R-CNN

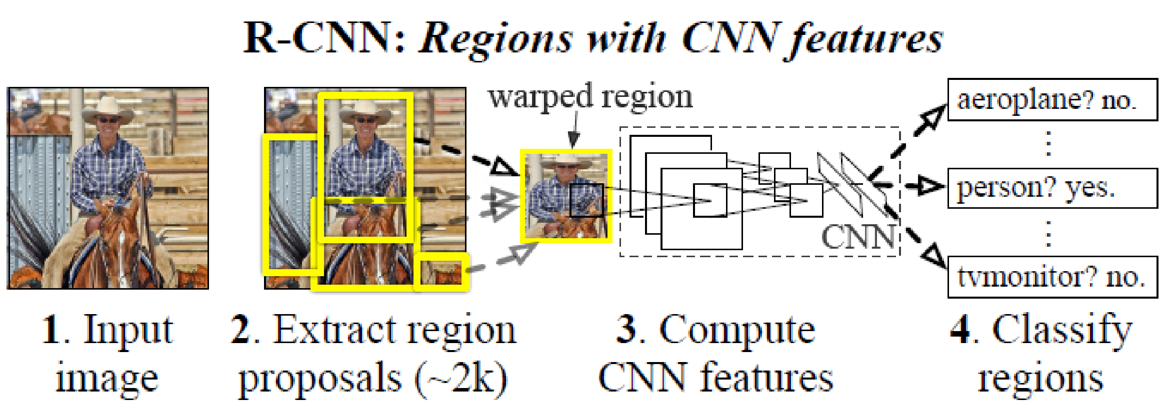
INTRODUCTION

R-CNN or Regions with CNN leverage the use of region proposals along with Convolutional Neural Networks for object detection.

Their approach combines two key insights:

1. Using CNNs to bottom-up region proposals in order to localize and segment objects.
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.

OBJECT DETECTION OVERVIEW



(1) Take an input image.

(2) Extracts around 2000 bottom-up region proposals.

(3) Computes features for each proposal using a large

convolutional neural network (CNN).

(4) Classifies each region using class-specific linear SVMs.

Feature Extraction-

1. A 4096-dimensional feature vector from each region proposal is extracted.
2. Features are computed by forward propagating a mean-subtracted 227*227 RGB image through five convolutional layers and two fully connected layers.
3. Regardless of the size or aspect ratio of the candidate region, all pixels are warped in a tight bounding box around it to the required size.

Test time detection-

At test time all the regions of interest are scored as described above. Given all scored regions in an image, a greedy non-maximum suppression (for each class independently) is applied that rejects a region if it has an IOU overlap with a higher scoring selected region larger than a learned threshold.

Training:

Supervised Pre Training: The CNN is trained on a larger auxiliary dataset (ILSVRC2012 classification) using image-level annotations only.

Domain Specific Fine Tuning:

1. SGD training of the CNN parameters using only warped region proposals.
2. The CNNs ImageNet specific 1000 way classification layer is replaced with (N+1) way classification layer(N is number of object classes plus 1 for background).
3. All region proposals with IoU greater than 0.5 overlap with a ground truth box are treated as positives for that box's class and rest as negatives.
4. Learning rate of 0.001.
5. In each SGD iteration, 32 positive windows (over all classes) and 96 background windows to construct a mini-batch of size 128 were sampled.

Object category classifiers

1. To deal with partially overlapping bounding boxes a IoU threshold of 0.3 was selected below which regions are classified as negative
2. Once features are extracted and training labels are applied, one linear SVM per class is optimized.

Object proposal transformations:

(A) (B) (C) (D) (A) (B) (C) (D)

Figure 7: Different object proposal transformations. (A) the original object proposal at its actual scale relative to the transformed CNN inputs; (B) tightest square with context; (C) tightest square without context; (D) warp. Within each column and example proposal, the top row corresponds to $p = 0$ pixels of context padding while the bottom row has $p = 16$ pixels of context padding.

Bounding Box Regression:

$$\boxed{\begin{aligned} P^i &= (P^i_x, P^i_y, P^i_w, P^i_h) \\ G &= (G_x, G_y, G_w, G_h) \end{aligned}} \quad (1)$$

$$\begin{aligned} t_x &= (G_x - P_x)/P_w \\ t_y &= (G_y - P_y)/P_h \\ t_w &= \log(G_w/P_w) \\ t_h &= \log(G_h/P_h). \end{aligned} \quad (2)$$

$$\begin{aligned} \hat{G}_x &= P_w d_x(P) + P_x \\ \hat{G}_y &= P_h d_y(P) + P_y \\ \hat{G}_w &= P_w \exp(d_w(P)) \\ \hat{G}_h &= P_h \exp(d_h(P)). \end{aligned} \quad (3)$$

$$\boxed{d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \phi_5(P)}$$

$$\mathbf{w}_\star = \operatorname*{argmin}_{\hat{\mathbf{w}}_\star} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2 \quad (4)$$

- Predicted proposal-*P*, Target proposal- *G*. *x, y, w,* and *h* stand for the coordinates of the center *(x, y)* and the width *w* and height *h* of the proposal.
- Transformations learned for ground truth shown in equation 2. The first two transformations specify a scale-invariant translation of the center of *P* — *x* and *y*, and the second two specify log space transformations of the width *w* and height *h*.
- $d_{\square}(P)$ -is the predicted transformation. Ĝ signifies the corrected predicted box calculated using the original predicted box *P* and the predicted transformation $d_{\square}(P)$.
- The predicted transformation $d_{\square}$(P) is modeled as a linear function of the $pool_5$ features — Φ_5. Hence, $d_{\square}(P) = w_{\square}^{T} \Phi_5(P)$ where $w_{\square}$ is the vector of learnable model parameters.

RESULTS:

This paper presents a simple and scalable object detection algorithm that gives a 30% relative improvement over the best previous results on PASCAL VOC 2012.