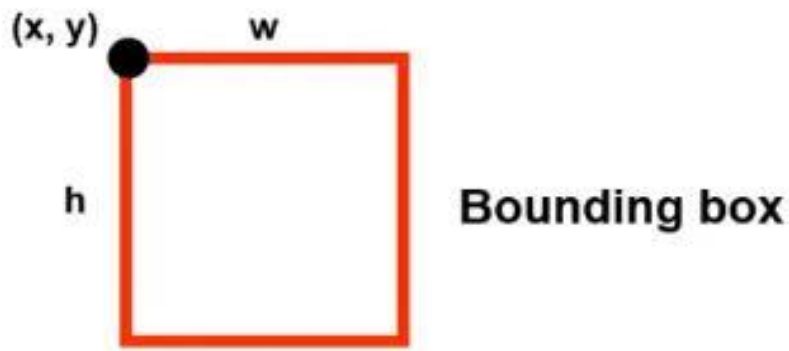
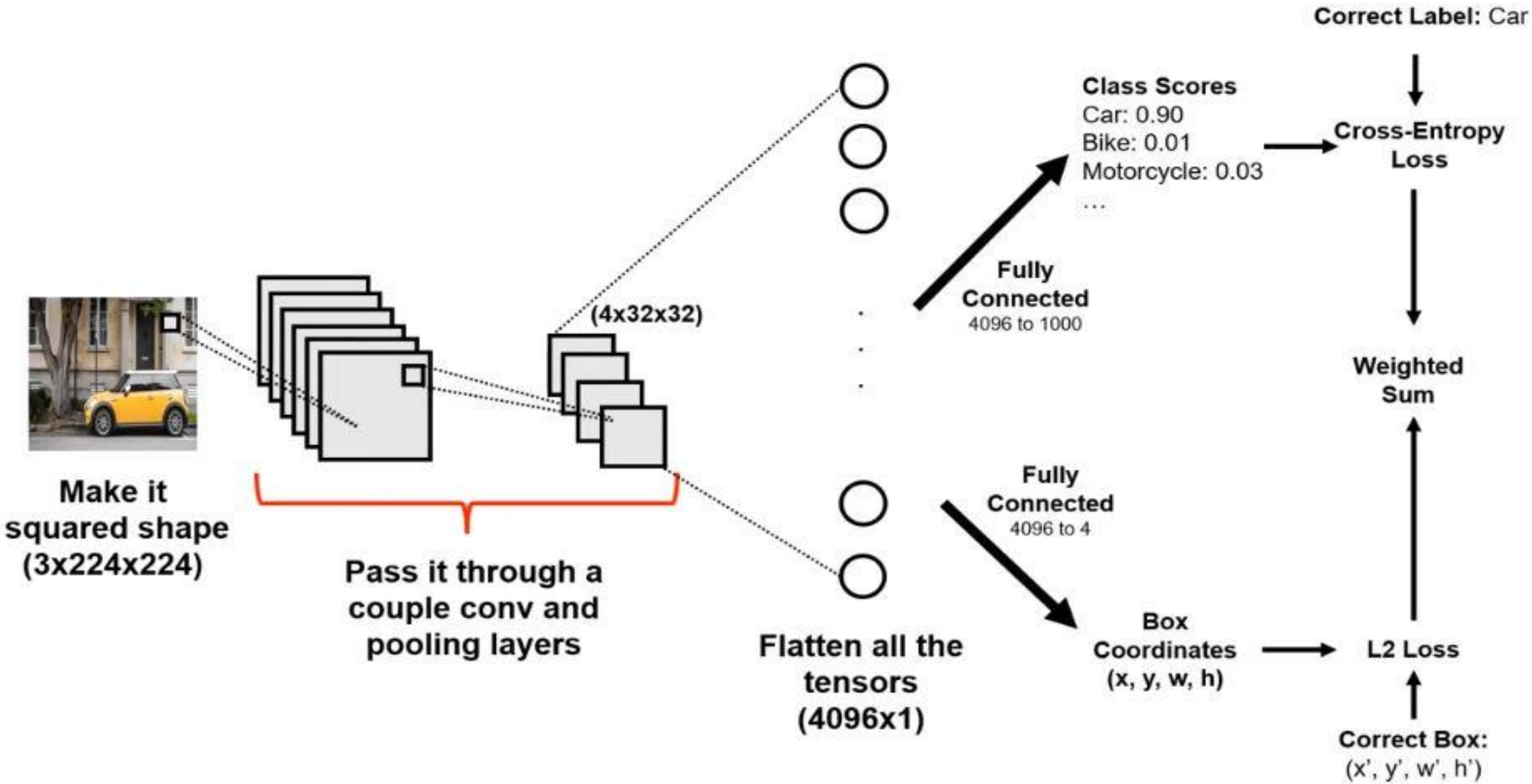

R-CNN

Our Goal

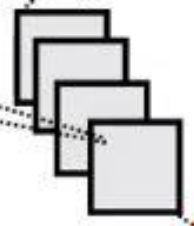


We receive input image


Correct Label: Car
Class Scores
Car: 0.90
Bike: 0.01
Motorcycle: 0.03
...
Cross-Entropy Loss
Fully Connected
4096 to 1000
(4x32x32)
Weighted Sum
Make it squared shape (3x224x224)
Pass it through a couple conv and pooling layers
Flatten all the tensors (4096x1)
Fully Connected
4096 to 4
Box Coordinates (x, y, w, h)
L2 Loss
Correct Box: (x', y', w', h')

Disadvantage: Only one object can be detected :(

Make it squared shape (3x224x224)



(4x32x32)

Pass it through a couple conv and pooling layers

Flatten all the tensors (4096x1)

Fully Connected 4096 to 1000

Class Scores
Car: 0.90
Bike: 0.01
Motorcycle: 0.03
…

Correct Label: Car

Cross-Entropy Loss

Weighted Sum → Loss

Fully Connected 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

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

Earliest Approach

Earliest Approach

Sliding Window

Earliest Approach

Classify this region!

Neural Network classifier

c + 1 outputs

Car
Motorcycle
Bike
…

Background

Earliest Approach

Classify this region!

Neural Network classifier

Mountain

Earliest Approach

Classify this region!

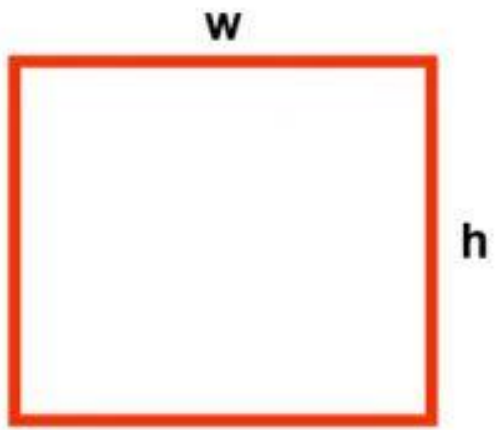
Neural Network classifier

Car

Earliest Approach

W

H



Possible Positions:

(W – w + 1) * (H – h + 1)

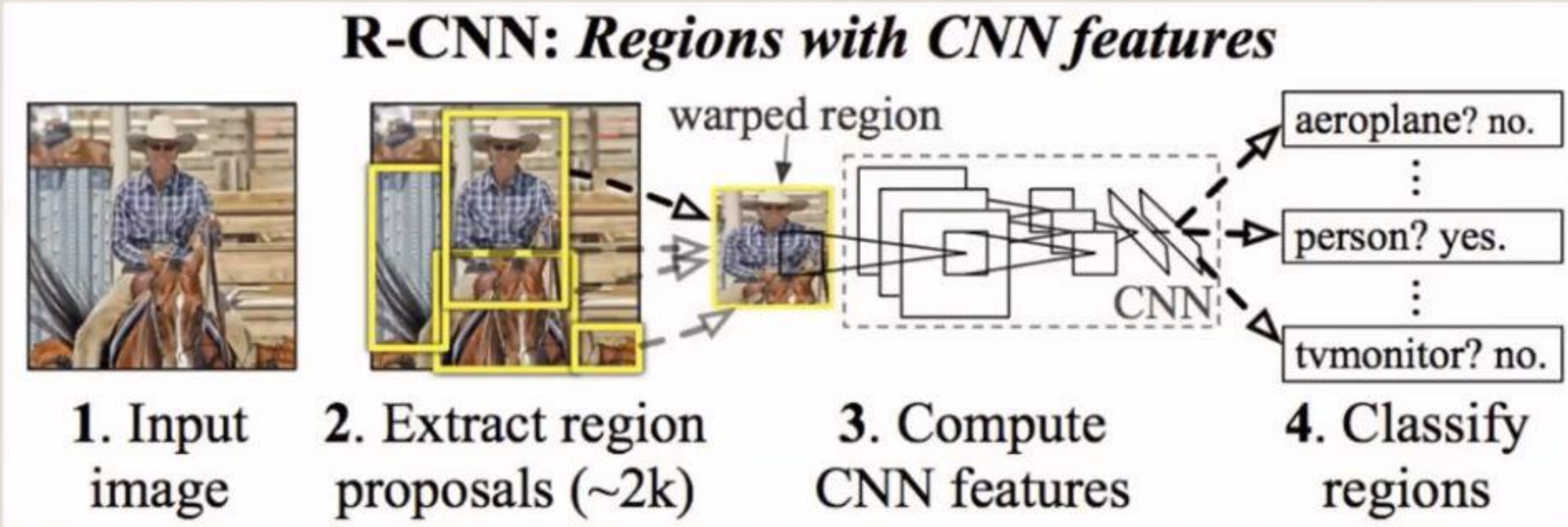
CNN as feature extractor

- What could be the problems?
  - Suppose we have a 600 x 600 image, if sliding window size is 20 x 20, then have (600-20+1) x (600-20+1) = ~330,000 windows
  - Sometimes we want to have more accurate results -> multi-scale detection
    - Resize image
    - Multi-scale sliding window

Disadvantages

- Very Slow
- Number of Picked windows is very huge
- For CNN classifier, needs to apply convolution to each window content
- Same object will be detected in multiple windows (with different Bounding Boxes)

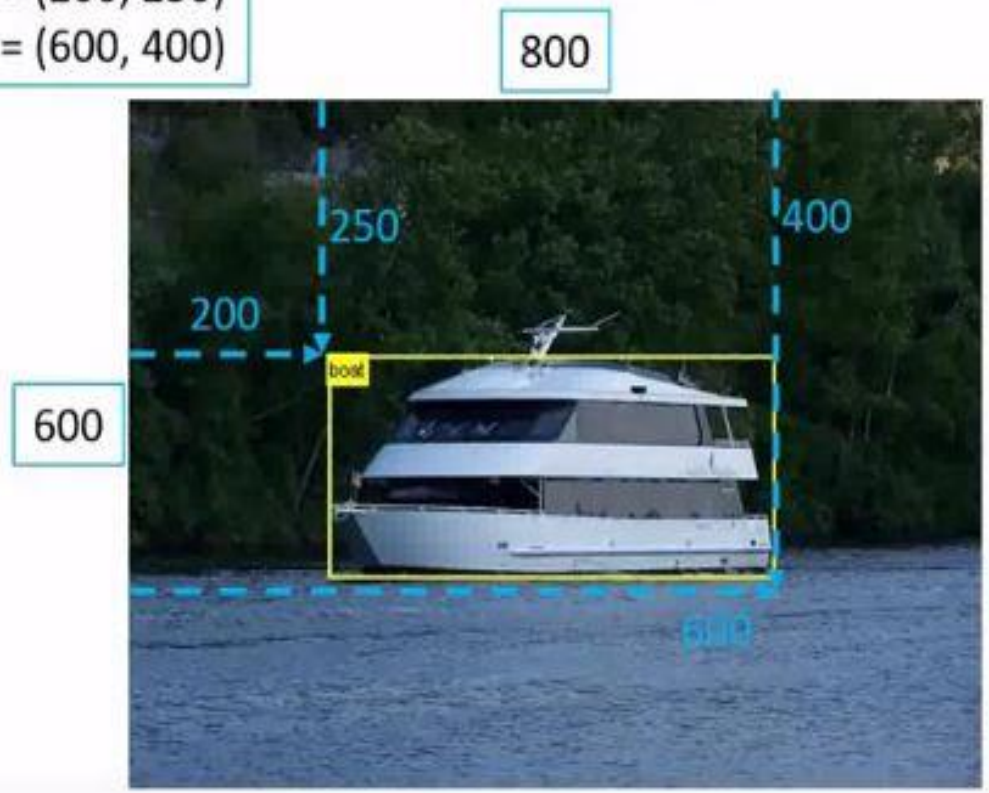
R-CNN

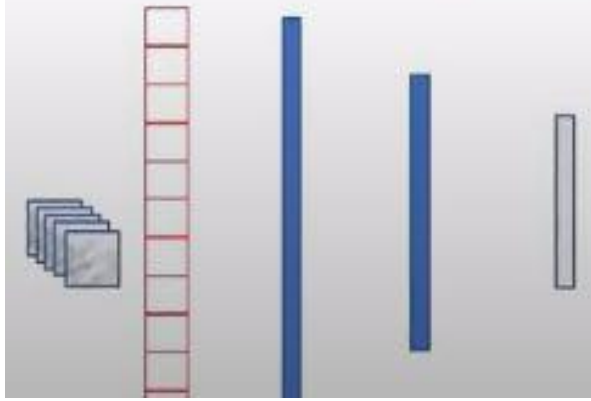


(Girshick, et al., 2014)

Bounding Box Regression Training

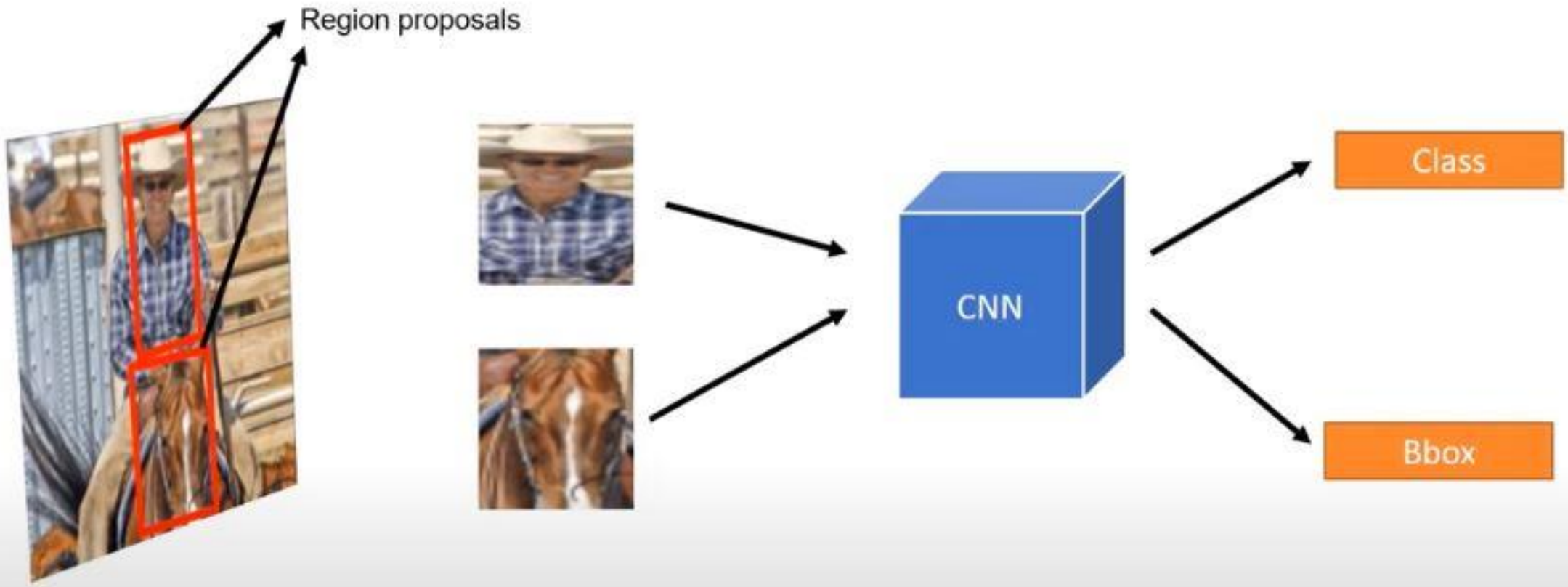
(x1,y1) = (200, 250)
(x2,y2) = (600, 400)





	x1	y1	x2	y2	L2 Loss				
Expected	200	250	600	400					
Prediction	0	0	800	600	$(200-0)^2$	$(250-0)^2$	$(600-800)^2$	$(400-600)^2$	182500
	100	150	700	450	$(200-100)^2$	$(250-150)^2$	$(600-700)^2$	$(400-450)^2$	32500
	210	245	590	405	$(200-210)^2$	$(250-245)^2$	$(600-590)^2$	$(400-405)^2$	250
	200	250	600	400	$(200-200)^2$	$(250-250)^2$	$(600-600)^2$	$(400-400)^2$	0

R-CNN



R-CNN

Region proposal: (p_x, p_y, p_h, p_w)

Bbox

Transform: (t_x, t_y, t_h, t_w)

Output: (b_x, b_y, b_h, b_w)

Translation:

$b_x = p_x + p_w t_w$ (Horizontal translation)

$b_y = p_y + p_h t_h$ (Vertical translation)

Log-space scale transform:

$b_w = p_w \exp(t_w)$ (Horizontal scale)

$b_h = p_h \exp(t_h)$ (Vertical scale)

Region Based CNN

R-CNN (Region proposal + CNN)

Selective Search

Color

Texture

Circles and Curves

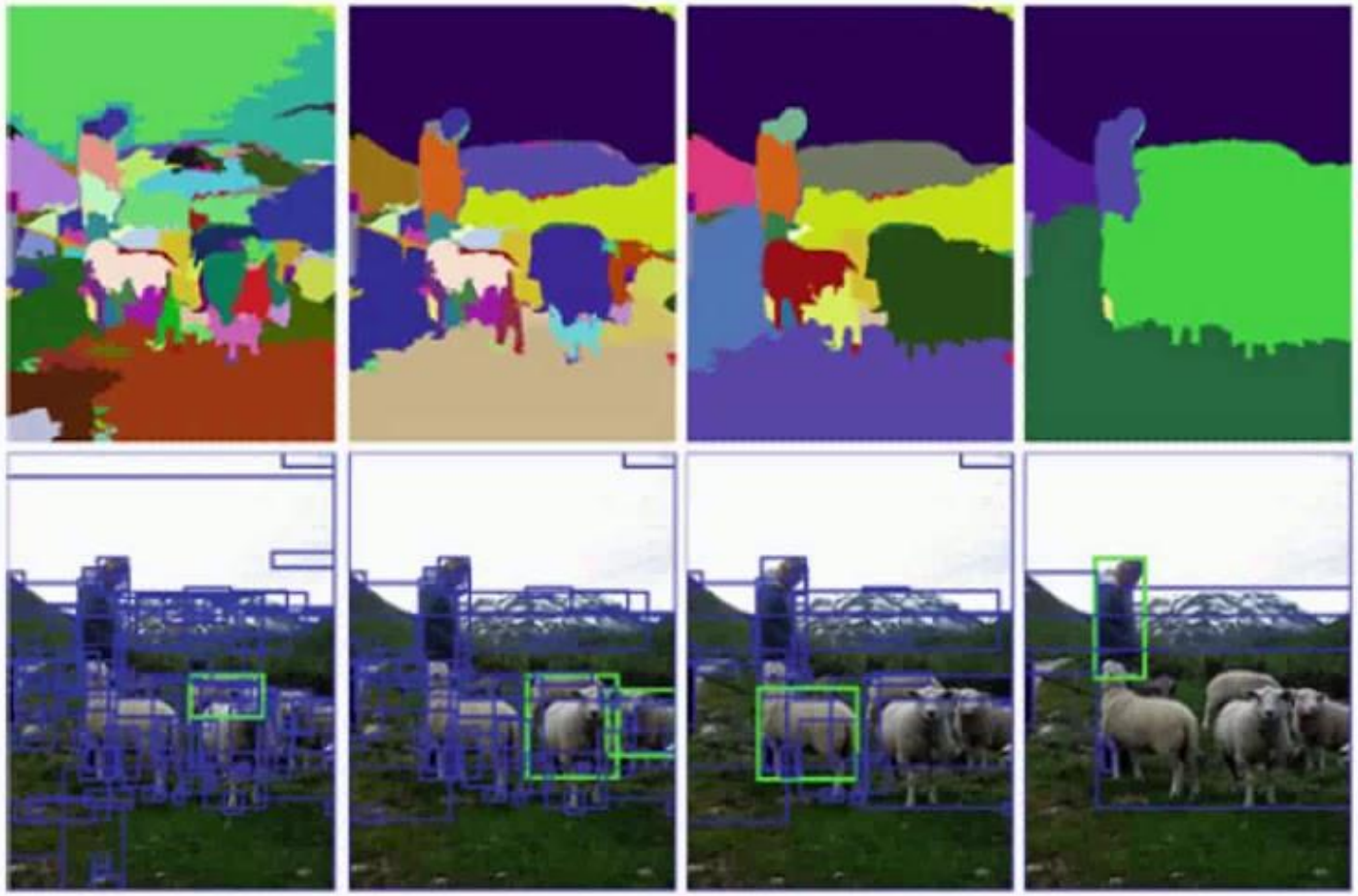
Selective Search (simplified)

- Group based on intensity of the pixels.

(Chandel, 2017)

- We cannot directly use the segmented image as region proposals!
- Group adjacent segments by similarity.

Region Proposal Techniques

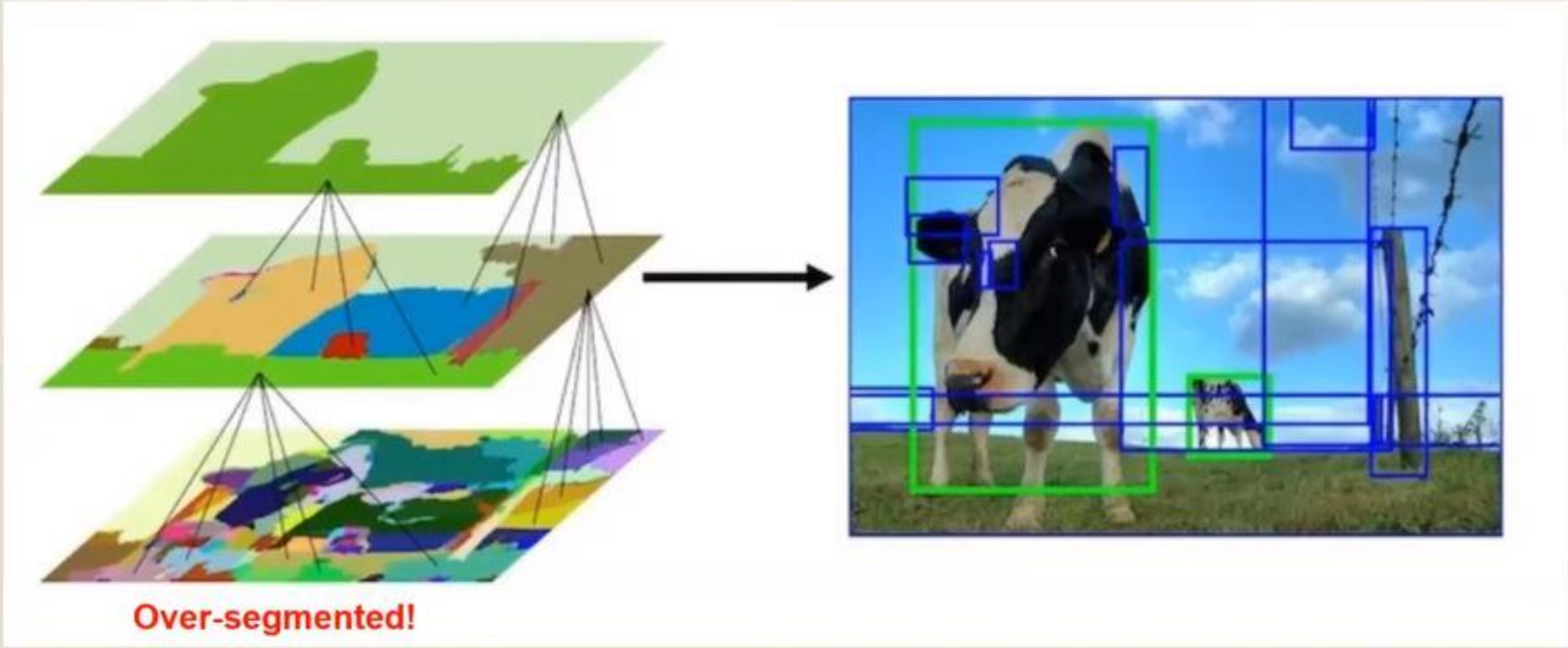


Superpixels Straddling

0	1	4	5	6	0	235	255
1	2	3	3	0	0	240	255
1	2	3	4	0	0	255	255
3	3	4	67	130	120	245	255
5	4	50	120	115	226	250	255
56	45	60	115	120	223	255	255
57	48	58	110	111	215	255	255
59	150	125	255	225	220	255	255

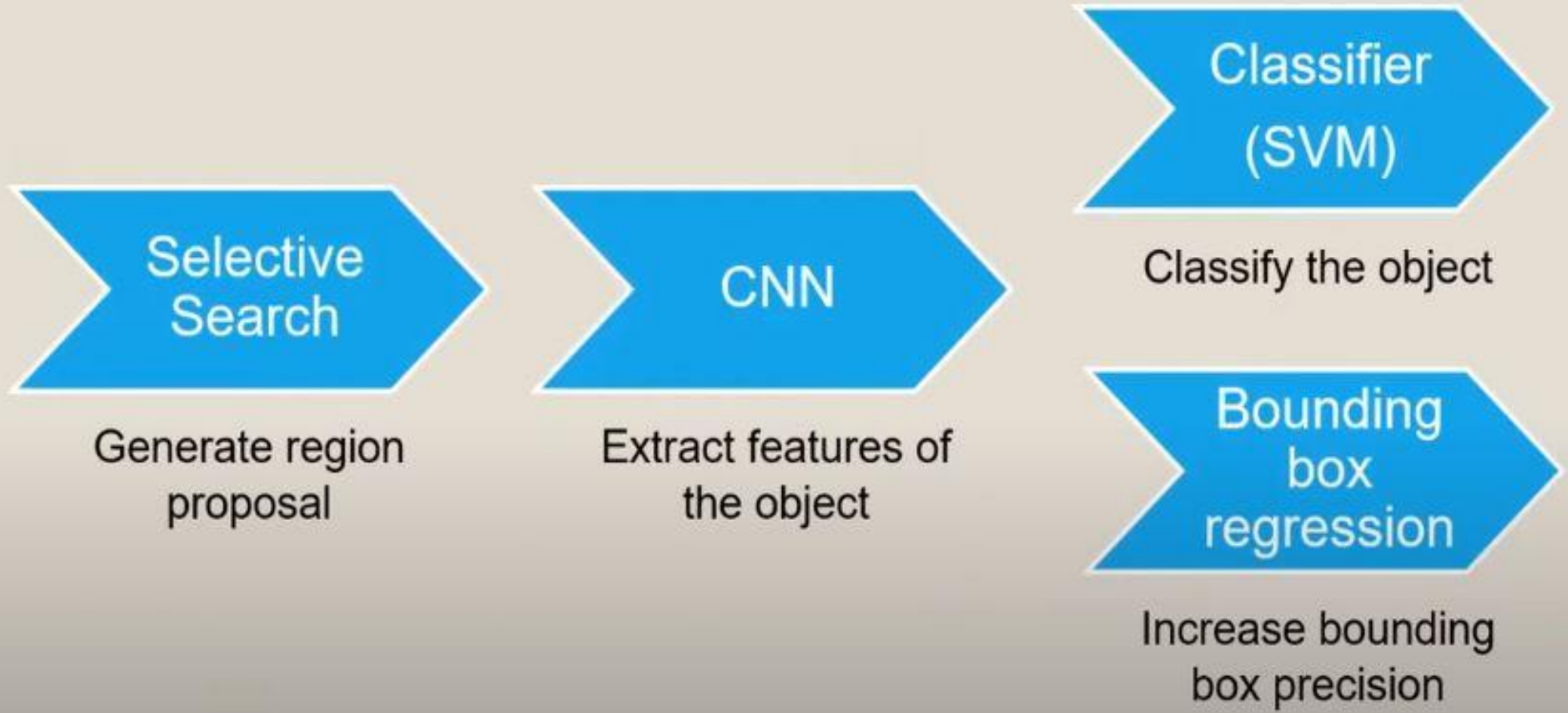
Selective Search for Object Recognition J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders

Selective Search (simplified)



(Chandel, 2017)

R-CNN in a Nutshell...



Disadvantages

- Very Slow
- Cropping of proposed regions
- Requires to apply convolution on each proposed image
- Same object will be detected in multiple windows (with different Bounding Boxes) due to NOT-Optimized Proposed regions

Solution

- Save time by doing CNN one time only on the whole input image