Region Proposal Networks (RPN)

Backbone of Faster R-CNN

Runwei Guan

rg6n20@soton.ac.uk

Institute of Deep Perception Technology

Jiangsu Industrial Technology Research Institute

September 13, 2021

Table of Contents

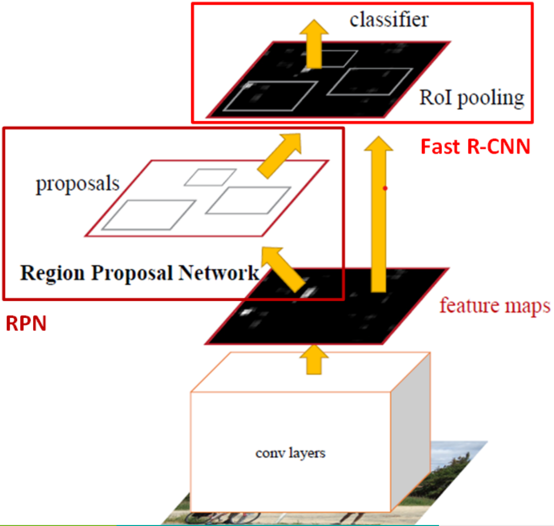
1. Introduction
   - Brief
2. Proposal Generation
   - RPN Intro
   - Translation-Invariant
   - Regression References
3. RPN Training
   - Loss Function
4. Implementation
   - Feature Sharing

- A high-efficient neural network for object proposal generation

- A high-efficient neural network for object proposal generation
- The backbone of Faster R-CNN[1]

- A high-efficient neural network for object proposal generation
- The backbone of Faster R-CNN[1]
- Component of MV3D[2], AVOD[3], etc

- A high-efficient neural network for object proposal generation
- The backbone of Faster R-CNN[1]
- Component of MV3D[2], AVOD[3], etc



RPN Intro

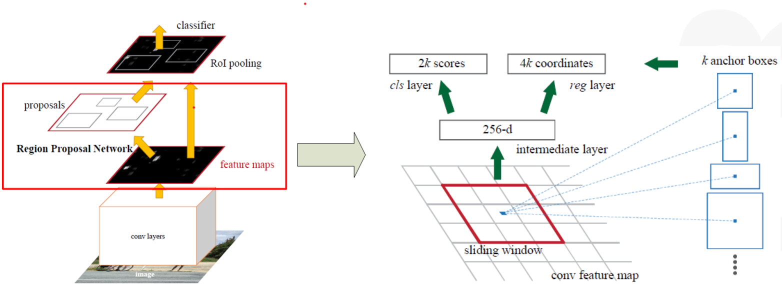


Figure: Mini Network for Region Proposal Generation

- Role: Tells the Fast R-CNN module where to look
- Input: Feature Map ; Output: 2 Classifiers' Results(reg and cls)
- Generate 9 anchor boxes in one sliding window
- Translation-Invariant

Translation-Invariant

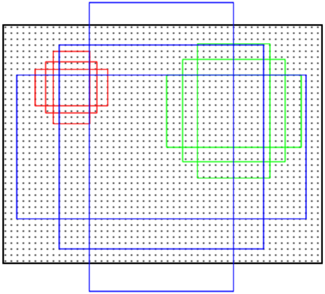


Figure: 9 anchors(3 scales and 3 aspect ratios)

- For the anchors
- For the functions that compute proposals relative to the anchors
- **Reduces the model size**
- **Better than MultiBox Method**

Multi-Scale Anchors as Regression References

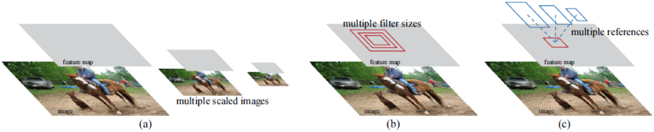


Figure: 3 methods using pyramids

Two Popular Methods **(time-consuming)**

1) Image/feature pyramids
2) Sliding windows of multiple scales/aspect ratios on the feature maps

Method in RPN **(cost-efficient)**

1) Anchor pyramid

Loss Function

Label Assignment

Positive Label

- Anchor/anchors with the highest (IoU) overlap with a ground-truth box
- An anchor that has an IoU overlap higher than 0.7 with any ground-truth box

Negative Label

- IoU ratio is lower than 0.3 for all ground-truth boxes

Others

- Anchors that are neither positive nor negative do not contribute to the training objective

Loss Function

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i L_{reg}(t_i, t_i^*) \qquad (1)$$

i: index of an anchor in a mini-batch
p_i: predicted probability of anchor i being an object
p_i^*: ground-truth label
t_i: a vector(4 parameterized coordinates of the predicted bounding box)
t_i^*: the ground-truth box with a positive anchor
L_{cls}: log loss over two classes(object *vs* not object)
L_{reg}: $smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases}$

Bounding Box Regression

$$
\begin{aligned}
t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a \\
t_w &= log(w/w_a), \quad t_h = log(h/h_a) \\
t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a \\
t_w^* &= log(w^*/w_a), \quad t_h^* = log(h^*/h_a)
\end{aligned}
$$

x, y: box's center coordinates
w: box's width
h: box's height

x, y, w, h: for predicted box
x_a, y_a, w_a, h_a: for anchor box
x^*, y^*, w^*, h^*: for ground-truth box

Training Details

Data sampling: “image-centric” sampling

Training method: Transfer learning
New layers initialization: Weights $\sim N(0, 0.01)$
Freezed layers: ZF net + VGG net

Optimizer: Stochastic gradient descent
Momentum: 0.9
Learning rate(first 60k): 0.001
Learning rate(last 20k): 0.0001
Weight decay: 0.0005

Sharing Features for RPN and Fast R-CNN

RPN: Region proposal generation
Fast R-CNN: Utilizes region proposals to do detection

- Share convolutional layers between 2 networks ✔
- Learn 2 seperate networks ✖

↓

Feature Shared Method

1) Alternating training
2) Approximate joint training
3) Non-approximate joint training

Alternating Training

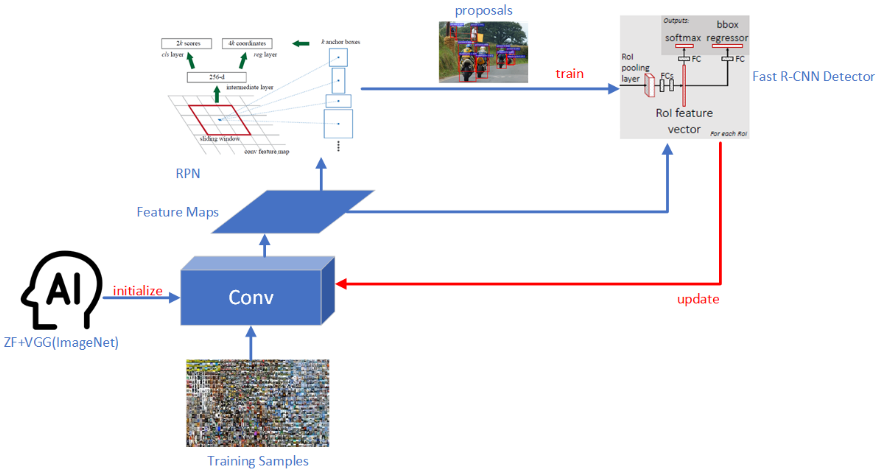


Figure: Alternating training

Approximate Joint Training

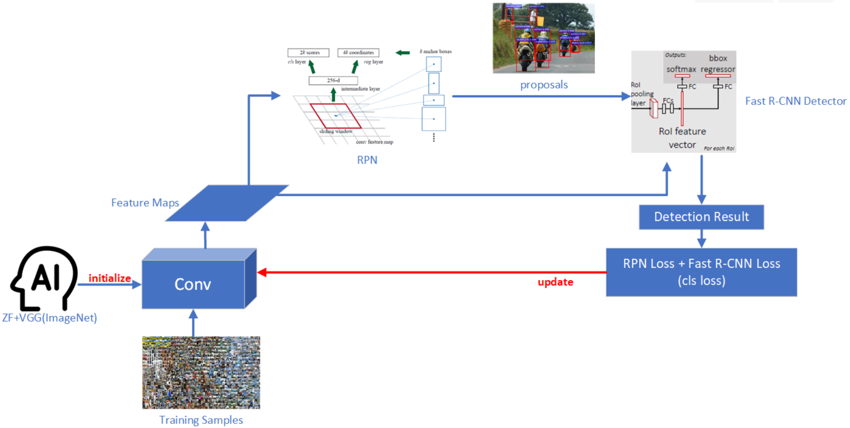


Figure: Approximate joint training

Non-approximate Joint Training

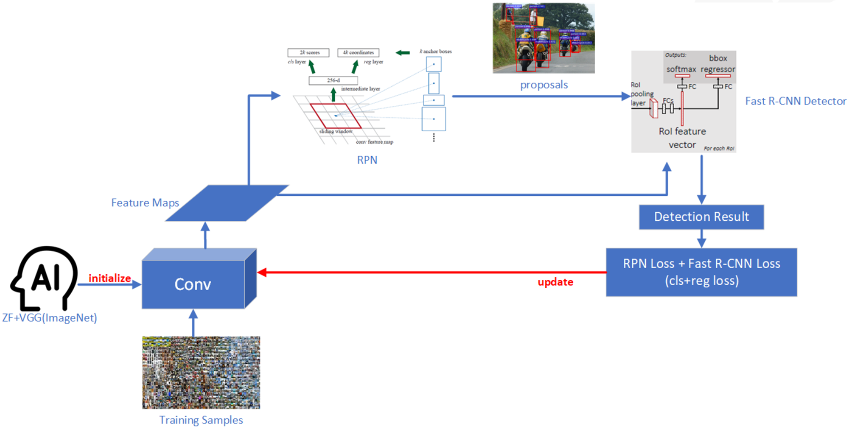


Figure: Non-approximate joint training

References

- Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: **Advances in neural information processing systems** 28 (2015), pp. 91–99.
- Xiaozhi Chen et al. "Multi-view 3d object detection network for autonomous driving". In: **Proceedings of the IEEE conference on Computer Vision and Pattern Recognition**. 2017, pp. 1907–1915.
- Jason Ku et al. "Joint 3d proposal generation and object detection from view aggregation". In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE. 2018, pp. 1–8.