# 6. R-FCN:Object Detection via Region-based Fully Convolution Networks explained

Region Based Detector의 대표적인 예시인 Fast / Faster R-CNN의 경우 region proposal을 통해 구한 RoI들을 각각 계산해주어야 한다는 단점이 존재함

R-FCN은 단점을 해결하여 높은 정확도를 달성함

## Introduction

- Translation Invariance


  - Invariance : 불변 → 이미지에서 객체의 모습이 달라져도 객체를 파악할 수 있는 성질을 의미
  - translation : 평행이동 → 이미지에서 각각의 픽셀값들이 일정 방향으로 이동한것을 의미
    - 모델은 위치나 객체의 모습에 불변하게 항상 존재를 인식할 수 있어야함
    - 따라서, 모델이 translation & invariance 성질을 얻는것은 굉장히 중요함
  - translation equivariance : 입력의 신호가 바뀌면 그에 따라 결과값도 바뀜 ↔ translation invariance 와 반대되는 개념

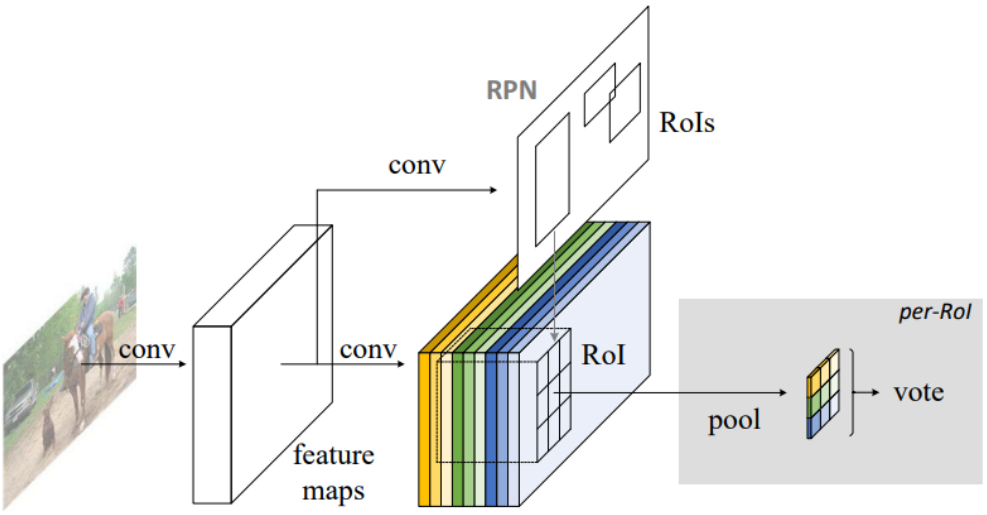
## Translation Invariance Dilemma

- Classification : translation invariance 가 중요함 → 객체의 위치가 바뀌어도 찾아야 하기 때문
- Object Detection : translation equivarianc 가 중요함 → localization이 중요해, 위치가 바뀌면 bounding box의 위치도 함께 조정돼야 하기 때문
- 2-stage detector들은 먼저 backborn network를 통해 feature map을 생성하고 구해진 RoI와 함께 Detection을 수행
    - Backborn Network(ResNet, VGG..)는 Classification을 목적으로 만들어 translation invariance한 성질을 갖게됨
    - 하지만 뒤 Detection Network에서 위치정보는 굉장히 중요함 → 위치가 변하더라도 위치에 대한
      정보를 가지고 있어야 함
    - 결국 위치정보를 포함하지 않은 값을 Detection Network에서 처리하기 때문에 모델의 정확도가 떨어짐 → 두 네트워크의 학습 성질이 달라 발생하는 문제 : Translation Invariance Dilemma

## RoI Pooling Layer For Object Detection

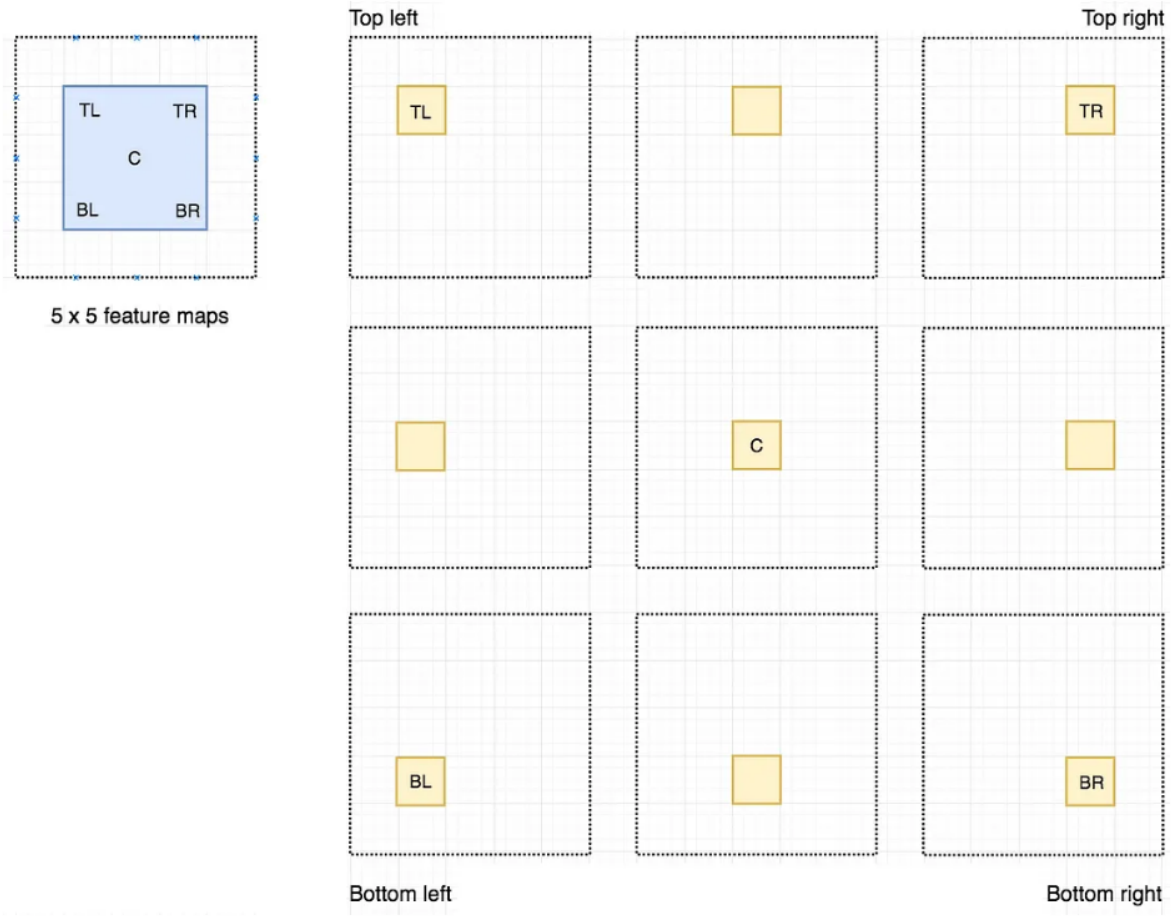
- Translation Invariance Dilemma를 해결하기 위해 제시된 방법이 RoI Pooling Layer를 네트워크에 삽입하는 것 → ResNet 논문에서 제시된 방법
    - Region Proposal 이후 RoI Pooling 진행
- 해당 연산을 통해 Backborn Network의 translation invariance 한 성질을 없앨 수 있다고 함
    - 따라서 RoI Pooling 뒤에 오는 Detection Network는 더 이상 translation invariance한 성질이 없어 정확도 향상
- 각각 의 모든 RoI에 대해 별도의 계산과정을 거치기 때문에 학습과 테스트 시 속도가 저하됨 → 이러한 문제를 해결하기 위해 R-FCN이라는 구조가 제안됨

## Network Architecture



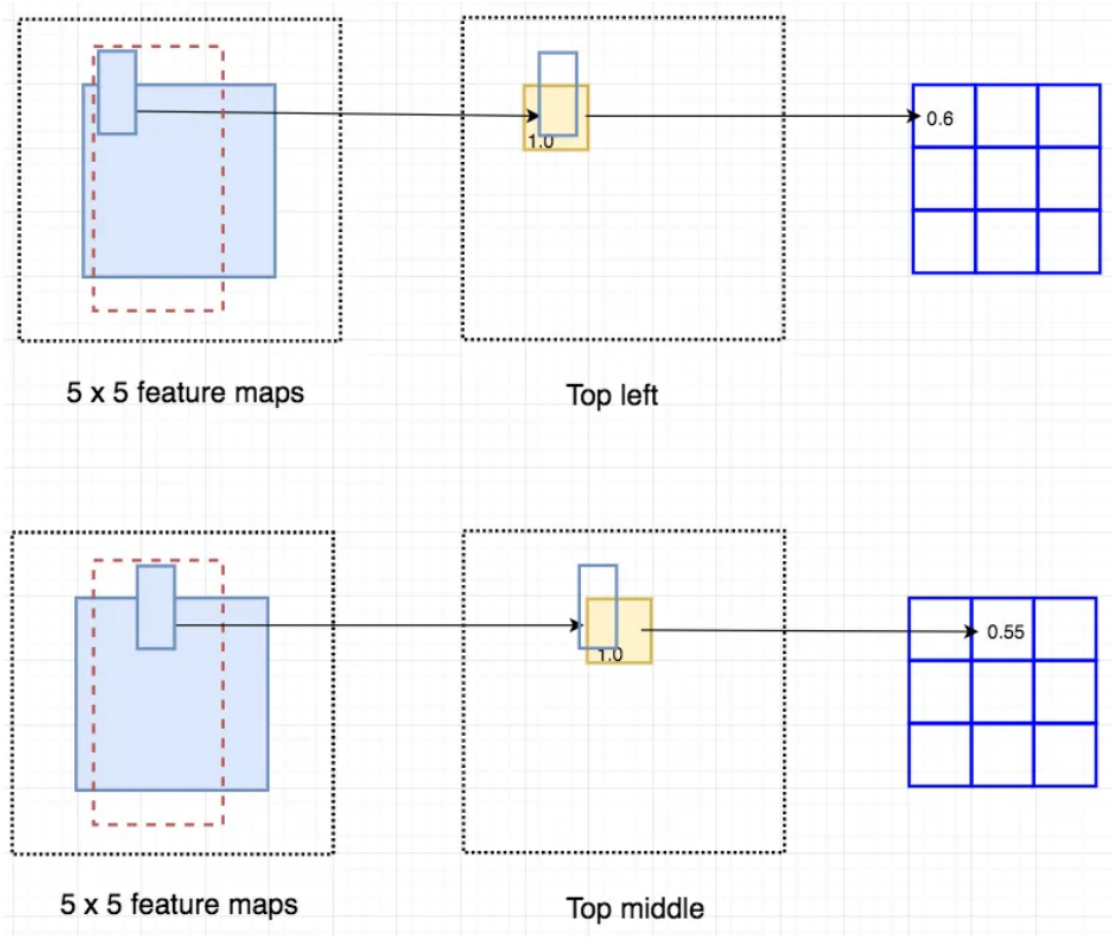

- Backborn Network를 통해 Feature Map을 계산 → ResNet101 사용
- 이후 Convolution Layer를 거쳐 k x k x ( C +1 ) 차원(1은 배경)의 position-sensitive score map이라는 것을 만들어 냄
  - 간단히 R-FCN에 translation variance 성질을 주기 위해 사용됨
  - 그 후 Position-Sensitive RoI Pooling을 적용
  - Fully Convolution Network 구조 이기 때문에 전체 이미지에 대해 연산으 공유할 수 있고 end - to - end 방식으로 학습할 수 있다는 장점이 존재

## Position-Sensitive Score Map

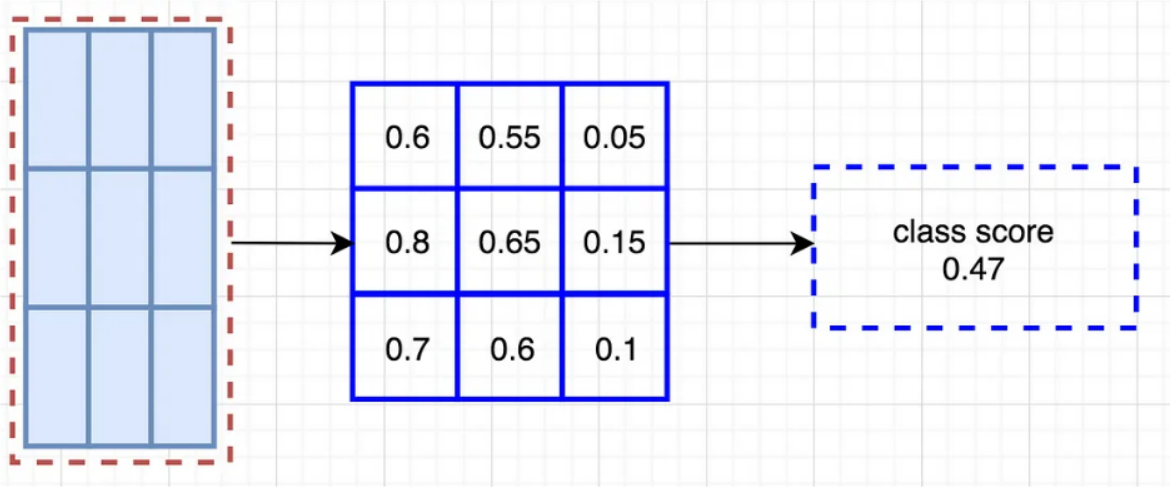



- 5 × 5 Feature map에서 파란색 영역은 모델이 탐지할 객체
  - R-FCN에서는 개체 영역을 3 x 3 영역으로 나눔 {top-left,  top-center, top-right,... bottom-left, bottom-center, bottom-right}
  - 각각의 영역에 전문화(specialize)된 9개의 feature map을 생성
  - YOLOv1에서 살펴보았듯이 각각의 영역에서의 detector들은 객체의 일정 부분을 탐지할 책임(responsible)을 지게 되는 것과 비슷

## Position-Sensitive RoI Pooling Layer




- 빨간색 점섬 영역은 구한 RoI를 의미 → 마찬가지로 3 x 3 영역으로 분할
  - 먼저 RoI에서 top-left영역과 score map에서의 top-left 영역을 매핑 → 위의 그림에서 첫 번째 과정과 같음
  - 이제 파란색 영역(RoI top-left)과 노란색 영역(score map top-left)이 얼마나 겹치는 가에 대해 점수를 계산
  - 논문에서는 겹쳐지는 부분이 60% 이상이라면 활성화 정도를 100%로 하고 40% 미만이면 0%로 활성화
  - 각각의 계산된 값을 2차원 배열에 저장




- 각각 9개의 값을 계산하고 class score는 모든 요소를 평균한 값을 가짐
  - 이러한 식의 연산을 position-sensitive RoI pooling이라고 함
- 만약 구별해야 하는 클래스가 배경을 포함해서 C + 1개라면 각각의 클래스는 3 x 3 score map을 가지고, 이후 각각의 클래스에 대해 calss score를 계산하고 softmax를 통해 확률값을 계산

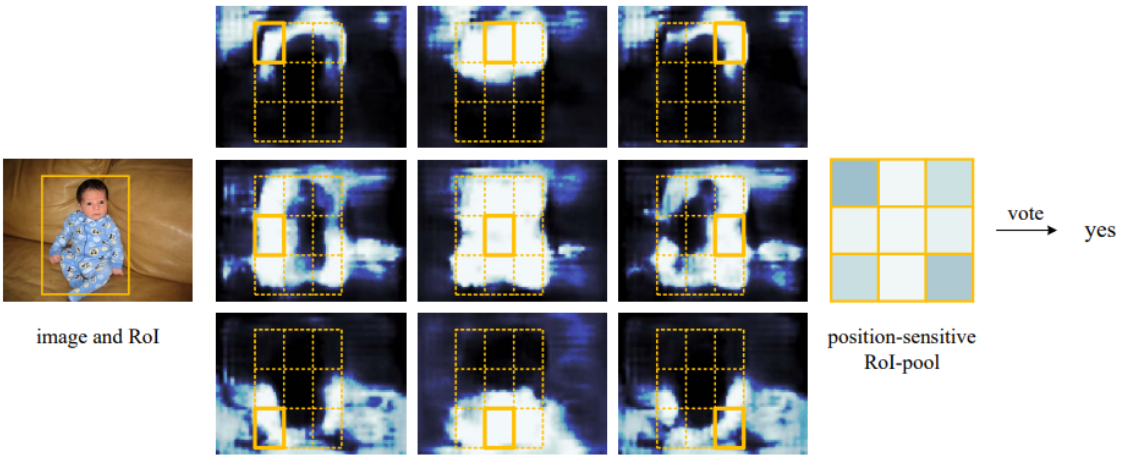



Figure 3: Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.

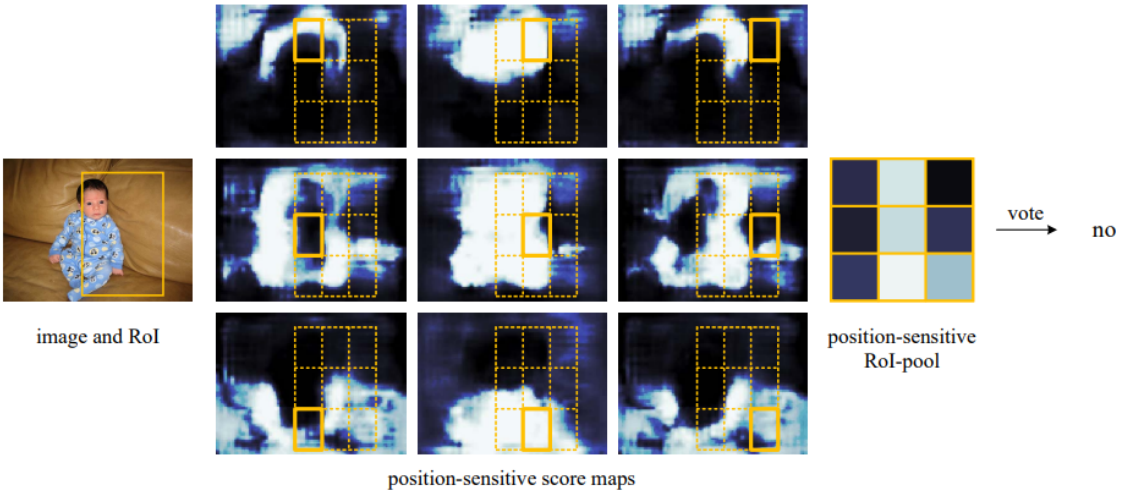



Figure 4: Visualization when an RoI does not correctly overlap the object.

## References

https://sncv.tistory.com/7

[1605.06409] R-FCN: Object Detection via Region-based Fully Convolutional Networks (arxiv.org)

Understanding Region-based Fully Convolutional Networks (R-FCN) for object detection | by Jonathan Hui | Medium

machine learning - What is translation invariance in computer vision and convolutional neural network? - Cross Validated (stackexchange.com)