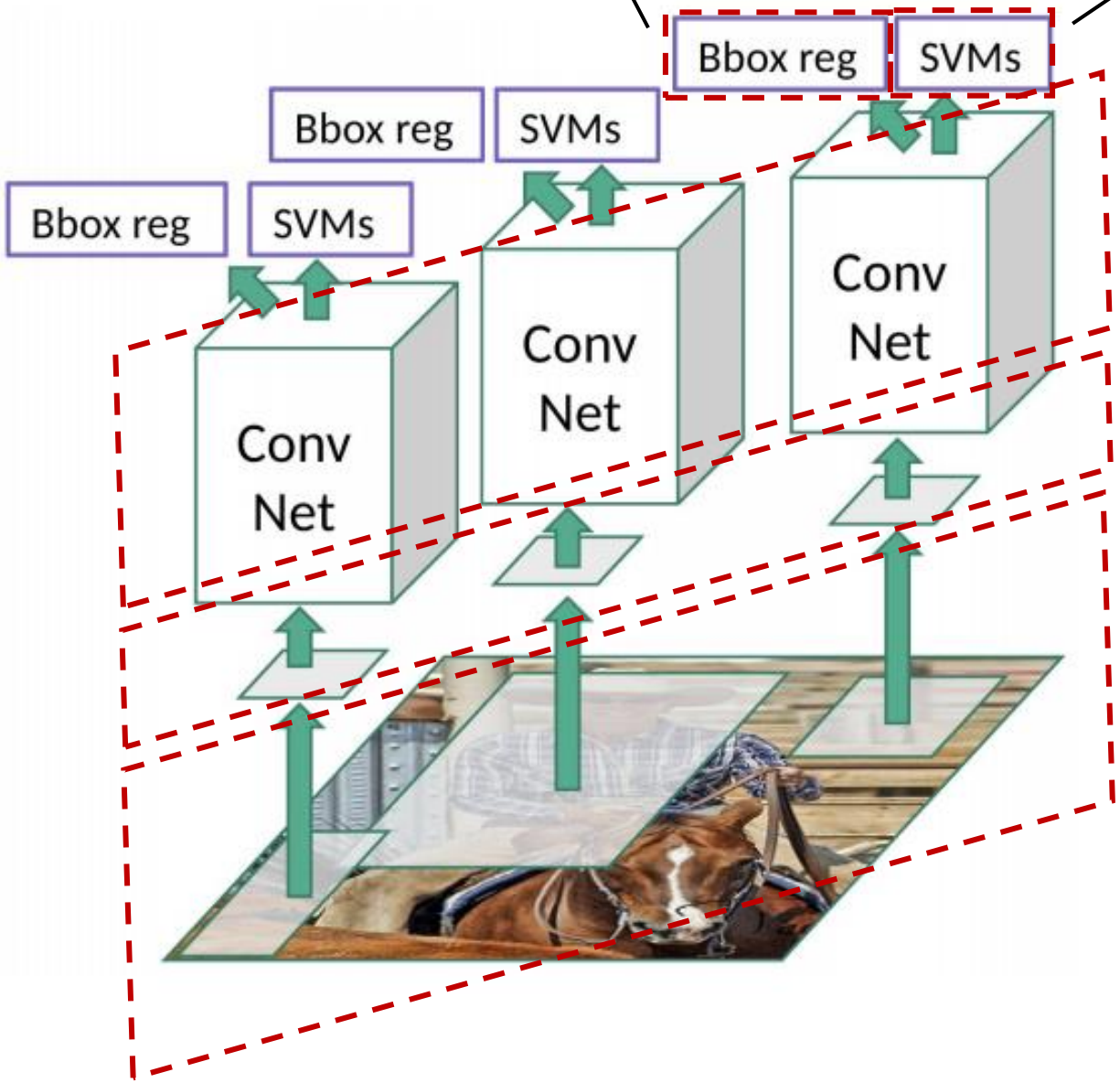
Object Detection by R-CNN



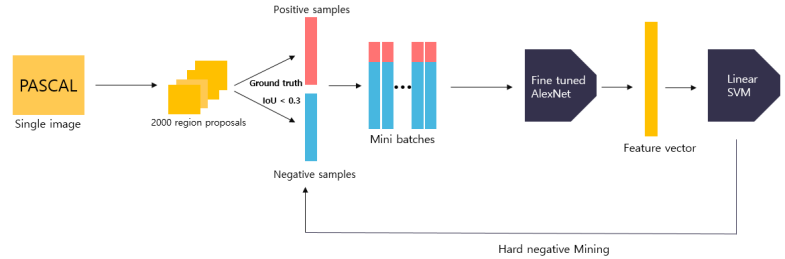
- R-CNN은 Object Detection을 수행하기 위해 최초로 딥러닝을 적용했습니다.
- Selective search를 사용하여 이미지 한 장당 2000개의 region proposals을 추출하니 학습 및 추론 속도가 매우 느립니다.
- Fine tuned AlexNet, Linear SVM, Bounding box regressor 3가지 모델을 사용하니 전체 구조와 학습 과정이 복잡합니다.

END

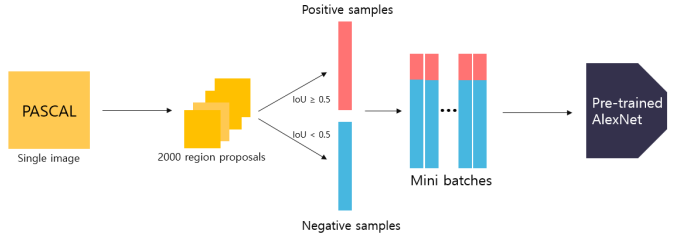




Training linear SVM using fine tuned AlexNet



Domain-specific fine-tuning



Wrapped Image Regions

- Selective search를 통해 만들어진 여러 사이즈들의 이미지를 227x227 사이즈로 통합

Region Proposal – Selective Search

- Efficient Graph-Based Image Segmentation
- Hierachical Grouping Algorithm