@Oct 7, 2020 Assignment

Status	Completed
Student ID	201950853
Student Name	JURAKUZIEV DADAJON BOYKUZI UGLI
Subject	Vision Intelligence

Summarize the difference among the networks of RCNN family.

- Summarize the difference of RCNN family

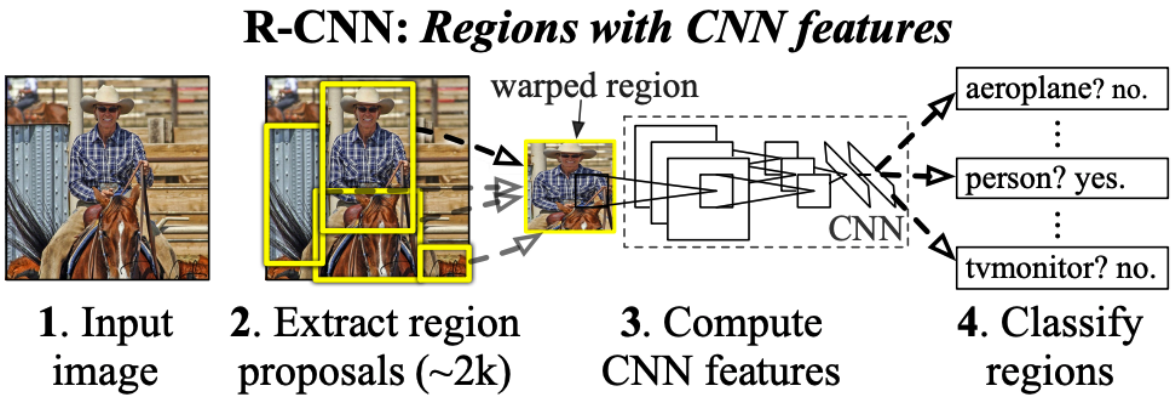
  RCNN → Fast-RCNN → Faster-RCNN → Mask-RCNN

RCNN

Author: Ross Girshick et al., 2014

Paper: Rich feature hierarchies for accurate object detection and semantic segmentation

The main idea is composed of two steps. First, using selective search, it identifies a manageable number of bounding-box object region candidates ("region of interest" or "RoI"). And then it extracts CNN features from each region independently for classification.

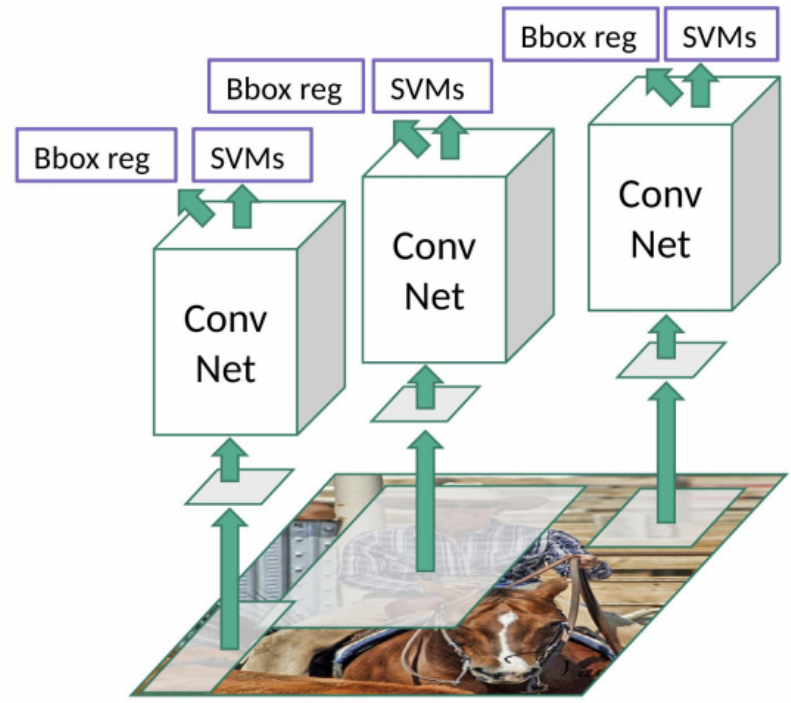


Model Workflow

How R-CNN works can be summarized as follows:

1. **Pre-train** a CNN network on image classification tasks; for example, VGG or ResNet trained on ImageNet dataset. The classification task involves N classes.

2. Propose category-independent regions of interest by selective search (~2k candidates per image). Those regions may contain target objects and they are of different sizes.
3. Region candidates are **warped** to have a fixed size as required by CNN.
4. Continue fine-tuning the CNN on warped proposal regions for K + 1 classes; The additional one class refers to the background (no object of interest). In the fine-tuning stage, we should use a much smaller learning rate and the mini-batch oversamples the positive cases because most proposed regions are just background.
5. Given every image region, one forward propagation through the CNN generates a feature vector. This feature vector is then consumed by a **binary SVM** trained for **each class** independently.The positive samples are proposed regions with IoU (intersection over union) overlap threshold >= 0.3, and negative samples are irrelevant others.
6. To reduce the localization errors, a regression model is trained to correct the predicted detection window on bounding box correction offset using CNN features.



Performance:

- It takes 2s to generate region proposals.
- The time spent computing region proposals and features is 13s/image on a GPU or 53s/image on CPU.

Architecture:

The system introduced in the paper consists of 3 modules.

1. **Category-independent region proposals generator** defines the set of candidate detections available to detector.

2. **CNN** extracts a fixed-length feature vector from each region.
3. **Set of class-specific linear SVMs** classify regions.

Implementation details:

- The main results are obtained using AlexNet, and compared to the results obtained using VGG (referred as O-Net). VGG outperforms AlexNet, but is about 7 times slower.
- Features are computed by forward propagating a mean-subtracted 227×227 RGB image through five-convolutional layers and 2 fully-connected layers. The feature vector has 4096 dimensions.
- ~2000 candidate regions are generated using Selective Search algorithm.
- Negative examples are built using IoU overlap threshold equal to 0.3. The threshold was selected by a grid search over {0, 0.1, ..., 0.5}.
- Affine image warping is used to compute a fixed-size CNN input from each region proposal, regardless of the region shape.

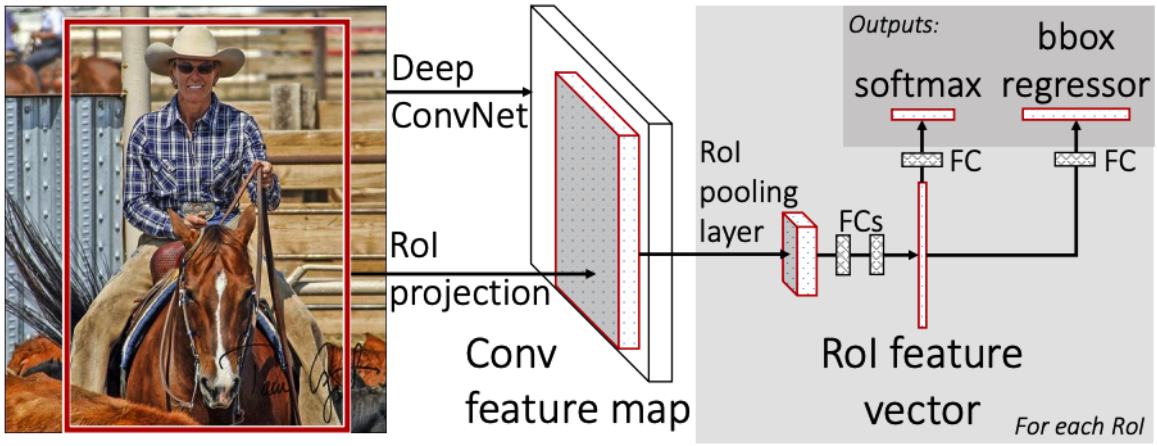
Limitations:

- Training is a multi-stage pipeline: (1) fine-tune ConvNet on object proposals; (2) fit SVMs to ConvNet features; (3) learn bounding-box regressors.
- Training is expensive in space and time. It is required to write to disk all features extracted from each object proposal in each image to train SVM and bounding-box regressors.
- Object detection is slow, because it performs a ConvNet forward pass for each object proposal, without sharing computations.
- Wrapped content of object proposals may result in unwanted geometric distortion, and as the result, the recognition accuracy can be compromised.

Fast-RCNN

Author: Ross Girshick, 2015

Paper: Fast R-CNN



- Given candidate regions, Fast R-CNN is trained in single stage. It jointly leans to classify object proposals and refine their spatial locations using **multi-task** loss.
- Training can update all network layers.
- No disk storage is required for feature caching.
- Similar to SPP-net, Fast R-CNN computes a convolutional feature map for the entire input image and then classifies each object proposal using a feature vector extracted from the shared feature map.
- According to the results, single-scale detection performs almost as well as multi-scale detection, and ConvNets are adept at directly learning scale invariance. The multi-scale approach offers only a small increase in mAP at a large cost in compute time.

Performance:

- Trains VGG16 network 9x faster than R-CNN, is 213x faster at test time, and achieves a higher mAP on PASCAL VOC 2012.
- Trains VGG16 network 3x faster than SPPnet, is 10x faster at test time, and is more accurate.

Architecture:

- Fast R-CNN takes as input an entire image and a set of object proposals.
- It processes the whole image with several convolutional and max-pooling layers to produce a feature map.
- For each object proposal RoI pooling layer extracts a fixed-length feature vector from the feature map.
- Each feature vector is fed into a sequence of fully connected layers that finally branch into two sibling output layers. One layer produces softmax probability estimates over **K** object classes plus a catch-all “background” class. The other layer outputs four real-valued number encoding refined bounding-box positions for each of the **K** object classes.

Implementation details:

- The RoI layer is simply the special-case of the spatial pyramid pooling layer used in SPP-nets in which there is only one pyramid level.
- To enable fine-tuning of all network layers, SGD mini-batches are sampled hierarchically, first by sampling **N** images and then by sampling **R/N** RoIs from each image. In the experiments R=128, N=2.
- In one batch, 25% of the RoIs are labeled with foreground object class. They are uniformly sampled from object proposals that have IoU overlap with a ground truth bounding box of at least 0.5.
- The remaining RoIs are background examples. They are sampled from object proposals that have a maximum IoU with ground truth in the interval [0.1, 0.5).
- During training, images are horizontally flipped with probability 0.5.
- The ground truth regression targets are normalised to have zero mean and unit variance, and the hyper-parameter controlling the balance between the two task losses is set to 1.
- SVD is used to reduce the time spend computing the fully connected layers.

Limitations:

- Still requires candidate regions as input.
- The running time of Fast R-CNN is reduced, exposing region proposal computation as a bottleneck.

Faster-RCNN

Author: Shaoqing Ren et al., 2016

Paper: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

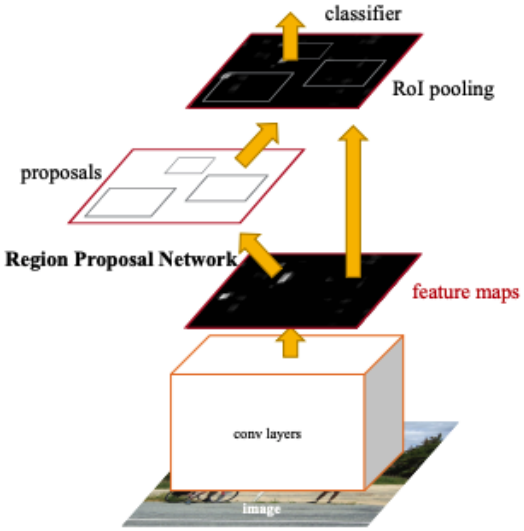


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

- Introduces :
  - a Region Proposal Network (RPN) that enables nearly cost-free region proposals.
  - Classification loss
- RPN is a fully-convolutional network that simultaneously predicts object bounds and objectness scores at each position. Objectness measures membership to a set of object classes vs. background.
- RPNs share convolutional layers with object detection networks.
- RPNs are trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection.

Performance:

- The marginal cost for computing proposals is small, e.g. 10ms per image.
- Using VGG, the detection works at 5fps frame rate (including all steps) on a GPU, using ZF – at 17fps.

Architecture:

- Convolutional feature maps are used for generating region proposals.
- On top of the conv features, RPNs are constructed by adding two additional conv layers.
- The first layer encods each conv map position into a short (e.g. 256-d) feature vector.
- The second layer at each conv map position outputs an objectness score and regressed bounds for k region proposals relative to various scales and aspect ratios at that

location (**k** = 9 is a typical value).

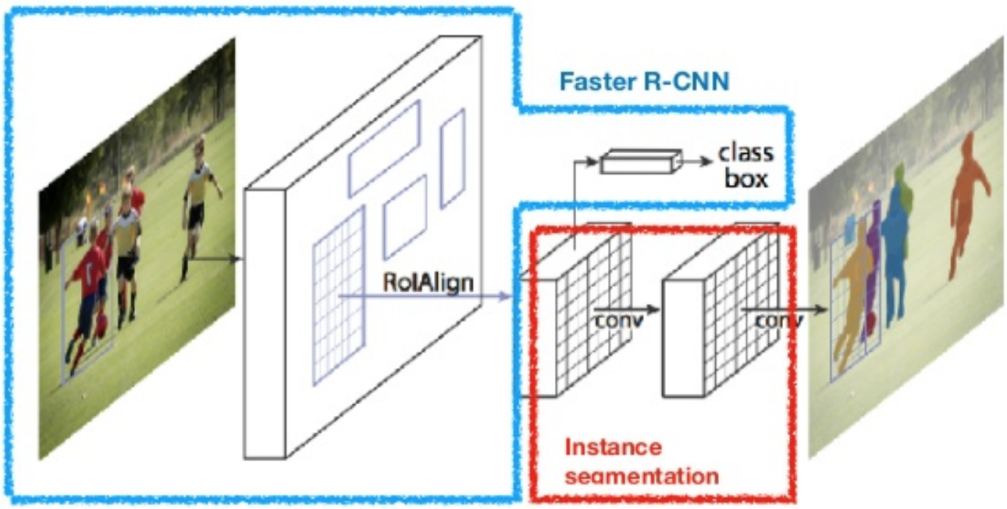
Implementation details

- At the training stage, alternate between fine-tuning for the region proposal task and then fine-tuning for object-detection, while keeping the proposals fixed.
- The first conv layer is 3×3 followed by ReLu, and the second conv layer is 1×1.
- Positive labels are assigned to two kinds of anchors: (1) the anchor(s) with the highest IoU overlap with a ground-truth box, (2) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box.
- Negative labels are assigned to a non-positive anchor if its IoU is lower than 0.3 for all ground truth boxes.
- Anchors that are neither positive or negative do not contribute to the training objective.
- Each mini-batch arises from a single image that contains many positive and negative anchors. In total 256 anchors are sampled. The ratio of positive and negative anchors in one mini-batch is 1:1. If there is not enough positive samples in an image, the batch is padded with negative samples.

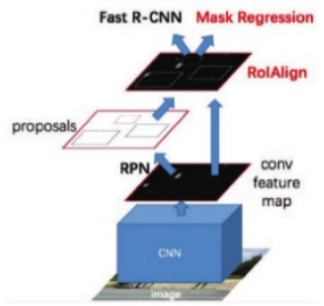
Mask-RCNN

Author: Kaiming He et al., 2018

Paper: Mask R-CNN



- Extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.



- Adds only a small overhead to Faster R-CNN, running at 5 fps.
- Easy to generalise to other tasks, e.g. to estimate human poses in the same framework.
- The network is allowed to generate masks for every class without competition among classes. The dedicated classification branch predicts class label used to select the output mask.

Performance:

- Runs at about 200ms per frame on a GPU, and training on COCO takes one to two days on a single 8-GPU machine.

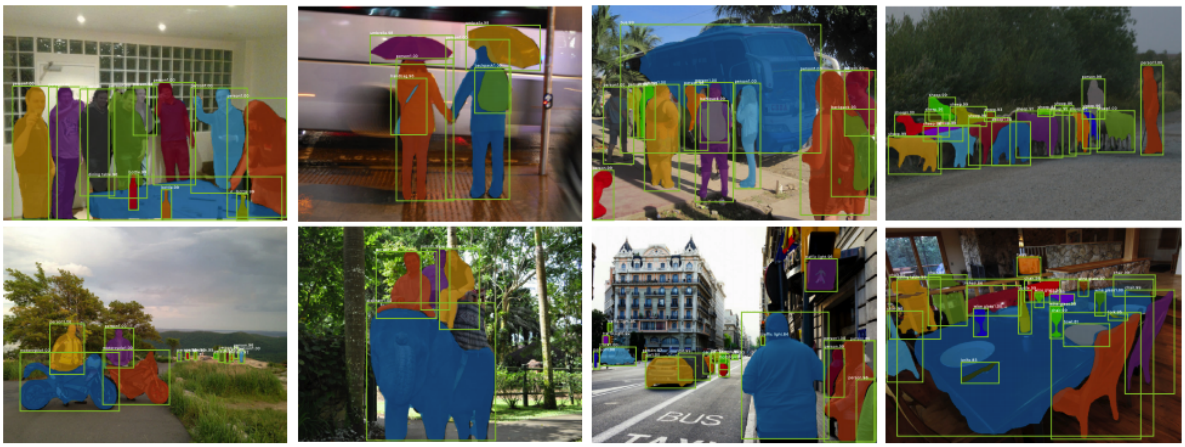


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

Architecture:

- The backbone architecture is used for feature extraction over the entire image. The results are show for ResNet, ResNetXt and FPN.
- The network head is applied separately to each RoI and is used for bounding box recognition (classification and regression) and mask prediction.
- The mask branch has a **Km²**dimensional output for each RoI, which encodes **K** binary masks of resolution **m**x**m**, one for each of the K classes.
- RoI pooling layer, used in Fast R-CNN and performing coarse spatial quantisation for feature extractions, is replaced with quantisation-free layer, preserving exact spatial locations (RoI align layer).

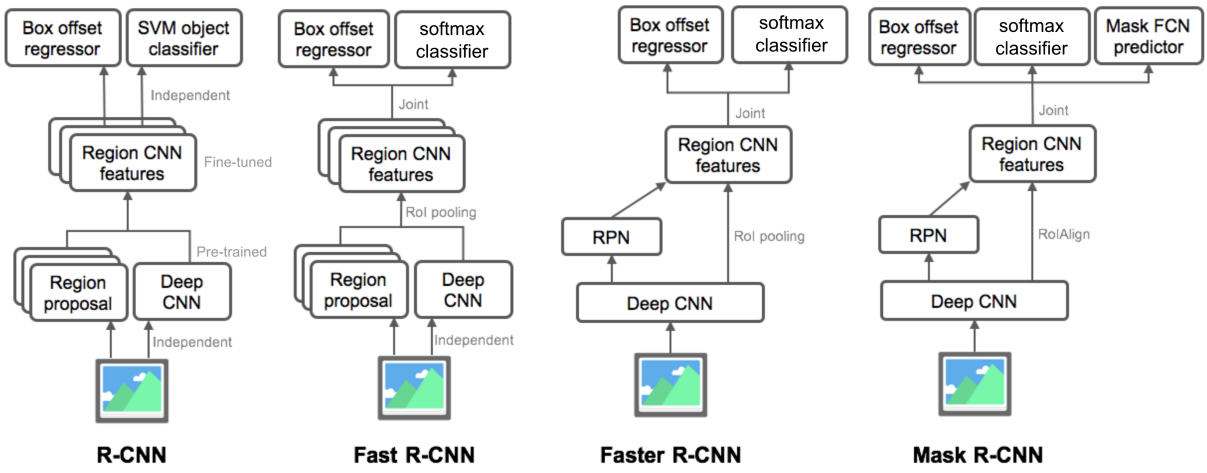
Implementation details:

- Hyper-parameters are set similar to Fast/Faster R-CNN.

- Each min-batch has 2 images per GPU and each image has **N** sampled RoIs, with a ratio of 1:3 of positive to negative. **N** is set depending on the architecture from 64 to 512. Total number of GPUs is 8.
- The results of keypoint detection are enhanced by using data distillation.

Summary

Finally, in this illustration one can track how one model evolves to the next version.



References

Rich feature hierarchies for accurate object detection and semantic segmentation

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context.

https://arxiv.org/abs/1311.2524

Fast R-CNN

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection

https://arxiv.org/abs/1504.08083

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image

https://arxiv.org/abs/1506.01497

Mask R-CNN

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in

https://arxiv.org/abs/1703.06870

R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms

Computer vision is an interdisciplinary field that has been gaining huge amounts of traction in the recent years(since CNN) and self-driving cars have taken centre stage. Another integral part of computer vision is object detection. Object detection aids in

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e