FAST R-CNN
(REGION-BASED CONVOLUTIONAL NETWORK)

WRITTEN BY ROSS GIRSHICK
PRESENTED BY MINSUNG CHO
TABLE OF CONTENTS

• Short Summary
• Assessment of Strengths
• Assessment of Weaknesses
• Questions Regarding the Work
SHORT SUMMARY

• The problem that is being addressed:
  • Training is a multi-stage pipeline
  • Training is expensive in space and time
  • Object detection is slow.

• Its significance:
  • 9x faster than R-CNN (0.3s to process images at runtime)
  • Higher mAP of 66% (vs. 62% for R-CNN)

• Approach used to solve the problems:
  • The RoI pooling layer
  • Initializing from pre-trained networks
  • Fine-tuning

• Setup:
  • CaffeNet with Python and C++
  • VOC07, 2010, and 2012 vs SPPNet, R-CNN, SegDeepM, and BabyLearning
WHAT IS CNN?

• Convolutional Neural Network
• Input -> Convolutional layer (transformed) -> output
• Features
  • Objects, textures, edges, shapes, etc.
• Layers = Filters
• Simple layers(edge 3x3) -> intermediate layers(corners, shapes) -> complicated layers(dog, cat)
WHAT IS RELU (RECTIFIED LINEAR UNIT) ACTIVATION FUNCTION AND BACKPROPAGATION?

• First, neural network consists of nodes and connections between the nodes that provides the weights and biases.

• Back propagation fills the empty parameters -> slope and intercept

• Bent activation node, zero until a point, rest is linear.

• Chain rule and gradient descent to provide the weights and biases for backpropagation.

• First determine the last parameter bias by gradient descent and with optimal past parameters.

• Then use derivative of SSR over the last parameter bias.
WHAT IS MAX POOLING & SOFTMAX

• Follows the convolutional layer output.
• Decreases the image pixel width and height.
• Defined by filter size (like 2x2) and stride (how many pixel to slide).

• SoftMax = higher dimension sigmoid.
• Goal is to transform the unbounded probability into a probability vector.
• It uses euler’s exponent.
WHAT IS R-CNN?

• Input image -> selective search -> initial segmentation -> after many iteration -> bounding box -> into the CNN

• Extract region proposal from bounding box -> crop -> CNN -> classification

• Problem addressed:
  Region loses detail when cropped to fit into the fixed-sized CNN.
  There could be too much bounding box to pass to the CNN and they have to pass one by one

• Significance of the problems:
  • Computationally expensive.
FAST R-CNN

• Approach used to solve the problems:
  • Still does selective search region proposals.
  • Passes the entire image into the deep ConvNet other than each RoI -> the bounding box is projected onto the RoI feature map -> RoI pooling layer makes the projection into a fixed size -> Fully connected network classification CNN

• Because CNN is just layers of filters and max pooling, it can estimate at the end of the image output.

• There may exist yet undiscovered techniques that allow dense boxes to perform as well as sparse proposals (wink wink)
FAST R-CNN

Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.
R-CNN VS FAST R-CNN
FINE TUNING

• Uses Stochastic gradient descent (just like regular gradient descent but random and better to update) for mini-batch sampling (25% of the RoIs).

• 2 output layers:
  • Discrete probability distribution per RoI over K+1 categories.
  • Bounding-box regression offset for each of the K object classes.

• Back propagation routes derivatives through the RoI pooling layer.

• SGD hyper-parameter: softmax & bounding-box initialized from zero-mean Gaussian distribution with standard deviations 0.01 and 0.001.

• Using both brute force(single scale) and image pyramid for scale invariance
ASSESSMENT OF STRENGTHS

• Higher detection quality (mAP) than R-CNN, SPPNet
• Training is single-stage, using a multi-task loss
• Training can update all network layers
• No disk storage is required for feature caching
• softmax slightly outperforming SVM
TRAINING AND TESTING TIME

- For VGG16, Fast R-CNN processes images 146× faster than R-CNN without truncated SVD and 213× faster with it.
- Training time is reduced by 9×, from 84 hours to 9.5.
- Compared to SPPnet, Fast RCNN trains VGG16 2.7× faster (in 9.5 vs. 25.5 hours) and tests 7× faster without truncated SVD or 10× faster with it.
- Fast R-CNN also eliminates hundreds of gigabytes of disk storage, because it does not cache features.
ASSESSMENT OF WEAKNESSES

• Assumptions that are used:
  • Brute force (single-scale)
  • Multi-task training
  • Training data size

• When the algorithm might fail:
  • Scale invariance - Larger networks with a single-scale offer the best speed / accuracy tradeoff.
  • The multi-scale approach offers only a small increase in mAP at a large cost in compute time.
  • Across all three networks we observe that multi-task training improves pure classification accuracy relative to training for classification alone.
  • Enlarging the training set improves mAP.
  • + Problems mentioned before
FASTER R-CNN (THE NEXT GEN)

- The entire image is fed through the convolutional layers for feature extraction just like the Fast R-CNN.
- The difference: the projection on the feature maps is not from the selective cells -> Uses RPN (Region Proposal Network) after the CNN output.
- Uses anchor and its intersection of union ratio 7:3 to determine the object bounding box.
- No selective cells -> the feature map -> passed into the RPN CNN -> gets object classification, bounding box, and whether an object is present -> RoI pooling and classifier.
QUESTIONS REGARDING THE WORK

• Later works that replaced Fast R-CNN:
  • Faster R-CNN -> Mask R-CNN Detectron(Facebook)
  • YOLO (You only look once)
  • Single Shot Detectors(SSD)