



# Going Faster: Faster R-CNN

Visão Computacional

Programa de Pós-Graduação em Ciência da Computação – UFMA

Prof. Geraldo Braz Junior

Baseado nas notas de aula do  
curso cs231n

# Classification + Localization



GRASS, CAT,  
TREE, SKY

No objects, just pixels



CAT

Single Object



DOG, DOG, CAT

Multiple Object



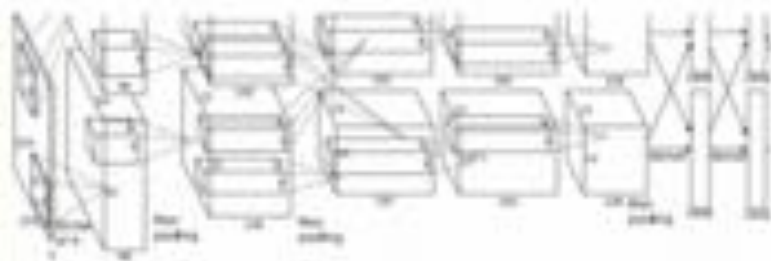
DOG, DOG, CAT

Image made with COCO public dataset

# Classification + Localization



This image is CC BY-SA licensed.



Fully  
Connected:  
4096 to 1000

**Class Scores**

Cat: 0.9  
Dog: 0.05  
Car: 0.01  
...

Correct label:  
Cat

Softmax  
Loss

Vector:  
4096

Fully  
Connected:  
4096 to 4

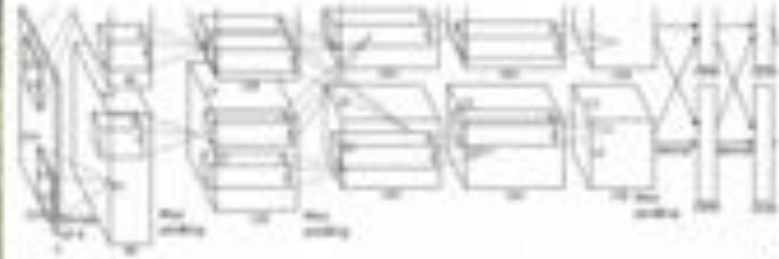
**Box  
Coordinates**  
(x, y, w, h)

L2 Loss

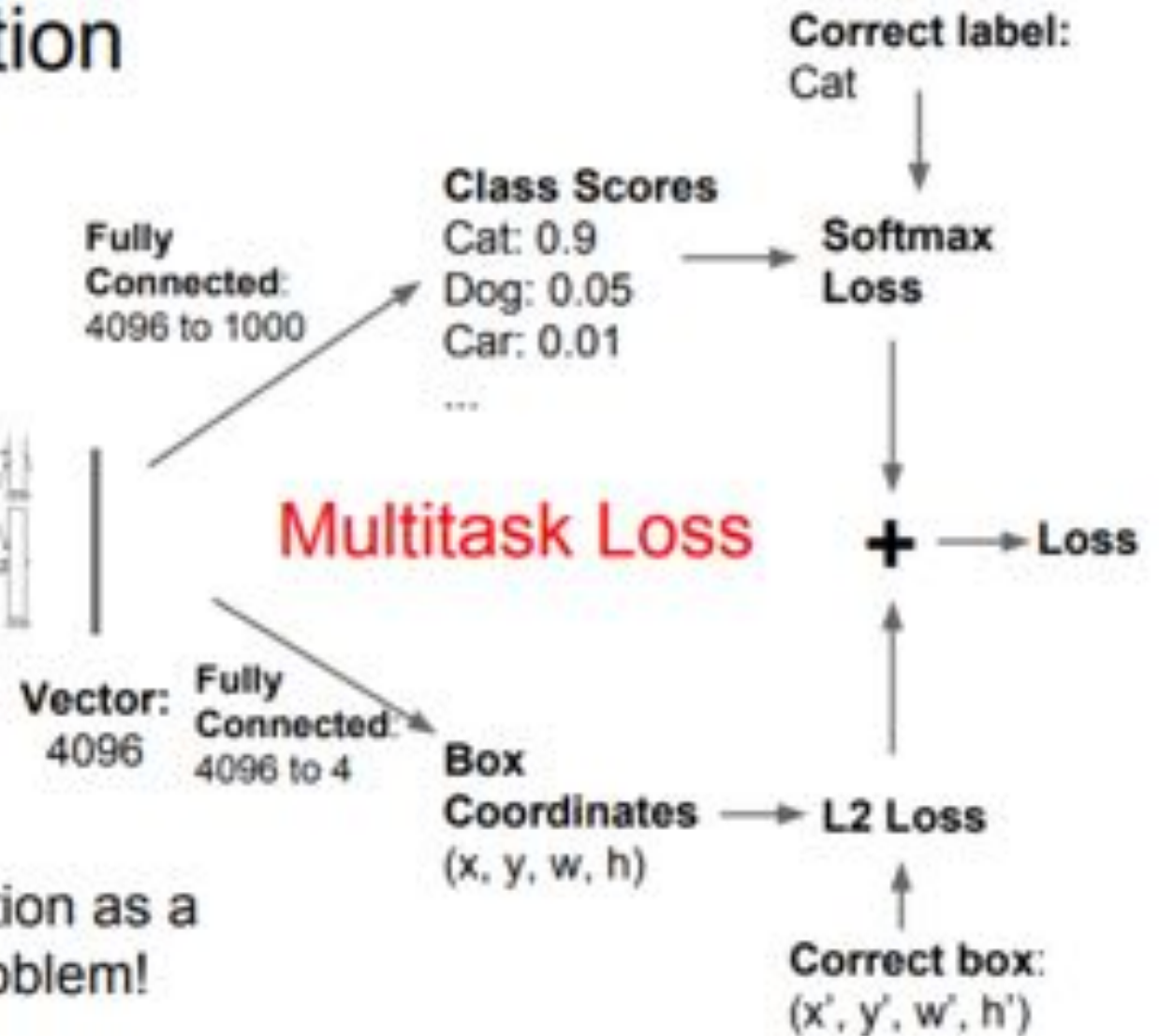
Correct box:  
(x', y', w', h')

Treat localization as a  
regression problem!

# Classification + Localization



This image is © 2012 Google, Inc.



Treat localization as a regression problem!

- A CNN usada é normalmente pré-treinada (Transfer Learning)

# Detecção de Objetos

Object Detection



GRASS, CAT,  
TREE, SKY

No objects, just pixels



CAT

Single Object



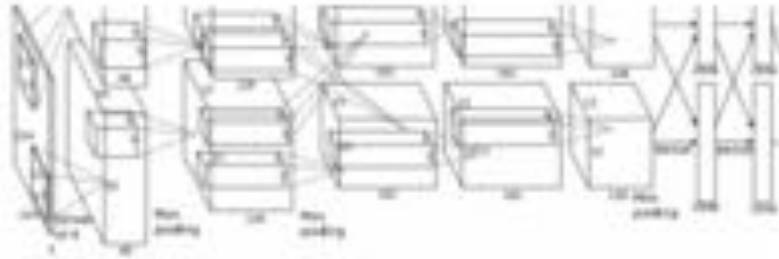
DOG, DOG, CAT

Multiple Object

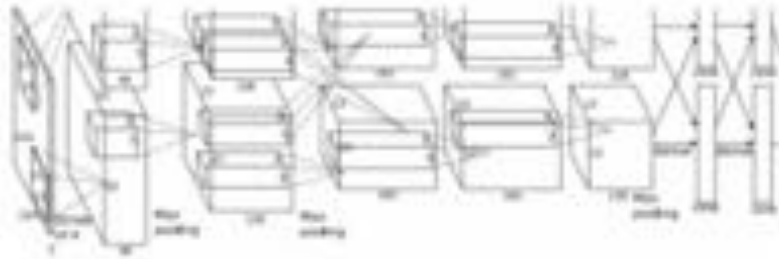


DOG, DOG, CAT

# Detecção de Objetos



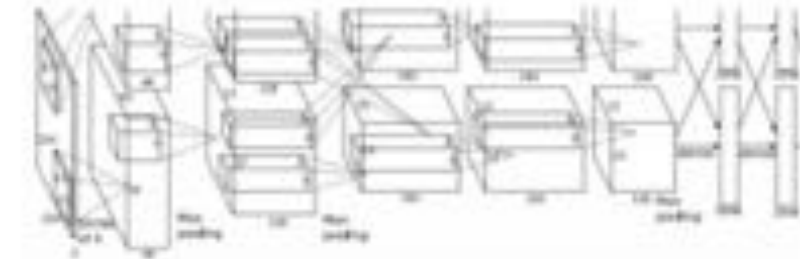
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)



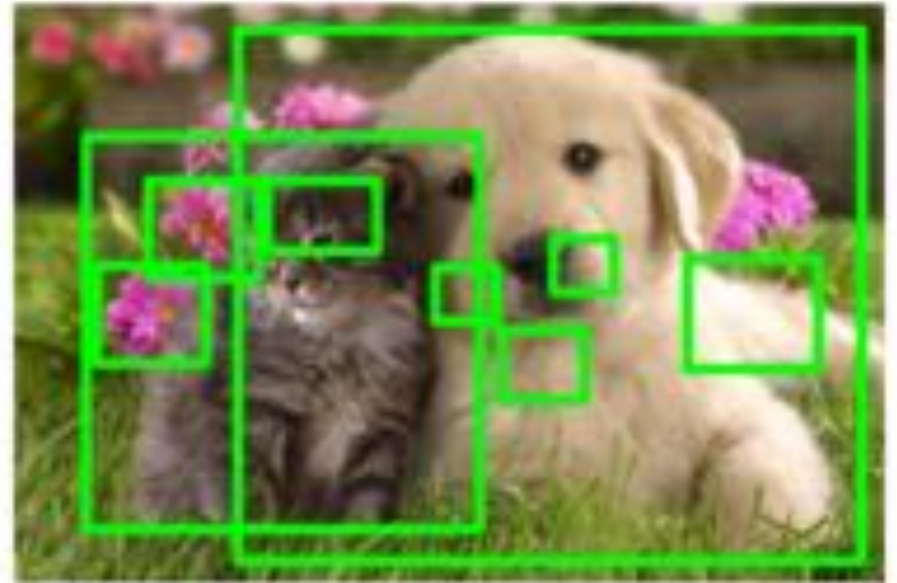
DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

....

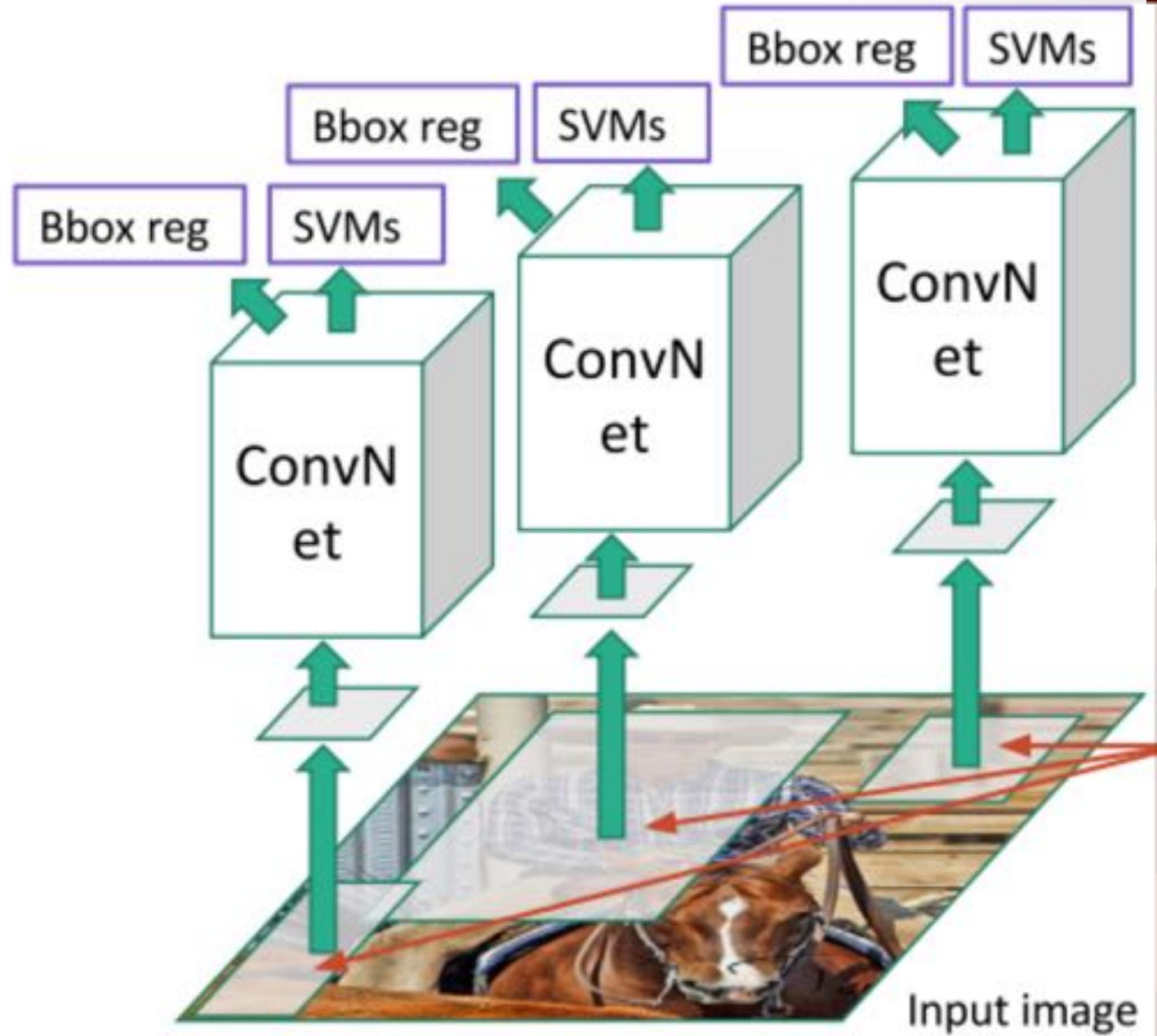
# Region Proposals

- Selective Search (aproximadamente 2k propostas por imagem)



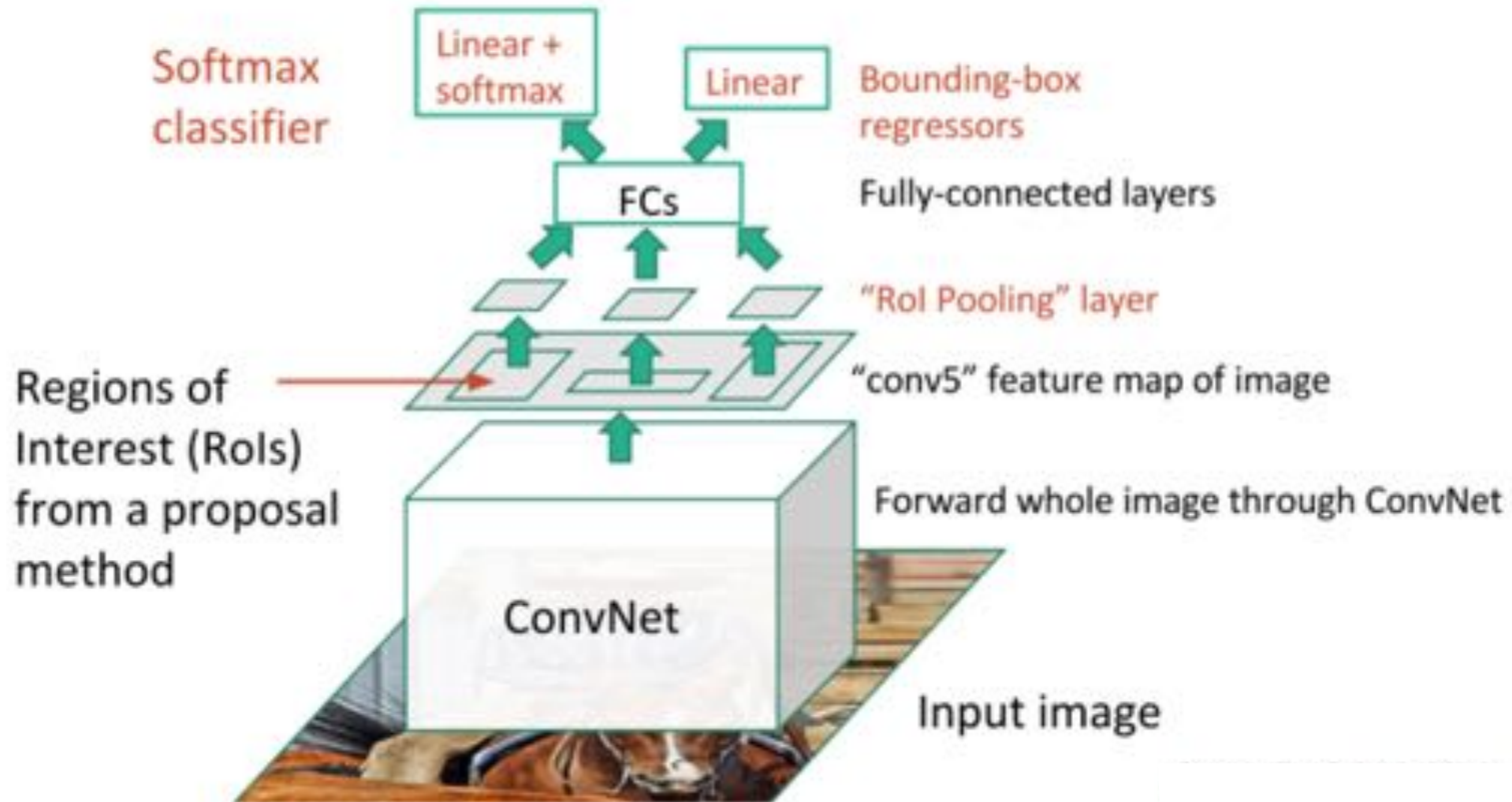
# R-CNN

Relativamente lento  
47s por imagem (usando  
VGG16)

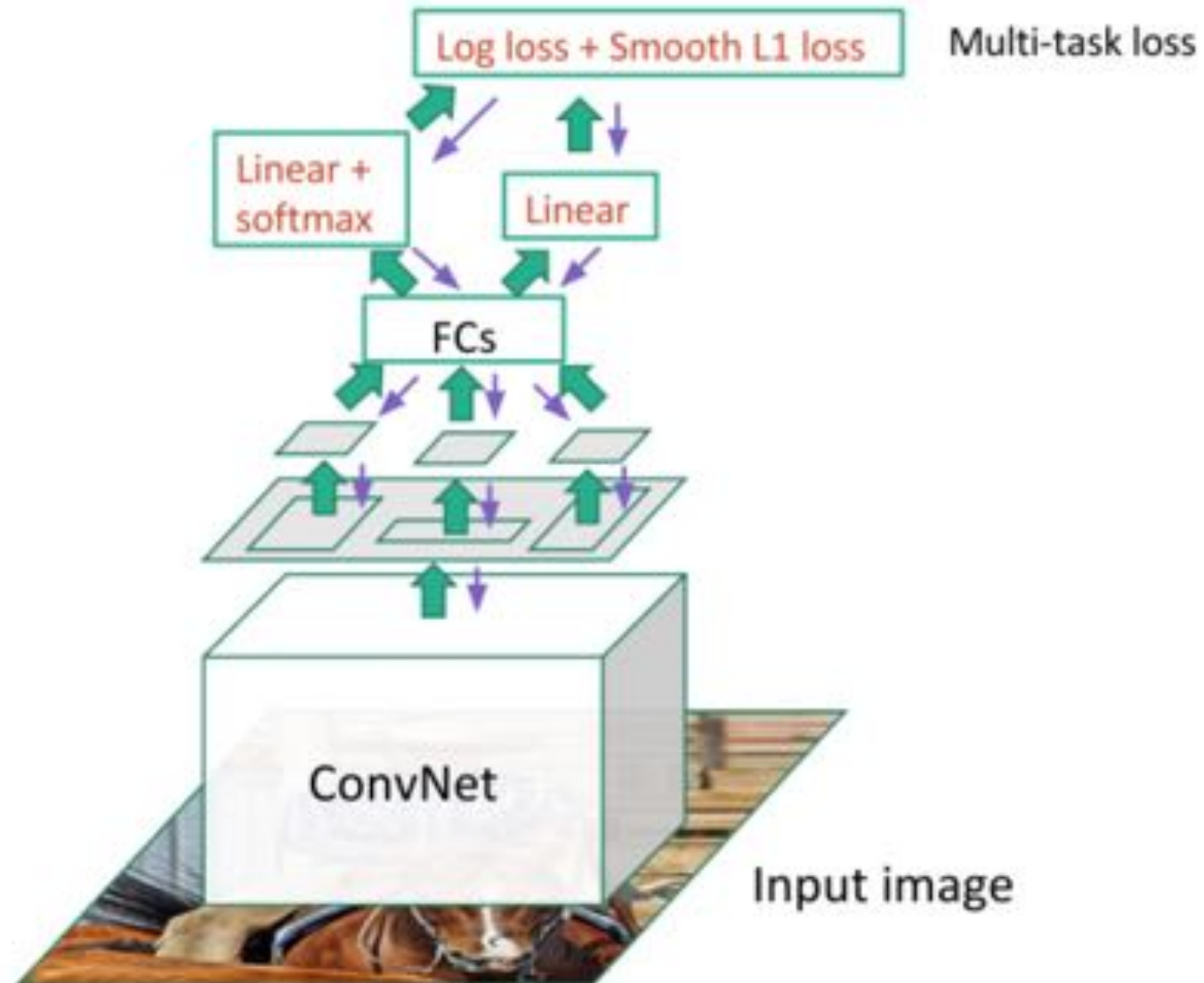




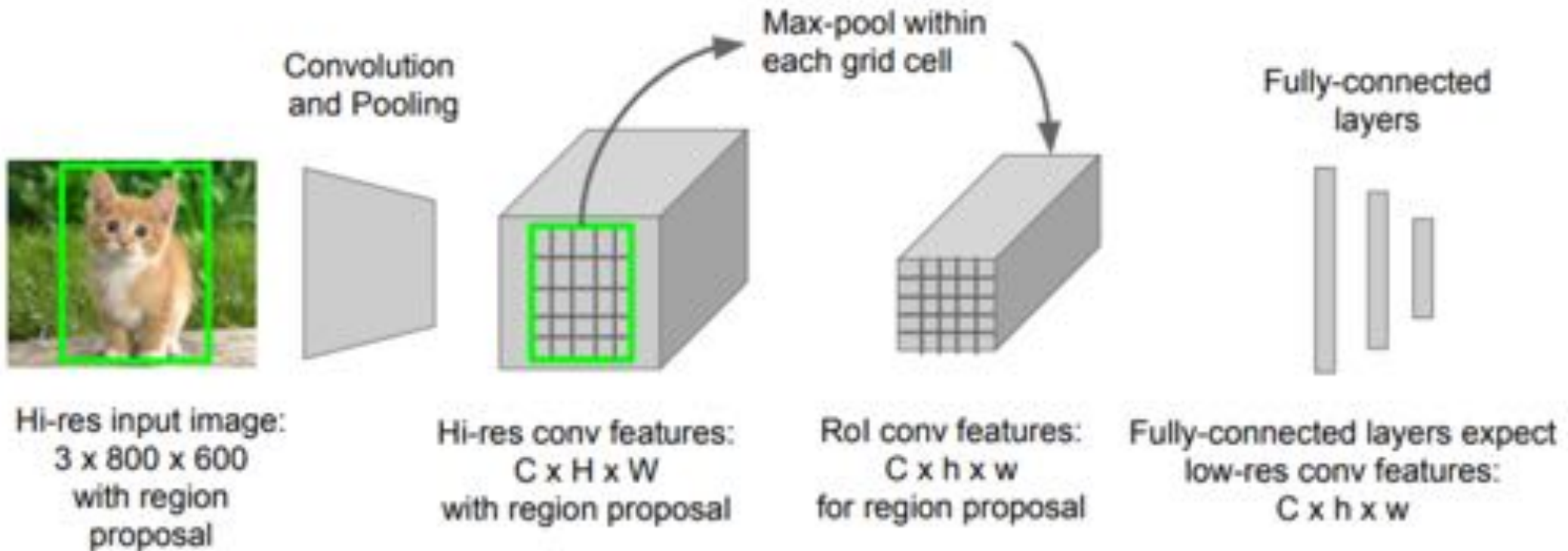
# Fast R-CNN

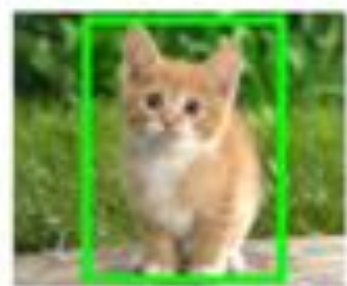


# Fast R-CNN



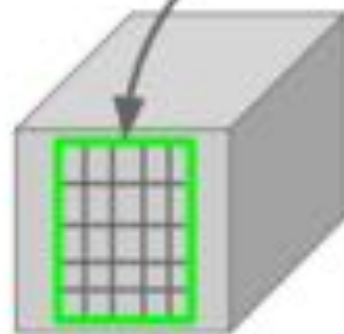
# Fast R-CNN: RoI pooling





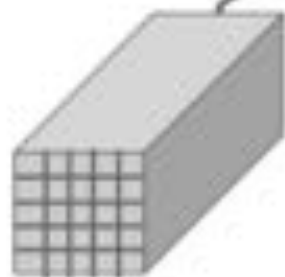
Hi-res input image:  
3 x 800 x 600  
with region  
proposal

Convolution  
and Pooling



Hi-res conv features:  
 $C \times H \times W$   
with region proposal

Can back propagate  
similar to max pooling



RoI conv features:  
 $C \times h \times w$   
for region proposal

Fully-connected  
layers

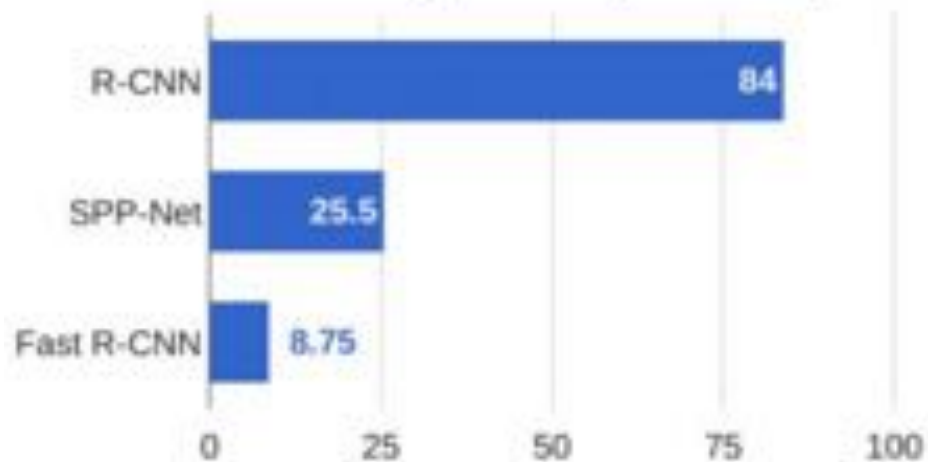


Fully-connected layers expect  
low-res conv features:  
 $C \times h \times w$

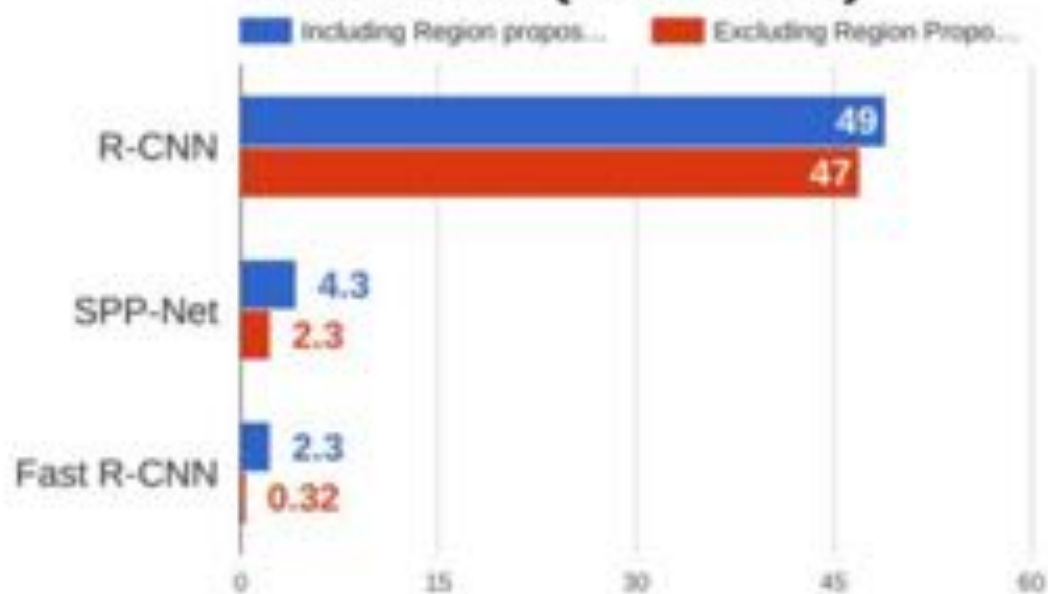
# Velocidade

## R-CNN vs SPP vs Fast R-CNN

Training time (Hours)

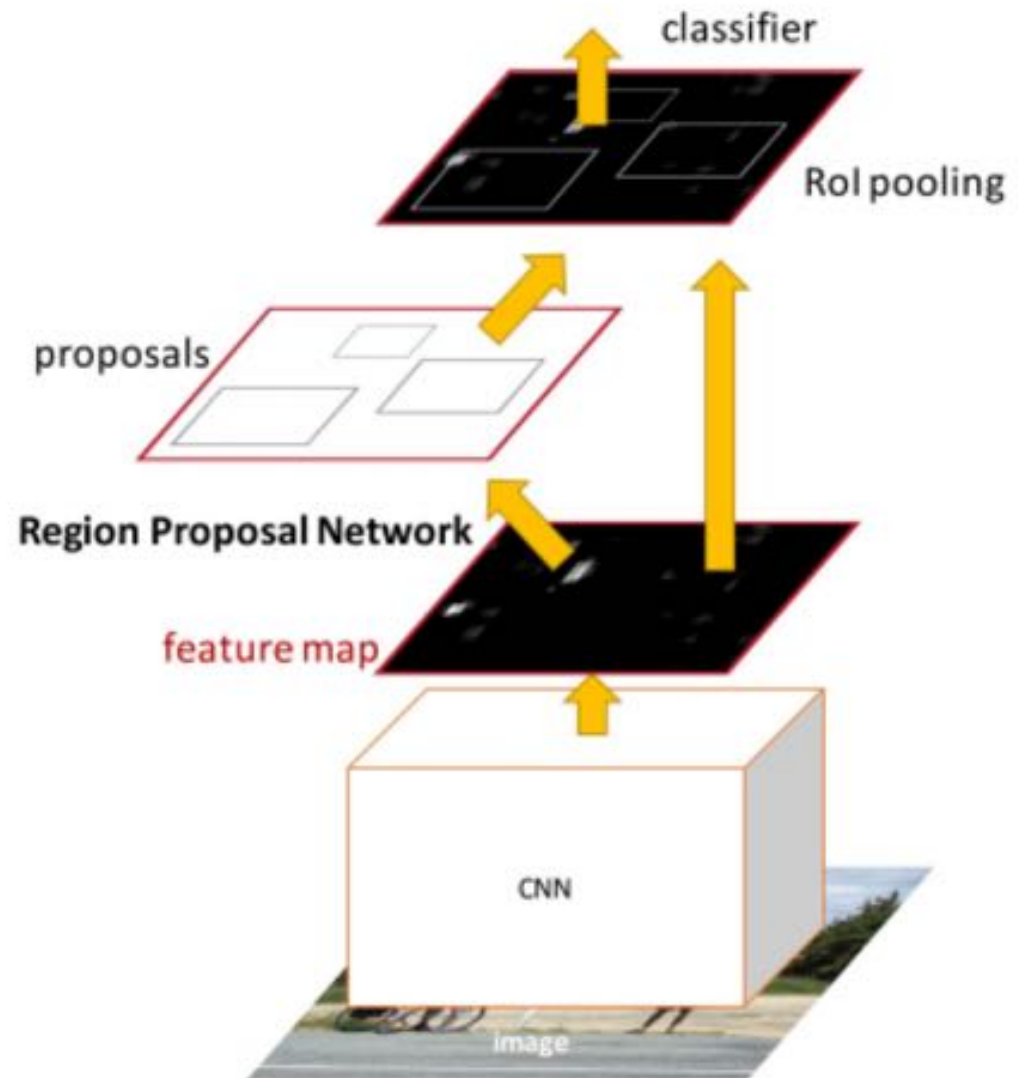


Test time (seconds)

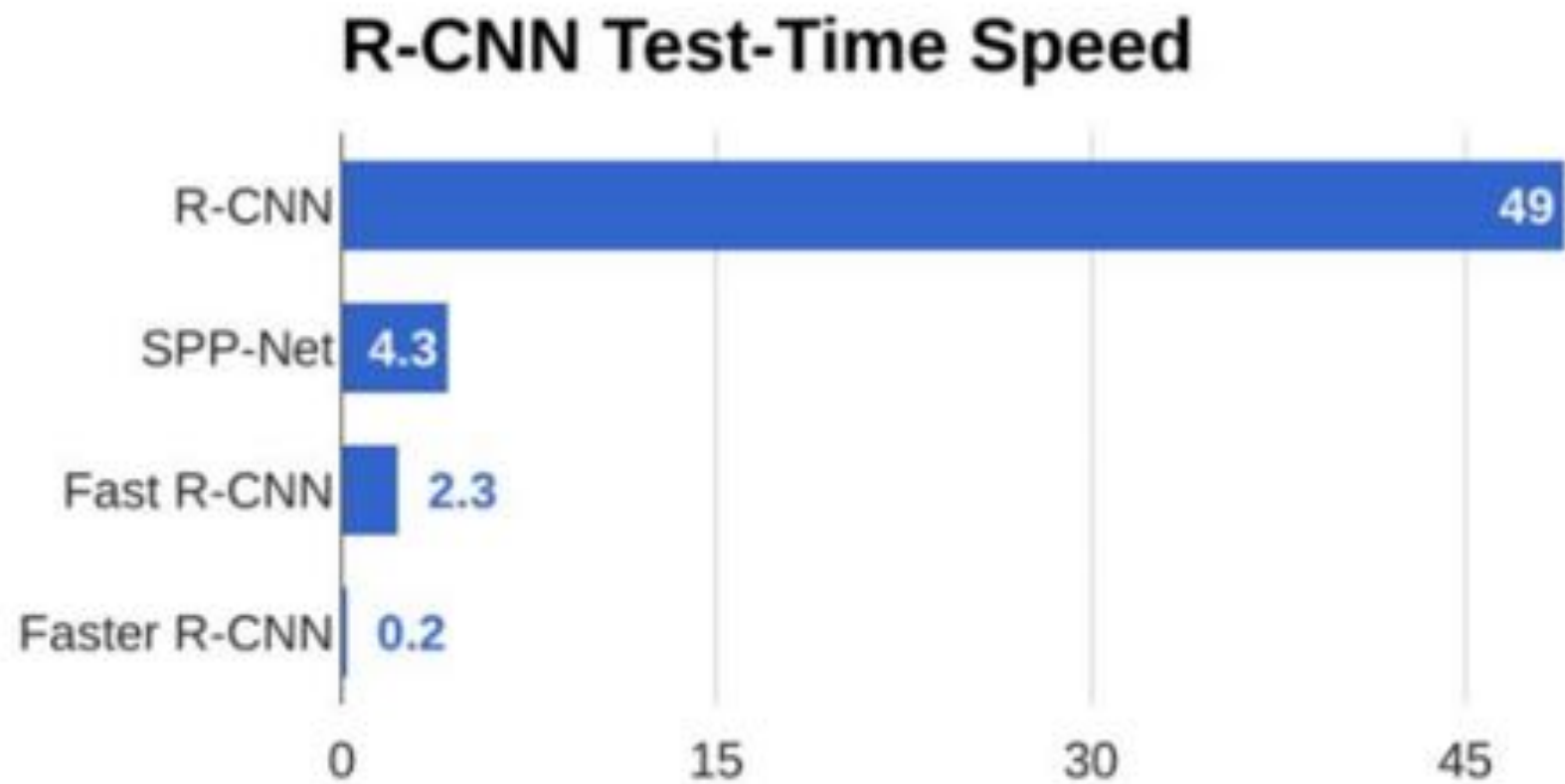


# Faster R-CNN

- Region Proposal Network: produz propostas baseadas no feature map
  - Usa a informação se tem fundo ou objeto para gerar uma proposta
- Dentro da mesma rede



# Velocidade



# Region Proposal Network: Steps

- 1) A imagem inteira passa por uma CNN o que gera um Feature Map



Convolutional  
Feature Maps





# Region Proposal Network: Steps

2) Sempre reduz o feature map para 512x3

3) Desliza uma janela pequena sobre o feature map

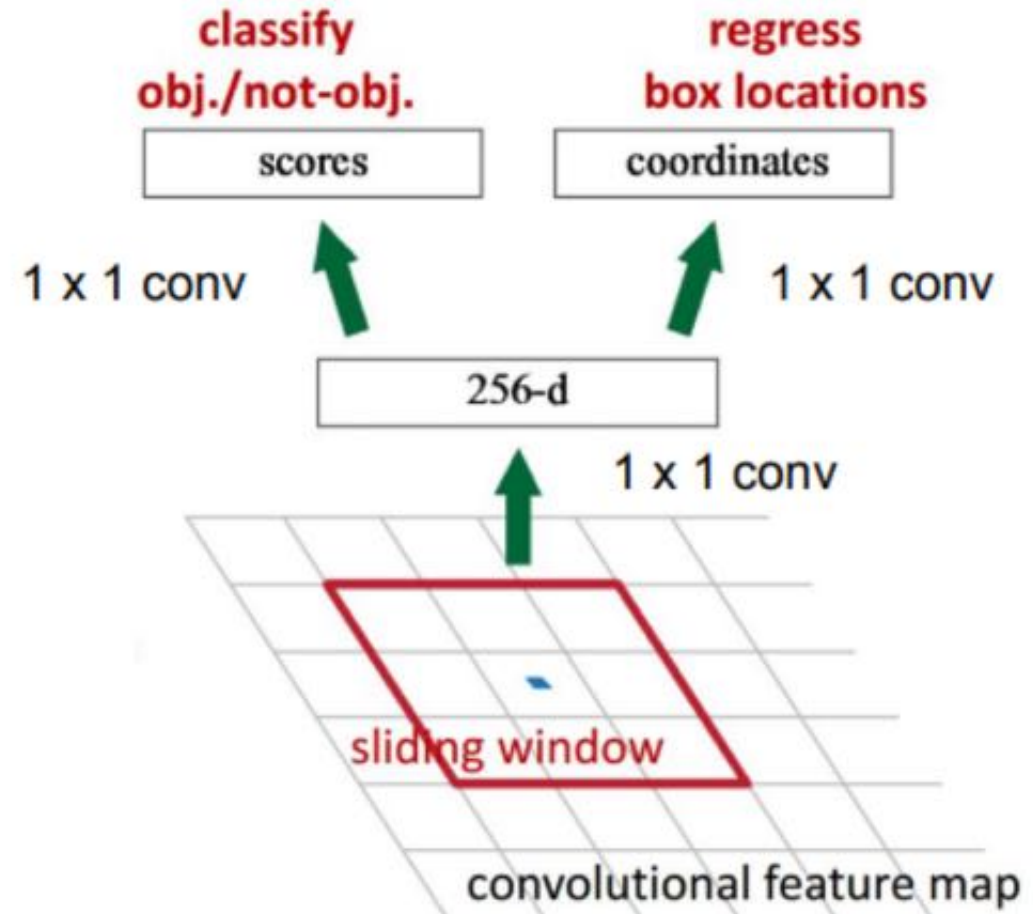
- Use stride 16, ou algo do tipo

4) Cria “âncoras” sobre cada janela

- Manda tudo isso para as duas cabeças a seguir:

5) Cria uma pequena rede:

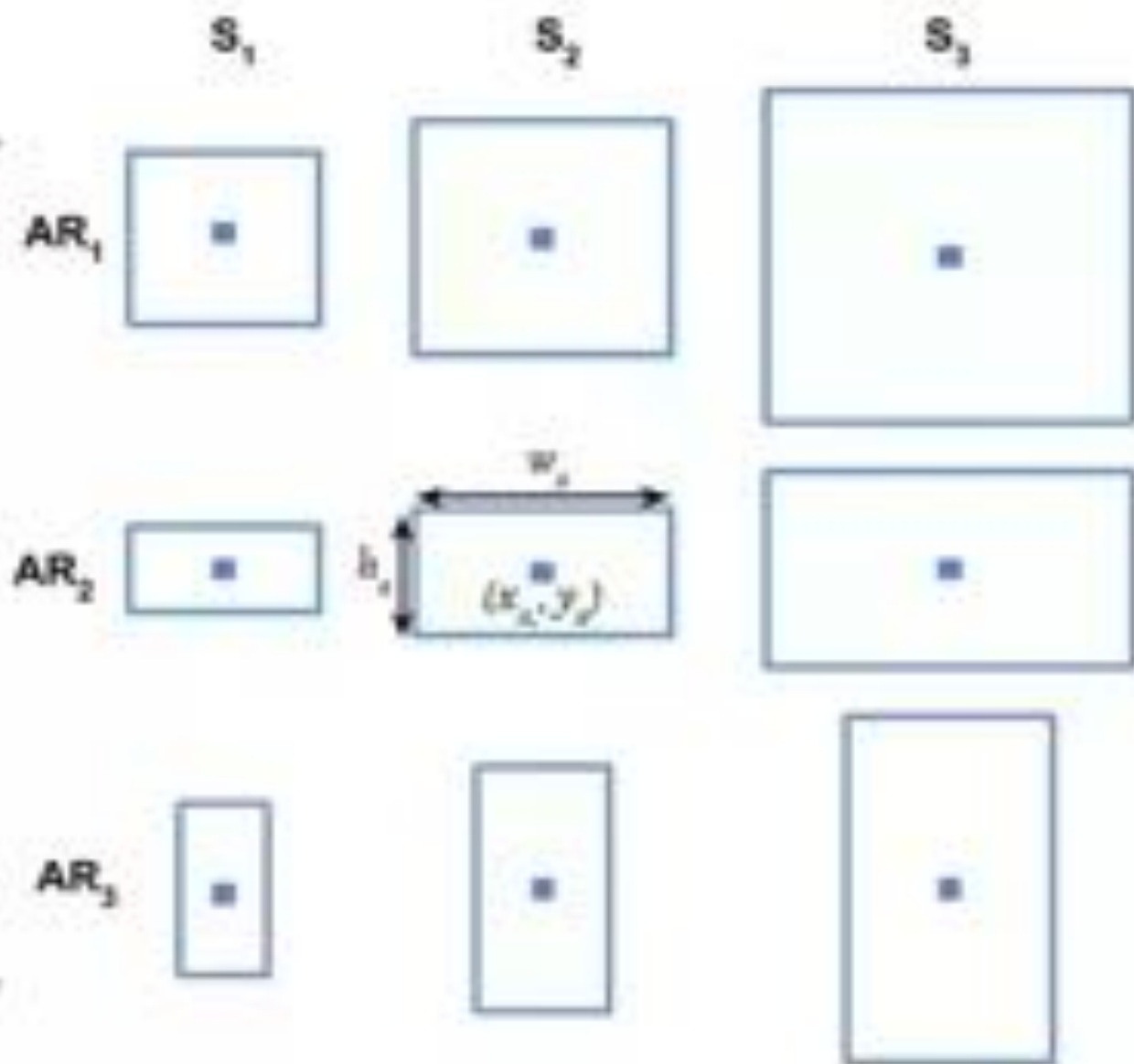
- Classificação (objeto ou fundo)
- Regressão do BBOX
- E daí saem as propostas



Generate 9 anchors for each **sliding window** on conv. feature map

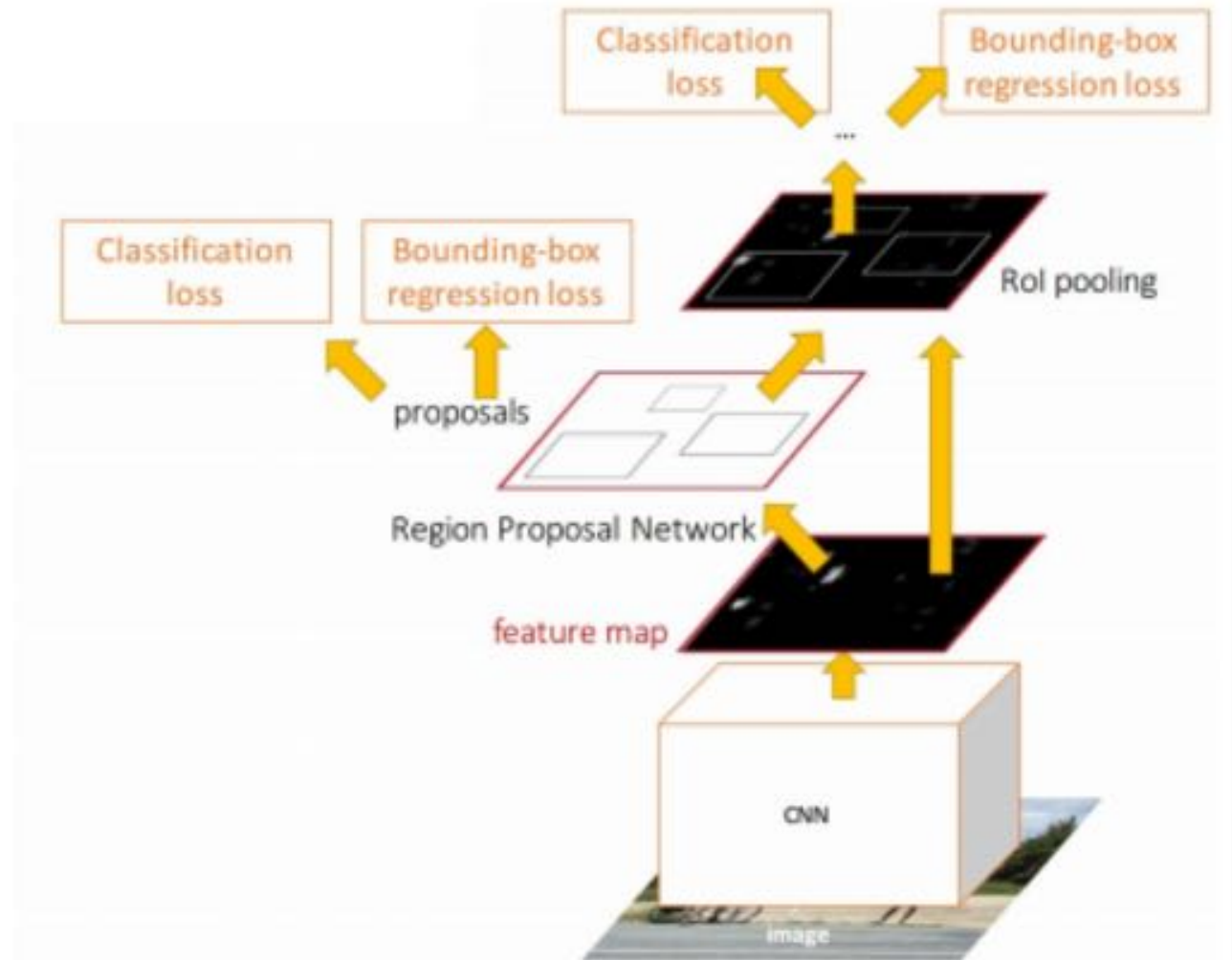


$w_a$ : anchor's width  
 $h_a$ : anchor's height  
 $(x_a, y_a)$ : anchor's center

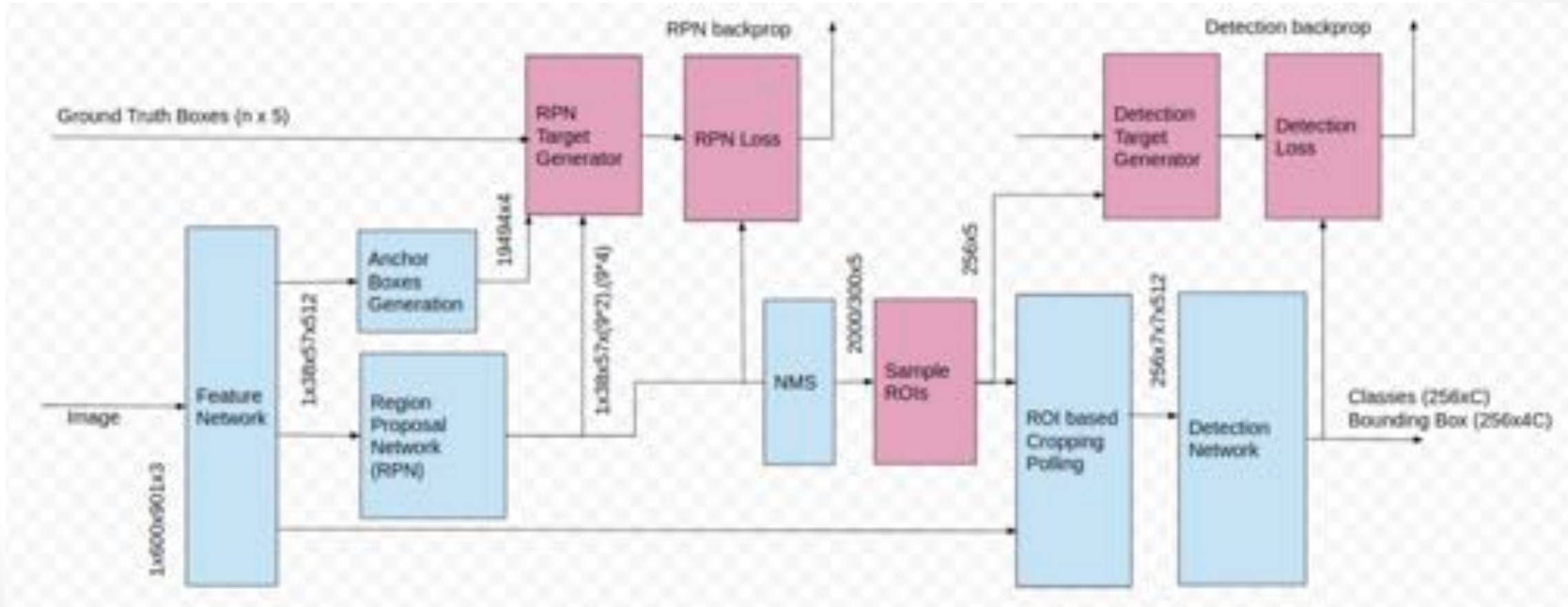


# Faster R-CNN

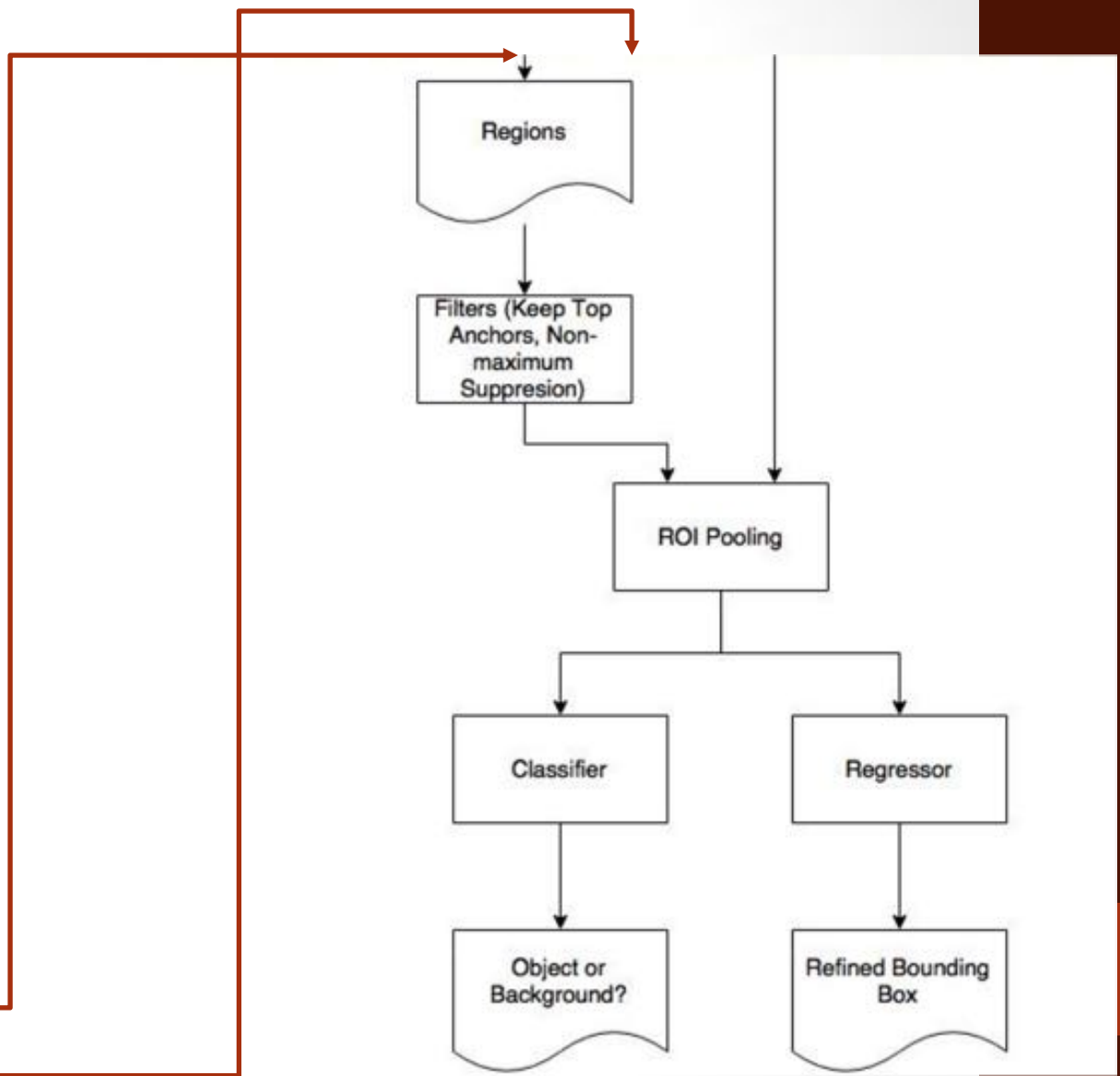
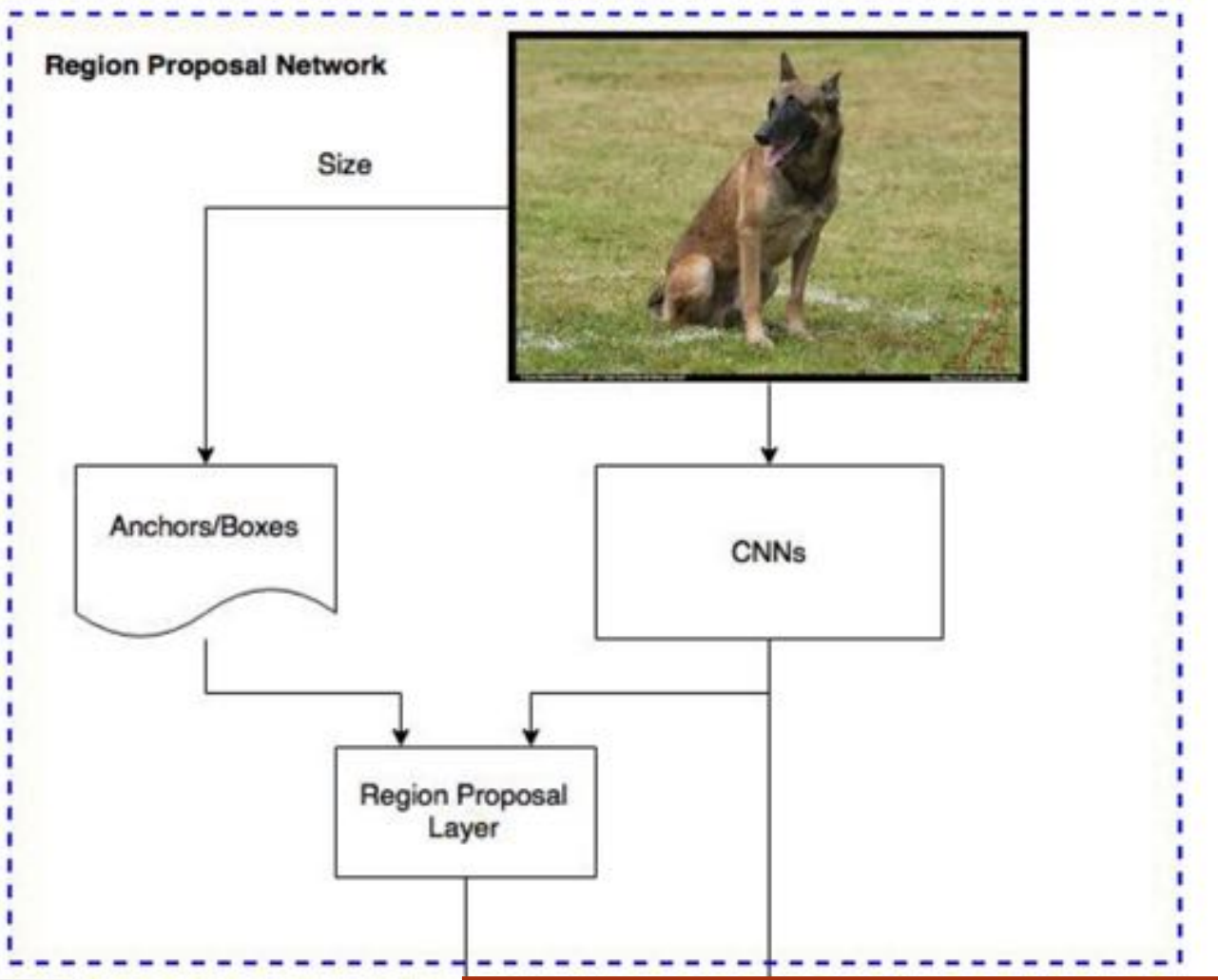
- E treina tudo junto !
  - Multi task loss
  - RPN class
  - RPN regressor
  - Fast R-CNN class
  - Fast R-CNN regressor



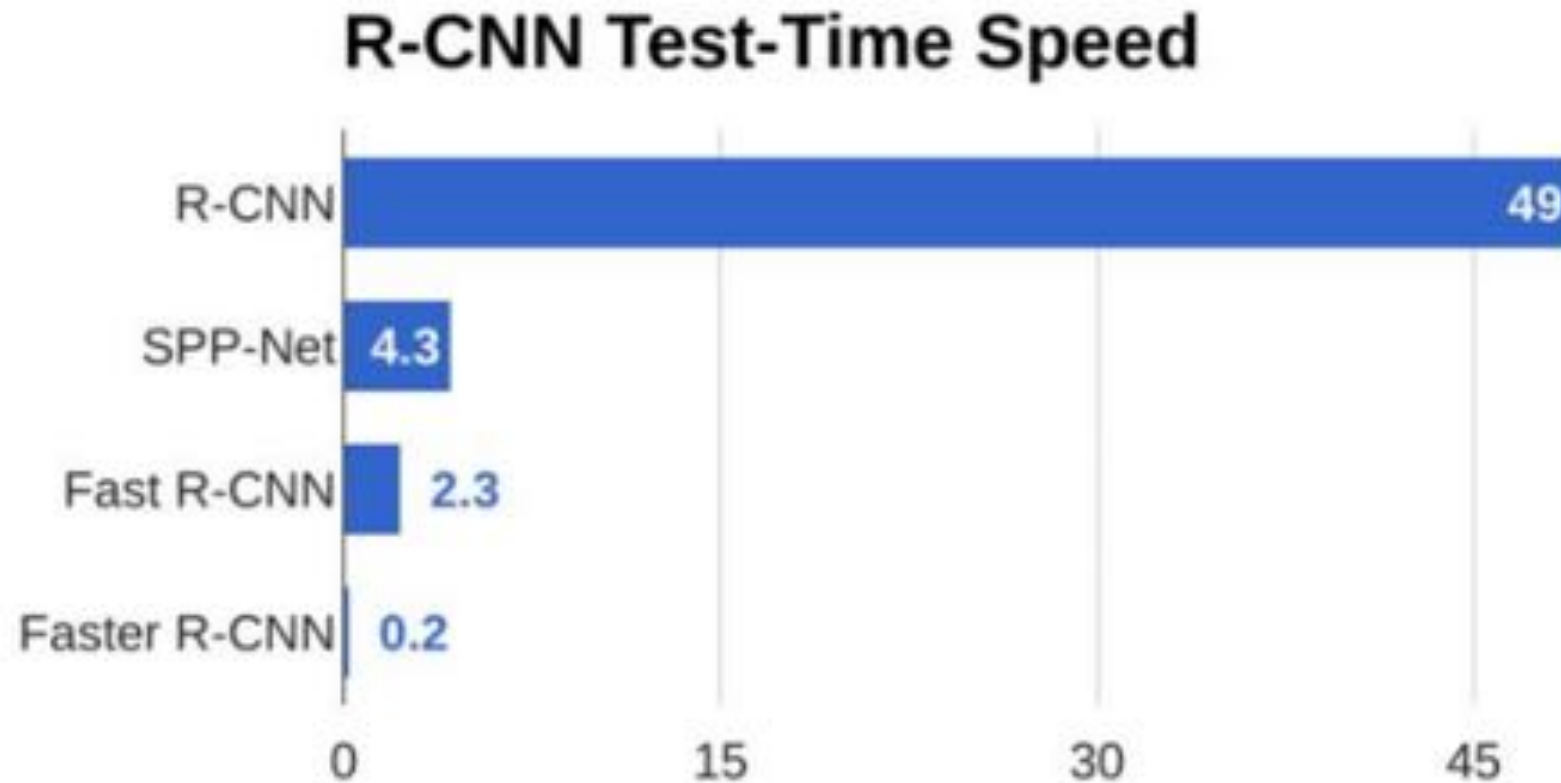
# Faster R-CNN



# Faster R-CNN



# E melhora a velocidade, apenas 1 treinamento e teste



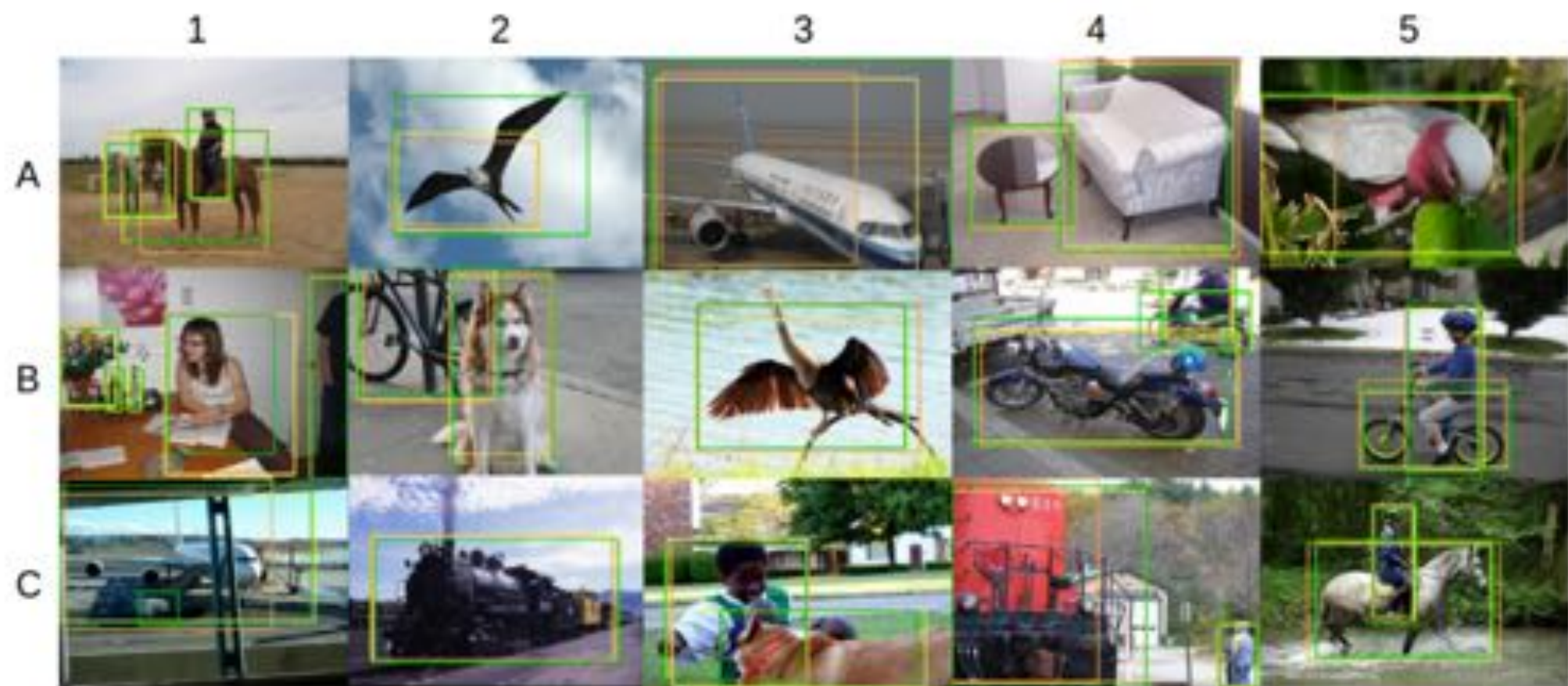


Figure 4. Visualization of iterative refinement model (Iteration 1, 2, and 3 are denoted by orange, yellow, and green)





- Cabuloso:

[https://www.youtube.com/watch?v=JC\\_vCqEoyeo](https://www.youtube.com/watch?v=JC_vCqEoyeo)

# Mais para ver?

- Este é apenas o começo
- Sugestões:
  - Mask R-CNN
  - YOLO. (real time?: [https://www.youtube.com/watch?v=4eIBisqx9\\_g](https://www.youtube.com/watch?v=4eIBisqx9_g))