Convolutional Neural Networks – Part 3 Object Detection

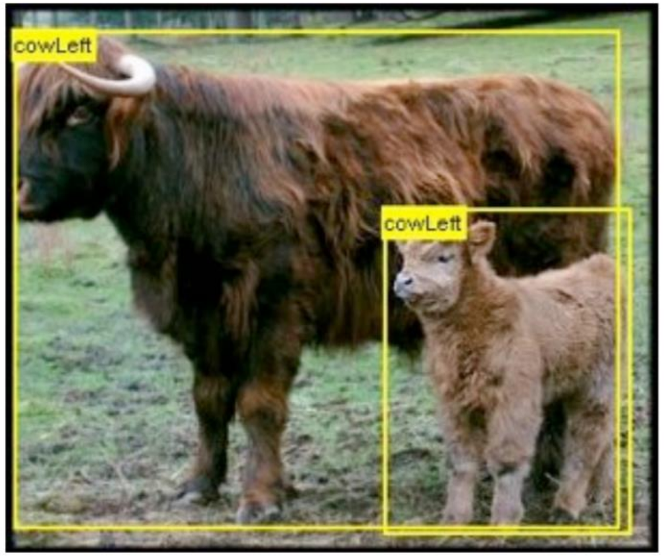
ECE 449

Object Detection

- Before deep learning
  - Deformable part models (DPM)
- Multi-stage
  - R-CNN
  - Fast R-CNN (one-stage)
  - Faster R-CNN
  - Mask R-CNN
- One-stage
  - You only look once (YOLO)
  - Single-shot detector (SSD)
  - CenterNet (anchor free)
- Non-maximum suppression (NMS)

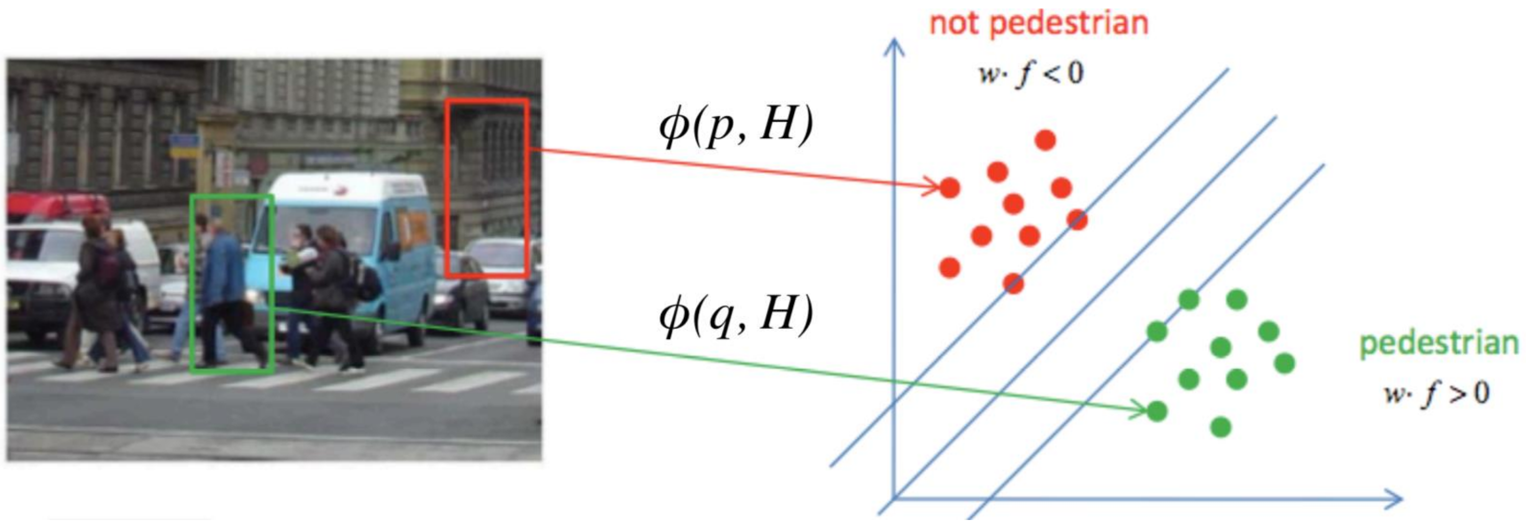
Object Detection

- PASCAL Challenge
  - Objects from 20 categories
    - person, car, bicycle, bus, airplane, sheep, cow, table, ...
  - Objects are annotated with bounding boxes



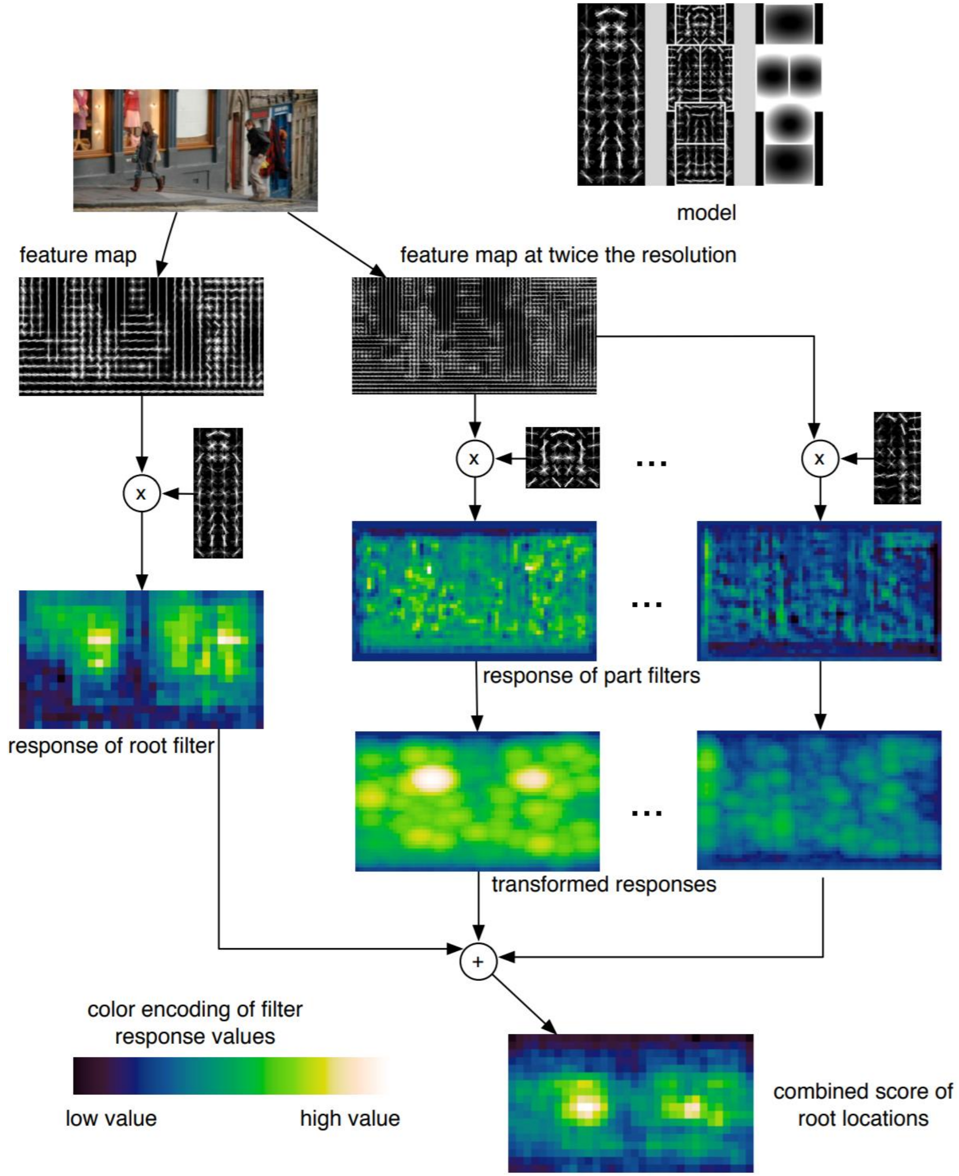
Deformable Part Models (DPM)

- Strategy
  - Sliding window approach, reduces object detection to binary classification
- Feature
  - Histogram of Gradient (HOG)
- Classifier
  - Linear SVMs



DPM

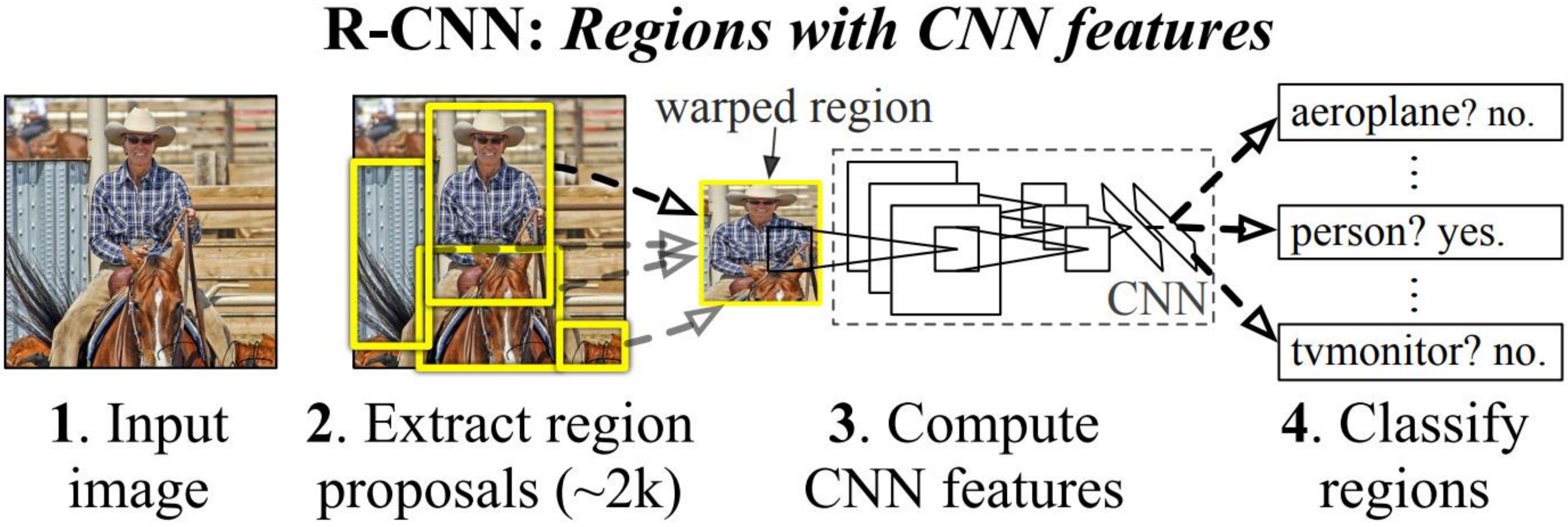
- Root filter + part filters



Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, *32*(9), 1627-1645.

R-CNN

- Classify region proposals



SVMs

R-CNN

- Generate region proposals
  - Selective search

Algorithm 1: Hierarchical Grouping Algorithm

DontPrintSemicolon **Input**: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using Felzenszwalb and Huttenlocher (2004) Initialise similarity set $S = \emptyset$;
foreach *Neighbouring region pair* (r_i, r_j) **do**
  Calculate similarity $s(r_i, r_j)$;
  $S = S \cup s(r_i, r_j)$;

while $S \neq \emptyset$ **do**
  Get highest similarity $s(r_i, r_j) = \max(S)$;
  Merge corresponding regions $r_t = r_i \cup r_j$;
  Remove similarities regarding r_i : $S = S \setminus s(r_i, r_*)$;
  Remove similarities regarding r_j : $S = S \setminus s(r_*, r_j)$;
  Calculate similarity set S_t between r_t and its neighbours;
  $S = S \cup S_t$;
  $R = R \cup r_t$;
Extract object location boxes L from all regions in R;

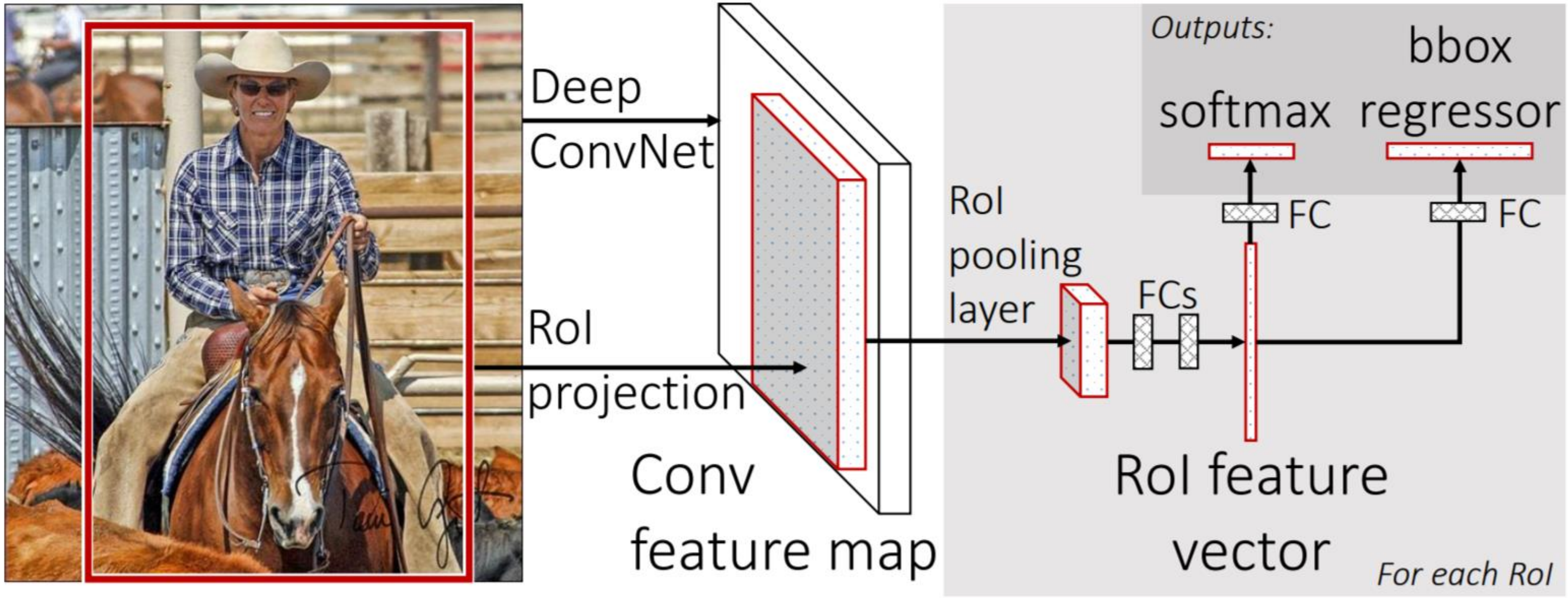
(a) (b)

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 154-171.

R-CNN

- Limitations
  - Training is a multi-stage pipeline
  - Training is expensive in space and time
  - Object detection is slow

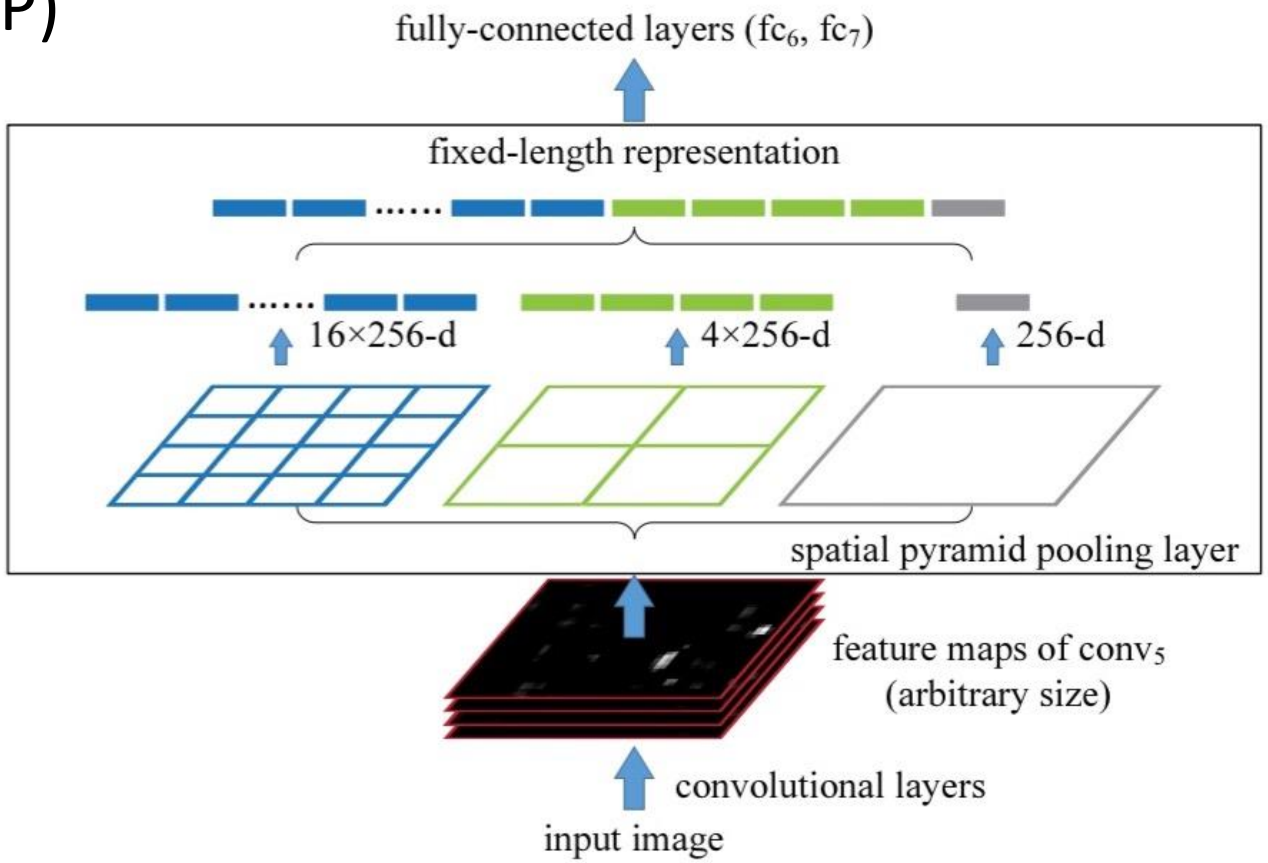
Fast R-CNN



Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Fast R-CNN

- Spatial pyramid pooling (SPP)
- One-stage training



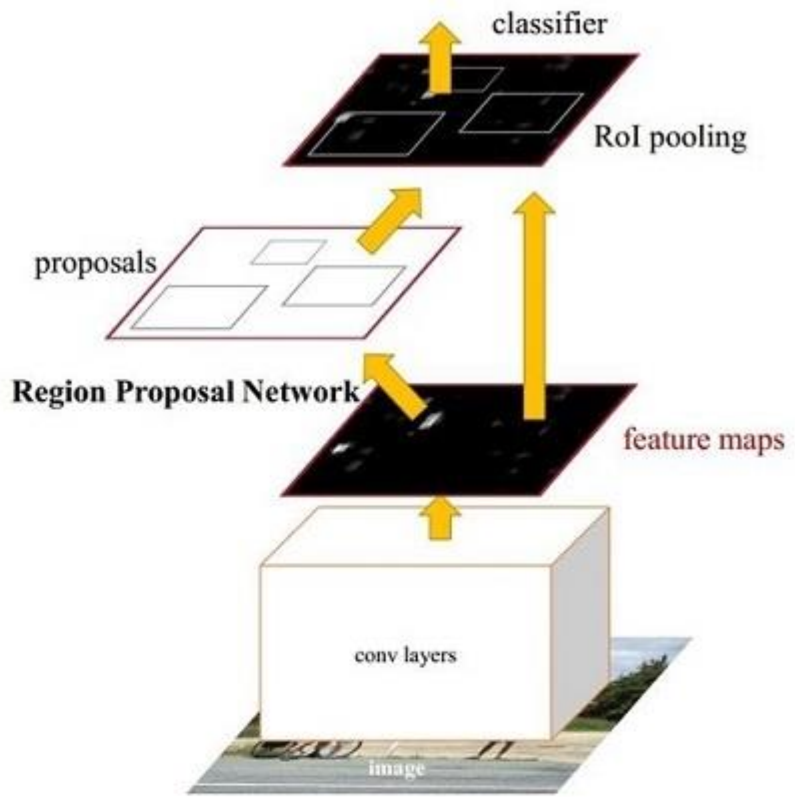
Fast R-CNN

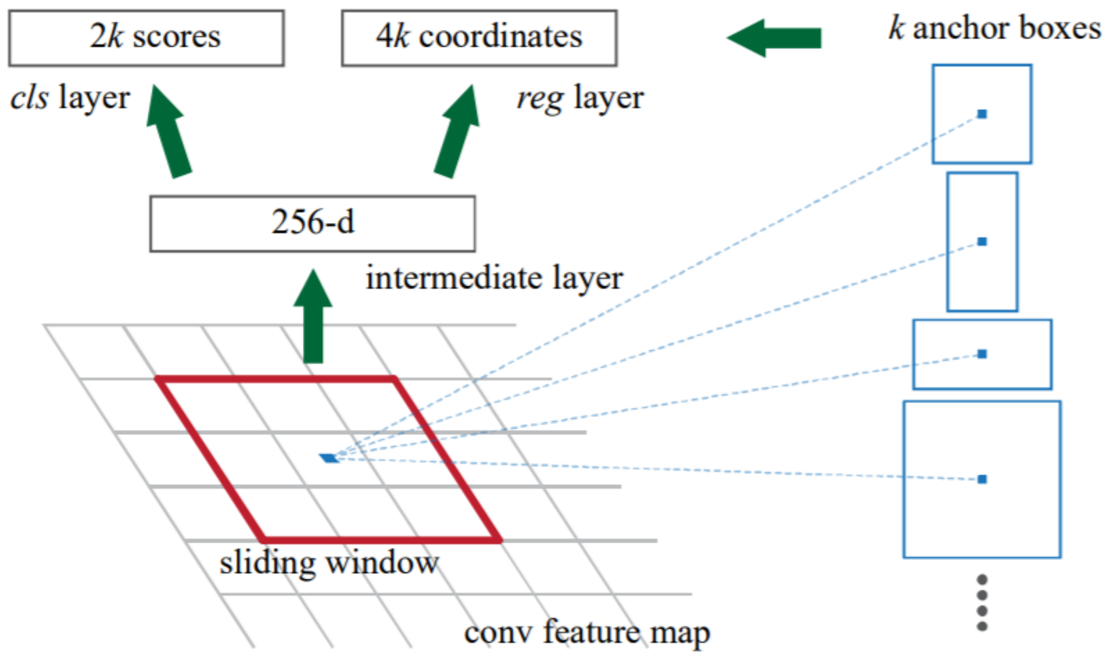
- Fast R-CNN vs R-CNN
  - One-stage
  - No SVMs
  - No warp / resize
  - Fast
- Limitations
  - Region proposals are not efficient

Faster R-CNN

- Region proposal network (RPN)

Objectness

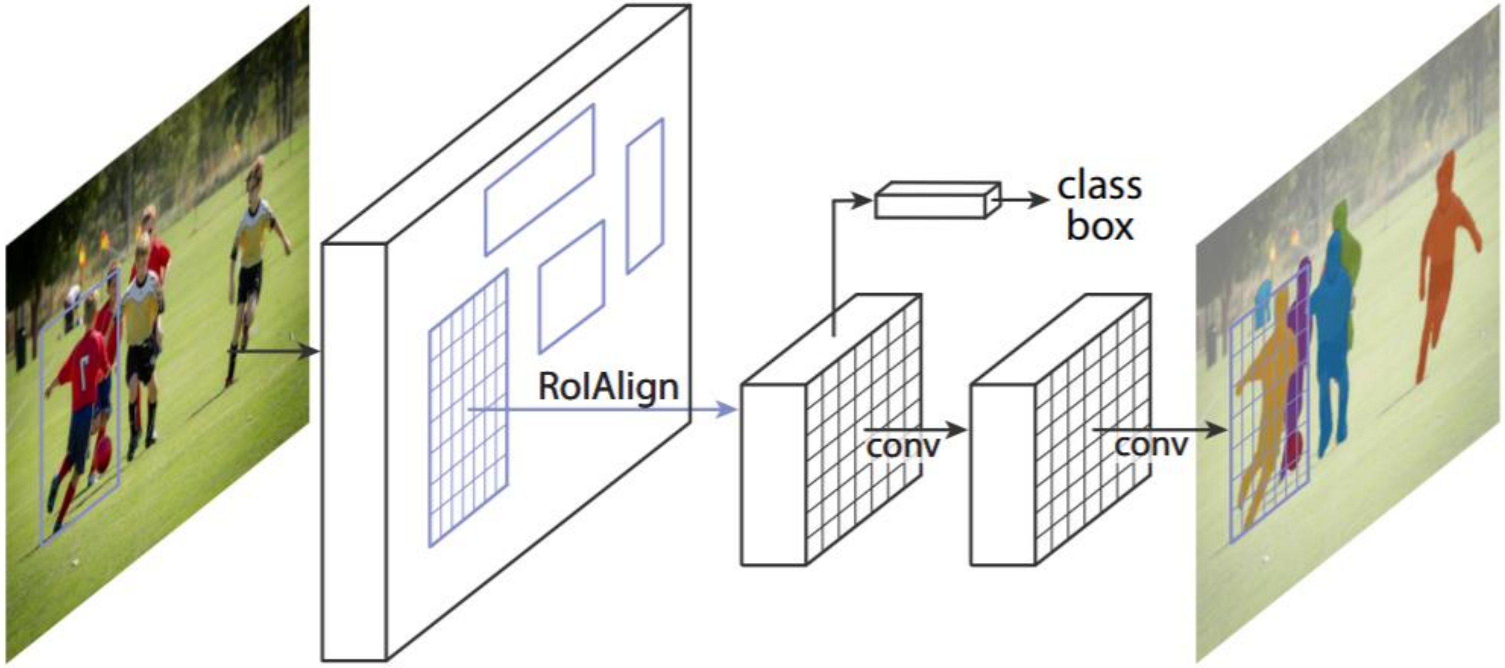




Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91-99.

Mask R-CNN

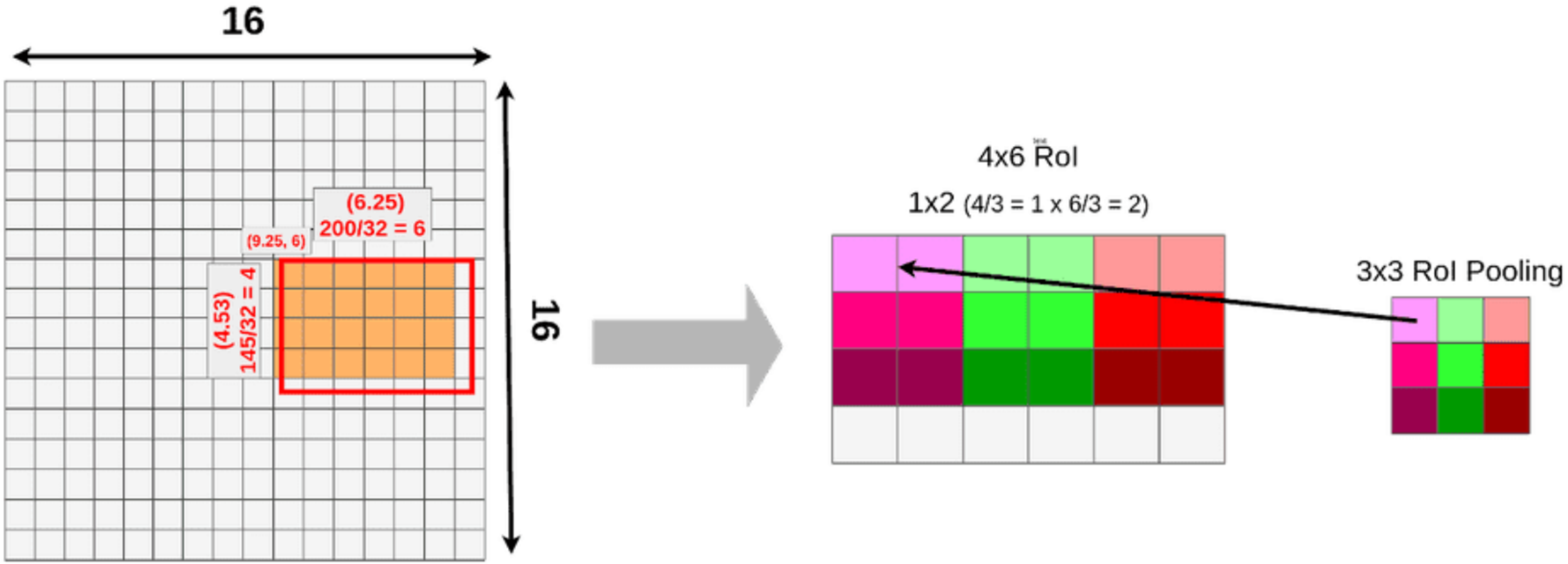
- Mask prediction
- ROI align



He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Mask R-CNN

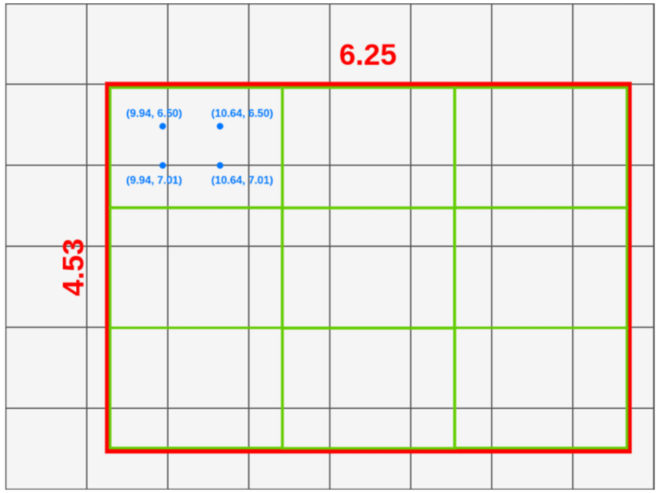
- **ROI pooling** vs ROI align



Quantization two times

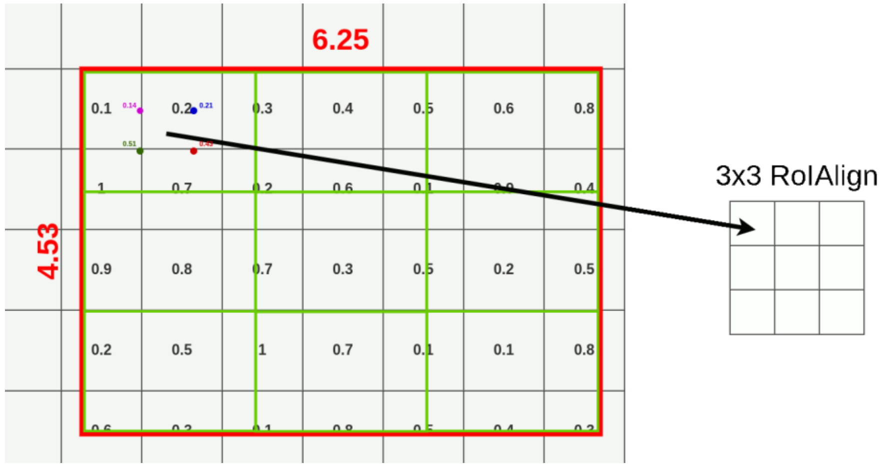
Mask R-CNN

- ROI pooling vs **ROI align**



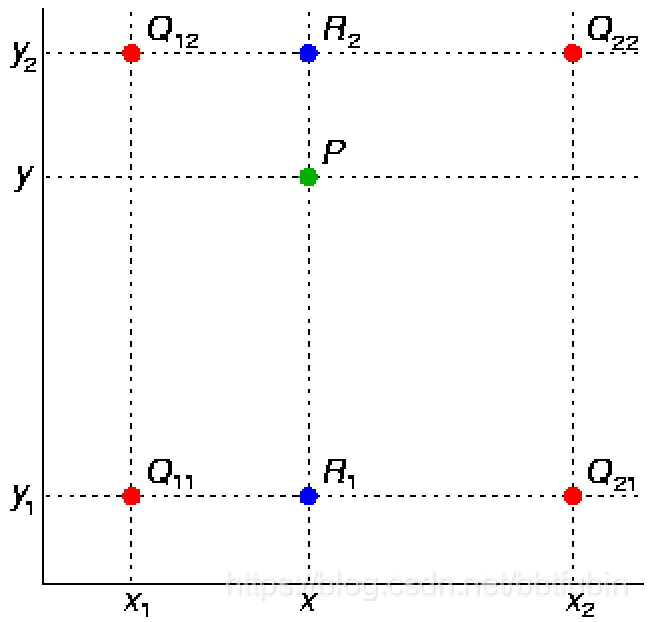
Sampling

Bilinear interpolation



Pooling

Bilinear Interpolation



$$f(x,y) \approx \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y)$$
$$+ \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) + \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1).$$

YOLO

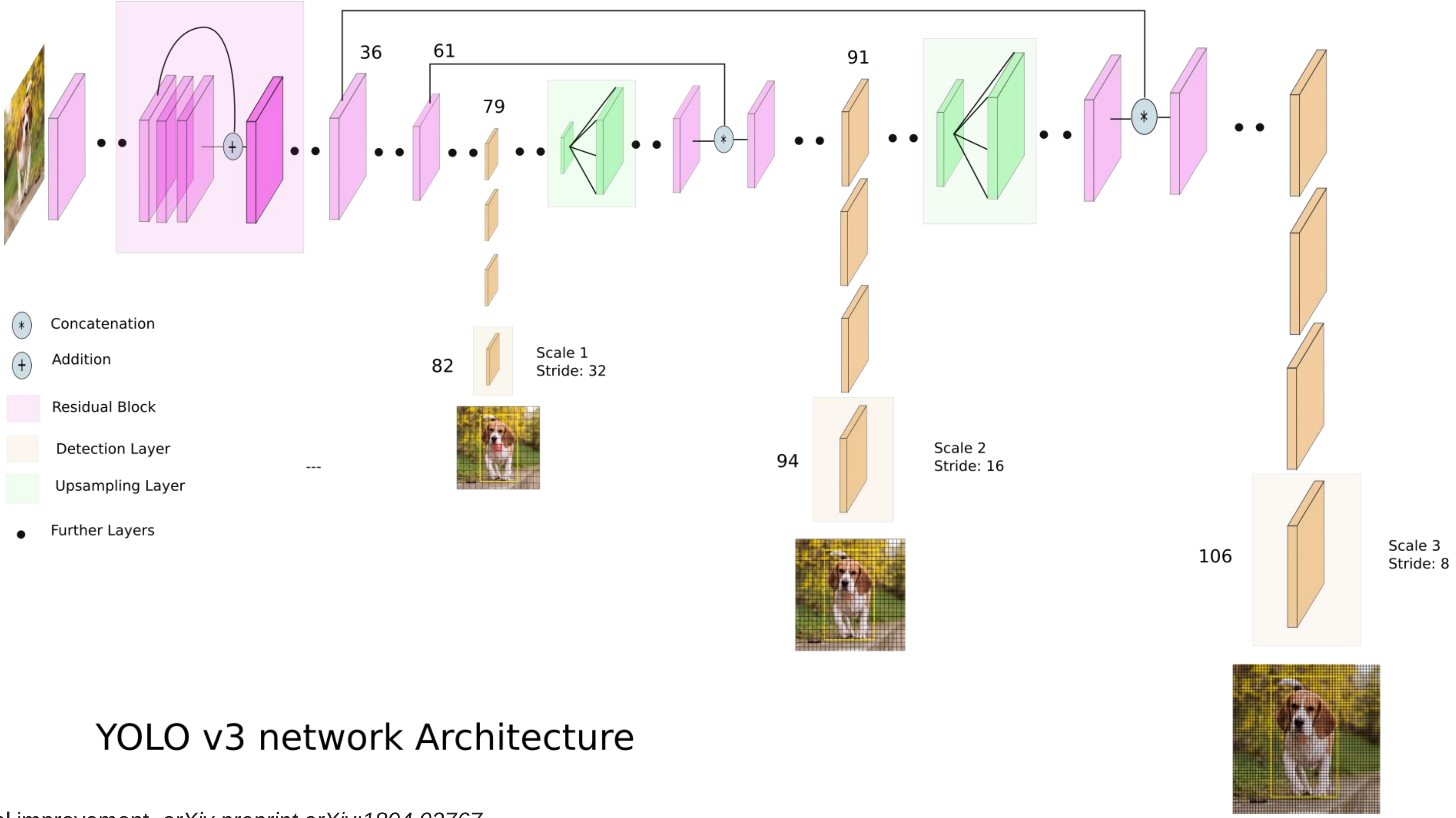
- Example
  - Scale 1: 7×7 grid
  - Each grid: (1+class+4)×k

Objectness

Class number

Bbox params

Anchors

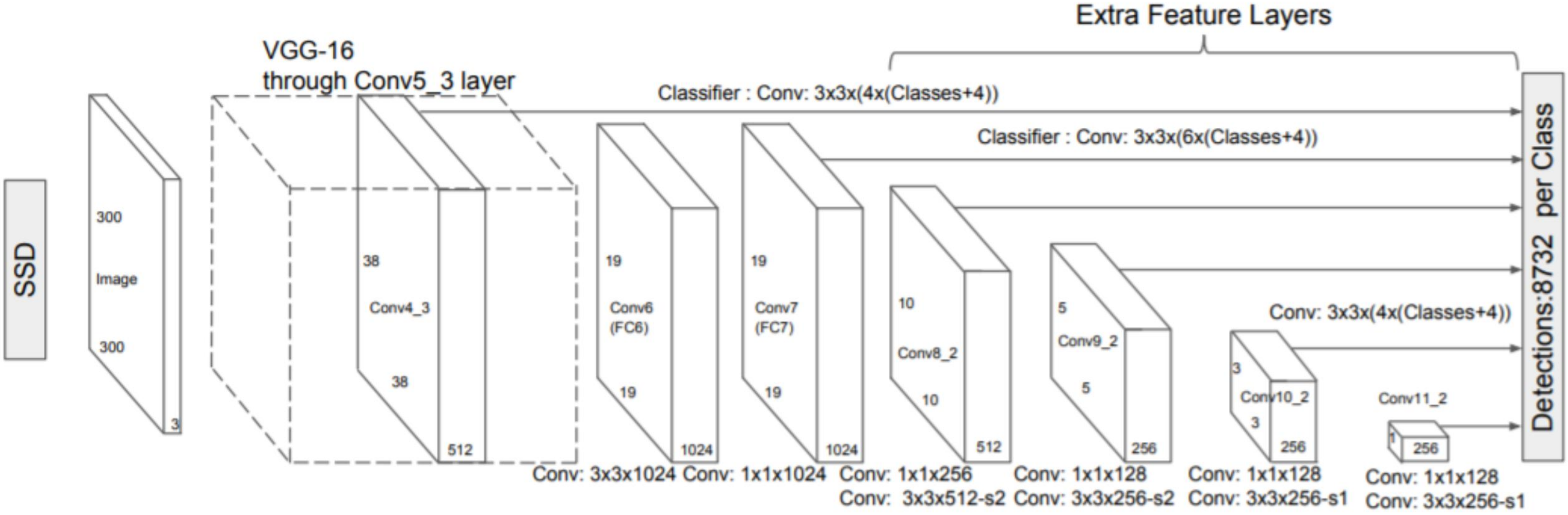


YOLO v3 network Architecture

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

SSD

- Multibox and multi-scale



Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

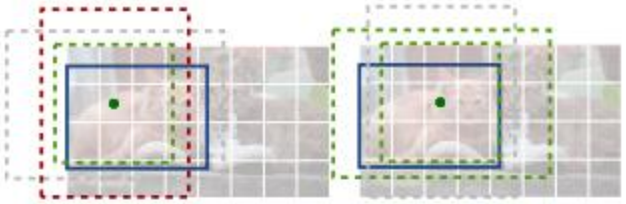
CenterNet

- Anchor vs anchor-free

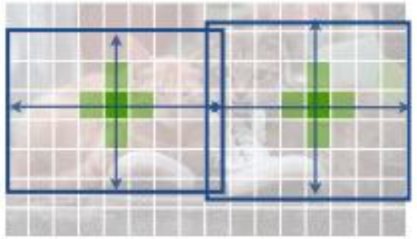
$$L_k = \frac{-1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \\ \quad \log(1 - \hat{Y}_{xyc}) & \text{otherwise} \end{cases}$$

$$L_{off} = \frac{1}{N} \sum_{p} \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right|$$

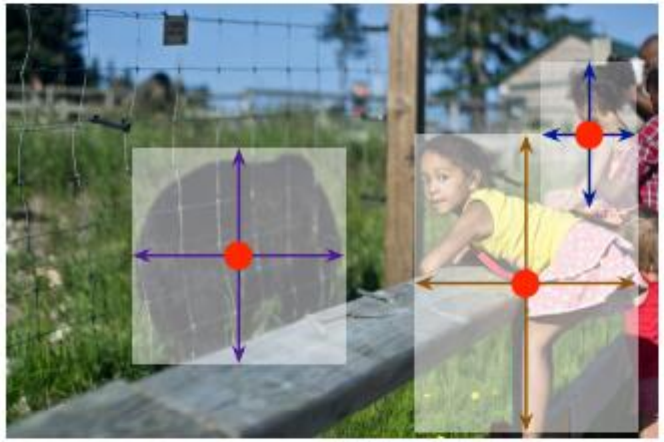
$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{p_k} - s_k \right|$$

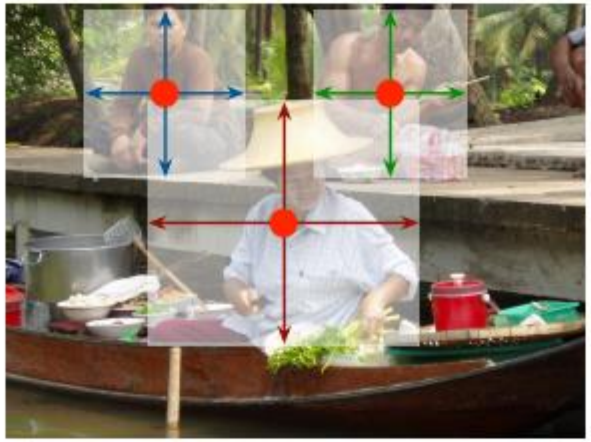


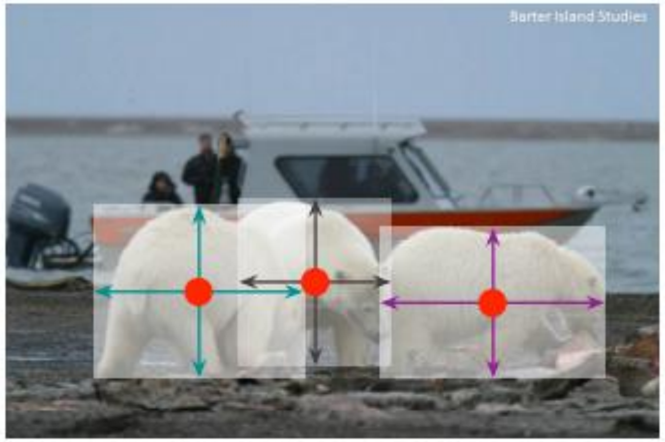
(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.



(b) Center point based detection. The center pixel is assigned to the object. Nearby points have a reduced negative loss. Object size is regressed.

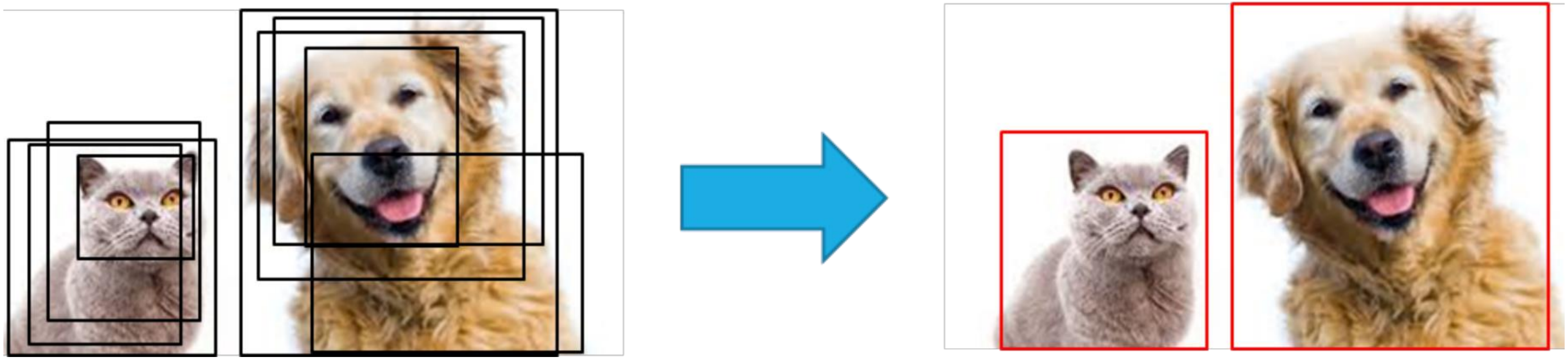






NMS

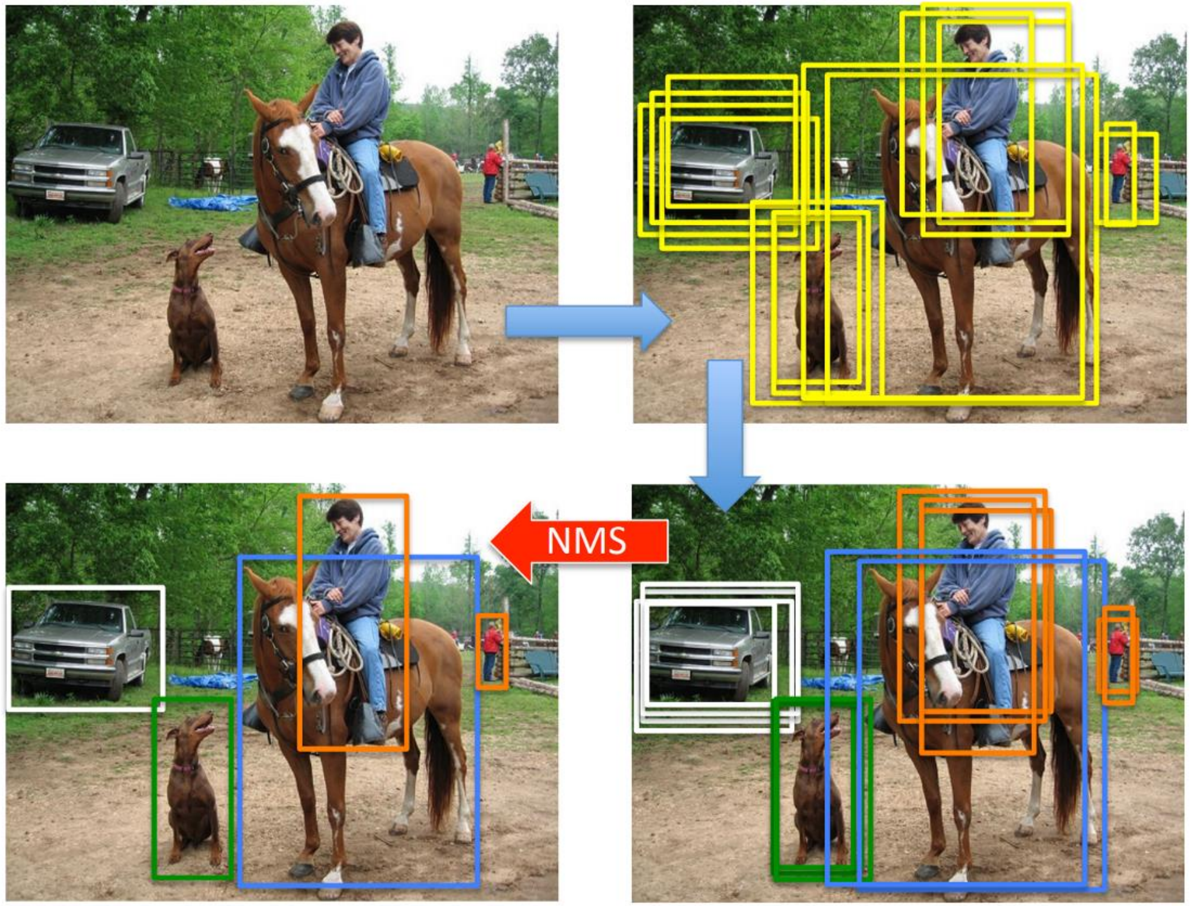
- What left?



NMS

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
    $B_{nms} \leftarrow \emptyset$
    for $b_i \in B$ **do**
        $discard \leftarrow \text{False}$
        for $b_j \in B$ **do**
            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then**
                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**
                    $discard \leftarrow \text{True}$
        if not $discard$ **then**
            $B_{nms} \leftarrow B_{nms} \cup b_i$
    return B_{nms}



Improve Your Detector Performance

- Add extra training data
- Use appropriate augmentation strategies
- Change the backbone architecture
- Modify the classification and regression methods
- Use proper loss functions
- …