Fast R-CNN

Link

arXiv

Summary

- Fast R-CNN improves upon R-CNN network for object detection by increasing both detection accuracy (in terms of mean average precision) and reducing training and inference time. They replace the multi-stage training pipeline in R-CNN that consists of CNN feature extractor, SVM classifier and bounding box regression with a single CNN that jointly learned to classify object proposals and refine their spatial positions. While R-CNN extracted feature for each object proposals in an image this network extracts features only once per image.
- To produce fixed size feature vector for each region proposals this paper introduced region of interest (RoI) pooling layer. RoI max pooling works by dividing the proposal region window of size $h \times w$ into a fixed sized $H \times W$ grid of sub-windows of approximately equal size and taking maximum in each sub-window into the corresponding output grid cell. So this is just like max pooling but here we want a fixed size output and so pool window size is varied accordingly. Since the gradients are back-propagated through the RoI pooling layer, all network weights can be trained with SGD. So fine-tuning is possible convolution layer weights
- The Fast R-CNN network has two sibling output layers, the first produces probability distribution over $k+1$ categories (k object and 1 background) and the second is bounding box regression offsets for each of k object classes. The loss of those two outputs are combined into single multi-task loss and used in back propagation.
- The fully connected layer can be truncated using SVD which reduces detection time by more than 30% with only a slight drop in accuracy.
- Multi-task training with both classification and bounding box regressors preform better than training them separately and combining.
- Single scale detection performs almost as well as multi scale detection suggesting that deep CNNs are adept at learning scale invariant features.
- Softmax classifier used in Fast R-CNN performs better than using a separate SVM classifier like in R-CNN.
- Sparse proposals produced by selective search performs better than dense search. Even when increasing number of proposals produced by selective search performance does not always increase, sometimes it also decreases which is surprising.