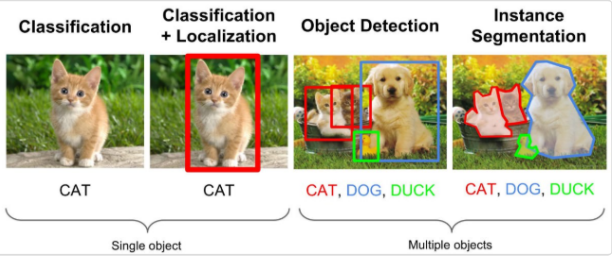
R-CNN

레퍼런스

(논문리뷰) R-CNN 설명 및 정리

컴퓨터비전에서의 문제들은 크게 다음 4가지로 분류할 수 있다. 1. Classification 2. Object Detection 3. Image Segmentation 4. Visual relationship 이중에서 4. Visual relationship은 나중에 다루고 먼저 위 3개의 차이를 살펴보자. Classification : Single object에 대해서 object의 클래스를 분류하는 문제이다. Classification + Localization : Single

https://ganghee-lee.tistory.com/35

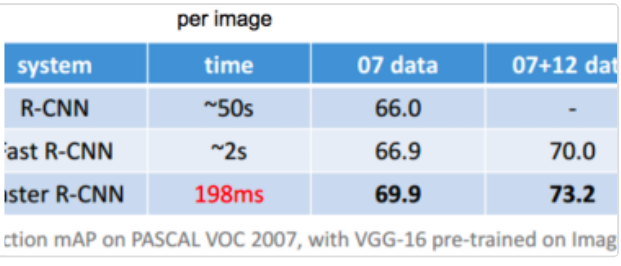


https://yeomko.tistory.com/13?category=888201

R-CNNs Tutorial

INTRODUCTION Object detection은 입력 영상이 주어졌을 때, 영상 내에 존재하는 모든 카테고리에 대해서 classification과 localization을 수행하는 것을 말합니다. 입력 영상에 따라 존재하는 물체의 개수가 일정하지 않고 0~N 개로 변하기 때문에 난이도가 높은 task로 알려져 있습니다. 본 글에서는 convolutional neural network(CNN) 기반의

https://blog.lunit.io/2017/06/01/r-cnns-tutorial/



R-CNN 프로세스

- image 를 입력받는다.
- selective search 알고리즘에 의해 regional proposal output 약 2000 개를 추출한다.
- 추출한 regional proposal output 을 모두 동일 input size 로 만들어주기 위해 warp 해준다.
  - 구체적으로 227*227 크기로 리사이즈(wrap) 한다. 박스의 비율은 고려하지 않는다.
- 2000개의 warped image 를 각각 CNN 모델에 넣는다.
  - 4096 차원의 특징 벡터를 추출한다.
- 각각의 convolution 결과에 대해 classification 을 진행하여 결과를 얻는다.
  - convolution 결과, 추출된 벡터를 가지고 각각의 클래스마다 학습시켜놓은 SVM classifier 를 통과시킨다.
- bounding box regression 을 적용하여 박스의 위치를 조정한다.

Abstract

image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that im-

a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: *Regions with CNN*

1. Introduction

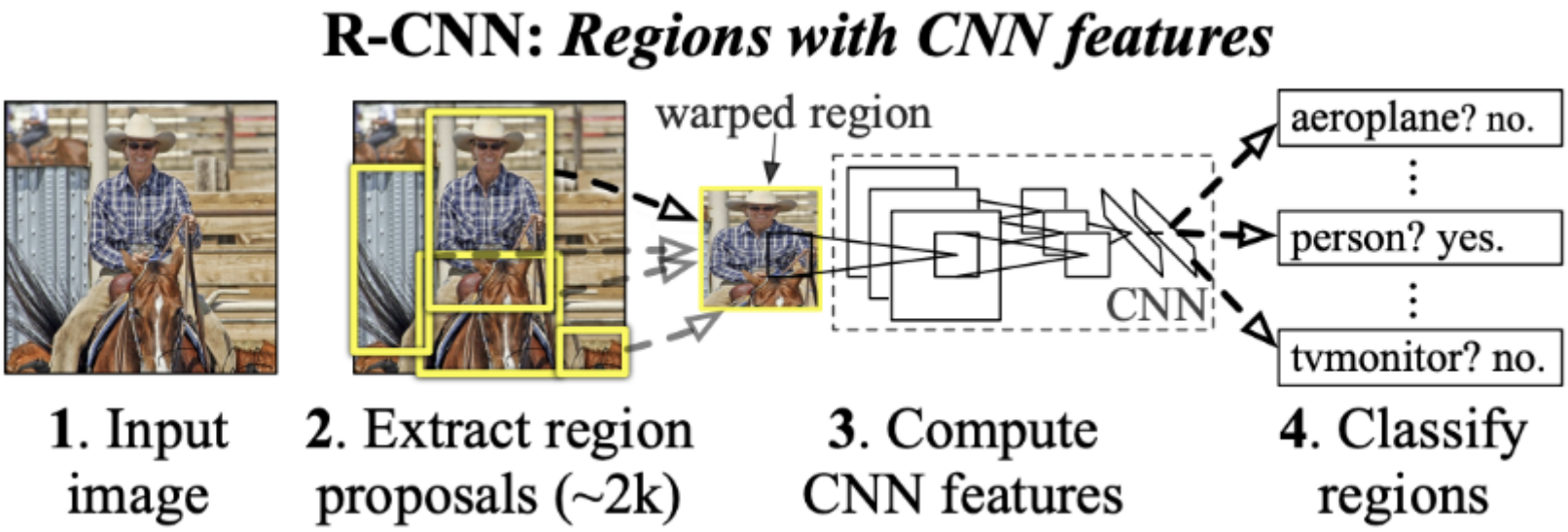
2012년 AlexNet 이 소개되면서 CNN 은 다시 떠오르기 시작했고, ImageNet 이라는 대용량 데이터셋 의 중요성이 화두가 되었다. 논문이 작성되던 당시 저자들의 생각은 다음과 같이 나타난다.

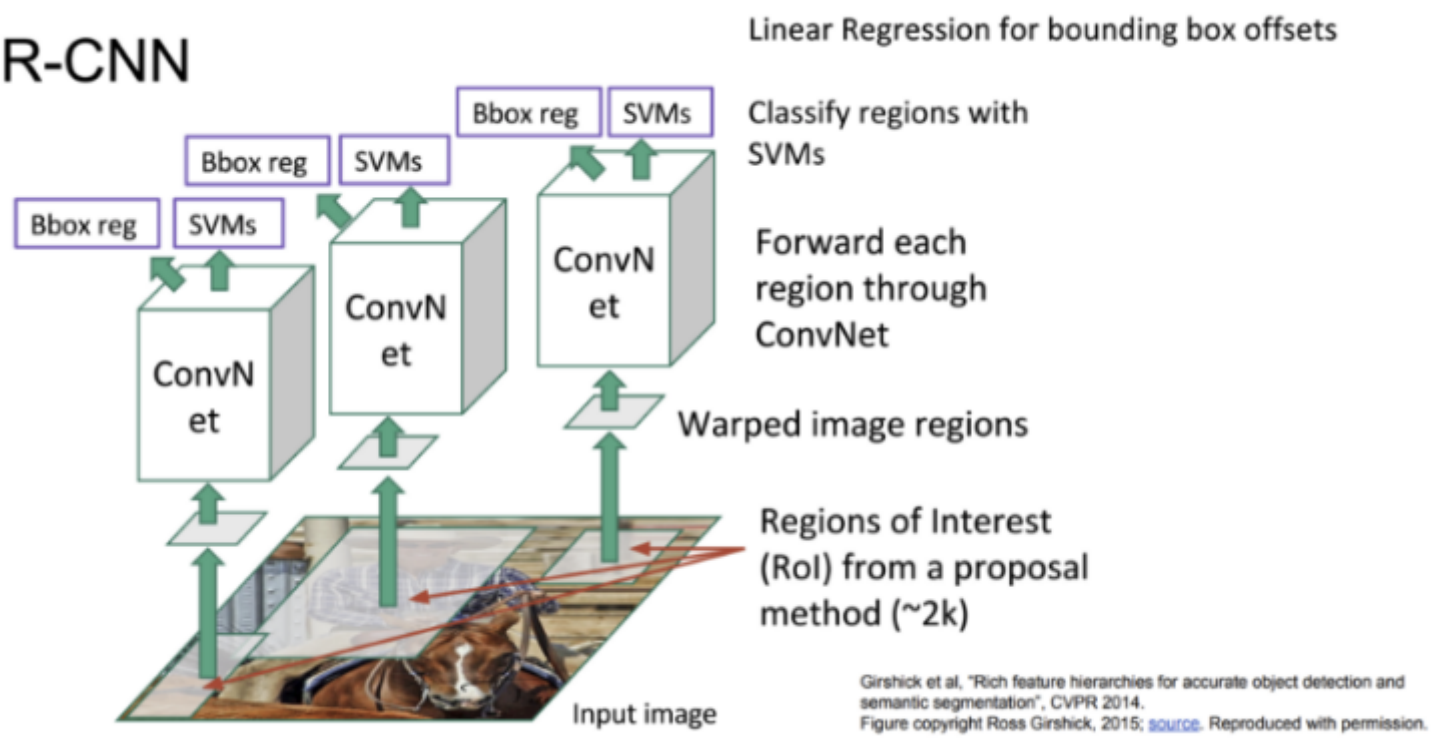
issue can be distilled to the following: To what extent do the CNN classification results on ImageNet generalize to object detection results on the PASCAL VOC Challenge?

We answer this question by bridging the gap between image classification and object detection. This paper is the first to show that a CNN can lead to dramatically higher object detection performance on PASCAL VOC as compared to systems based on simpler HOG-like features. To achieve this result, we focused on two problems: localizing objects with a deep network and training a high-capacity model with only a small quantity of annotated detection data.

CNN 을 Object detection 에 활용하기 위한 저자들의 시도

Instead, we solve the CNN localization problem by operating within the "recognition using regions" paradigm [21], which has been successful for both object detection [39] and semantic segmentation [5]. At test time, our method generates around 2000 category-independent region proposals for the input image, extracts a fixed-length feature vector from each proposal using a CNN, and then classifies each region with category-specific linear SVMs. We use a simple technique (affine image warping) to compute a fixed-size CNN input from each region proposal, regardless of the region's shape. Figure 1 presents an overview of our method and highlights some of our results. Since our system combines region proposals with CNNs, we dub the method R-CNN: Regions with CNN features.





A second challenge faced in detection is that labeled data

is scarce and the amount currently available is insufficient for training a large CNN. The conventional solution to this problem is to use *unsupervised* pre-training, followed by supervised fine-tuning (e.g., [35]). The second principle contribution of this paper is to show that *supervised* pre-training on a large auxiliary dataset (ILSVRC), followed by domain-specific fine-tuning on a small dataset (PASCAL), is an effective paradigm for learning high-capacity CNNs when data is scarce. In our experiments, fine-tuning for detection

R-CNN 이 미완성 모델임에도 의미가 있는 이유.

Our system is also quite efficient. The only class-specific computations are a reasonably small matrix-vector product and greedy non-maximum suppression. This computational

Before developing technical details, we note that because R-CNN operates on regions it is natural to extend it to the task of semantic segmentation. With minor modifications,

2. Object detection with R-CNN

Our object detection system consists of three modules. The first generates category-independent region proposals. These proposals define the set of candidate detections available to our detector. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region. The third module is a set of class-specific linear SVMs. In this section, we present our design decisions for each module, describe their test-time usage, detail how their parameters are learned, and show detection results on PASCAL VOC 2010-12 and on ILSVRC2013.

2.1. Module design

Region Proposals

object 가 있을 법한 영역을 찾는 모듈

- selective search (rule-basoed algorithm)
  - 색상, 질감, 영역 크기 등을 이용
  - 주변 픽셀 간의 유사도를 기준으로 Segmentation 을 만들고, 이를 기준으로 물체가 있을 법한 박스를 추론한다.
  - 작은 영역들을 합쳐서 더 큰 segmented area 를 만든다.
  - 작업을 반복하여 최종적으로 2000 개의 region proposal 을 생성.
  - CNN을 통과해서 나올 때, output 사이즈를 동일하게 만들기 위해, CNN에 넣기 전에 같은 사이즈(224*224)로 wrap 시킨다.

CNN

각각의 영역으로부터 고정된 크기의 Feature Vector 를 뽑아낸다.(고정된 크기의 output 을 얻기 위해 warp 작업을 통해 크기를 찌그러뜨려서 동일 input_size 로 만들고 CNN에 넣는다.)

- CNN 은 AlexNet 의 구조를 거의 그대로 가져왔다.
- CNN을 거쳐 최종적으로 4096 차원의 feature vector 를 뽑아낸다.

논문의 저자들은 ImageNet 데이터셋으로 미리 학습된 CNN 모델을 가져온 다음, fine-tune 하는 방식을사용했다. fine-tune 시에는 실제 object detection 을 적용할 데이터셋에서 ground truth 에 해당하는 이미지드을 거쳐와 학습시켰다.

- 마지막 레이어를 object detection 의 클래스 수 n 과 아무 물체도 없는 배경까지 포함한 n+1로 맞추었다.

SVM

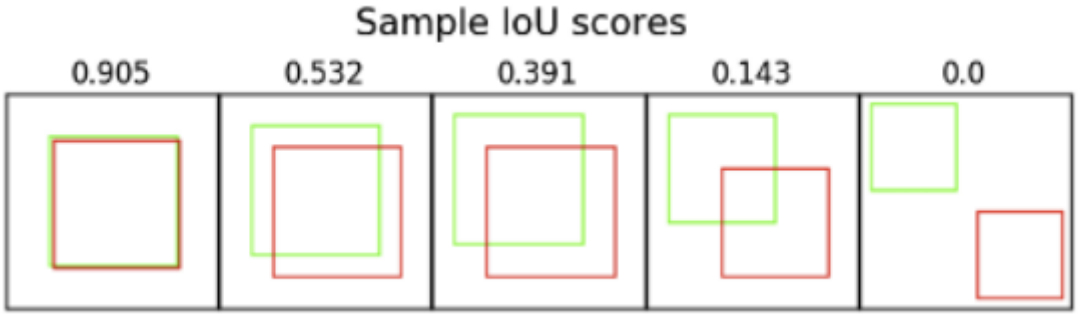
classification 을 위한 선형 지도학습 모델

- CNN 으로부터 feature 가 추출되면 Linear SVM 을 통해 classification 을 진행한다. 논문이 쓰인 당시에는 softmax 보다 SVM 이 더 좋은 성능을 보였기에 SVM을 사용했다.
- SVM 을 붙여서 학습시키는 기법은 더 이상 사용되지 않는다.

Non-Maximum Suppression

동일한 object 에 여러 개의 박스가 있다면, 가장 스코어가 높은 박스만 남기고 나머지는 제거해야 한다. 서로 다른 두 박스가 동일한 물체를 감싸고 있을 때 IoU 를 사용하면 두 박스가 동일한 물체를 가리키고 있음을 판별할 수 있다.

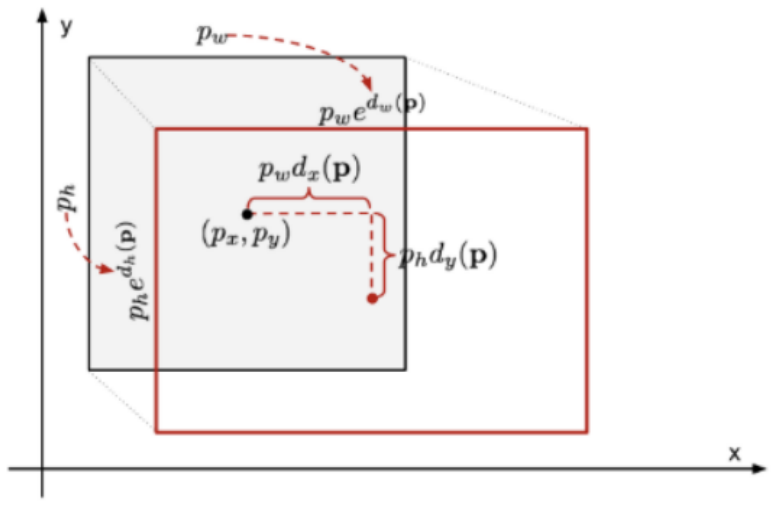
IoU(Intersection over Union) 은 두 박스의 교집합을 합집합으로 나눠준 값이다. 두 박스가 일치할 수록 1에 가까운 값이 나오게 된다.



논문에서는 IoU 가 0.5 보다 크면 동일한 물체를 대상으로 한 박스로 판단했다.

Bounding box regression

selective search 로 만든 bounding box 는 정확하지 않기 때문에 물체를 비교적 정확히 감싸도록 조정해주는 bounding box regression 이 존재한다.



하나의 박스를 나타낼 때, (x, y) 는 이미지의 중심점, (w, h) 는 너비와 높이로 표기할 수 있다.

$$P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$$

Ground truth 에 해당하는 박스도 다음과 같이 표기할 수 있다.

$$G = (G_x, G_y, G_w, G_h).$$

P에 해당하는 박스를 최대한 G에 가깝도록 이동시키는 함수를 학습시키는 것이 목표이고, 박스가 input 으로 들어왔을 때, x,y,w,h 를 각각 이동시켜주는 함수들을 표현하면 다음과 같다.

$$d_x(P),\ d_y(P),\ d_w(P),\ \text{and}\ d_h(P).$$

(x, y) 는 점이기 때문에 이미지의 크기에 상관없이 위치만 이동시켜주면 된다. 너비와 높이는 이미지의 크기에 비례하여 조정해야주어야 한다.

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

우리가 얻고자 하는 함수는 d(input) 함수이다. 논문의 저자들은 해당 함수를 구하기 위해 앞서 CNN 을 통과할 때 pool5 레이어에서 얻어낸 특징 벡터를 사용한다. 그리고 함수에 학습 가능한 weight vector 를 주어 계산한다.

$$d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P)$$

weight 를 학습시킬 loss function 은 MSE 에 L2 norm 을 추가한 형태이다. 논문의 저자들은 λ 를 1000 으로 설정했다.

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2$$

t 는 P를 G로 이동시키기 위해서 필요한 이동량을 의미한다.

$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$
$$t_w = \log(G_w/P_w)$$
$$t_h = \log(G_h/P_h).$$