FAST R-CNN

Paper summary

This paper proposes a Fast Region-based CNN (FAST R-CNN) method for object detection. Fast R-CNN build on previous work to efficiently classify object proposals using deep convolutional network. Compared to image classification, object detection is a more challenging task that requires more complex methods to solve. R-CNN and SPPnet archives excellent object detection accuracy, however it has notable drawbacks. R-CNN follows a multi stage training pipeline. It trains its modules separately. R-CNN first fine tune a ConvNet using Log loss then it fits SVMs to ConvNet features and Box Regressor for bounding box regression. For SVM and bounding box regressor training, extracted features are written to a disk. These required hundreds of gigabytes of storage. For that reason R-CNN is expensive in space, time and also in speed. Because it performs forward pass for each object proposal without sharing computation. SPPnet also has drawbacks like R-CNN. Unlike R-CNN the fine tuning algorithm proposed in SPPnet cannot update the convolutional layer that precedes the spatial pyramids pooling. This limits the accuracy of very deep networks. The authors of this paper come up with an approach to solve these problems.

Model Architecture and working process :

Fast R-CNN is an enhanced version of the R-CNN model. The Fast R-CNN model consists of a CNN with its final pooling layer replaced by “ROI pooling” layer and its final FC layer is replaced by two branches –a(K+1) category softmax layer branch and a category-specific bounding box regression branch. Instead of training different modules separately the modules are trained in an end-to-end manner. CNN for extract features , SVM for classification and Box regressor for bounding box regression are trained as a single network.

First instead of applying convolutional on each object proposal CNN is first applied only once on the image. Once image features are extracted using the CNN in the ROI feature extraction is applied on the feature map using the object region proposals. Once every feature vector is extracted for each object proposal region the vectors are fed into a sequence of FC layers and produce two branches of outputs. A single branch is used to output a class of object and another branch is used to produce four dimensional output where each set of four values here encodes refined bounding box positions of one of the classes. The RoI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of H × W, where H and W are layer hyper-parameters that are independent of any particular RoI. Fast R-CNN network undergoes three transformations. First, the last max pooling layer is replaced by a RoI pooling layer that is configured by setting H and W to be compatible with the net’s first fully connected layer(for VGG16). Second, the network’s last fully connected layer and softmax are replaced with the two sibling layers described earlier. Third, the network is modified to take two data inputs: a list of images and a list of RoIs in those images.

In Fast R-CNN training, stochastic gradient descent (SGD) mini batches are sampled hierarchically, first by sampling N images and then by sampling R/N RoIs from each image. Critically, RoIs from the same image share computation and memory in the forward and backward passes. In this paper authors explore two ways of achieving scale invariant object detection: (1) via “brute force” learning and (2) by using image pyramids. During multi-scale training, they randomly sample a pyramid scale each time an image is sampled as a form of data augmentation. Authors used truncated SVD for fast detection.Truncated SVD reduces the parameter count and this simple compression method gives good speedups when the number of RoIs is large.

Results:

Fast R-CNN achieved state of the art performance at the time of its publication on VOC12, VOC10 and VOC 07. Fast R-CNN processes images 45x faster than R-CNN at test time and 9x faster at train time. It also trains 2.7x faster and runs test images 7x faster than SPPnet. Using truncated SVD the detection time reduces by more than 30% with just a 0.3 drop in mAP.

Multi task training improves from +0.8 to +1.1 mAP points, showing a consistent positive effect

Contributions:

1. Higher detection quality (mAP) than R-CNN, SPPnet.
2. Training in single stage, using a multi task loss,
3. Training can update all network layers,
4. No disk required.

Limitations:

1. SGD mini batch strategy may cause slow training convergence because RoIs from the same image are correlated.
2. The multi-scale approach offers only a small increase in mAP at a large cost in compute time
3. It depends on Selective Search to generate region proposals. Selective search cannot be customized on a specific object detection task and it may not detect all objects in the dataset.