R-FCN-3000

《R-FCN-3000 at 30fps: Decoupling Detection and Classification》

Bharat Singh Hengduo Li Abhishek Sharma Larry S. Davis

http://www.cs.umd.edu/~bharat/rfcn-3k.pdf

张海鹏

2018-02-12

- **Performance/Speed**
  - 34.9%mAP(+18% than YOLO9000, ImageNet), 30fps
- **Motivation/Intuition(v.s. R-FCN)**
  - localization: position-sensitive filters are shared across different object classes
  - classification: position-sensitive filters aren’t needed
- **Generalization**
  - performance(objectness) increases with the number of training object classes

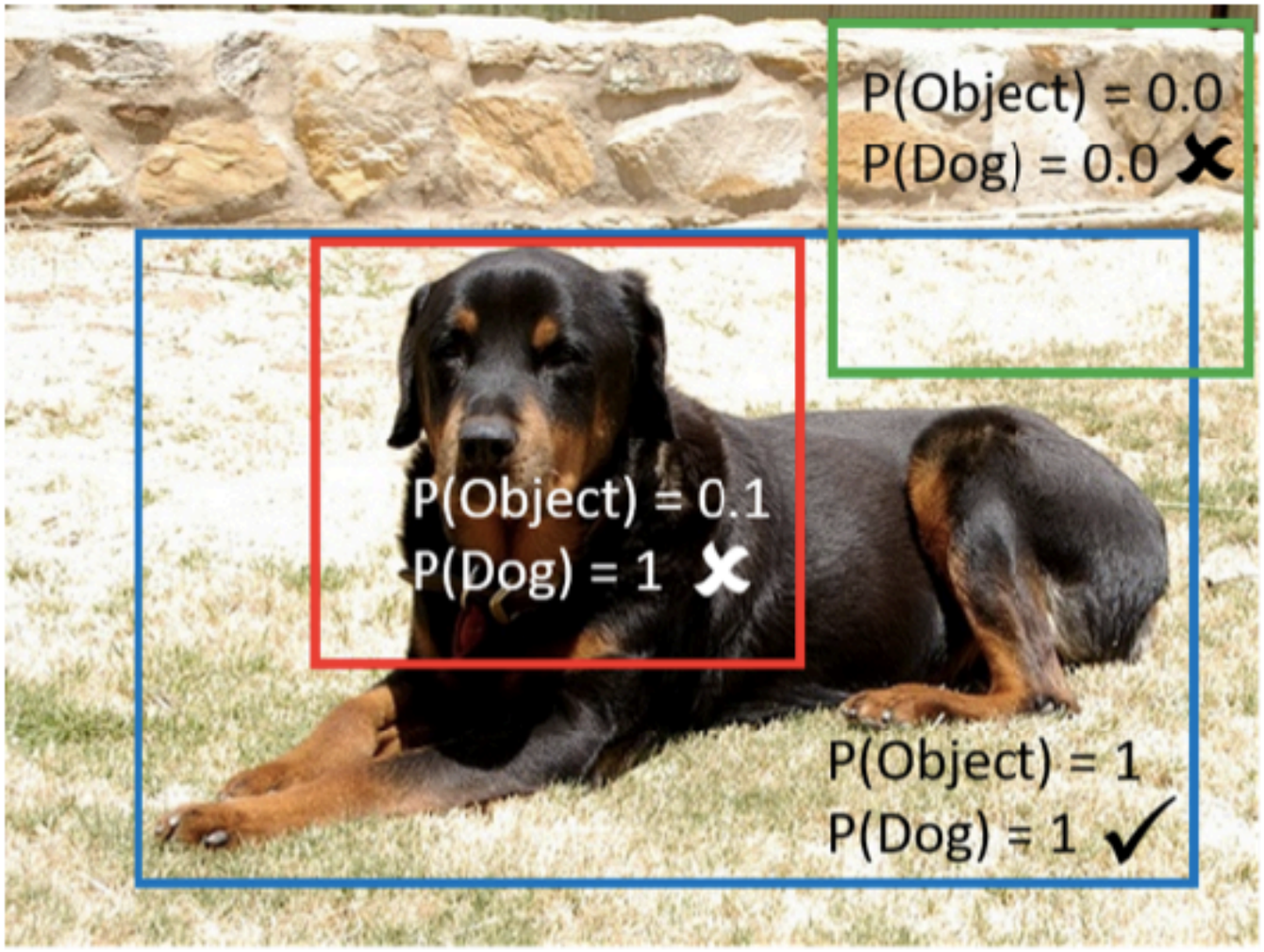
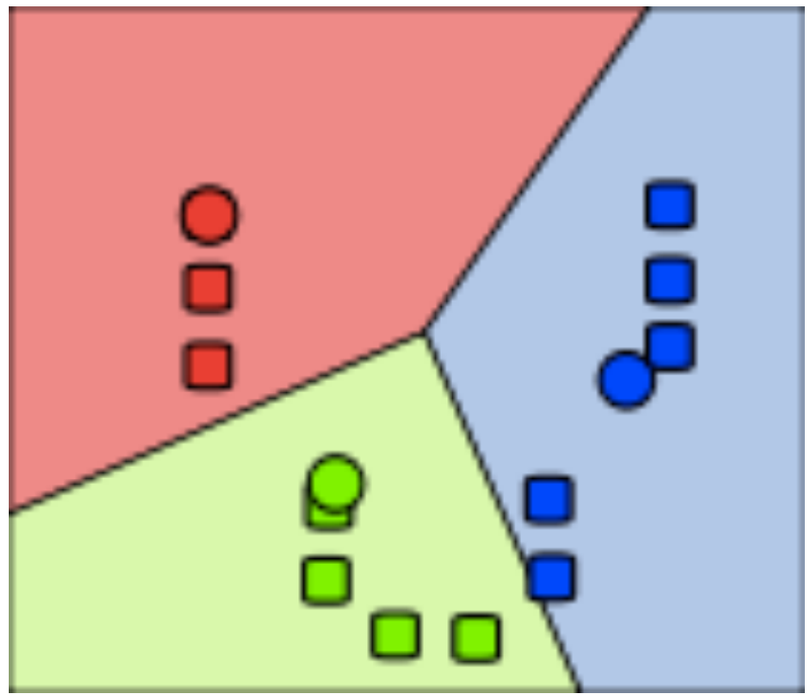


Figure 1. We propose to decouple classification and localization by independently predicting objectness and classification scores. These scores are multiplied to obtain a detector.

How to get representation of super-class?



K-means(ResNet-101, 2048)

Detection score=objectness score * **classification score**

Classification prob=**super-class prob** * category prob in super-class

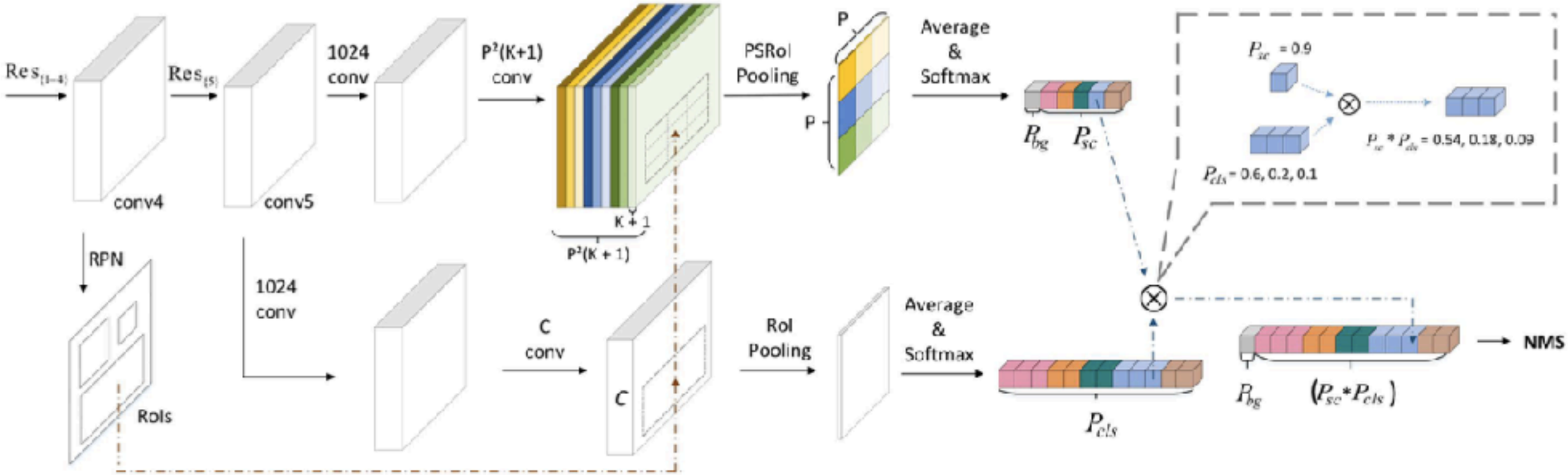


Figure 2. R-FCN-3000 first generates region proposals which are provided as input to a super-class detection branch (like R-FCN) which jointly predicts the detection scores for each super-class (sc). A class-agnostic bounding-box regression step refines the position of each RoI (not shown). To obtain the semantic class, we do not use position-sensitive filters but predict per class scores in a fully convolutional fashion. Finally, we average pool the per-class scores inside the RoI to get the classification probability. The classification probability is multiplied with the super-class detection probability for detecting 3000 classes. When K is 1, the super-class detector predicts objectness.

- **Conclusion and Future Directions**

  - How can we accelerate the classification stage of R-FCN-3000 for detecting **100,000 classes**?

  - **A typical image contains a limited number object categories**-how to use this prior to accelerate inference?

  - What changes are needed in this architecture **if we also need to detect objects and their parts**?

  - Since it is expensive to label each object instance with all valid classes in every image, can we learn robust object detectors **if some objects are not labelled in the dataset**?