Faster R-CNN

Comment

It achieves *translation-invariance*. It can perform real-time detection. It is not clearly a priori in terms of the convergence of shared (of RPN and Fast R-CNN) network

Abstract

Region Proposal Network (RPN) shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. FC network simultaneously predicts object bounds and objectness scores at each position. It can detect in 5-17 fps and gains 70.4% mAP using 300 proposals per image.

1. Introduction

- Region proposals are the computational bottleneck. In Faster R-CNN, the proposal computation is nearly cost-free given network's computation.
- Like Fast R-CNN, RPNs is constructed by adding two additional conv layers
  - one that encodes each conv map position into a short feature vector
  - one that, at teach conv map position, outputs an objectness score and regressed bounds for k region proposals relative to various scales and aspect ratio (practically, $k = 9$)
- To unify RPNs with Fast R-CNN object detection networks, it alternates between fine-tuning for the region proposal task and then fine-tuning for object detection, while keeping the proposals fixed.

2. Related Work

- OverFeat method predicts the box coordinates for the localization task that assumes a single object. The fc layer detects multiple class-specific objects.
- MultiBox methods generate region proposals which are simultaneously predicted with multiple (e.g, 800) boxes.
- Shared computation is widely used for efficient, yet accurate visual recognition.

3. Region Proposal Networks

- It takes an image of any size as input and outputs a set of rectangular object proposals with objectness score.
- The ultimate goal is to share compuation with a Fast R-CNN object detection network
- A box-regression layer *(reg)* and a box-classification layer *(cls)*

Translation-Invariant Anchors

- At each sliding-window, sized $n \times n$ (e.g $n = 3$), it simultaneously predict k region proposals. *reg* has $4k$ outputs, *cls* does $2k$ scores of objectness / non-objectness for each proposal.
- The k proposals are parameterized *relative* to k reference boxes, called ***anchors***.
- For a conv feature map of size $W \times H$ (typically ~2,400) there are WHk anchors in total.
- Practically, it uses 3 scales and 3 aspect ratios, yielding $k = 9$ anchors at each sliding position.
- Both in terms of the anchors and the functions, it is ***translation invariant***, computing proposals relative to the *anchors*.

A Loss Function for Learning Region Proposals

- It assigns a positive label to two kinds of anchors:
  - the anchor(s) with the highest IoU with a ground-truth box
  - an anchor that has IoU higher than 0.7 with any ground-truth box
- It assigns a negative label to non-positive anchor if IoU is lower than 0.3 for all ground-truth boxes.
- This implies there can be neither positive nor negative anchors which do not contribute to the training objective at all.
- The loss function for an image is,

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i*)$$

  where i is the index of an anchor in a mini-batch and p_i is the predicted probabilty of anchor i being an object, the ground-truth label p_i^* is 1 if the anchor is positive, and is 0 if the anchor is negative, t_i is predicted bounding box, and t_i^* is the ground-truth box, $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ where R is the robust loss function.
- For regression,

$$\begin{bmatrix} t_x & t_y & t_w & t_h \\ t_x^* & t_y^* & t_w^* & t_h^* \end{bmatrix} = \begin{bmatrix} (x - x_a)/w_a & (y - y_a)/h_a & \log(w/w_a) & \log(h/h_a) \\ (x^* - x_a)/w_a & (y^* - y_a)/h_a & \log(w^*/w_a) & \log(h^*/h_a) \end{bmatrix}$$

- In previous works, a bounding-box regression is performed on features pooled from *arbitrarily* sized regions, and the regression weights are *shared* by all region sizes.
- In Faster R-CNNm, the feature has the *same* spatial size ($n \times n$) on the feature maps. To account for varying sizes, a set of k bounding-box regressors are learned. Each regressor is responsible for one scale and one aspect ratio, and the k regressors do not share weights. It is still possible to predict boxes of various sizes even though the features are of a fixed size/scale.

Optimization

The RPN is likely to bias towards negative samples as they are dominate, it randomly samples 256 mini-batches with keeping the ratio of *up to* $1:1$ (pos:neg)

Sharing Conv Features for Region Proposals and Object Detection

- It is not easy to define a single network that includes both RPN and Fast R-CNN and to optimize it jointly with backpropagation. Neither is it clearly a priori if learning Fast R-CNN while simultaneously changing the proposal mechanism will converge.
- Here is pragmatic 4-step training algorithm via alternating optimization
    1. Train RPN as described above, initialized with an ImageNet pre-trained model and fine-tuned end-to-end for the proposal.
    2. Train a separate detection network by Fast R-CNN using the proposals generated by step-1 RPN. It is also initialzed with the pre-trained from step-1.
    3. Use detection network to initialize RPN training, but fixing the sharable conv layers and only fine-tuning the unique layers of RPN.
    4. Keeping the shared conv layers fixed, fine-tune the fc layers of the Fast R-CNN.