Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

1 Citation

TODO: Put citation here.

https://www.cv-foundation.org/openaccess/content$_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper$

2 Abstract

We get 53.3% mean average precision on PASCAL VOC object detection by building a three stage system (1) propose regions, (2) extract features with a pretrained (on ImageNet) AlexNet CNN fine-tuned to PASCAL VOC, (3) SVM classifier to label extracted features.

3 Introduction

SIFT and HOG are good features with analogs in the human visual system. However, they don't have hierarchical layers like the human visual system. Convolutional Neural Networks (CNNs) have been dominating ImageNet, so maybe we can use them for object detection.

Solving object detection as a regression problem doesn't usually work and using a sliding window classifier has its own issues (e.g. need to find object boundaries, need to handle multiple scales). Our approach basically uses a class-independent region proposal system, a CNN for feature extraction, and an SVM to classify. Since training data is scarce, we use a CNN pretrained on ImageNet.

What makes the CNN good? It's not the fully-connected layers because you can get rid of like 94% of the weights in them and still get good performance. The answer is the convolutional layers.

4 Object Recognition with R-CNN

There are many region proposal algorithms - pick whatever you like. We use selective search because that let's us compare against previous work.

For the CNN, we use AlexNet pretrained on ImageNet. Given a region proposal, we get a bounding box (with some padding), warp it to fit the CNN's input size, and subtract the mean pixel activation (over training set).

At test-time, we get like 2000 region proposals, use the warping process above, feed them into the CNN, and feed the resulting feature vectors into the SVM. This gives a class label for each region. We apply greedy non-maxima suppression for each class independently where we reject a region if its intersection-over-union with a higher-scored region exceeds a chosen threshold.

To process an image, it takes about 13 seconds on a GPU and 53 seconds on a CPU.

Let's look at the training process. Given a pretrained AlexNet, we replace the 1000-way softmax with a 21-way softmax (for PASCAL VOC). To get the ground truth, we consider each region proposal for an image. If the proposal overlaps a ground truth bounding box (from PASCAL VOC) by ≥ 0.5 intersection-over-union, then it's a positive example for the bounding box class. Otherwise it's background. For stochastic gradient descent, we make minibatches with 32 positive windows and 96 background windows (biasing towards positive examples because they are rare).

Now that we have the fine-tuned AlexNet, let's move on to the SVM. There's a reason we use an SVM instead of the CNN classifier output and there's a reason we select SVM ground truths differently than CNN ground truths (not discussed in this paper). To pick ground truths, we examine a region and see if it overlaps with a ground truth bounding box by at least 0.3 intersection-over- union (this threshold was picked with grid search cross validation and it's pretty sensitive, so don't mess with it) - if so, it's positive for the class. Otherwise it's background.

Comparing against PASCAL VOC state of the art, we get a 30% relative improvement in mean average precision.

5 Visualization, ablation, and modes of error

First layer filters are easy to visualize - they find edges and opponent colors.

For the other units, we take this approach. We feed lots of regions through them and see which regions each unit fires for. For each unit, we examine the regions that produced the strongest activation. For pooling layer 5, which we call pool_5, we found units that detect faces, red blobs, dogs/holes, text on a sign, and lens flare.

Fine-tuning boosts mAP by 8%. Most of this gain comes from the fully connected layers, not the pool_5 layer, which indicates that CNNs pretrained on ImageNet generalize well to other tasks.

We measure how often the mistake was caused by poor localization, confusion with a similar category, confusion with non-similar category, or background. See supplementary materials.

We also trained a linear regression model that uses the pool_5 features to predict a new bounding box and we found it boosts mAP by like 5%.

6 Semantic Segmentation

To see how our system does on PASCAL VOC segmentation, we use the CMPC algorithm to generate regions (state of the art does thhis). To classify, we consider two approaches. First, we warp the region as described before and extract features with CNN. Second, we only consider foreground pixels (setting background to the mean so they become zero after mean subtraction) and extract features with CNN. If you then stack the features from both approaches, then you get best performance. We then train 20 (one per class, excludes background) support-vector-regressors to compute a score for each class (we did this because state of the art does this). We beat state of the art for about half classes and basically tie on the rest.

7 Conclusion

We found that region proposals + fine-tuned pretrained CNN + SVM does really well at object detection.