

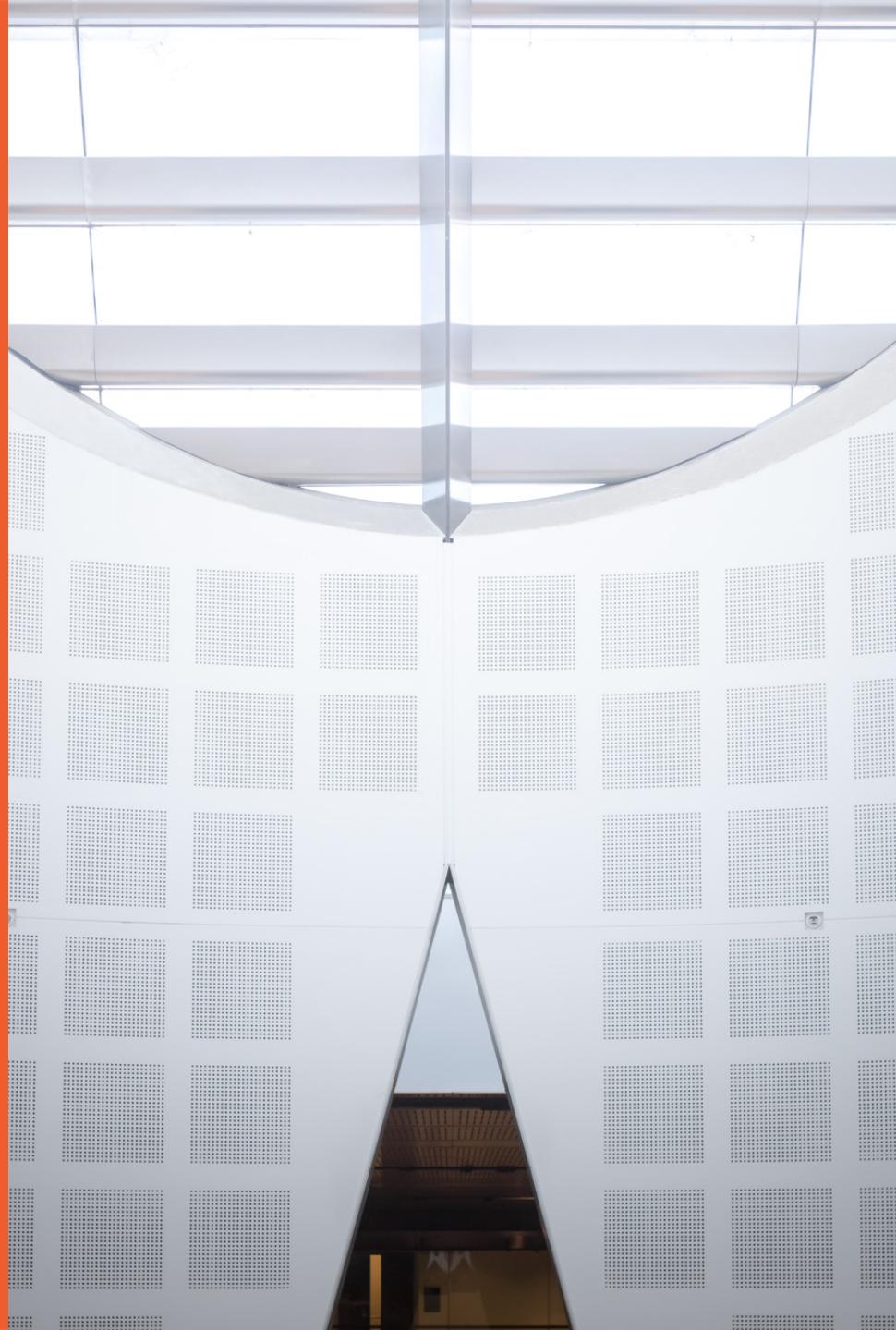
Deep Learning Applications

Dr Chang Xu

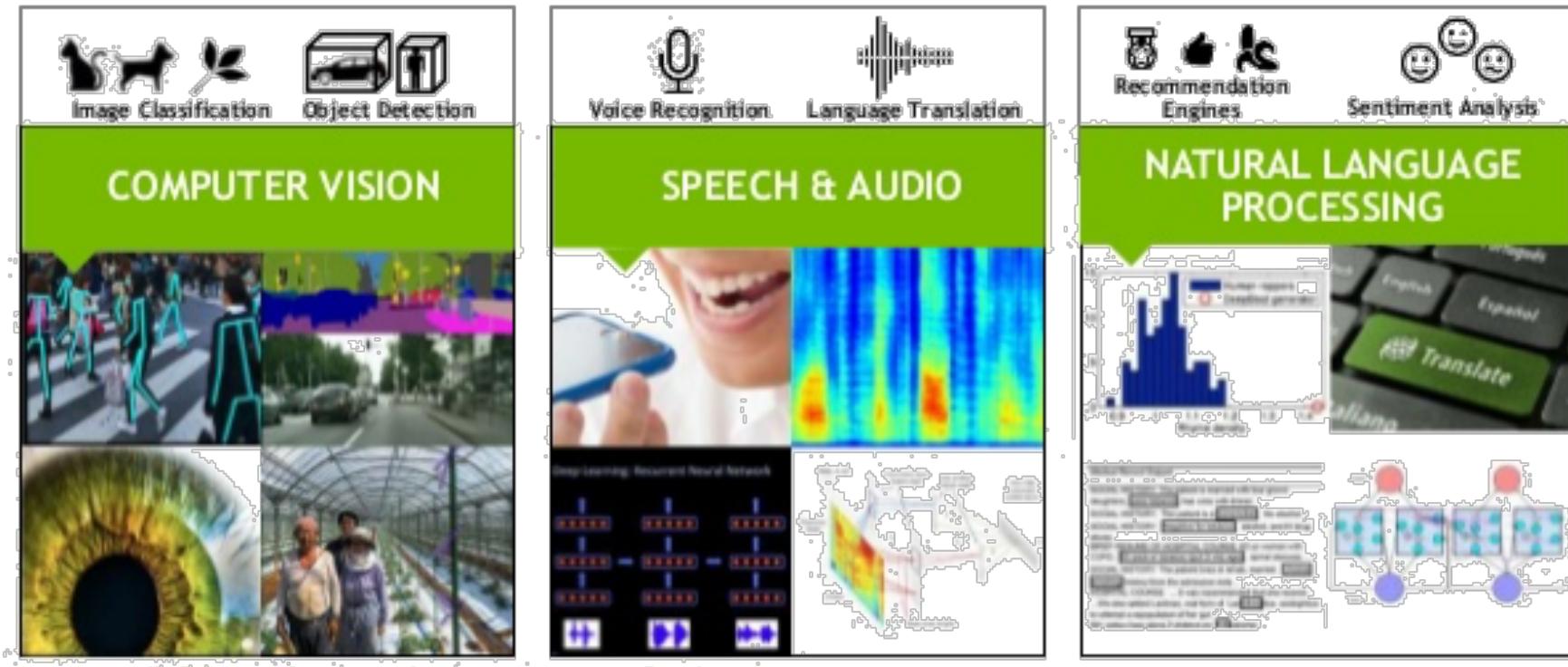
School of Computer Science



THE UNIVERSITY OF
SYDNEY

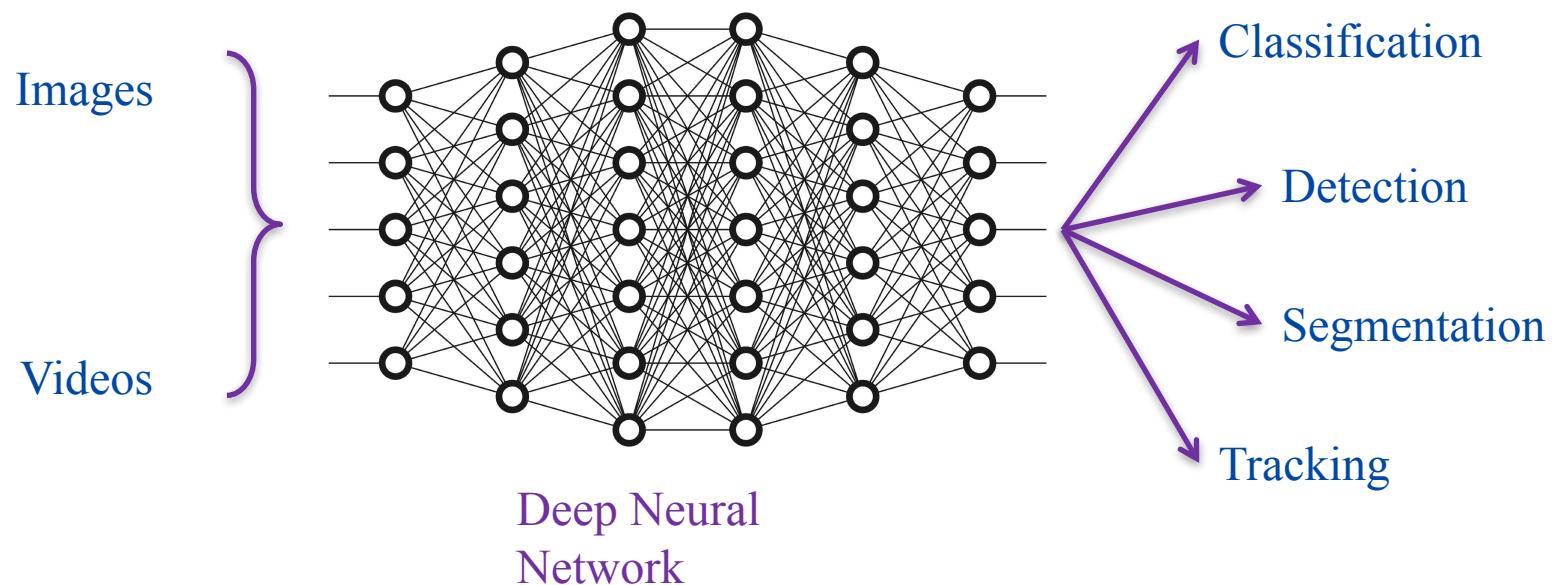


Applications



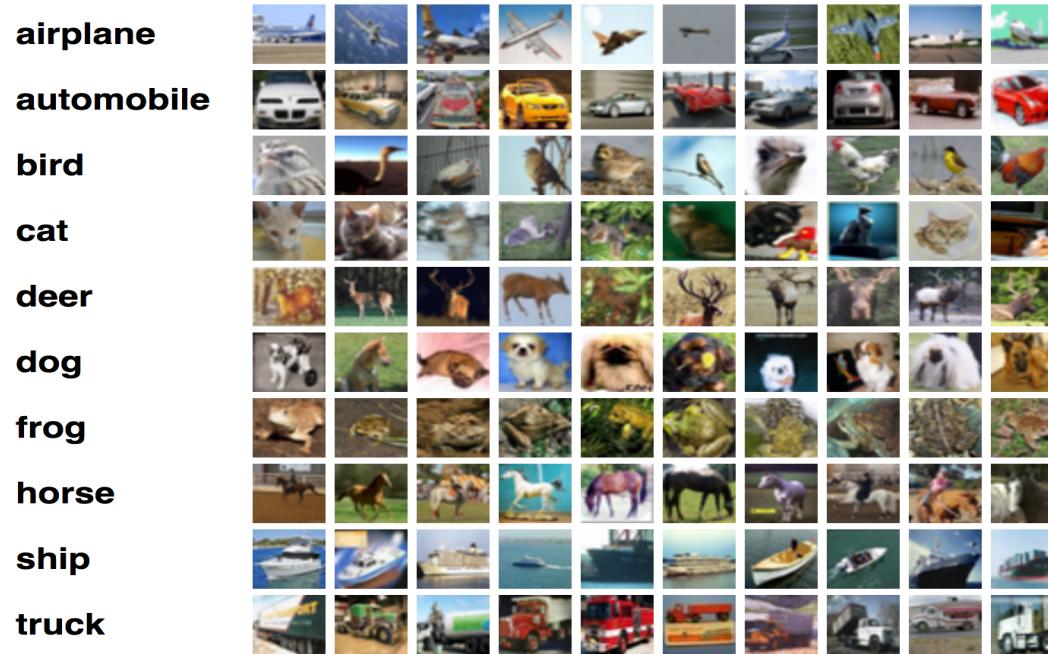
Introducing Deep Learning for Computer Vision

Deep convolutional neural network (DCNN) is the key concept for introducing deep learning to the development of computer vision. By imitating the biological nervous systems, deep neural networks can provide unprecedented ability to interpret complicated data patterns and thus effectively tackle various computer vision tasks.



Classification

Goal: Assign a label to an input image based on a fixed set of categories.



Credit To: <https://medium.com/@tifa2up/image-classification-using-deep-neural-networks-a-beginner-friendly-approach-using-tensorflow-94b0a090cccd4>

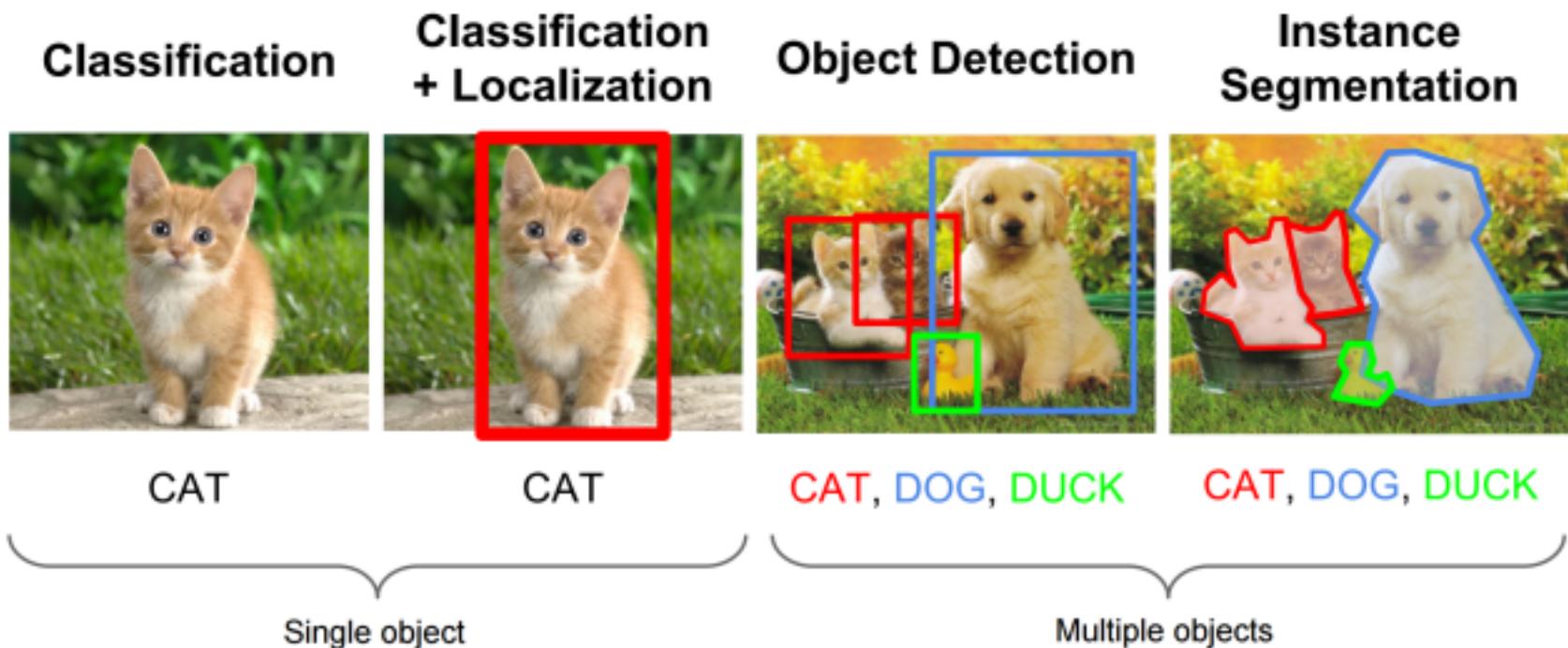
Deep Learning-based Classification

Suppose W is the parameter set of a deep neural network. Given an input image I , classification can be tackled by:

$$y = f(W, I)$$

where y is the predicted class label and f represents a series of operations parameterized based on W for processing the input image.

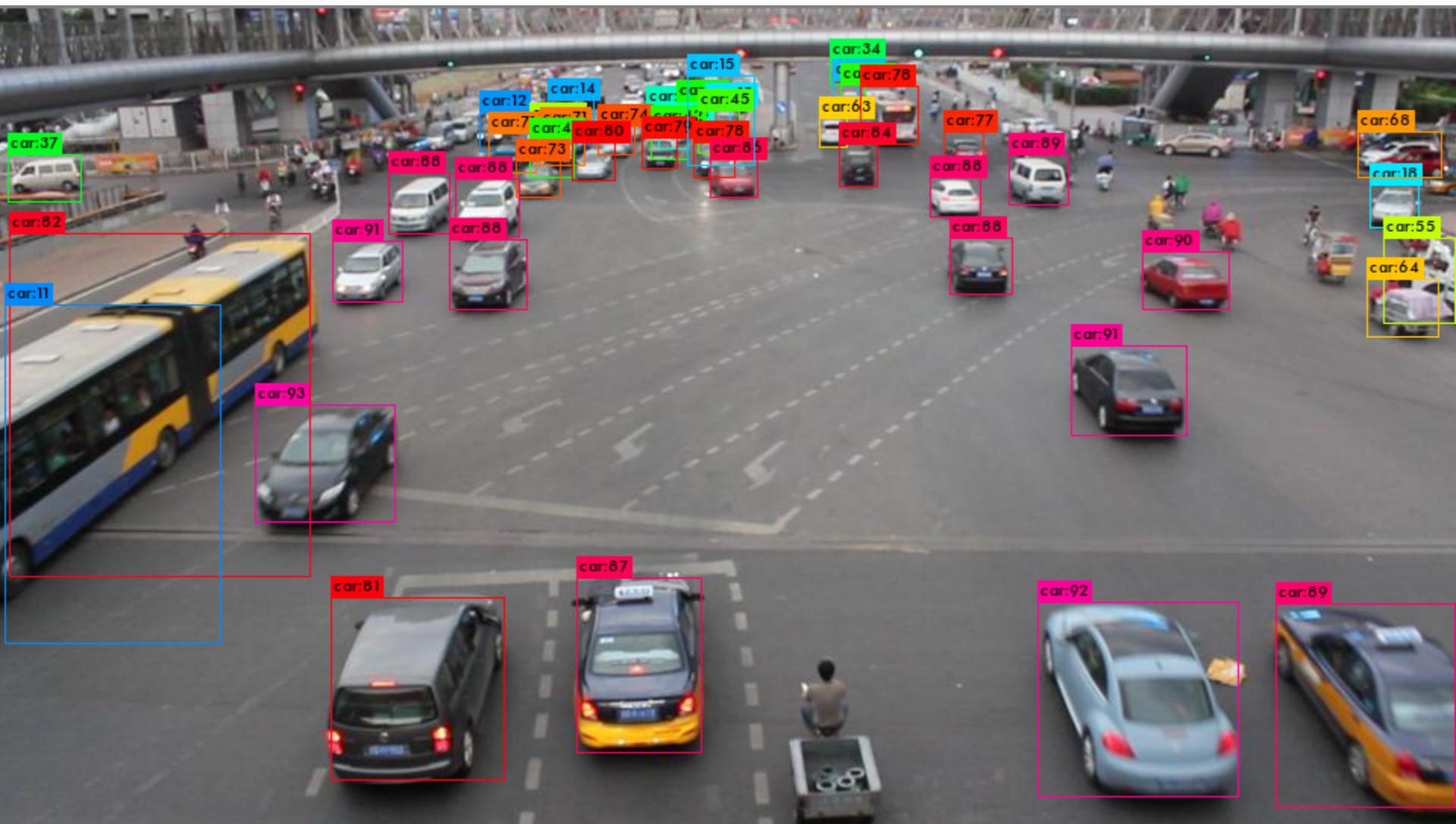
Computer Vision Tasks



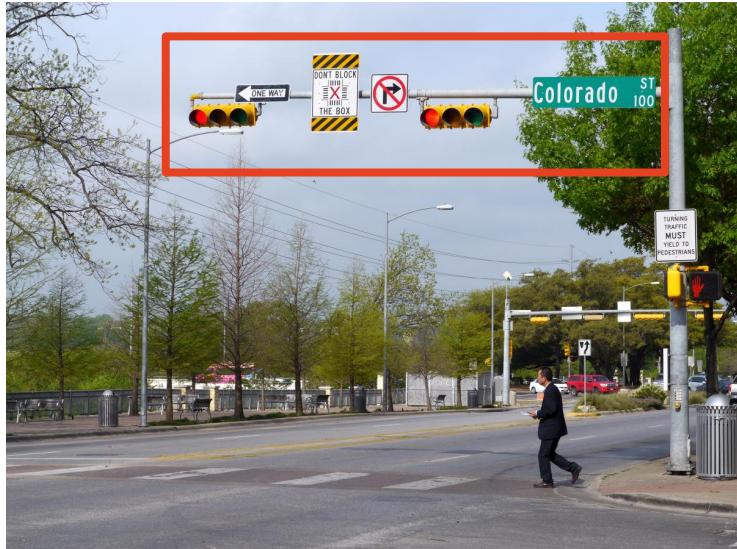
Pedestrian Detection



Car Detection



Other Applications



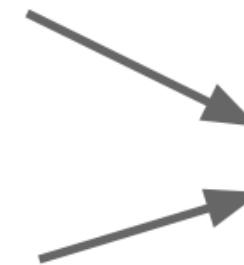
Localize objects with regression

Input: image



Neural Net
→

Output:
Box coordinates
(4 numbers)



Loss:
L2 distance

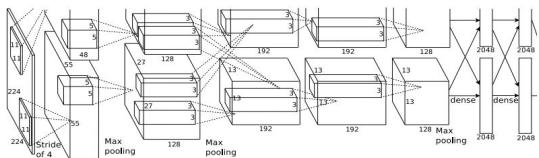
Correct output:
box coordinates
(4 numbers)

Only one object,
simpler than detection

Classification with Localization

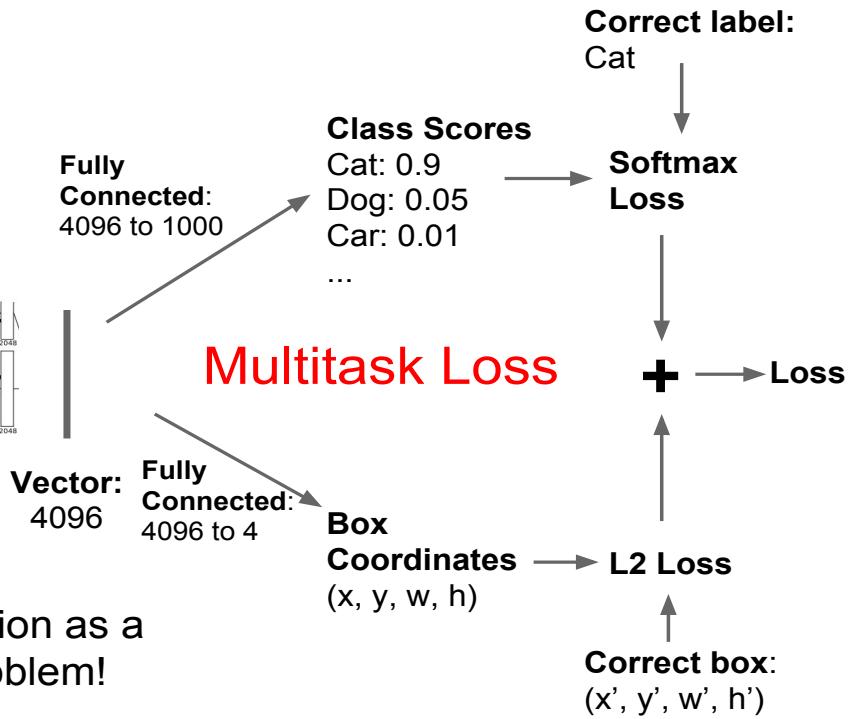


This image is CC0 public domain



Treat localization as a regression problem!

It is assumed that there is only one object in an input image.

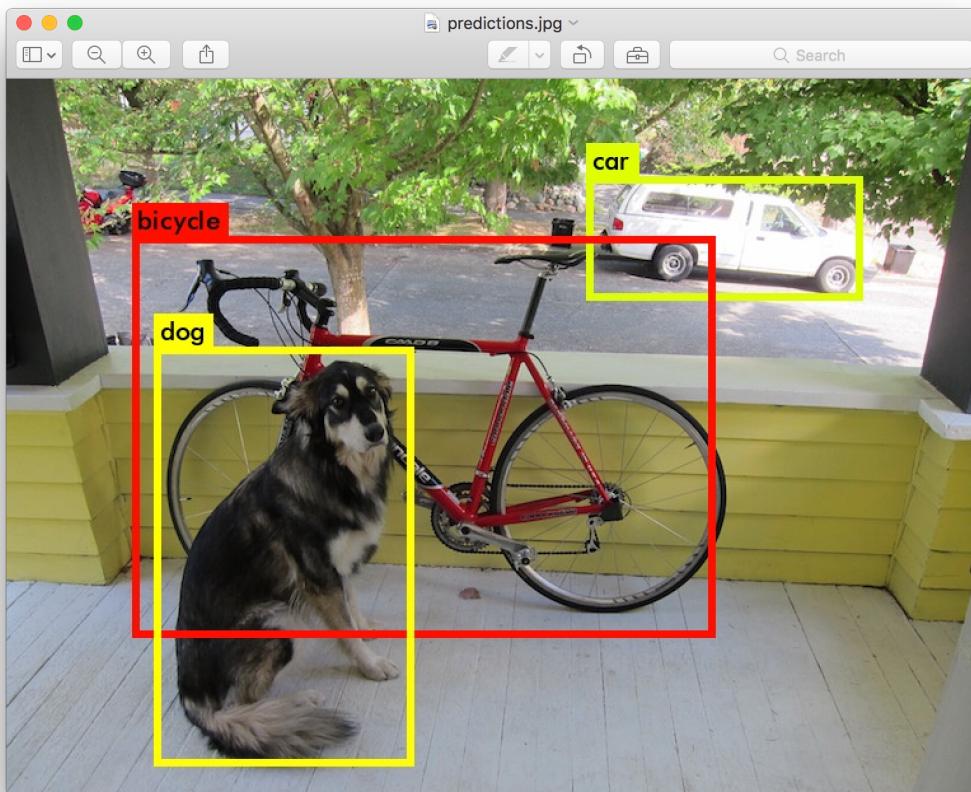


COCO DATASET FORMAT



Detection

Goal: Detect semantic objects of certain classes in images.

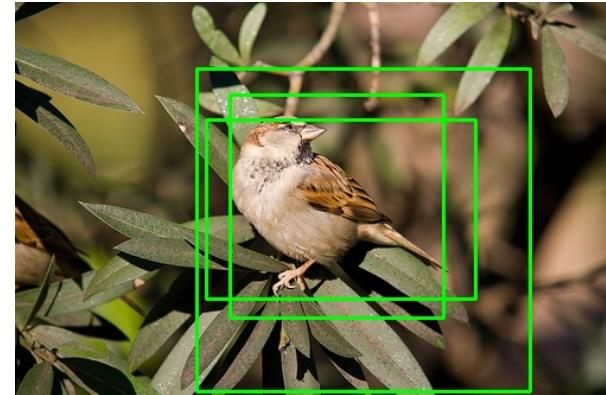


- Region CNN
- SPPNet
- Fast RCNN
- Faster RCNN
- Mask R-CNN

Typical architecture



Proposals



1. Region proposal: Given an input image, find all possible places where objects can be located. The output of this stage should be a list of bounding boxes of likely positions of objects. These are often called region proposals or regions of interest.

2. Final classification: for every region proposal from the previous stage, decide whether it belongs to one of the target classes or to the background. Here we could use a deep convolutional network.

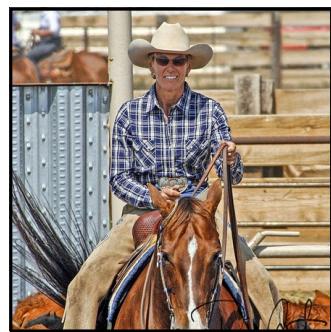
Deep Learning-based Detection

Given a deep neural network parameterized by W , the goal of object detection is to predict a set of bounding boxes that may contain objects together as well as the objects' categories.

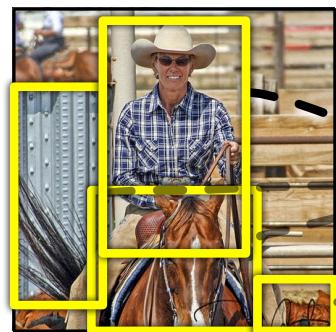
$$\{y_i, (x_1, y_1, x_2, y_2)_i\} = f(W, I)$$

where y_i is the predicted class label and $(x_1, y_1, x_2, y_2)_i$ describes the four coordinates (i.e. the top-left coordinate and the bottom-right coordinate) of the estimated bounding box for the i -th result.

Region CNN (R-CNN)

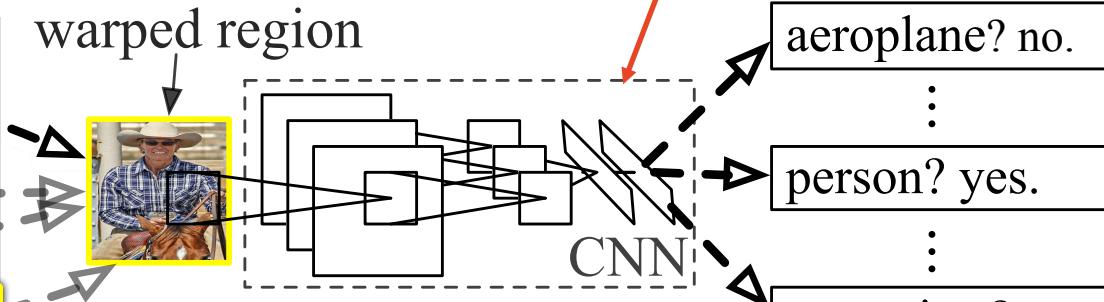


1. Input image



2. Extract region proposals (~2k)

warped region



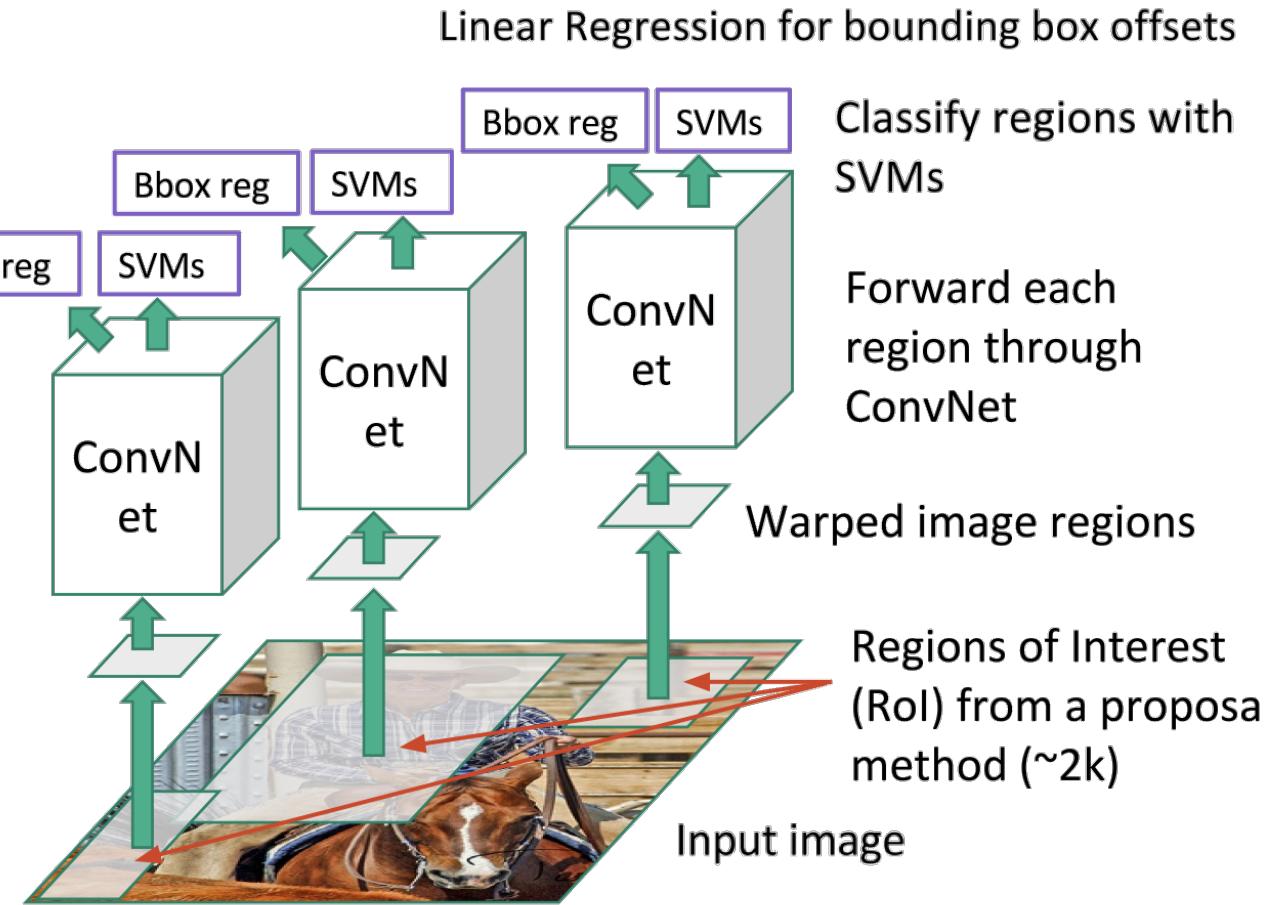
3. Compute CNN features

4. Classify regions

Object Detection to Image Classification

[Girshick14] R. Girshick, J. Donahue, S. Guadarrama, T. Darrell, J. Malik: **Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation**, CVPR 2014

Region CNN (R-CNN)



- Duplication of calculation
- Not in end-to-end manner
- Slow
- Fixed image size (227x227)

*Selective Search

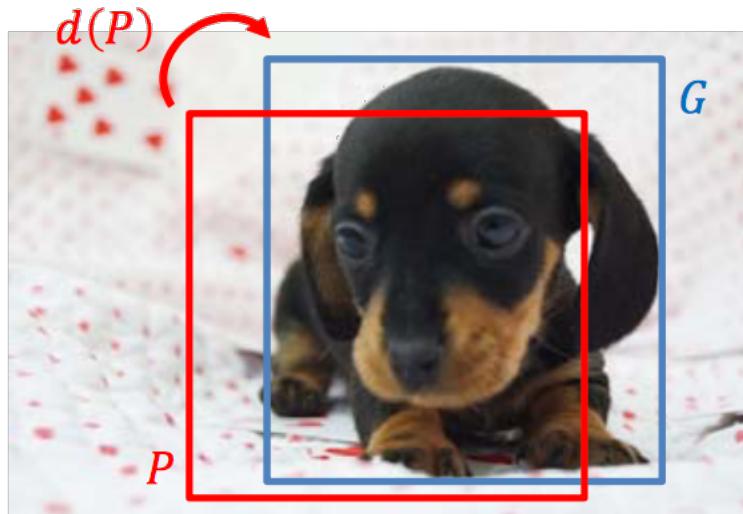
- Motivation
 - Sliding window approach is not feasible for object detection with convolutional neural networks.
 - We need a more faster method to identify object candidates.
- Finding object proposals
 - Greedy hierarchical superpixel segmentation
 - Diversification of superpixel construction and merge
 - Using a variety of color spaces
 - Using different similarity measures
 - Varying staring regions



[Uijlings13] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders: **Selective Search for Object Recognition**. IJCV 2013

*Bounding Box Regression

- Learning a transformation of bounding box
 - Region proposal: $P = (P_w, P_h, P_w, P_h)$
 - Ground-truth: $G = (G_x, G_y, G_w, G_h)$
 - Transformation: $d(P) = (t_x, t_y, t_w, t_h)$



$$\hat{G}_x = P_w d_x(P) + P_x$$

$$\hat{G}_y = P_h d_y(P) + P_y$$

$$\hat{G}_w = P_w \exp(d_w(P))$$

$$\hat{G}_h = P_h \exp(d_h(P))$$

$$d_i(P) = \mathbf{w}_i^T \phi_5(P)$$

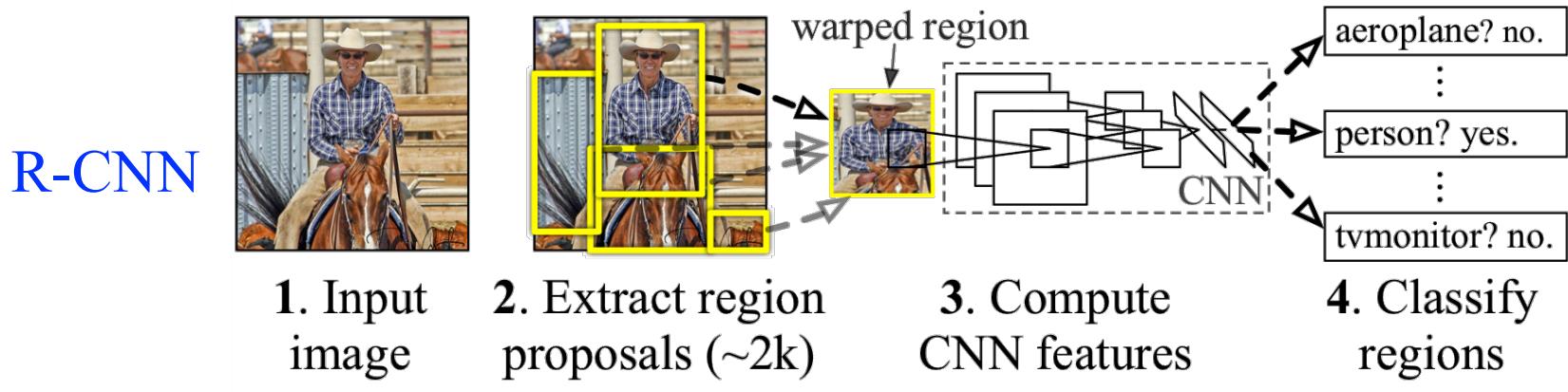
CNN pool5 feature

$$\mathbf{w}_i^* = \operatorname{argmin}_{\mathbf{w}_i} \sum_{k=1}^N \left(t_i^k - \mathbf{w}_i^T \phi_5(P^k) \right)^2 + \lambda \|\mathbf{w}_i\|^2$$

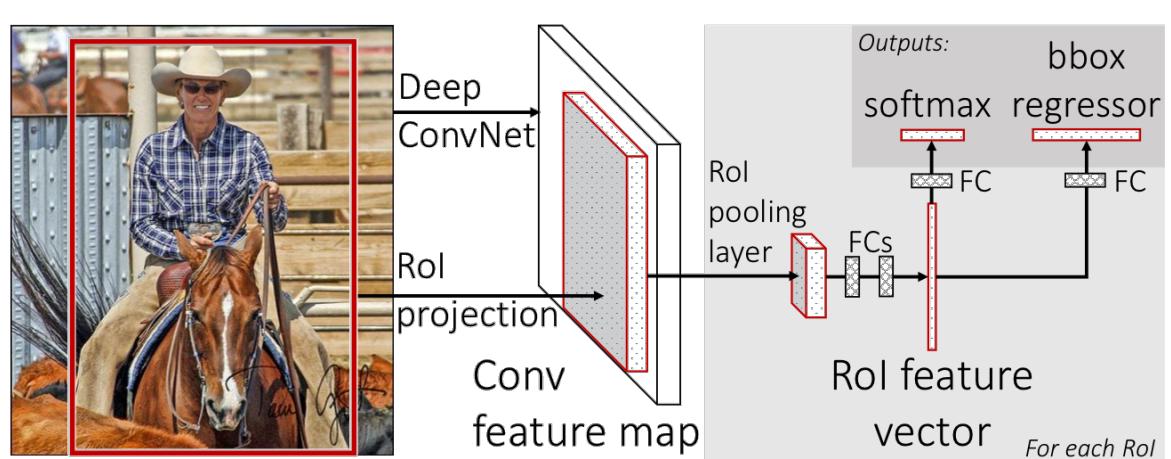
Fast R-CNN

Drawback of R-CNN and the modification:

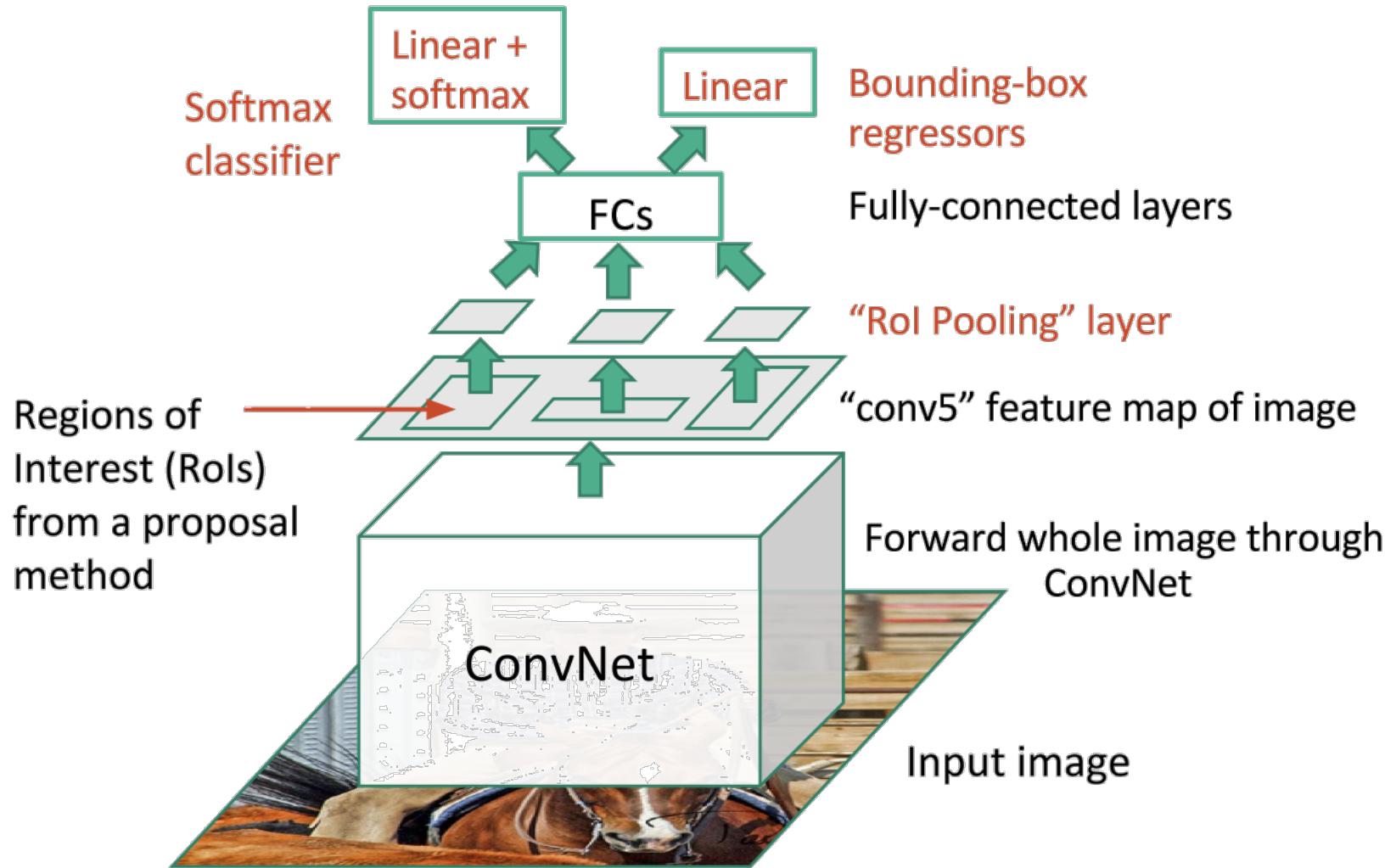
1. Multi-stage training. → End-to-end joint training.
2. Expensive in space and time. → Convolutional layer sharing.
3. Test-time detection is slow. → Single scale testing



Fast R-CNN



Fast R-CNN



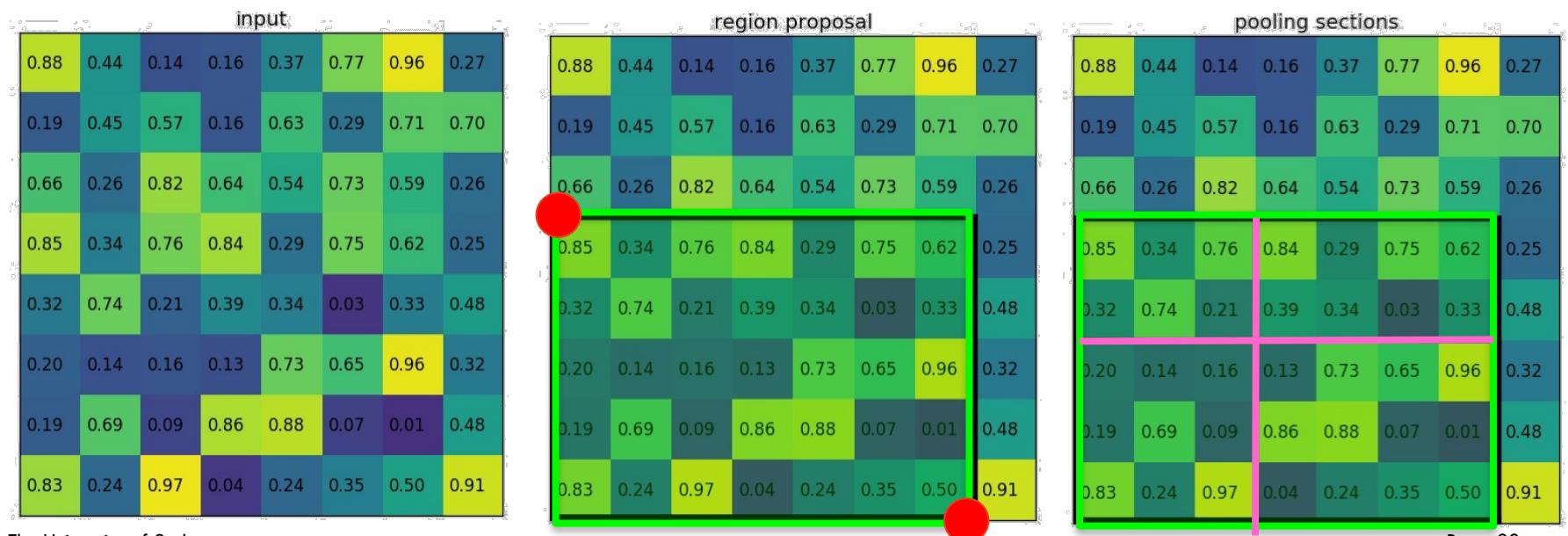
Region of Interest Pooling

A type of max-pooling. The output always has the same size.



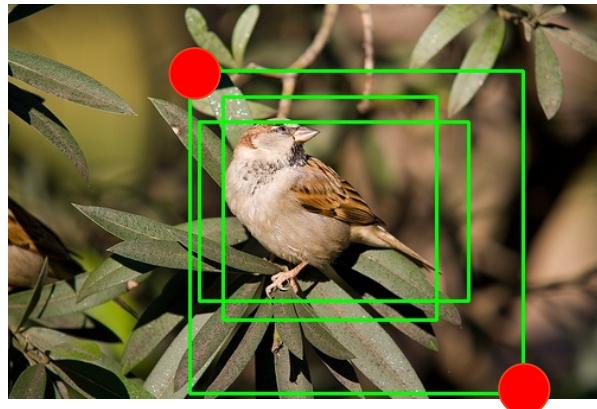
Input to ROI pooling layer:
1. A fixed-size feature map
2. A list of regions of interest

a region proposal
 8×8 feature map \longrightarrow 2×2 output



Region of Interest Pooling

A type of max-pooling. The output always has the same size.



Input to ROI pooling layer:

1. A fixed-size feature map
2. A list of regions of interest

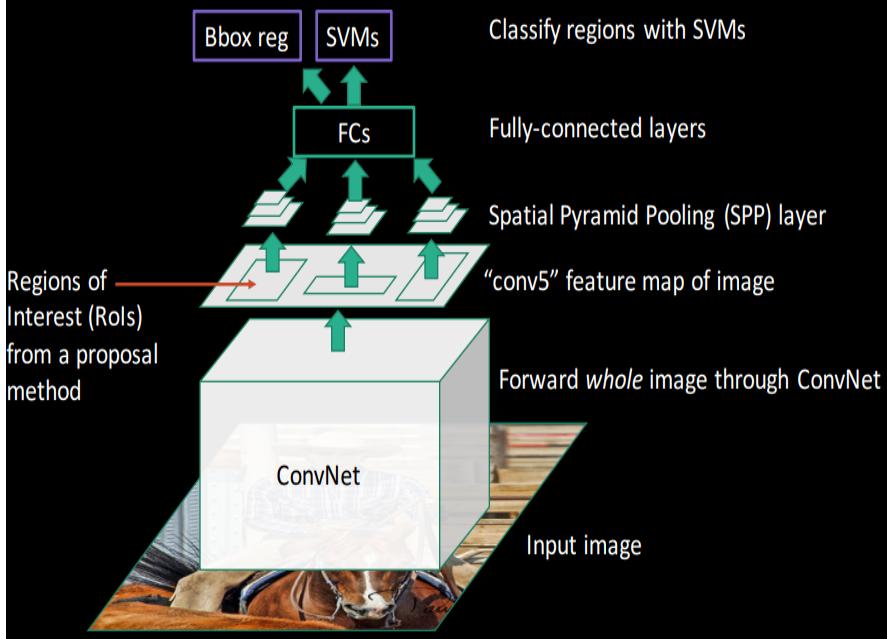


input								
0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27	
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70	
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26	
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25	
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48	
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32	
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48	
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91	

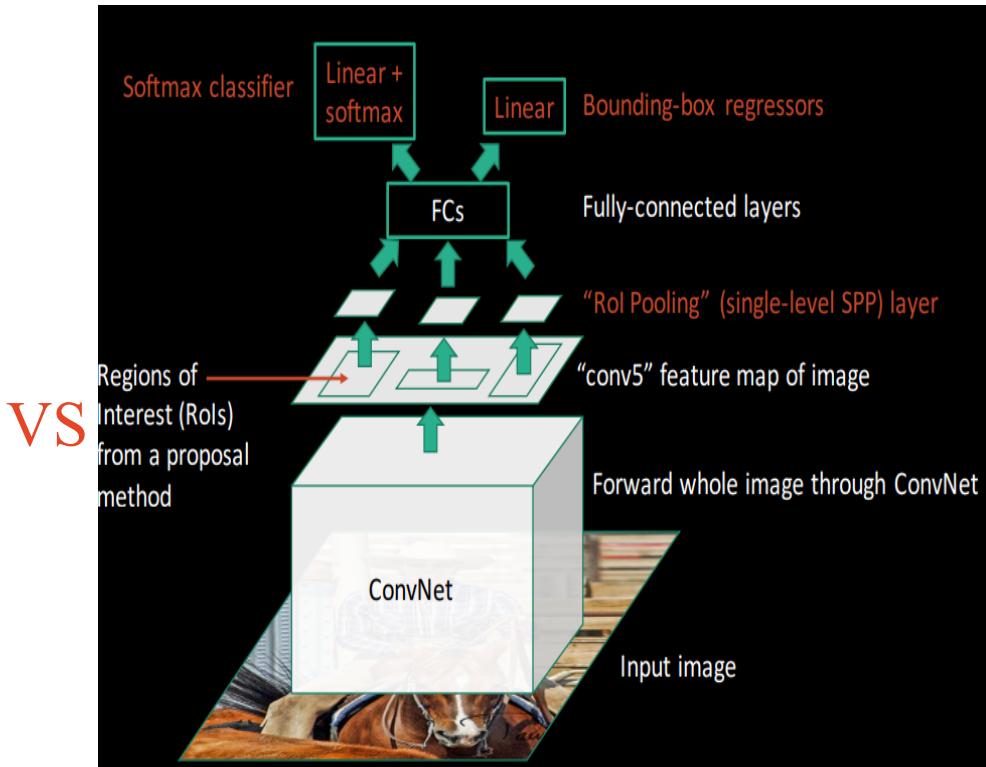
- Used for object detection tasks
- Reuse the feature map from CNNs
- Speed up both train and test time
- Train detection model in an end-to-end manner

Fast R-CNN

SPP-net



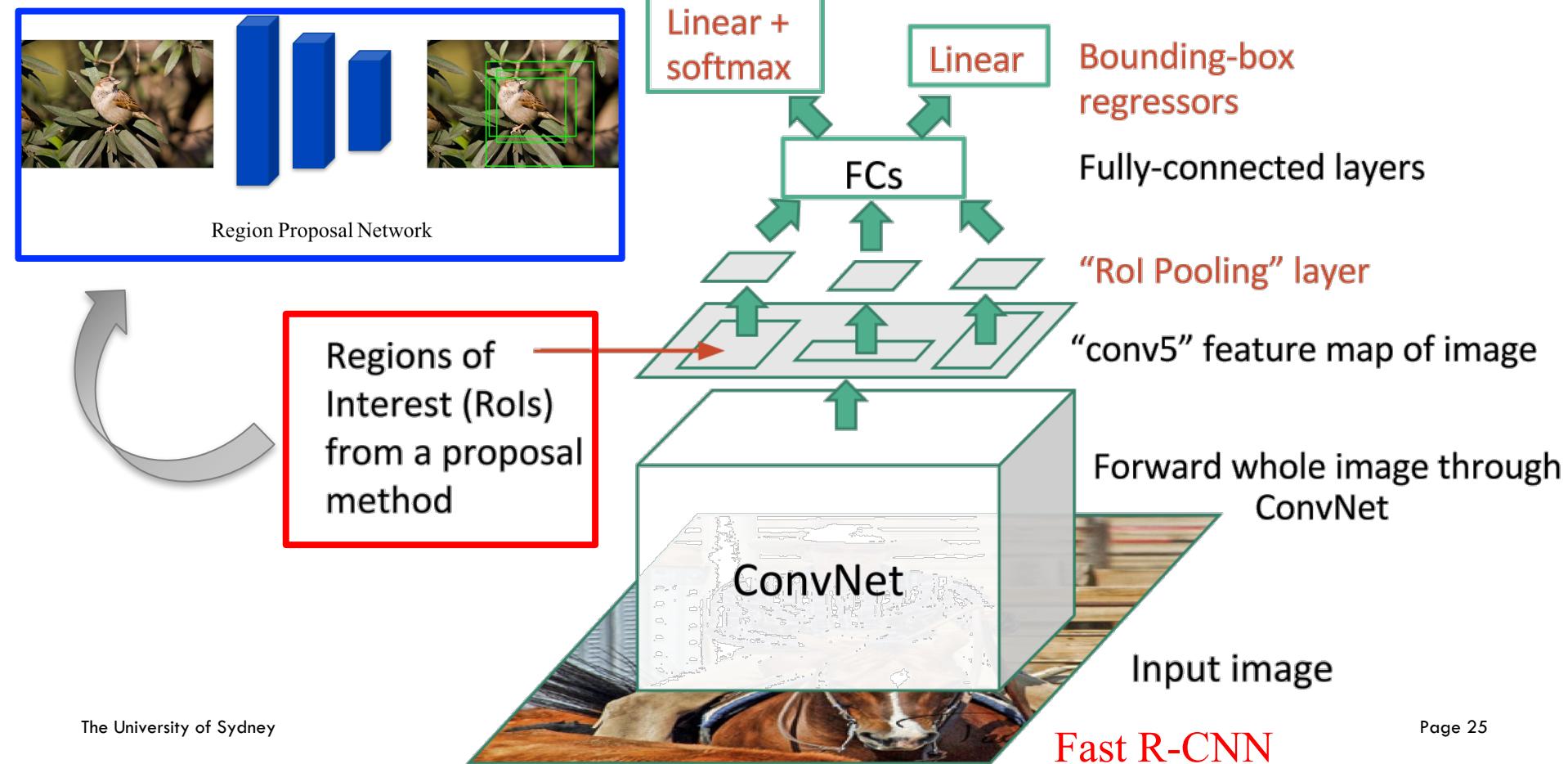
VS



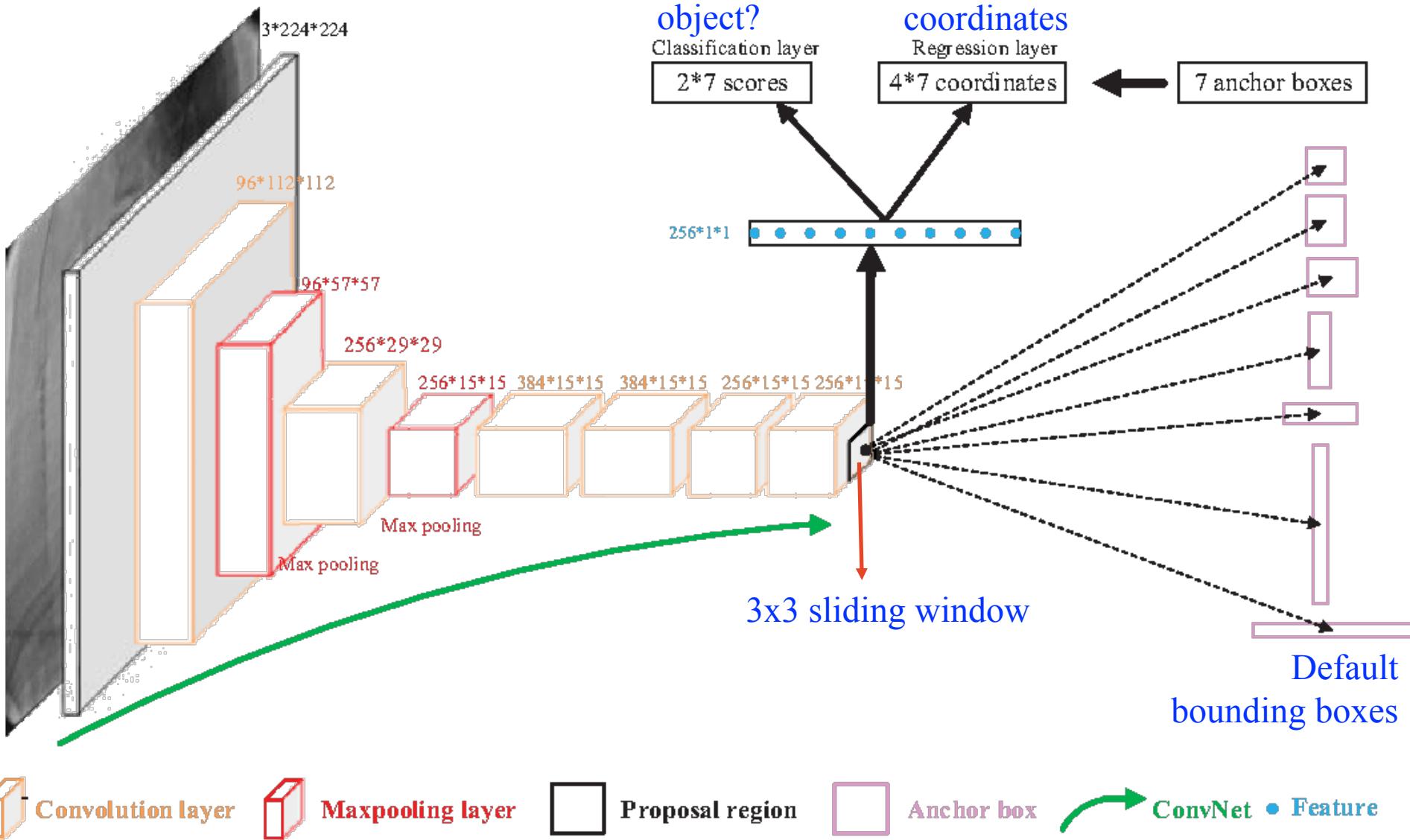
Credit To: <http://www.robots.ox.ac.uk/~tvg/publications/talks/fast-rcnn-slides.pdf>

Faster R-CNN

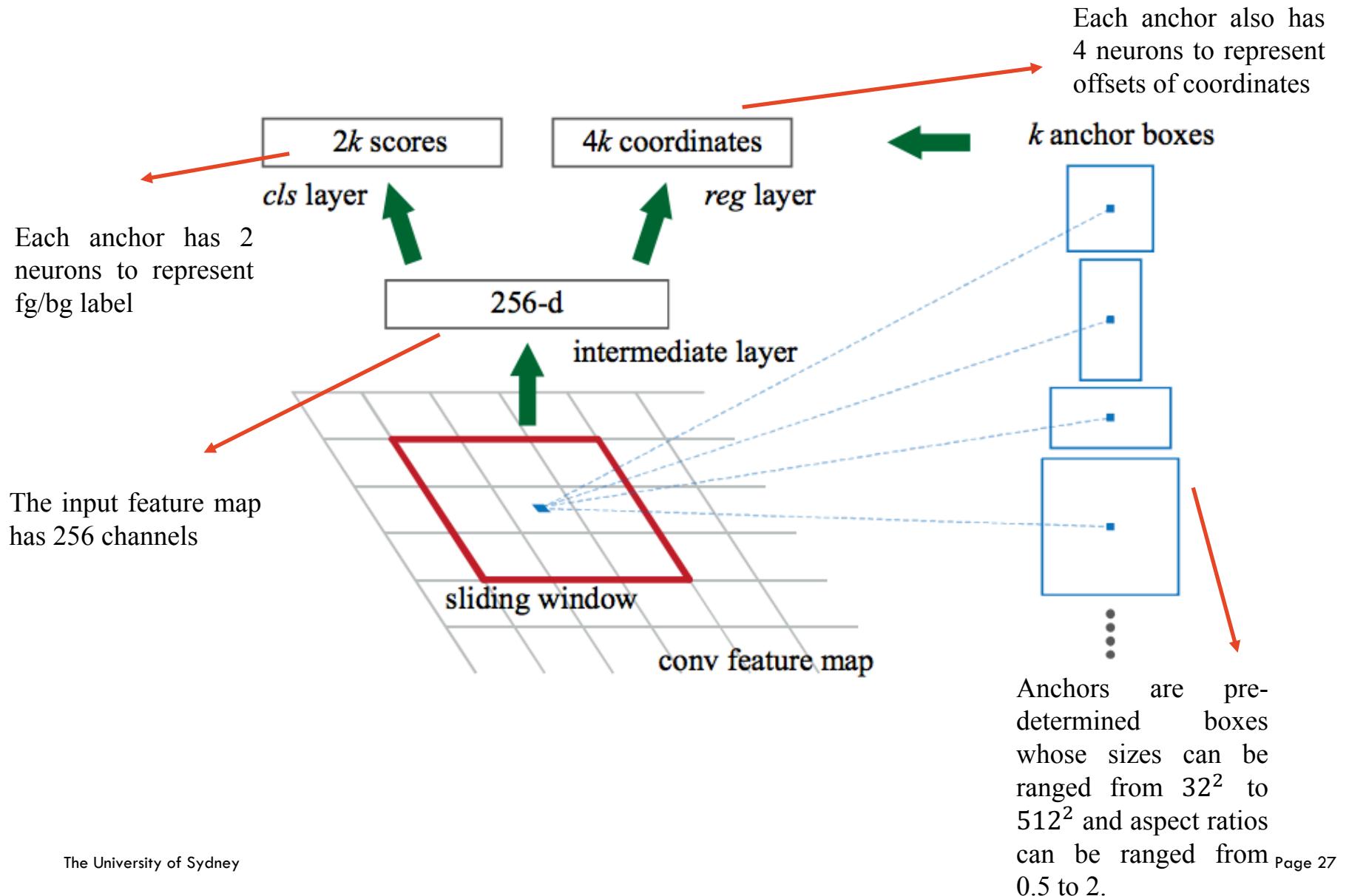
Replace the slow selective search algorithm with a fast neural net - region proposal network (RPN).



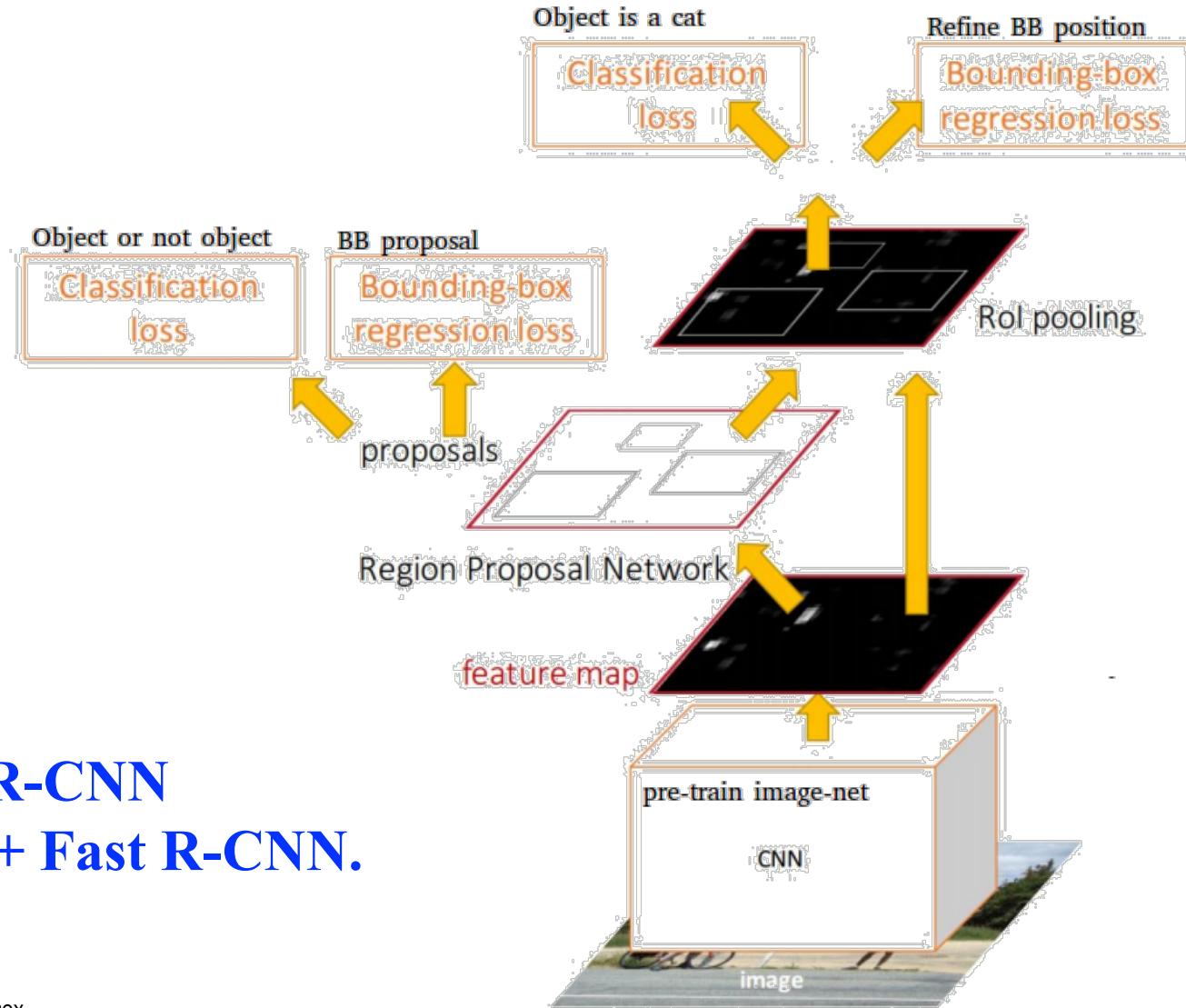
Region Proposal Network



Region Proposal Network

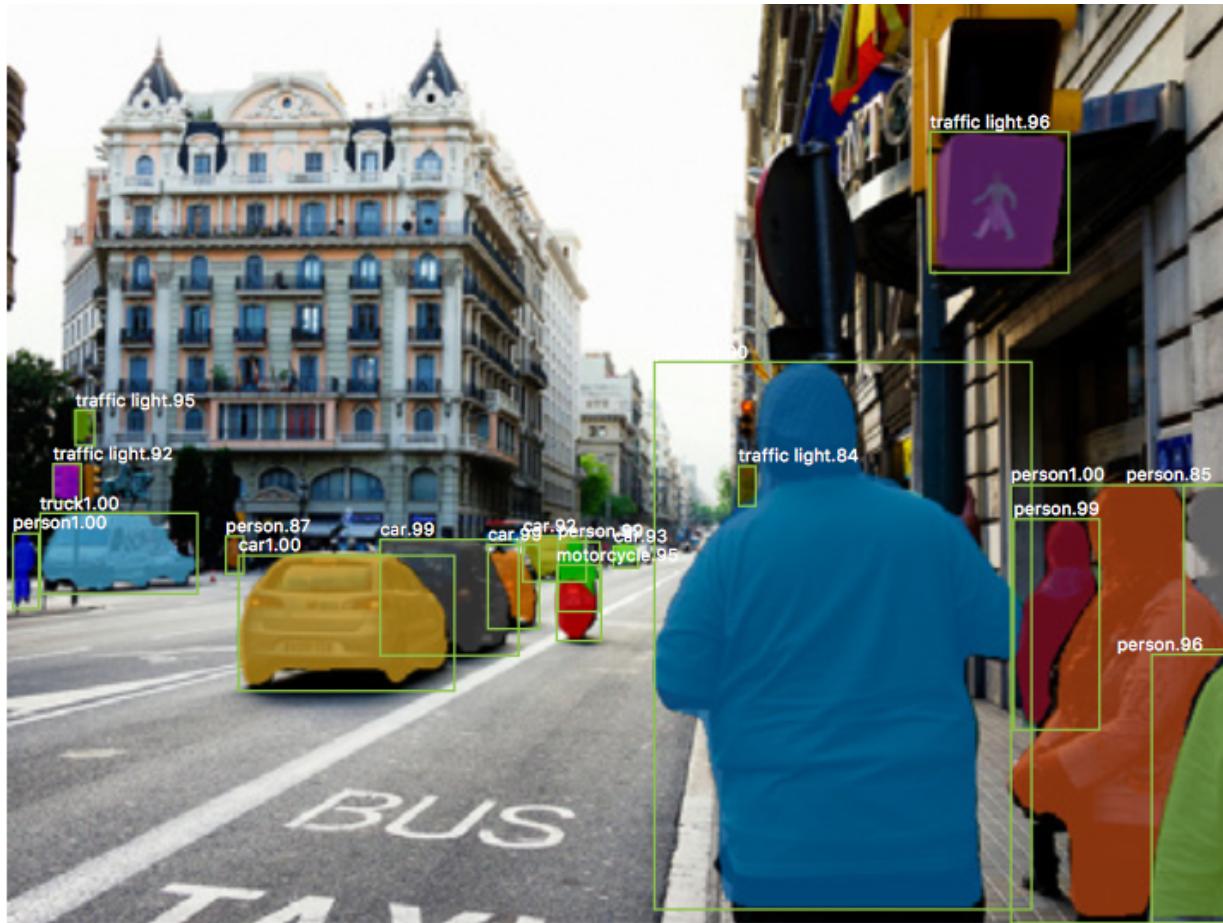


Faster R-CNN

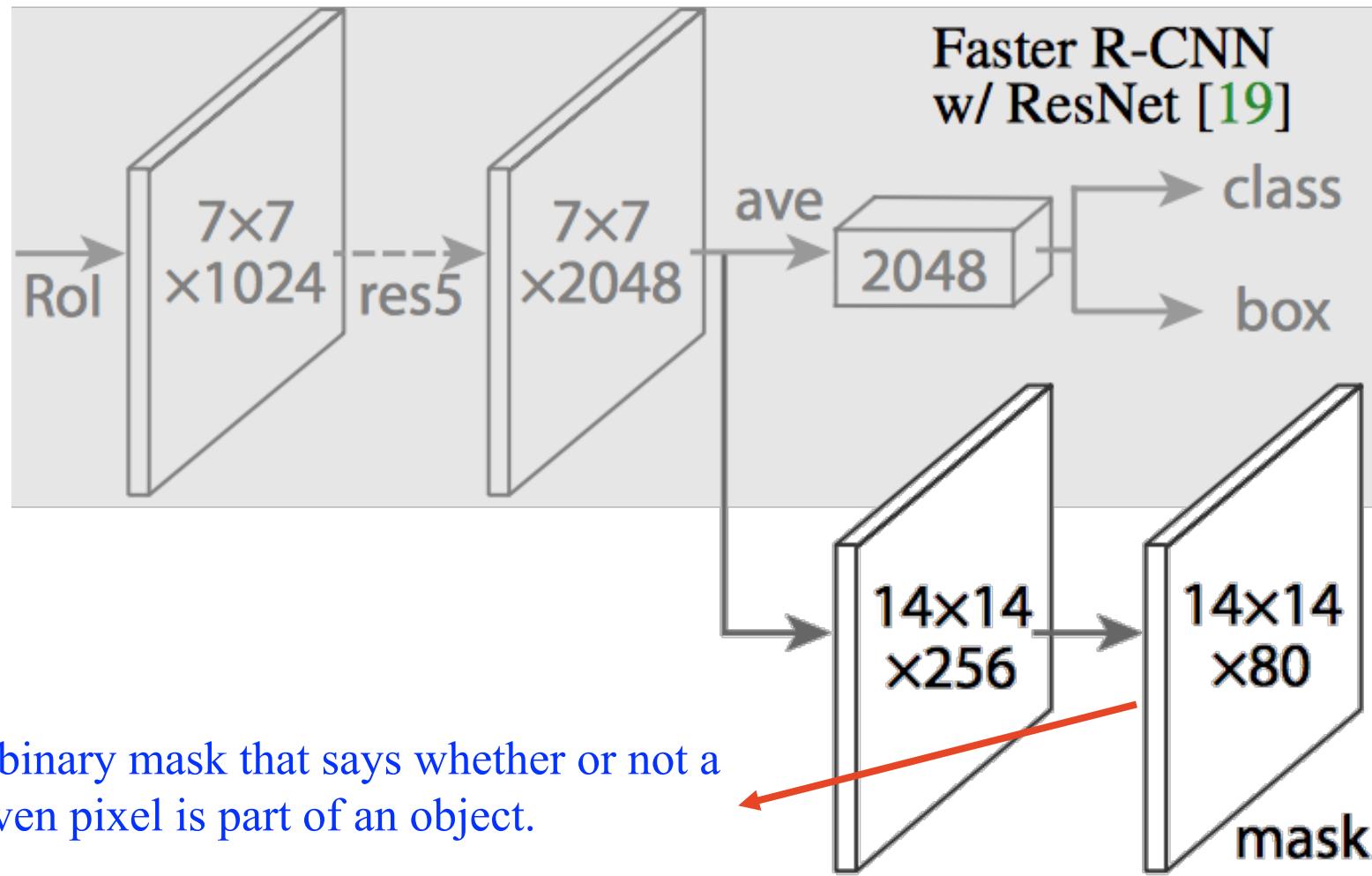


Mask R-CNN

Image instance segmentation is to identify, at a pixel level, what the different objects in a scene are.



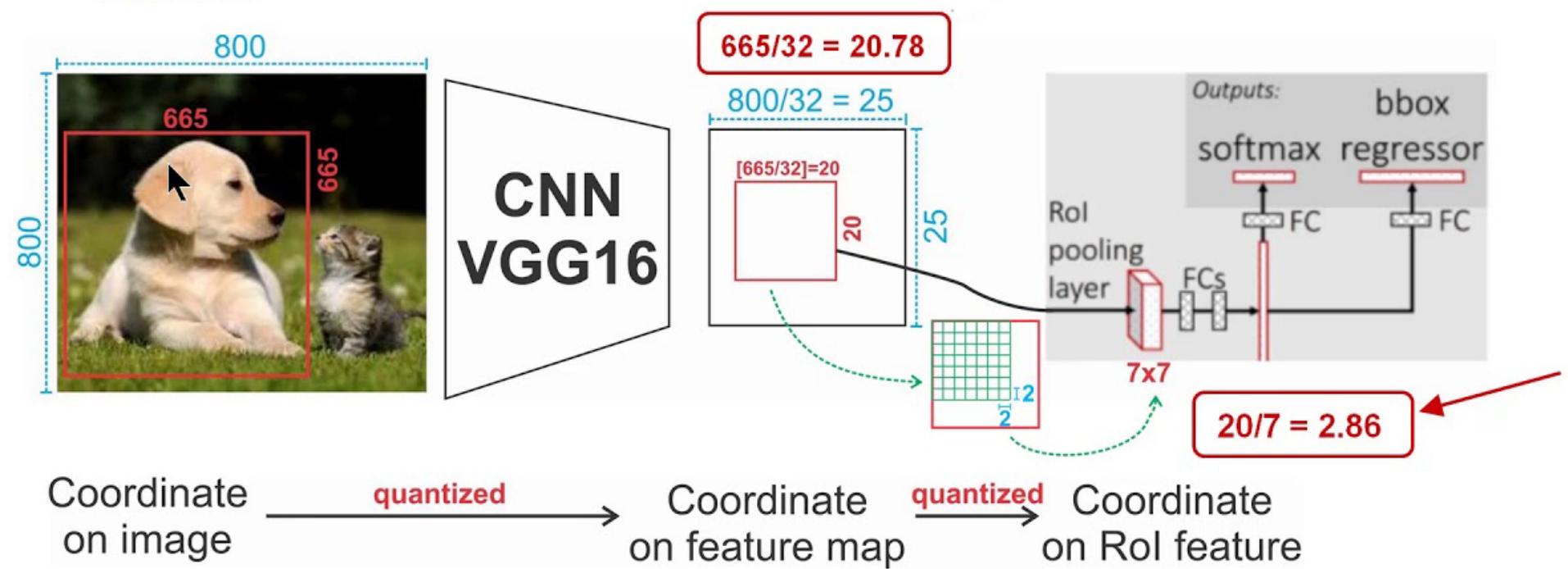
Mask R-CNN



RoIAlign

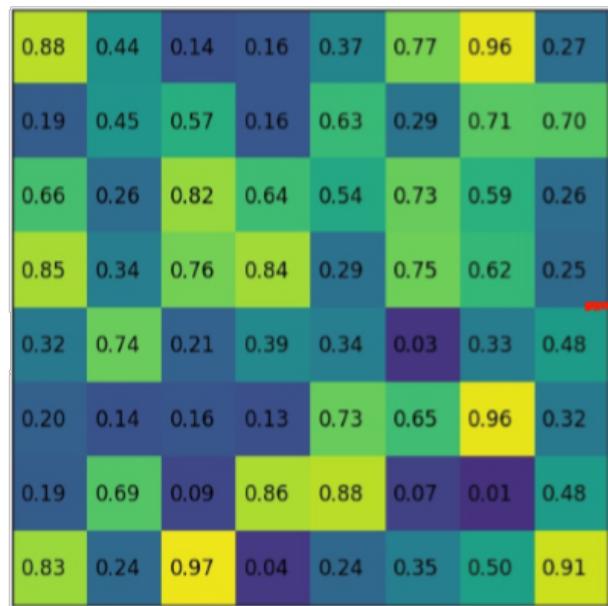
Realigning RoI Pooling to be More Accurate.

“RoiPool”

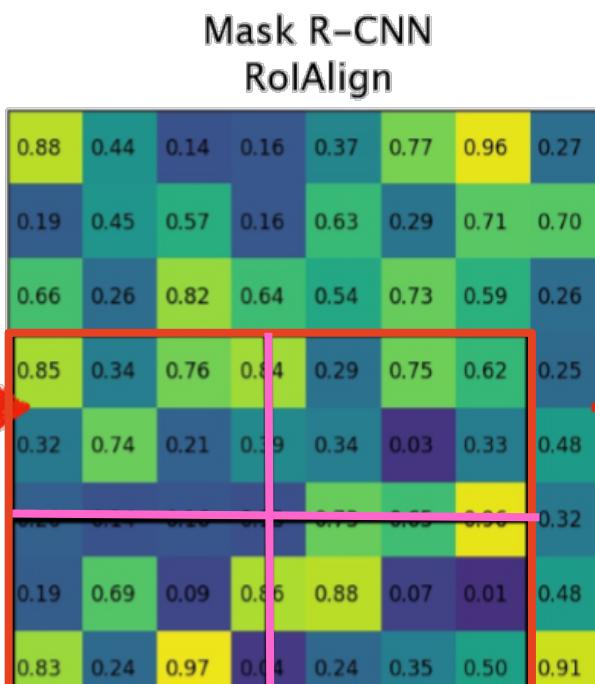


RoIAlign

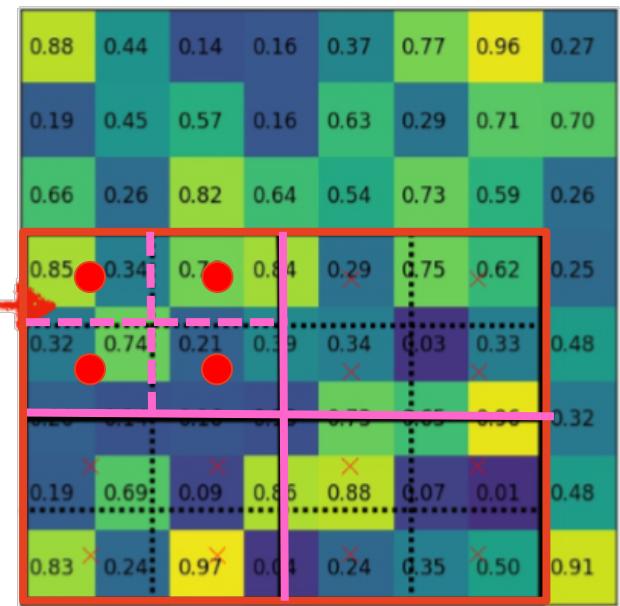
1) RoIAlign



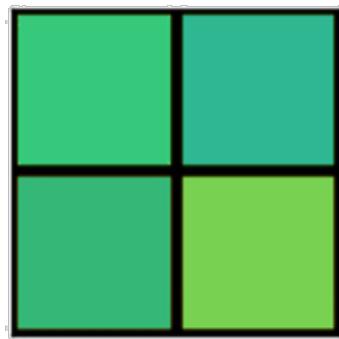
Input activation



Region projection and pooling sections

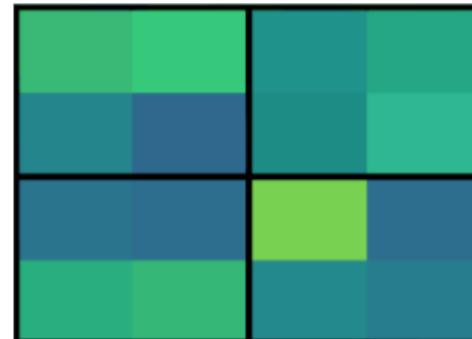


Sampling locations



The University of Sydney

Max pooling output

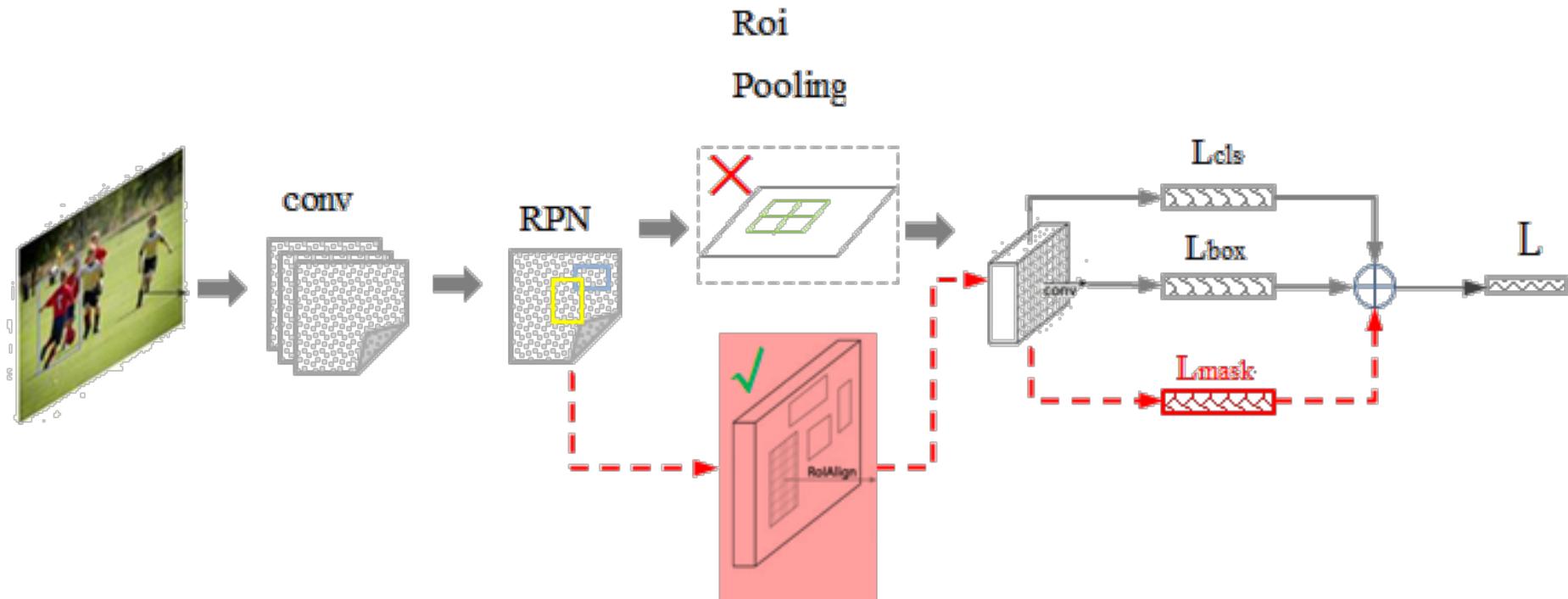


Bilinear interpolated values

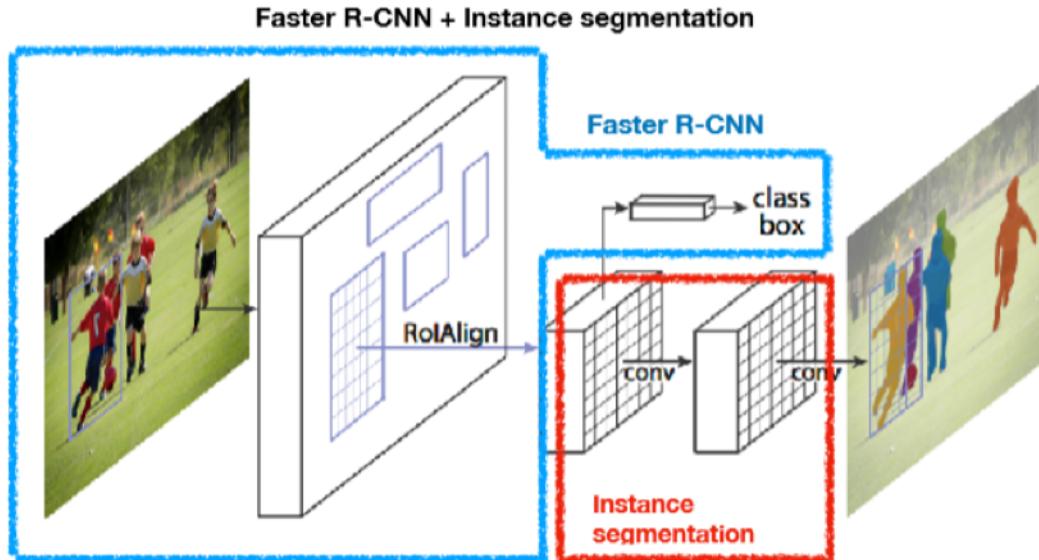
bilinear interpolation

RoiAlign

Realigning RoI Pooling to be More Accurate.



Mask R-CNN



Faster R-CNN

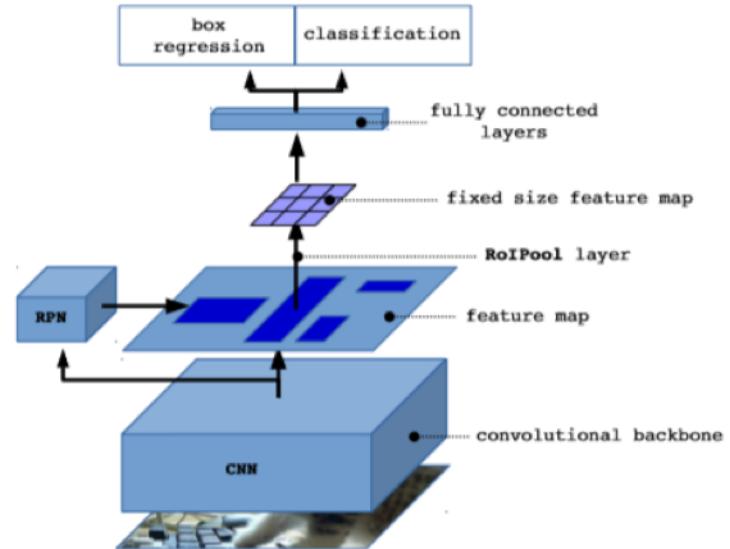


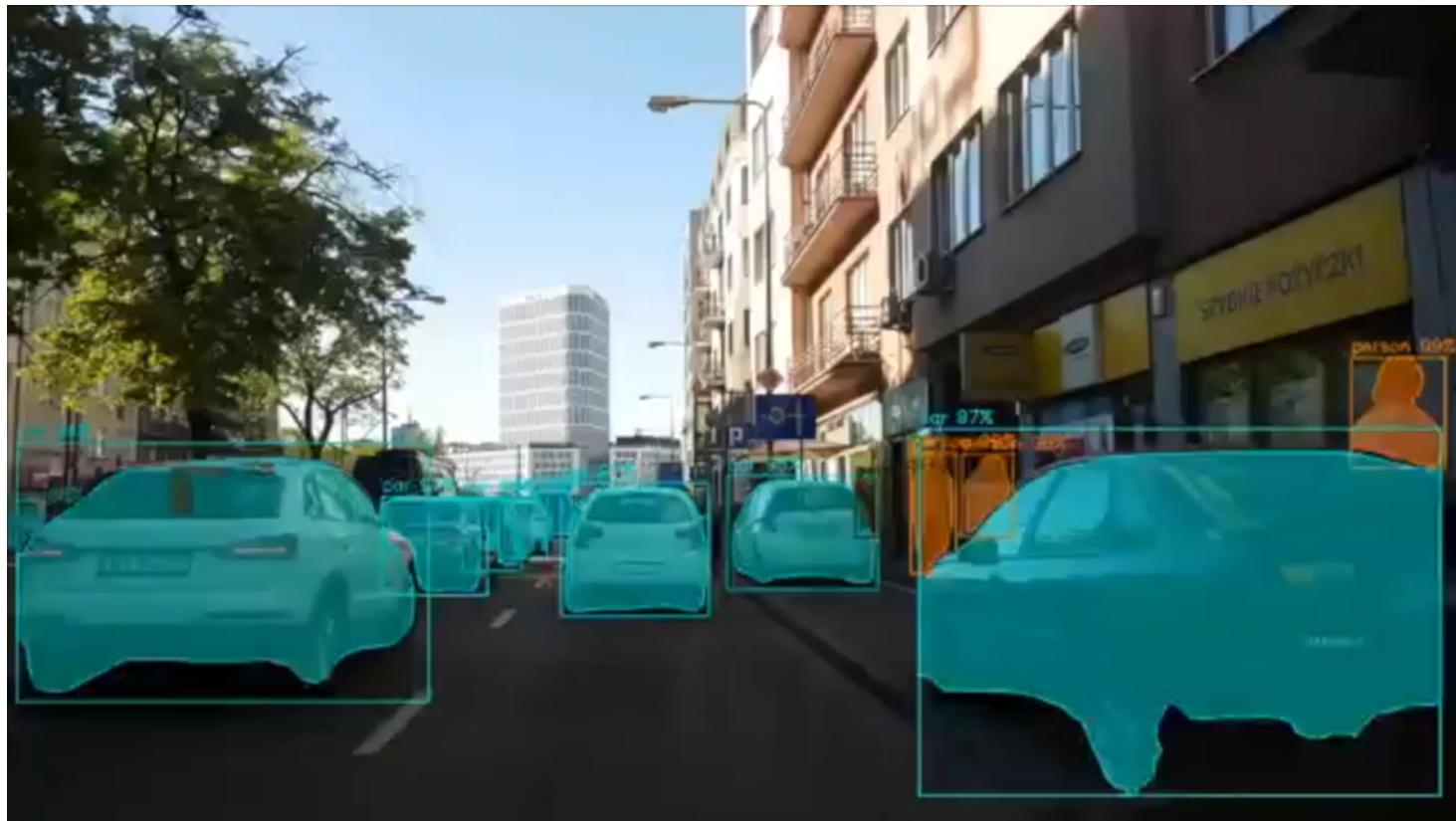
Figure 1. The Mask R-CNN framework for instance segmentation.

Mask R-CNN
= Instance Segmentation+ Faster R-CNN.

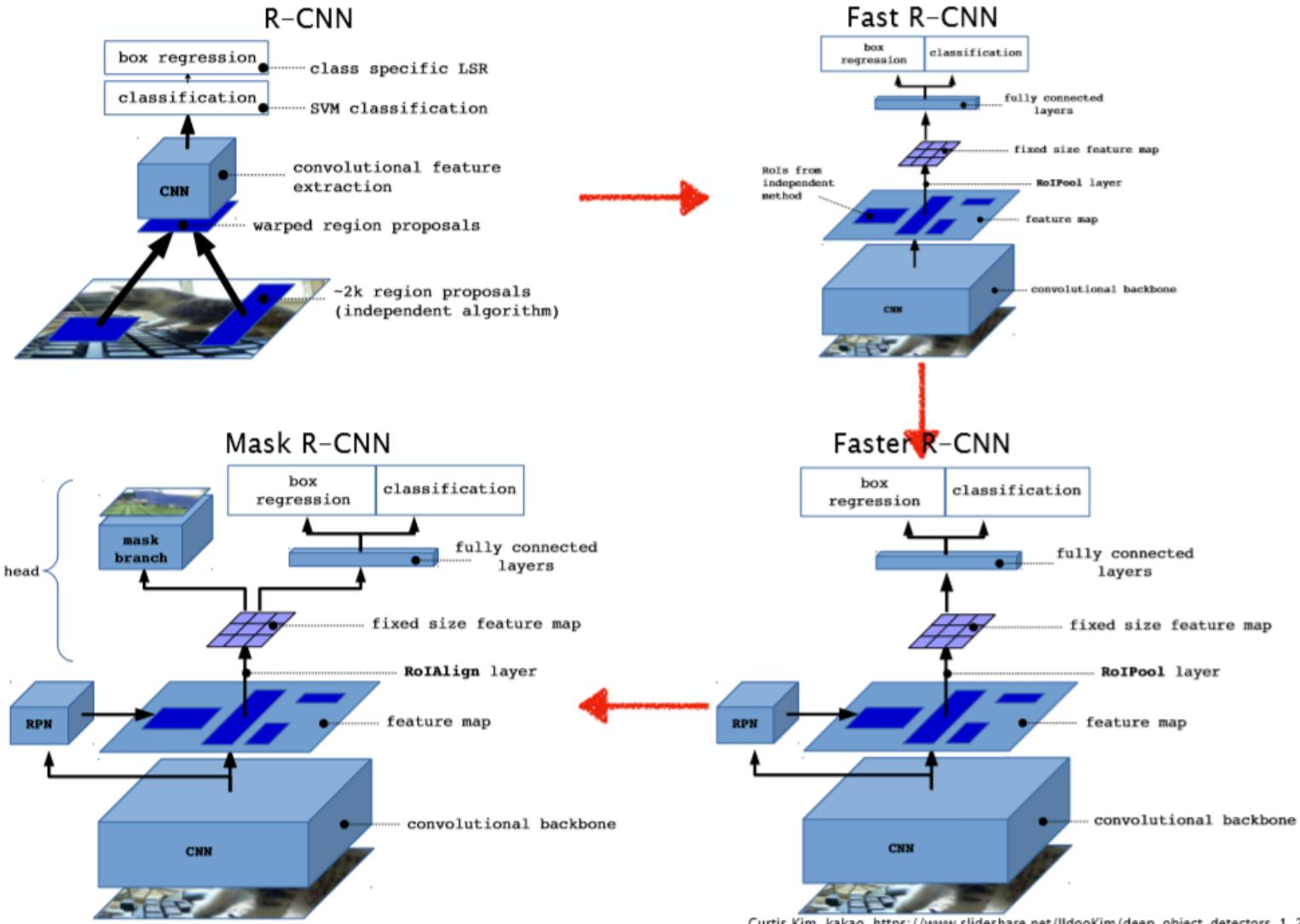
Credit To: <https://www.slideshare.net/windmdk/mask-rcnn>

Mask R-CNN

Demo



From R-CNN to Mask R-CNN



Curtis Kim, kakao, <https://www.slideshare.net/lldooKim/deep-object-detectors-1-20166>

Credit To: <https://www.slideshare.net/windmdk/mask-rcnn>

Page 36