

EE 4211 Computer Vision

Lecture 12B: Object detection with deep learning

Semester A, 2020-2021

Schedules

Week	Date	Topics
1	Sep. 4	Introduction/Imaging
2	Sep. 11	Image enhancement in spatial domain
3	Sep. 18	Image enhancement in frequency domain (HW1 out)
4	Sep. 25	Morphological processing
5	Oct. 2	Image restoration(HW1 due)
6	Oct. 9	Image restoration
7	Oct. 16	Midterm (no tutorials this week)
8	Oct. 23	Edge detection (HW2 out, illustrate the project)
9	Oct. 30	Image segmentation (HW2 due)
10	Nov. 6	Face recognition with PCA, LDA (tutorial on deep learning framework)
11	Nov. 13	Face recognition based on deep learning Image segmentation based on deep learning (tutorial on coding)
12	Nov. 20	Object detection with traditional methods (Quiz) Object detection based on deep learning
13	Nov. 27	Project presentation and summary
14	Dec. 4	Review and Summary

Lecture outline

- RCNN
- Fast-RCNN
- Faster-RCNN
- Mask-RCNN

Detection Competitions

- Pascal VOC
- COCO
- ImageNet ILSVRC

VOC: 20 classes

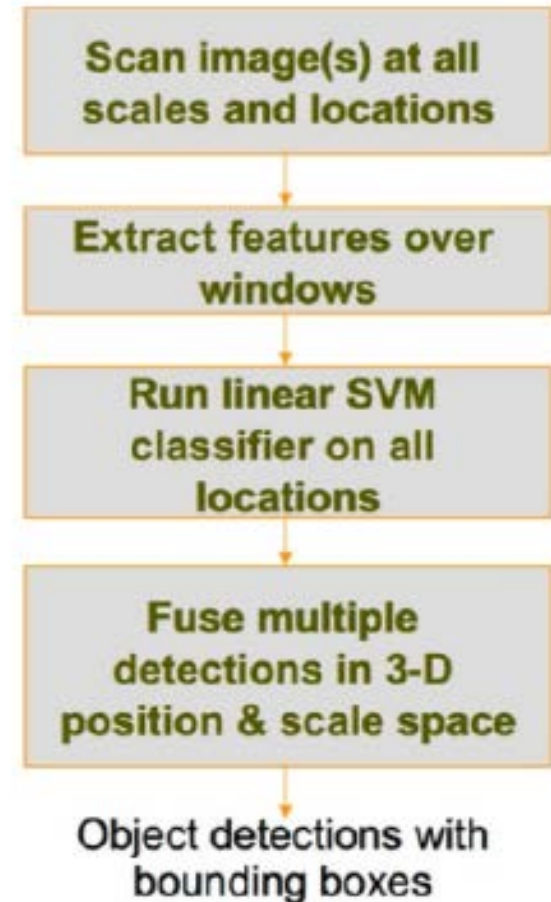
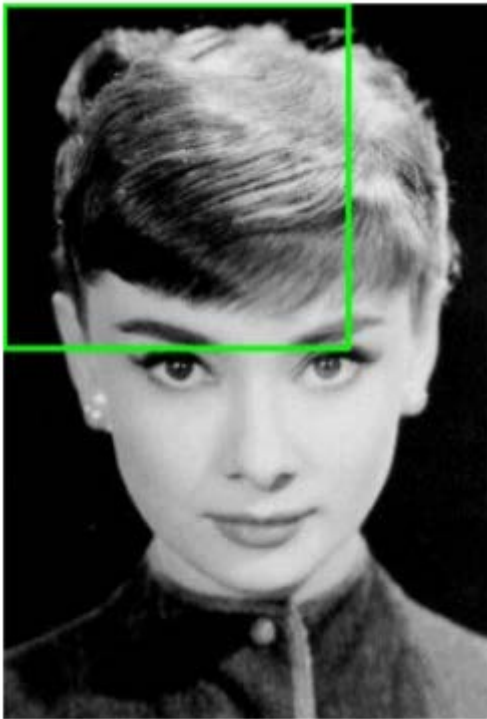


COCO: 200 classes



Before Deep Learning

- Sliding windows
- Score every sub-window. (HOG)



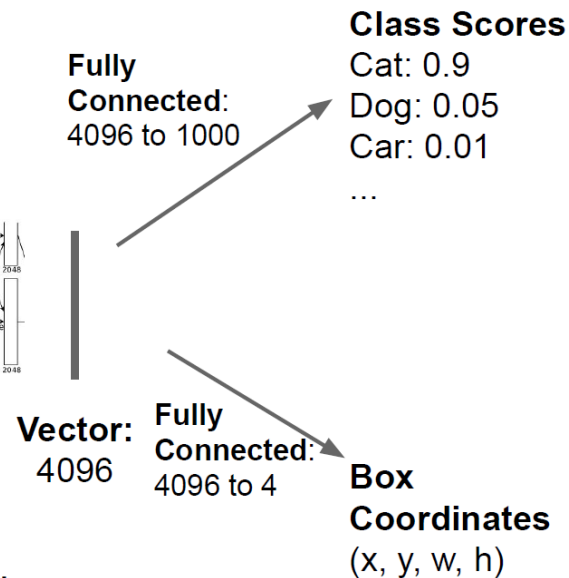
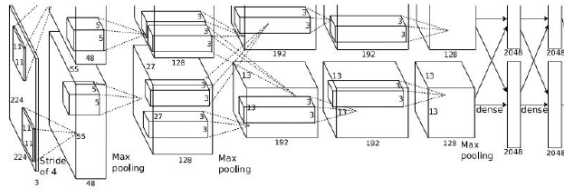
Object Detection: Single Object

Object Detection: Single Object

(Classification + Localization)



[This image is CC0 public domain](#)



Treat localization as a regression problem!

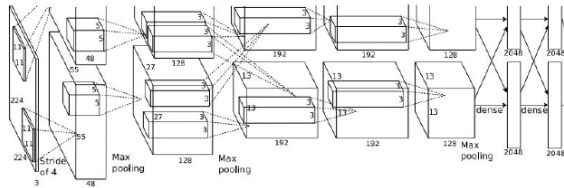
Object Detection: Single Object

Object Detection: Single Object

(Classification + Localization)



[This image is CC0 public domain](#)



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!

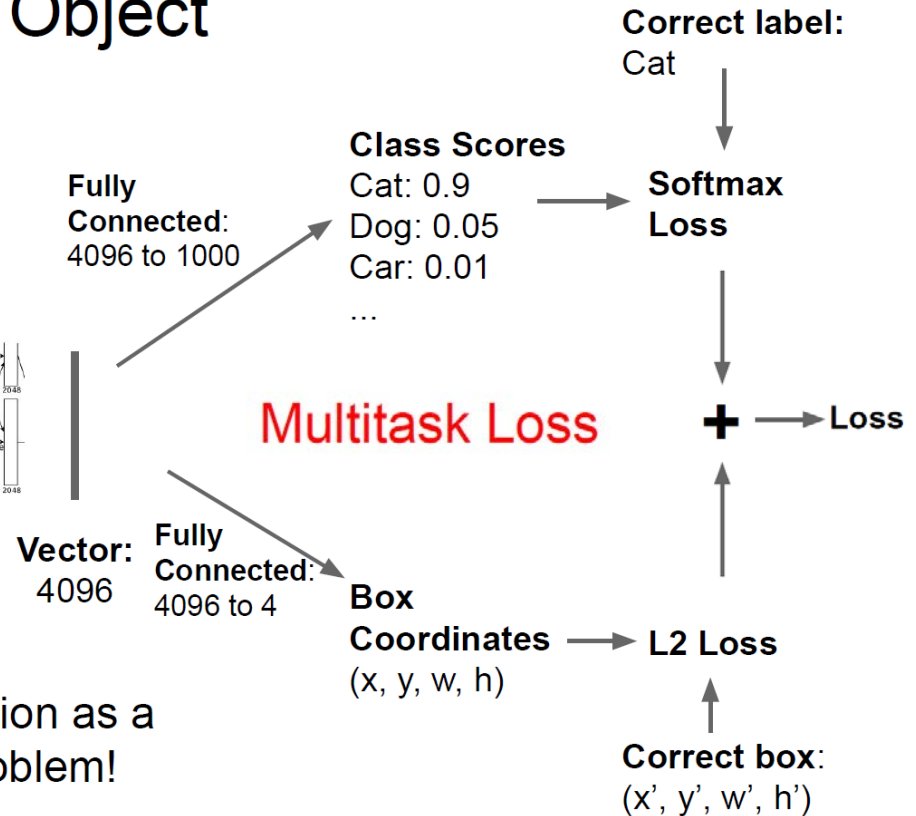
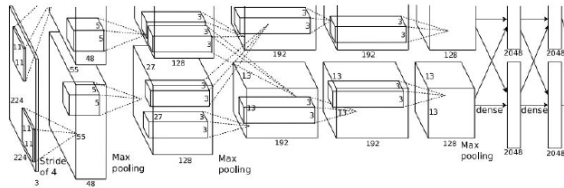
Object Detection: Single Object

Object Detection: Single Object

(Classification + Localization)



[This image is CC0 public domain](#)



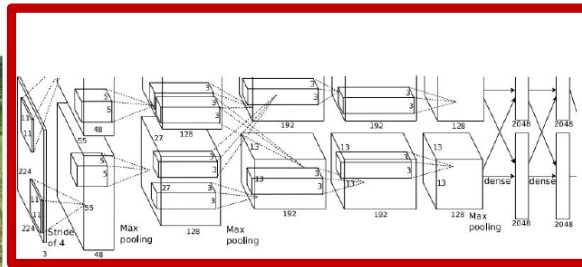
Object Detection: Single Objects

Object Detection: Single Object

(Classification + Localization)



[This image is CC0 public domain](#)



Often pretrained on ImageNet
(Transfer learning)

Treat localization as a
regression problem!

Fully
Connected:
4096 to 1000

Class Scores

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

Correct label:
Cat

Softmax
Loss

Multitask Loss

Vector:
4096

Fully
Connected:
4096 to 4

Box
Coordinates
(x, y, w, h)

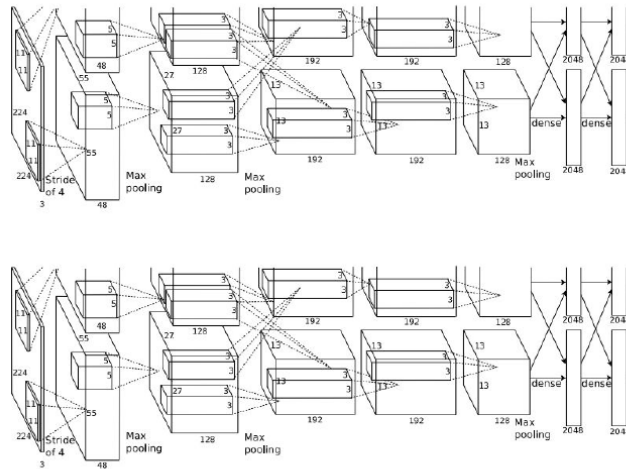
L2 Loss

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

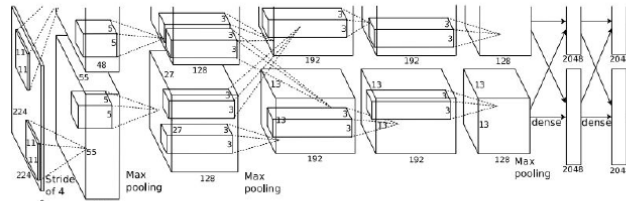
+

Loss

Object Detection: Multiple Objects



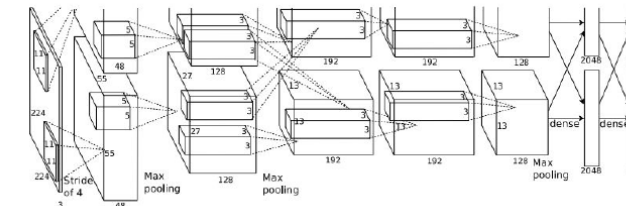
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)



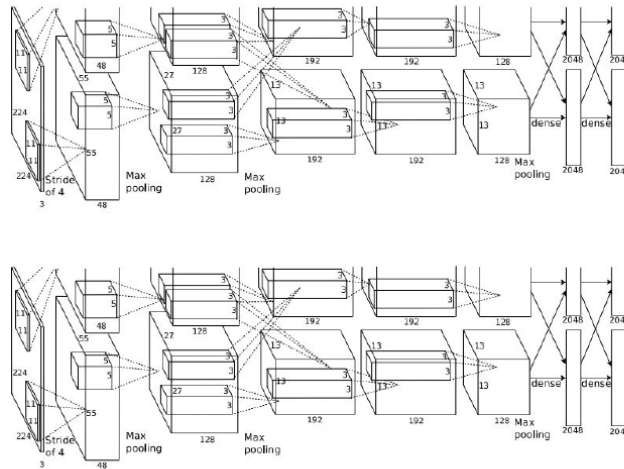
DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

...

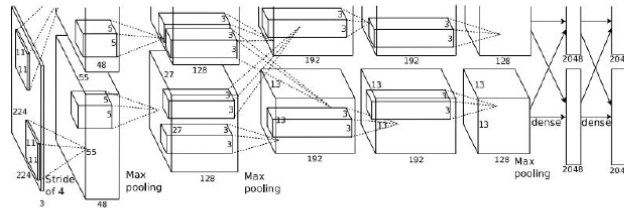
Object Detection: Multiple Objects

- Each image needs a different number of outputs



CAT: (x, y, w, h)

4 numbers

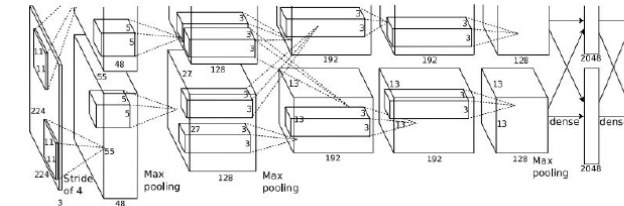


DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)

12 numbers



DUCK: (x, y, w, h)

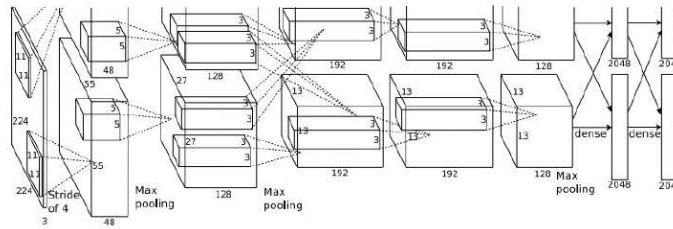
DUCK: (x, y, w, h)

....

Many numbers!

Object Detection: Multiple Objects

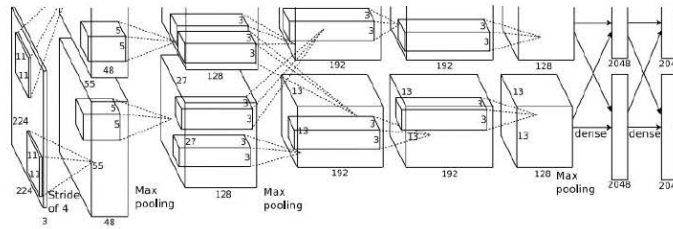
- Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection: Multiple Objects

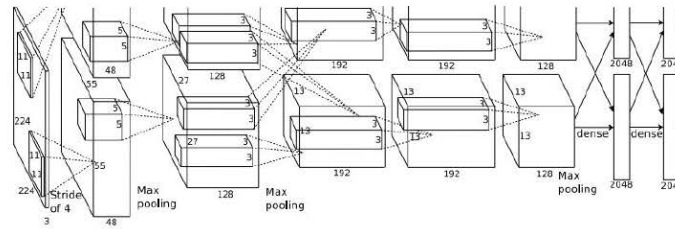
- Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? Yes
Cat? NO
Background? NO

Object Detection: Multiple Objects

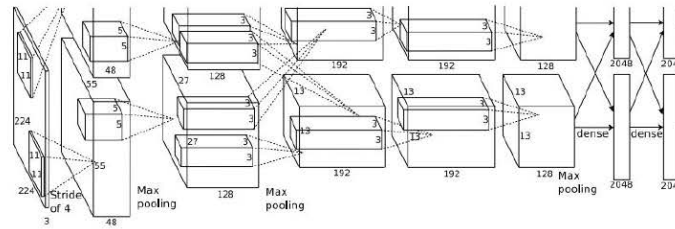
- Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? Yes
Cat? NO
Background? NO

Object Detection: Multiple Objects

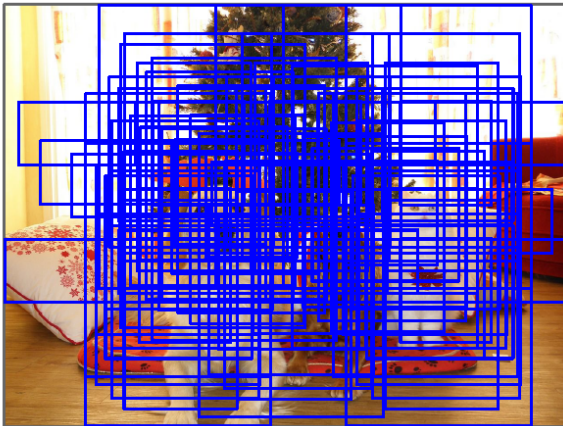
- Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



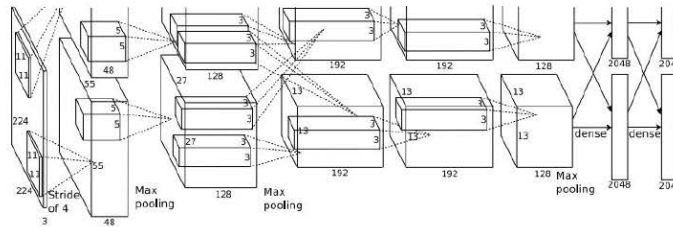
Dog? NO
Cat? Yes
Background? NO

Object Detection: Multiple Objects

- Apply a CNN to many different crops of the image, CNN classifies each crop as object or background
- Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive



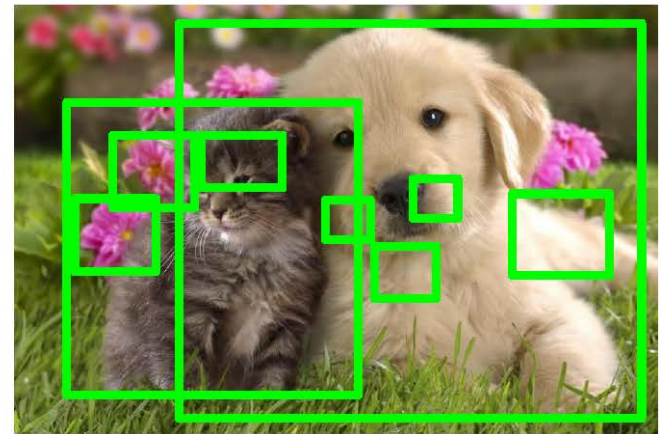
Sliding Window



Dog? NO
Cat? Yes
Background? NO

Region Proposals: Selective Search

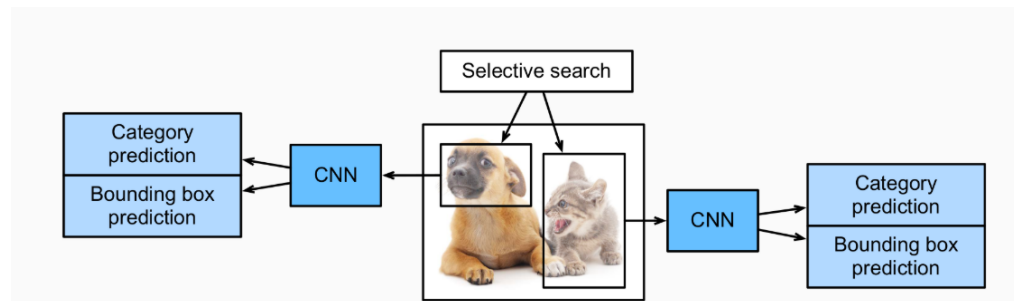
- Find image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



R-CNN

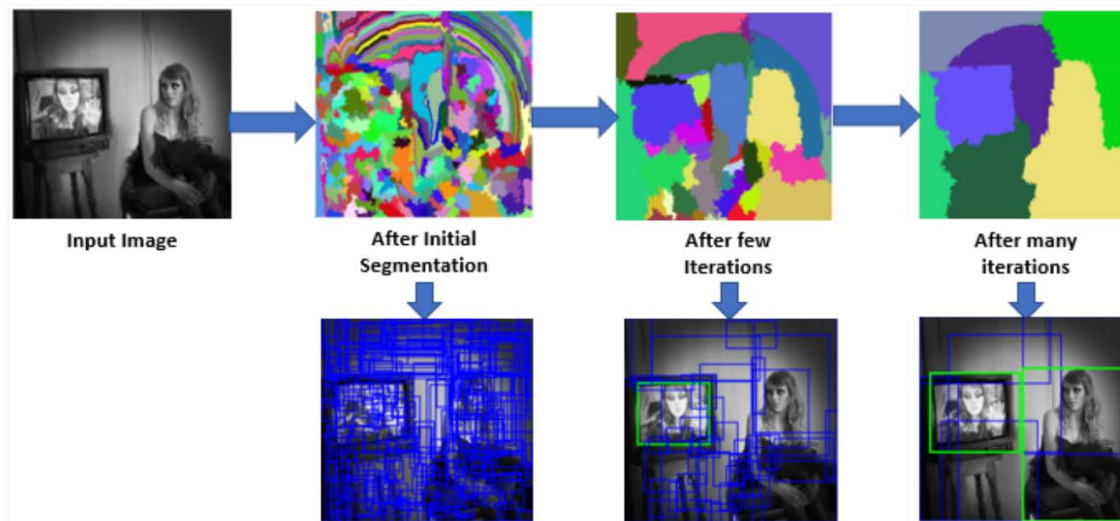
■ Region-CNN model (R-CNN)

- **Selective search** is performed on the input image to select multiple high-quality proposed regions [\[Uijlings et al., 2013\]](#). These proposed regions are generally selected on multiple scales and have different shapes and sizes.
- A pre-trained CNN is selected and placed output the features extracted from the proposed regions.
- The features and labeled category of each proposed region are combined as an example to train multiple support vector machines for **object classification** AND **train a linear regression model** for ground-truth bounding box prediction.

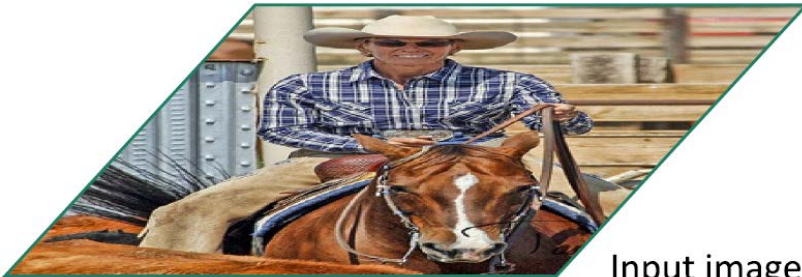


R-CNN

- Step1: Generate initial sub-segmentation of input image using the method describe by *Felzenszwalb et al* in his paper “Efficient Graph-Based Image Segmentation “
- Step2: Recursively combine the smaller similar regions into larger ones. We use Greedy algorithm to combine similar regions to make larger regions.
- Step3: Use the segmented region proposals to generate candidate object locations.

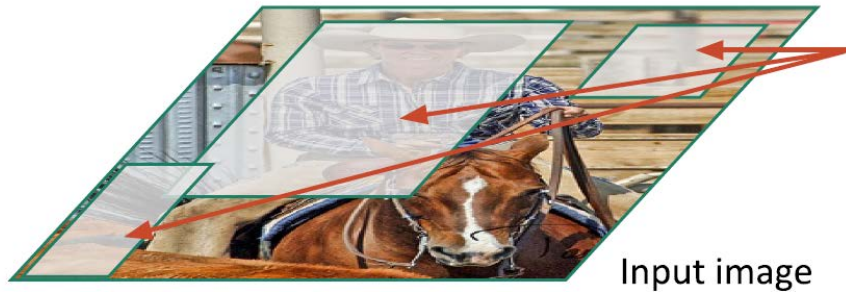


R-CNN



Input image

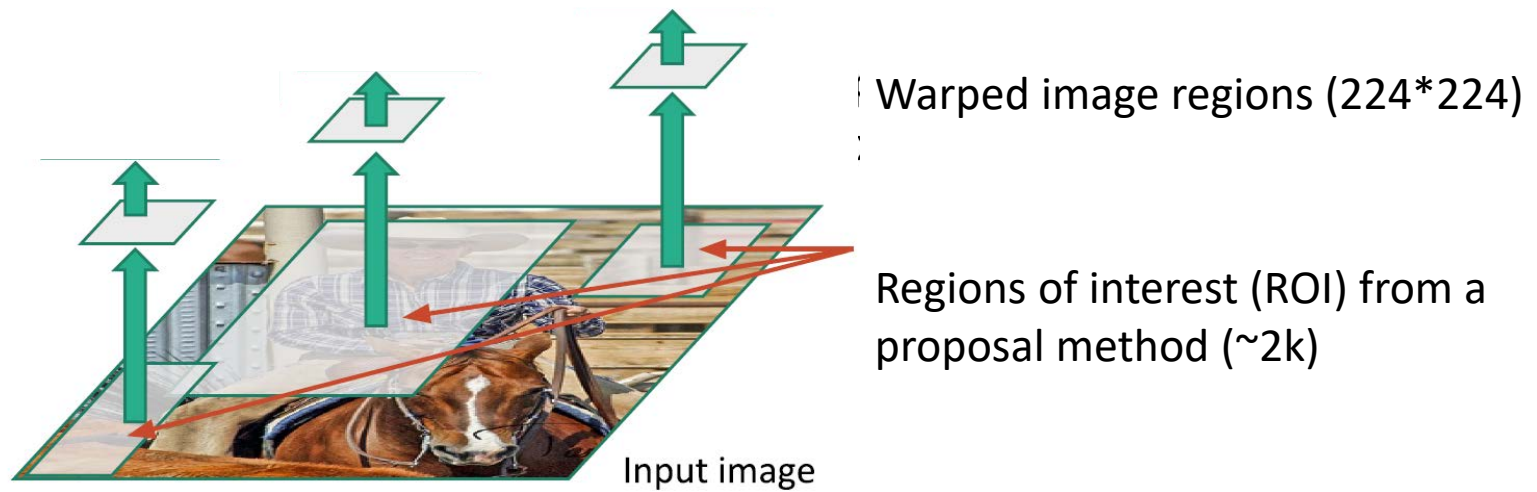
R-CNN



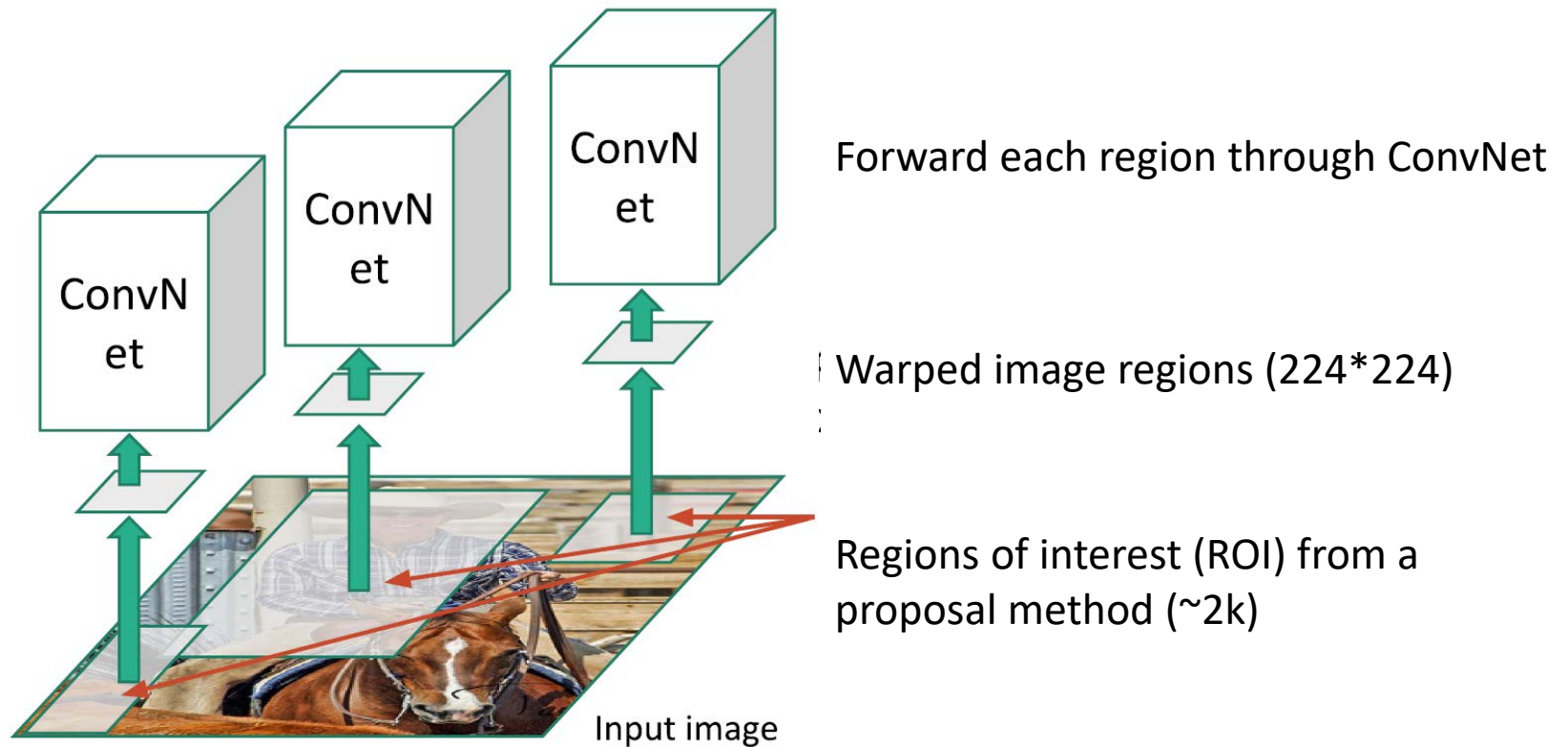
Regions of interest (ROI) from a
proposal method (~2k)

Input image

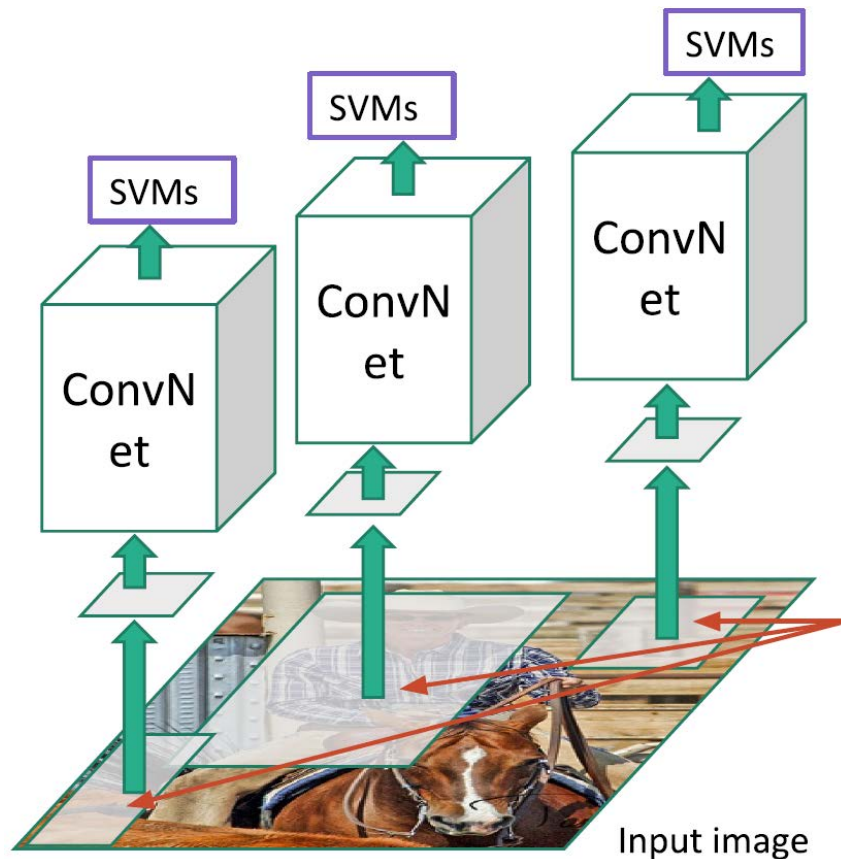
R-CNN



R-CNN



R-CNN



Classify regions with SVMs

Forward each region through ConvNet

Warped image regions (224*224)

Regions of interest (ROI) from a proposal method (~2k)

R-CNN

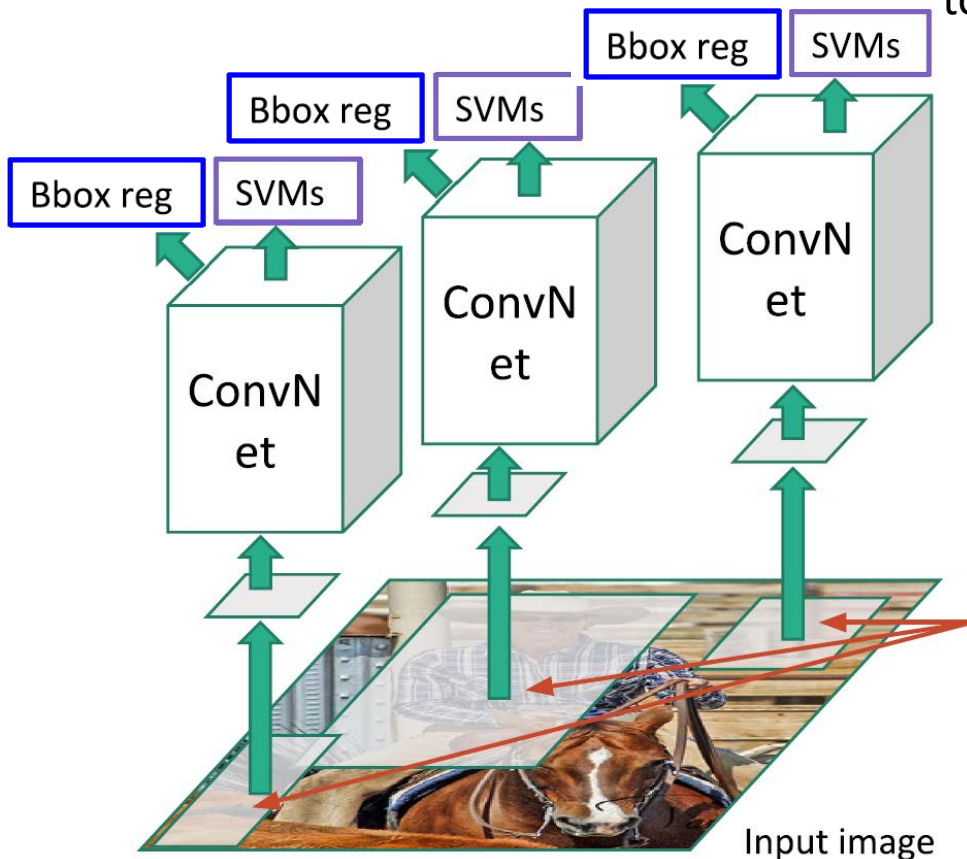
Linear progression: Predict corrections to ROI: 4 numbers (dx, dy, dw, dh)

Classify regions with SVMs

Forward each region through ConvNet

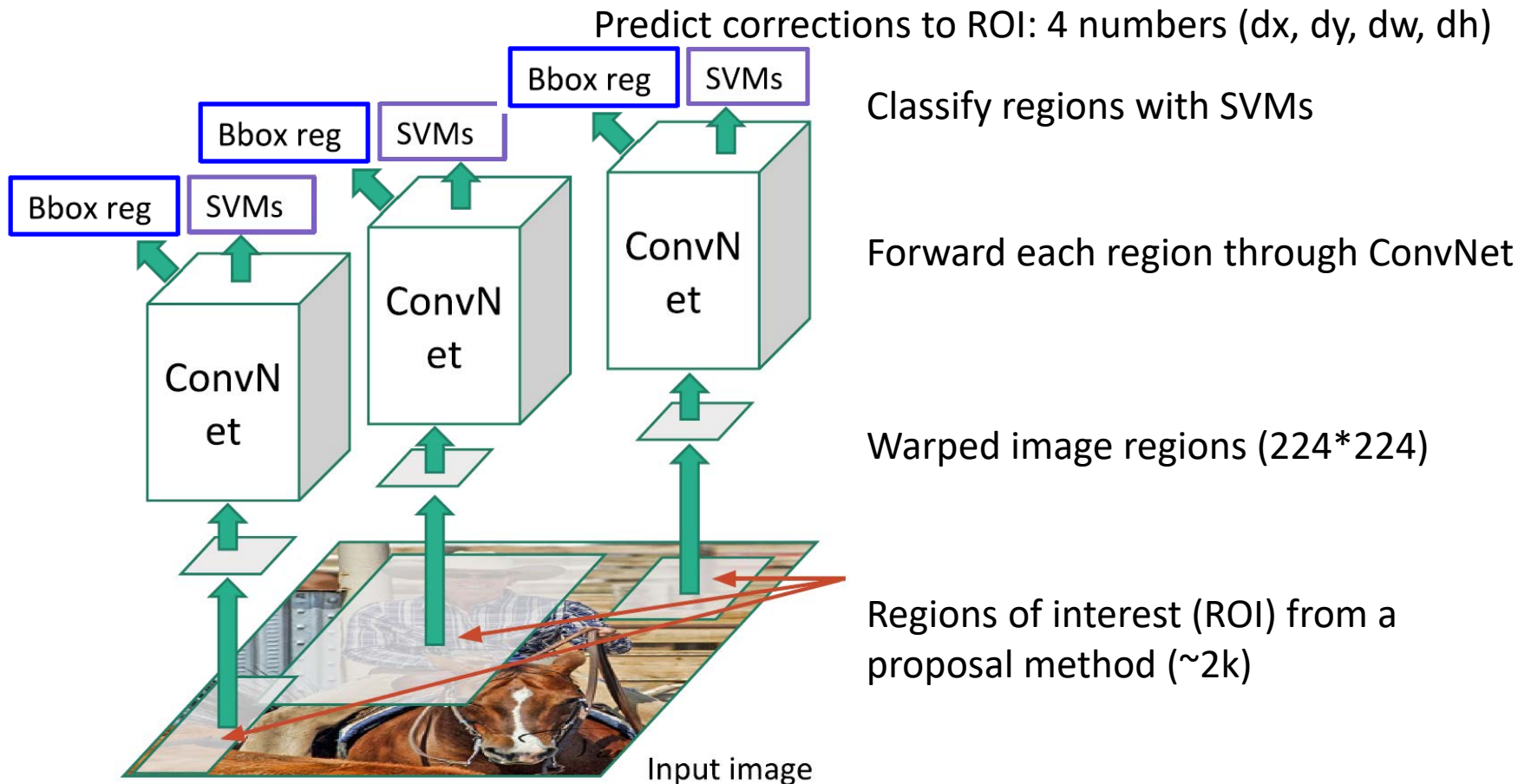
Warped image regions (224*224)

Regions of interest (ROI) from a proposal method (~2k)



R-CNN

- Problem: Very slow! Need to do $\sim 2k$ independent forward passes for each image

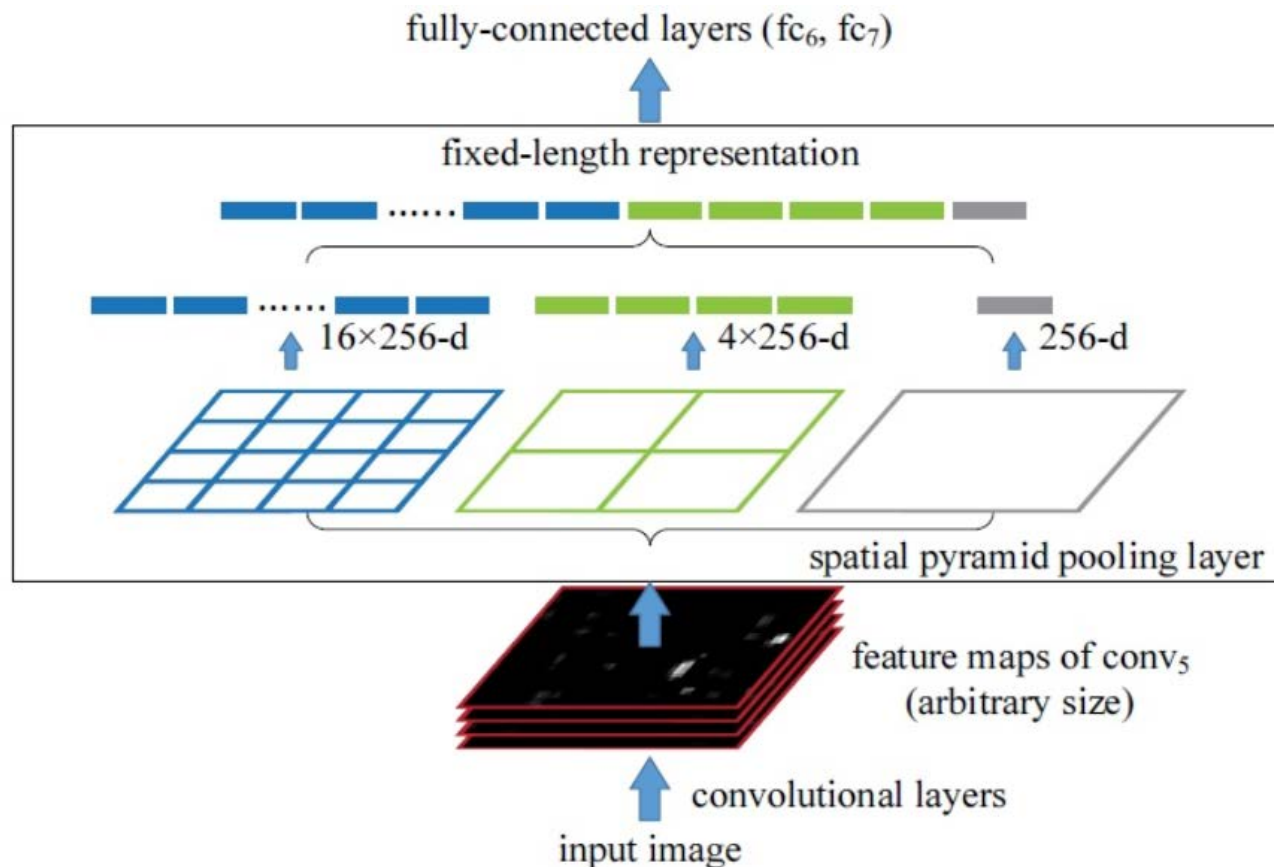


R-CNN: Problems

- Ad hoc training objectives
 - Fine-tune network with softmax classifier (log loss)
 - Train post-hoc linear SVMs (hinge loss)
 - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
 - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
 - Fixed by SPP-net [He et al. ECCV14]

SPPNet

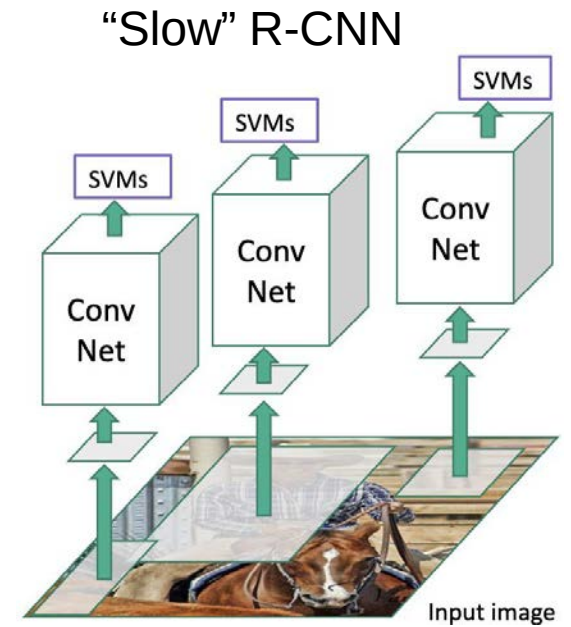
- Spatial pyramid pooling
- Generate a fixed-length representation regardless of image size/scale.
- Pyramid pooling is also robust to object deformations.



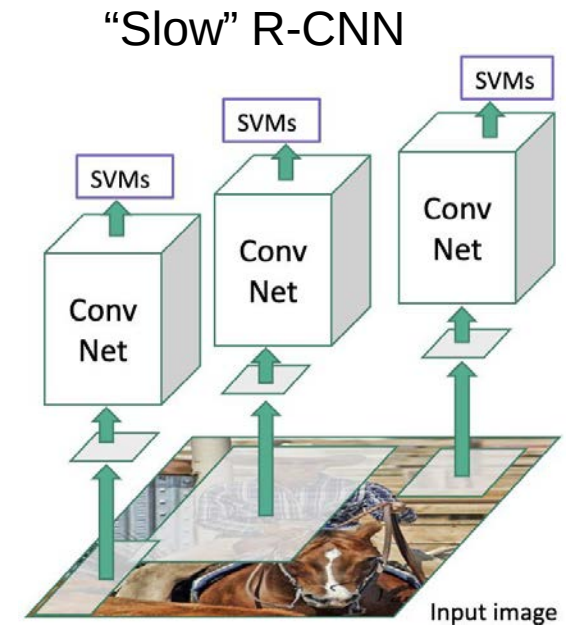
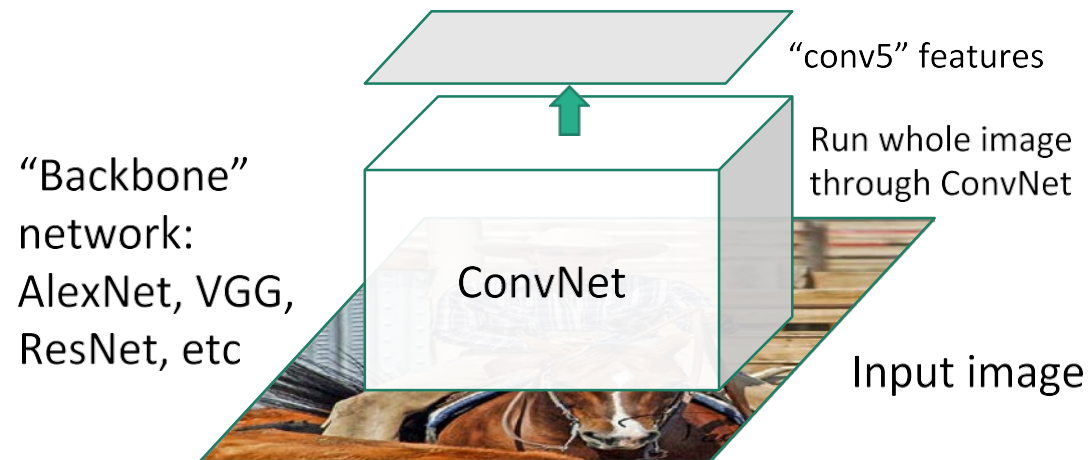
Fast R-CNN



Input image

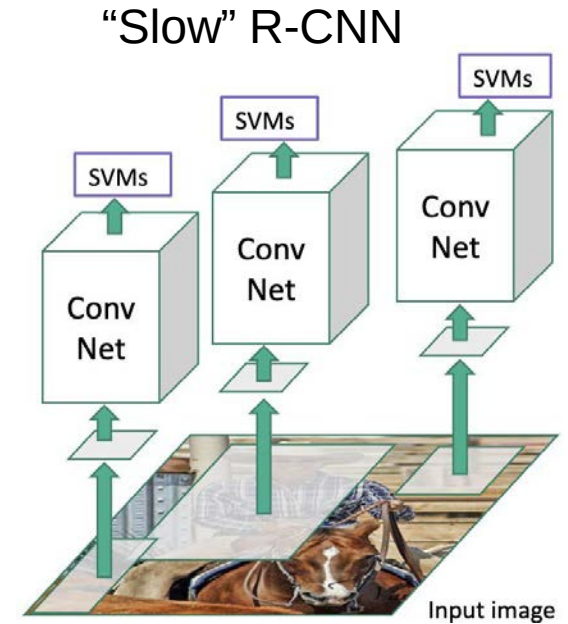
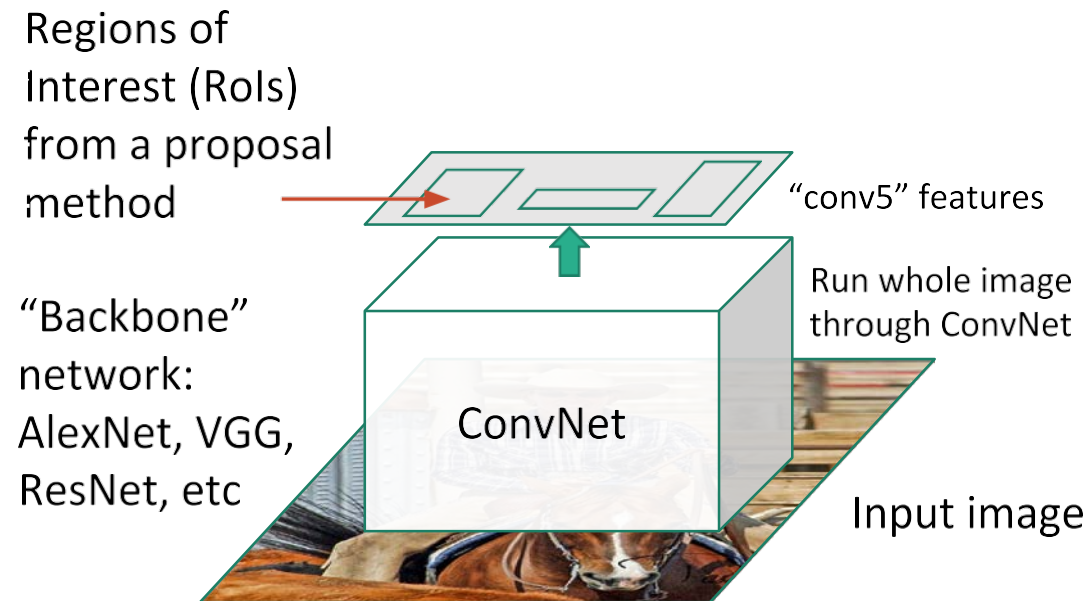


Fast R-CNN



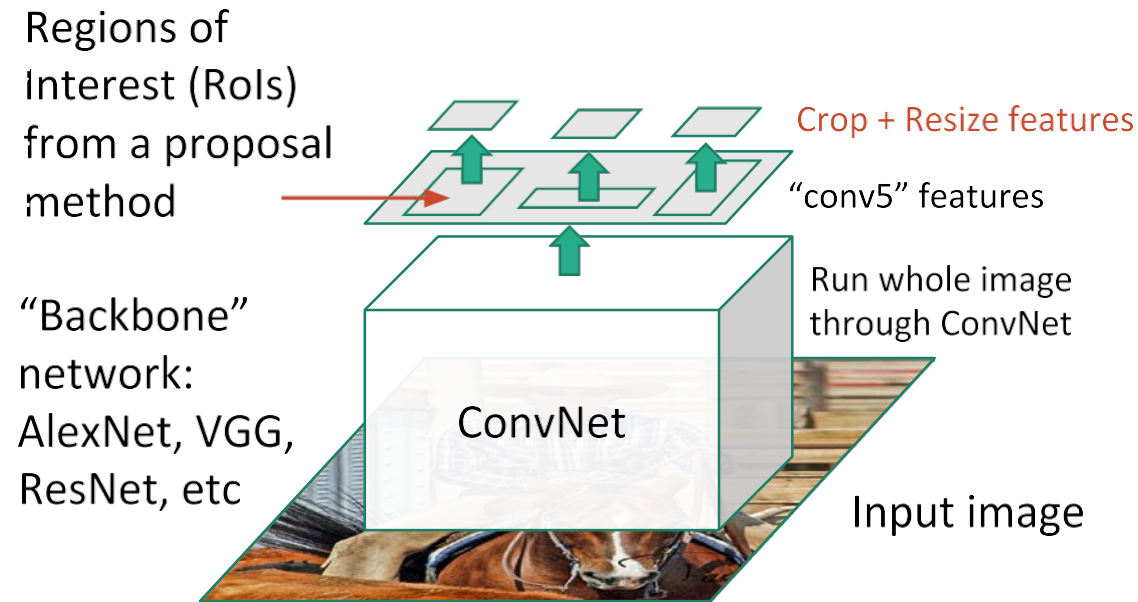
Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

Fast R-CNN

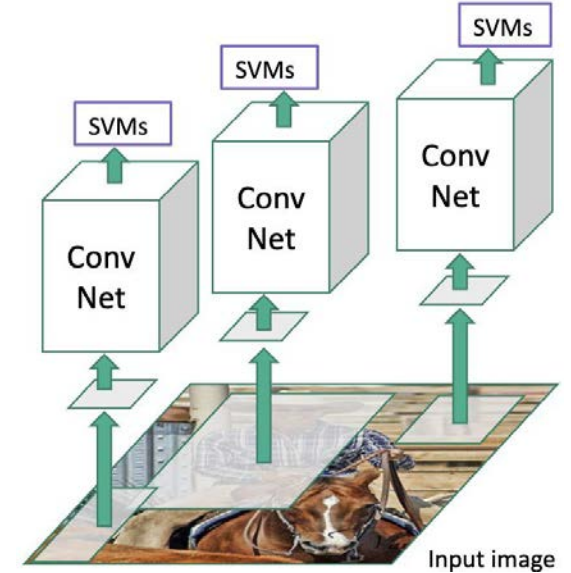


Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

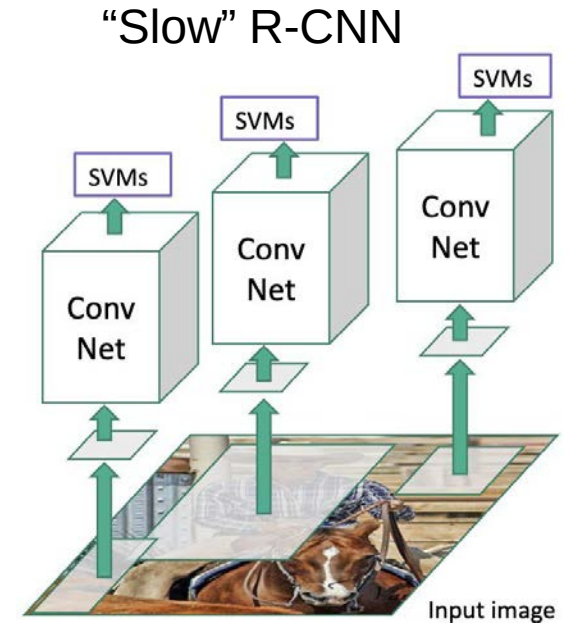
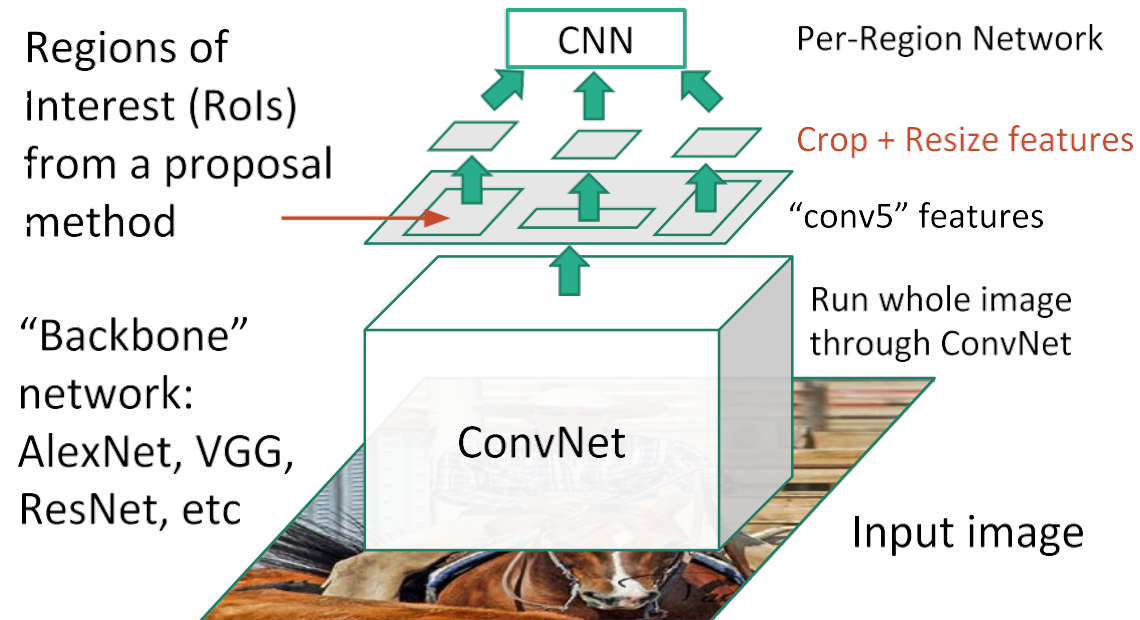
Fast R-CNN



"Slow" R-CNN

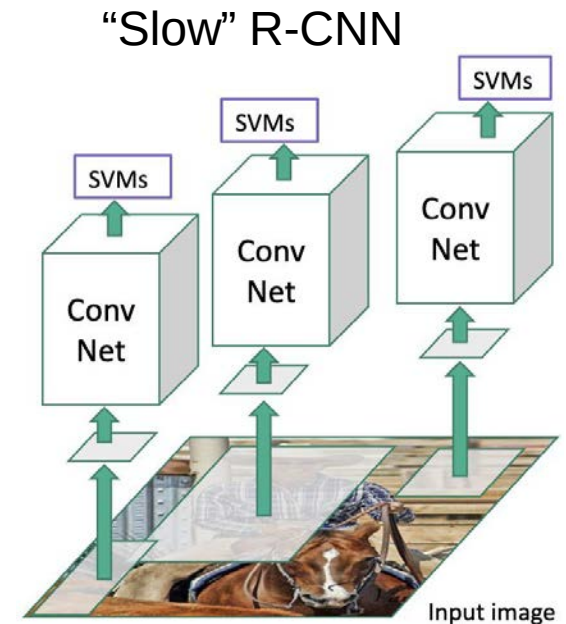
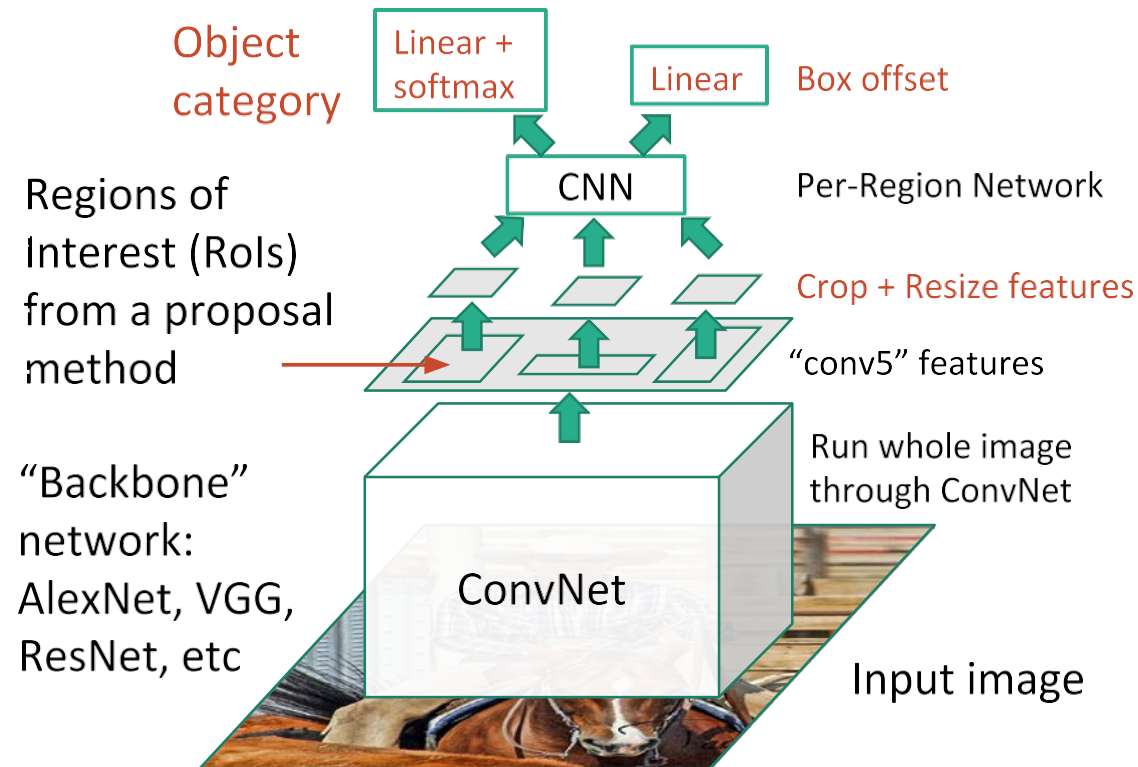


Fast R-CNN



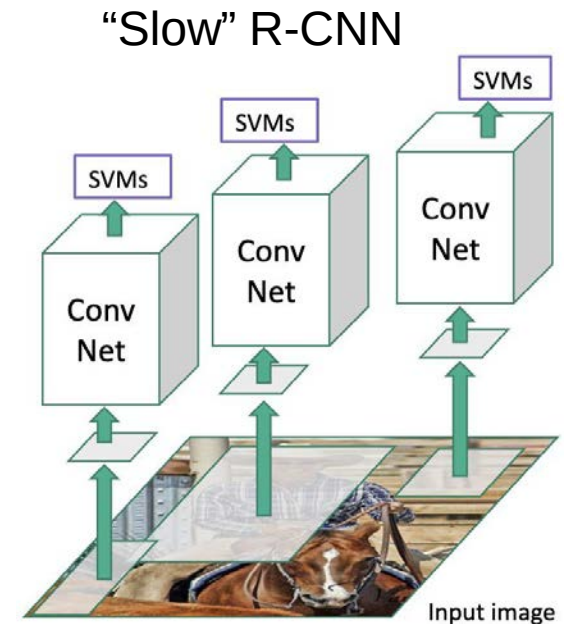
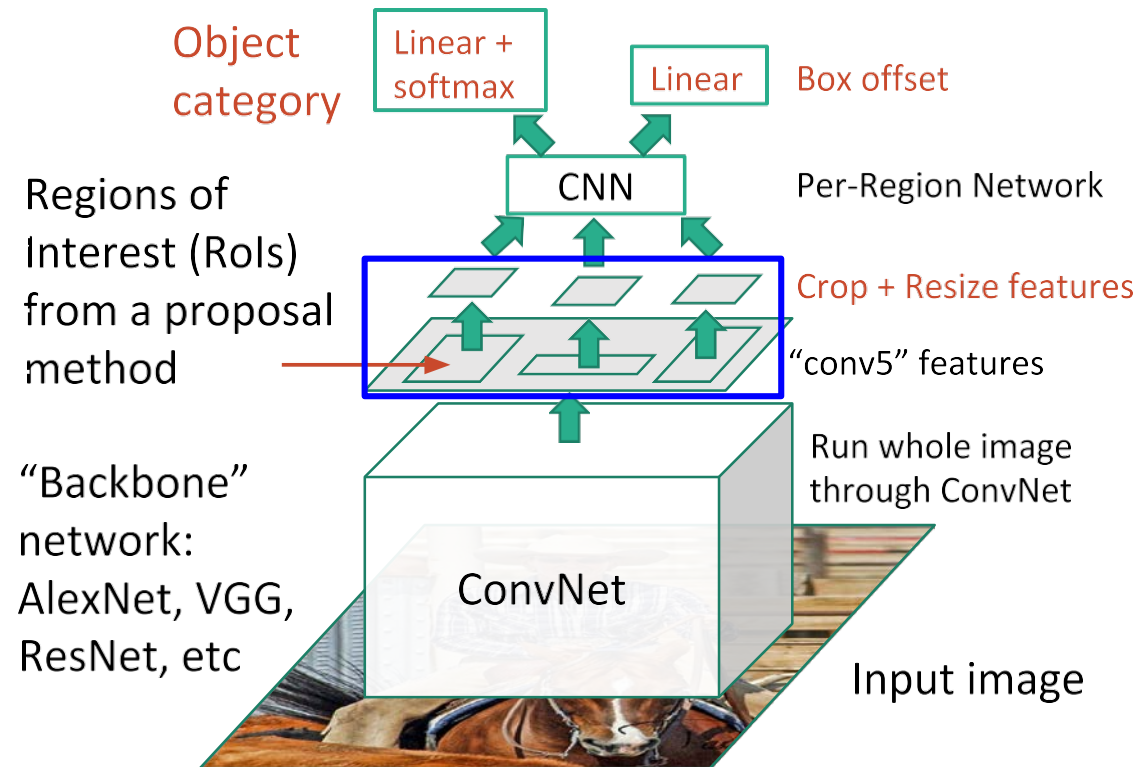
Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

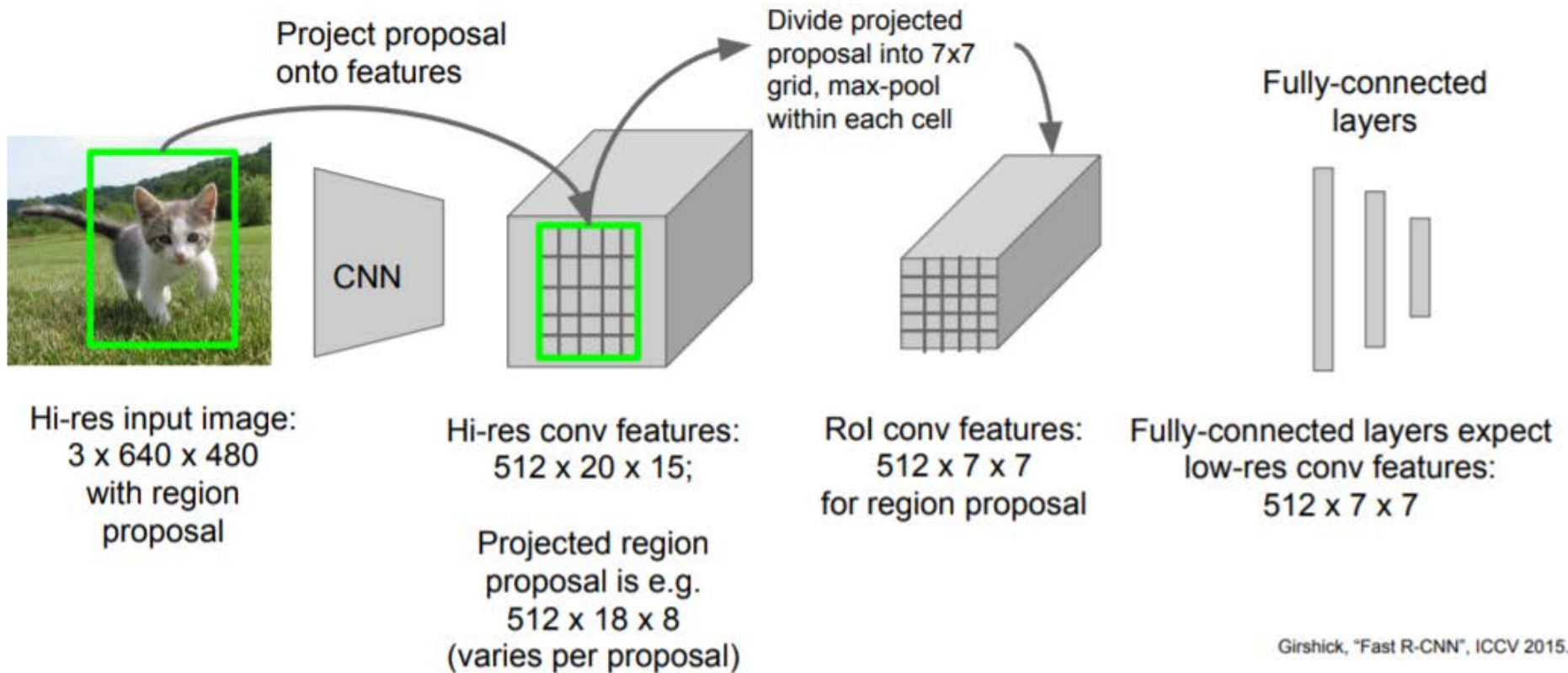
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

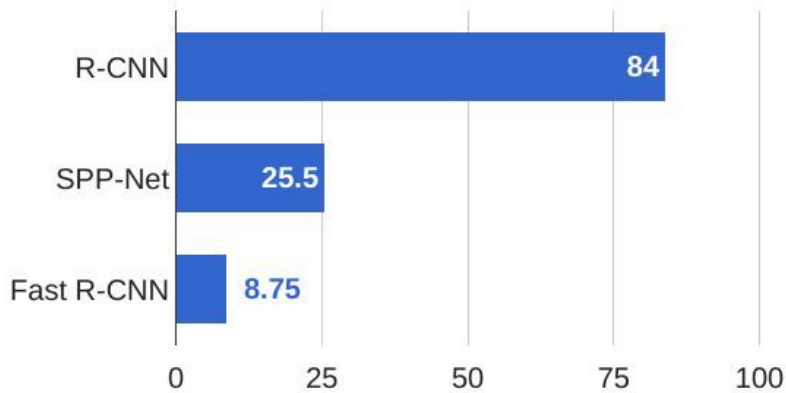
Fast R-CNN

■ ROI

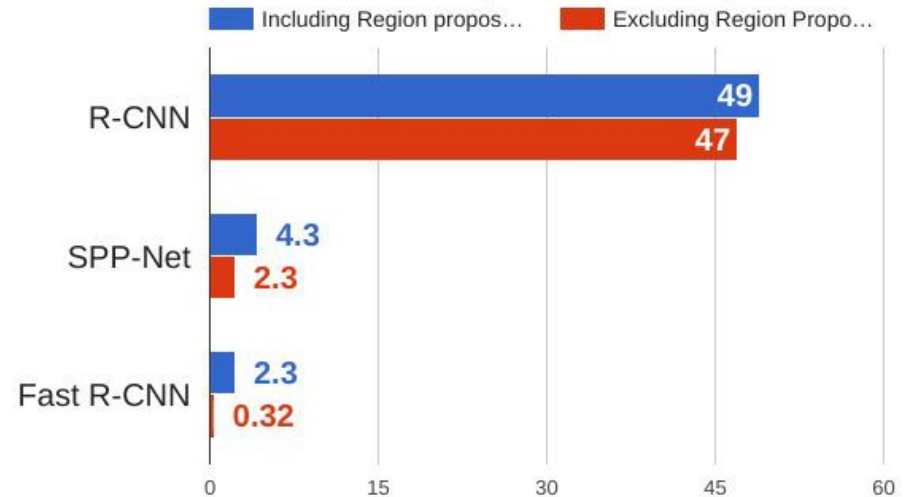


R-CNN vs Fast R-CNN

Training time (Hours)



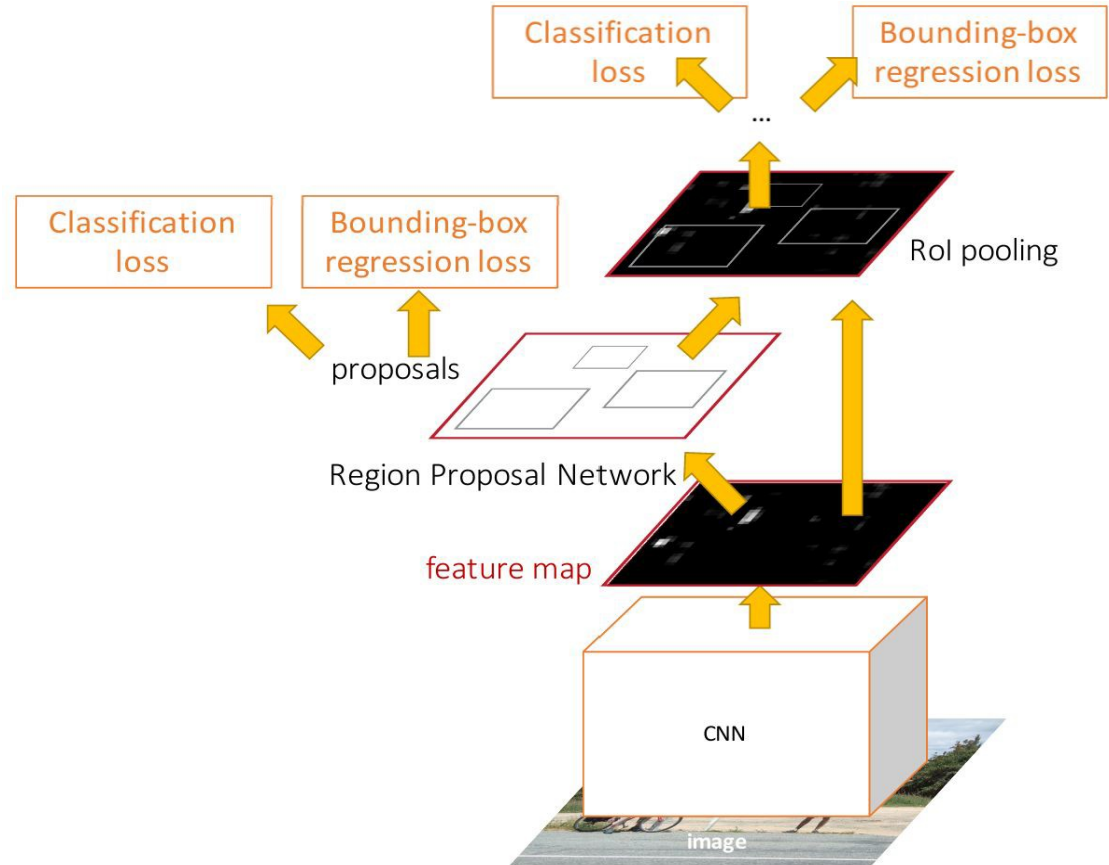
Test time (seconds)



Problem:
Runtime dominated
by region proposals!

Faster R-CNN

- Insert Region Proposal Network (RPN) to predict proposals from features
- Otherwise same as Fast R-CNN: Crop features for each proposal, classify each one



Region Proposal Network

- Make CNN do proposals!



Input Image
(e.g. 3 x 640 x 480)

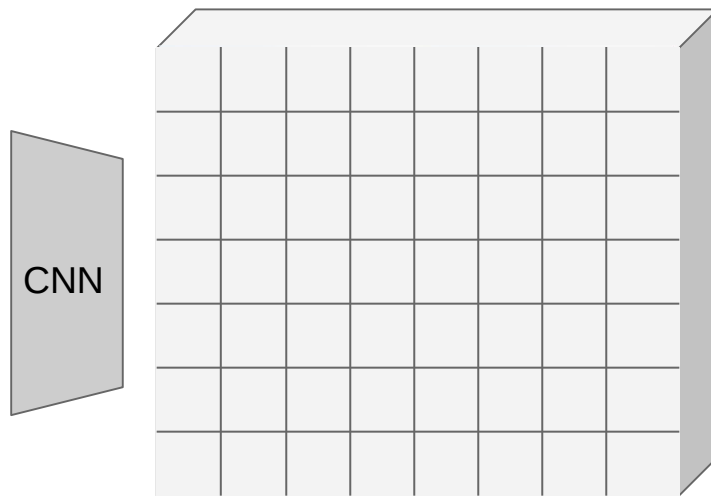


Image features
(e.g. 512 x 20 x 15)

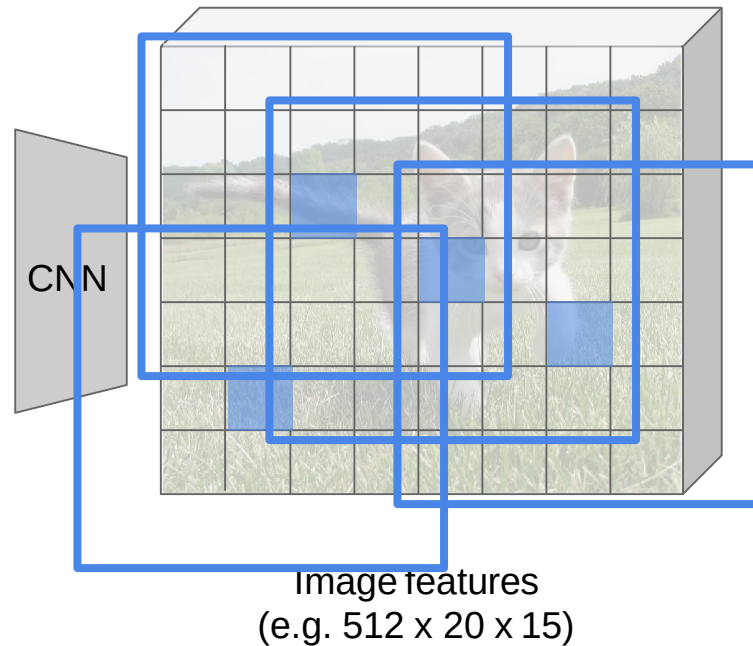
Region Proposal Network

- Make CNN do proposals!

Imagine an **anchor box** of fixed size at each point in the feature map



Input Image
(e.g. 3 x 640 x 480)

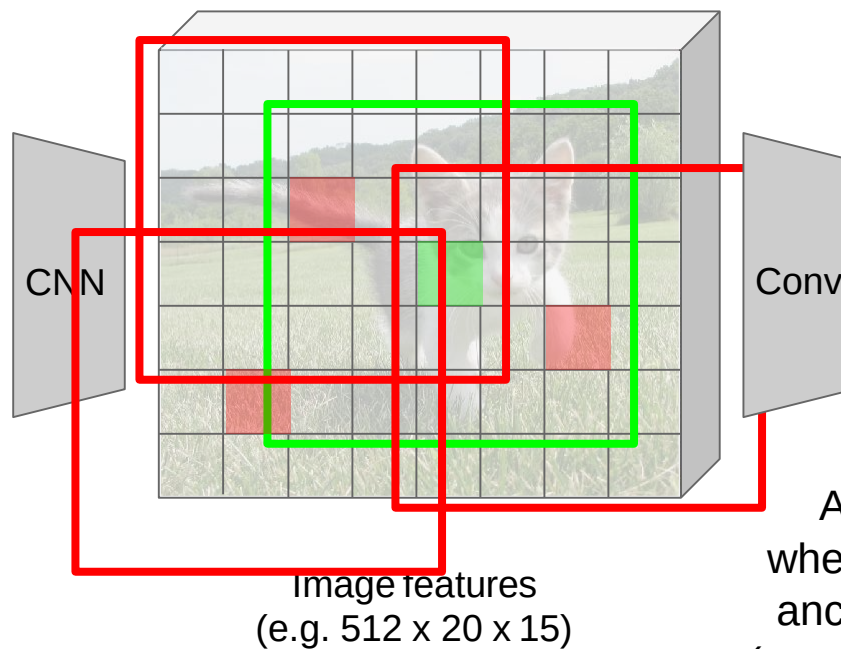


Region Proposal Network

- Make CNN do proposals!



Input Image
(e.g. 3 x 640 x 480)



Imagine an **anchor box**
of fixed size at each
point in the feature map

Anchor is an object?
1 x 20 x 15

At each point, predict
whether the corresponding
anchor contains an object
(per-pixel logistic regression)

Region Proposal Network

- Make CNN do proposals!



Input Image
(e.g. 3 x 640 x 480)

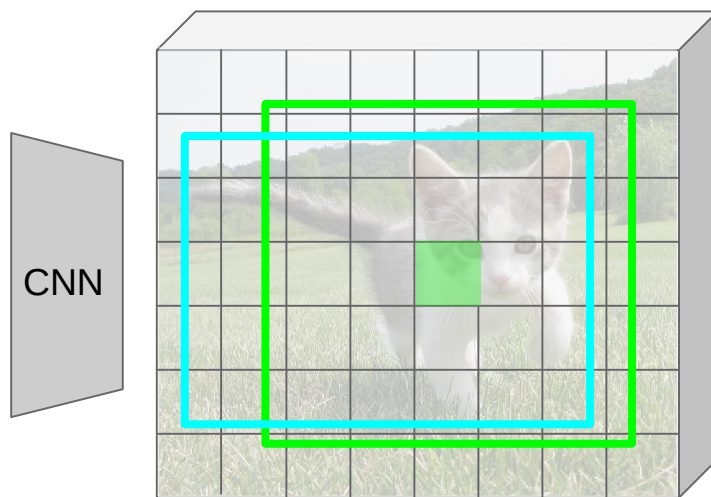


Image features
(e.g. 512 x 20 x 15)

Imagine an **anchor box**
of fixed size at each
point in the feature map

Anchor is an object?
1 x 20 x 15

Box transforms
4 x 20 x 15

For positive boxes, also predict
a transformation from the
anchor to the ground-truth box
(regress 4 numbers per pixel)

Region Proposal Network

- Make CNN do proposals!

In practice use K different anchor boxes of different size / scale at each point



Input Image
(e.g. $3 \times 640 \times 480$)

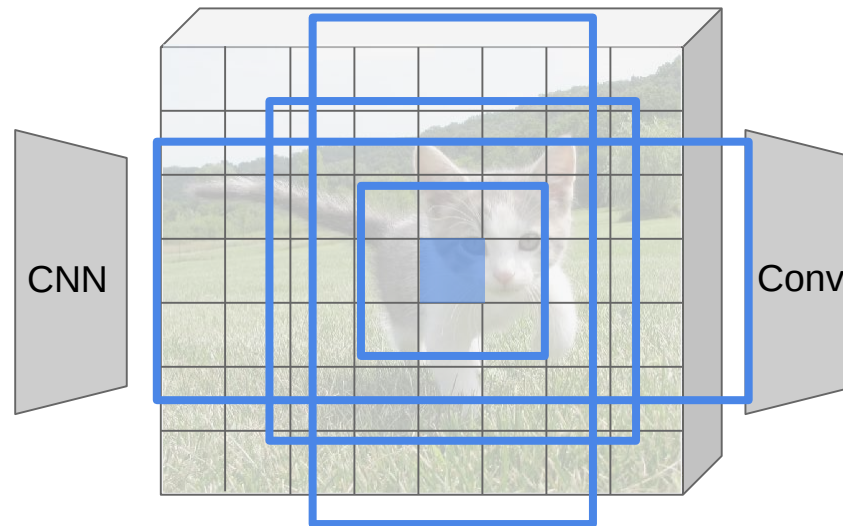


Image features
(e.g. $512 \times 20 \times 15$)

Anchor is an object?
 $K \times 20 \times 15$

Box transforms
 $4K \times 20 \times 15$

Region Proposal Network

- Make CNN do proposals!

In practice use K different anchor boxes of different size / scale at each point



Input Image
(e.g. $3 \times 640 \times 480$)

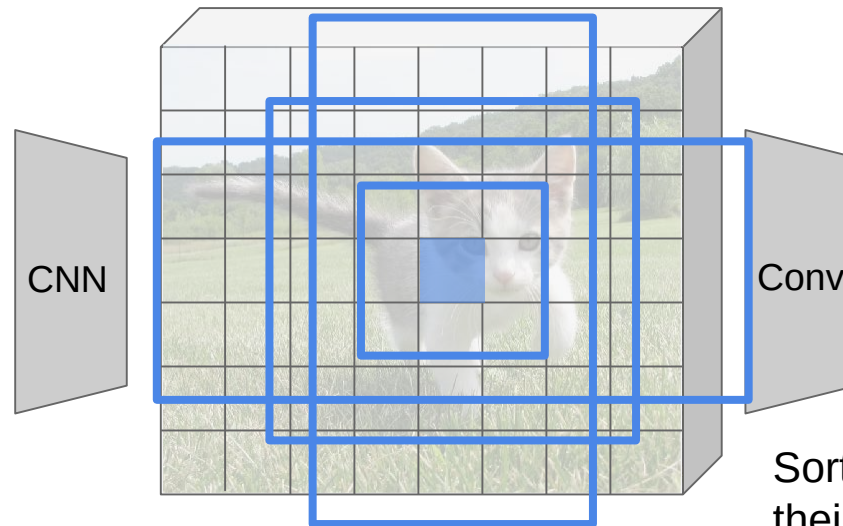


Image features
(e.g. $512 \times 20 \times 15$)

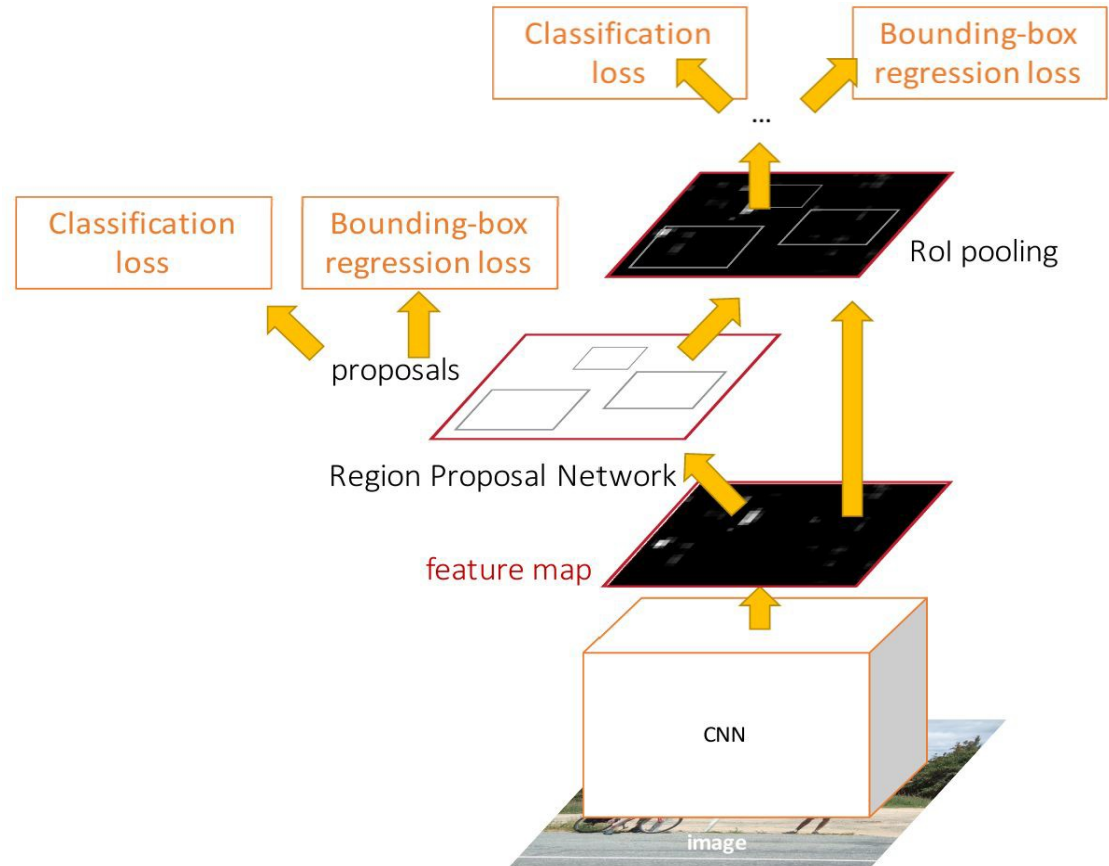
Anchor is an object?
 $K \times 20 \times 15$

Box transforms
 $4K \times 20 \times 15$

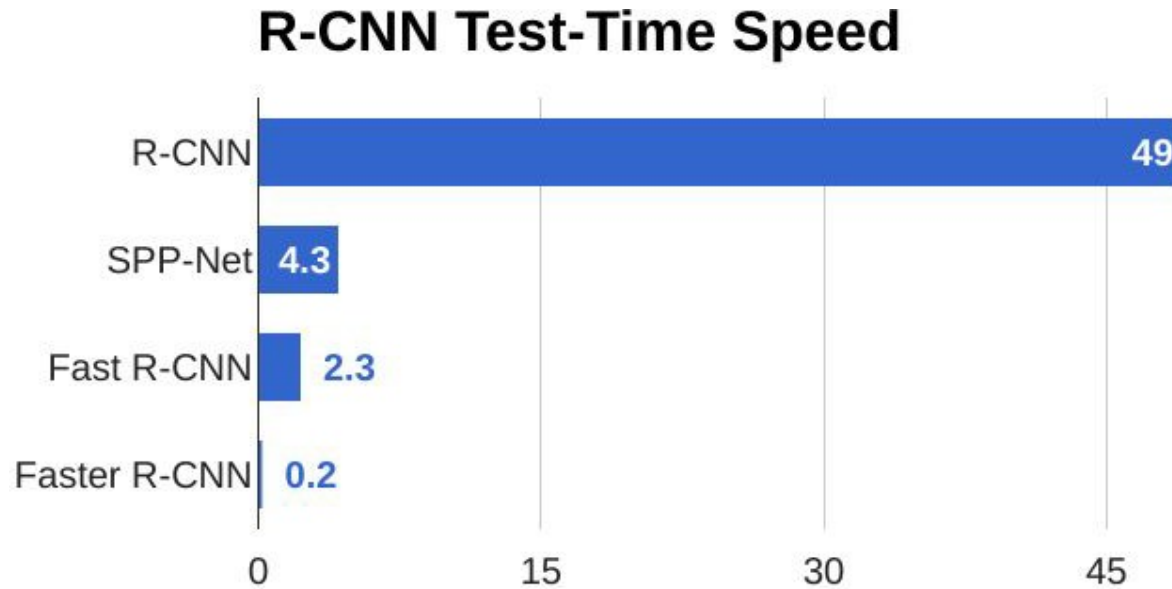
Sort the $K \times 20 \times 15$ boxes by their “object” score, take top ~ 300 as our proposals

Faster R-CNN

- Jointly train with 4 losses:
 - RPN classify object / not object
 - RPN regress box coordinates
 - Final classification score (object classes)
 - Final box coordinates

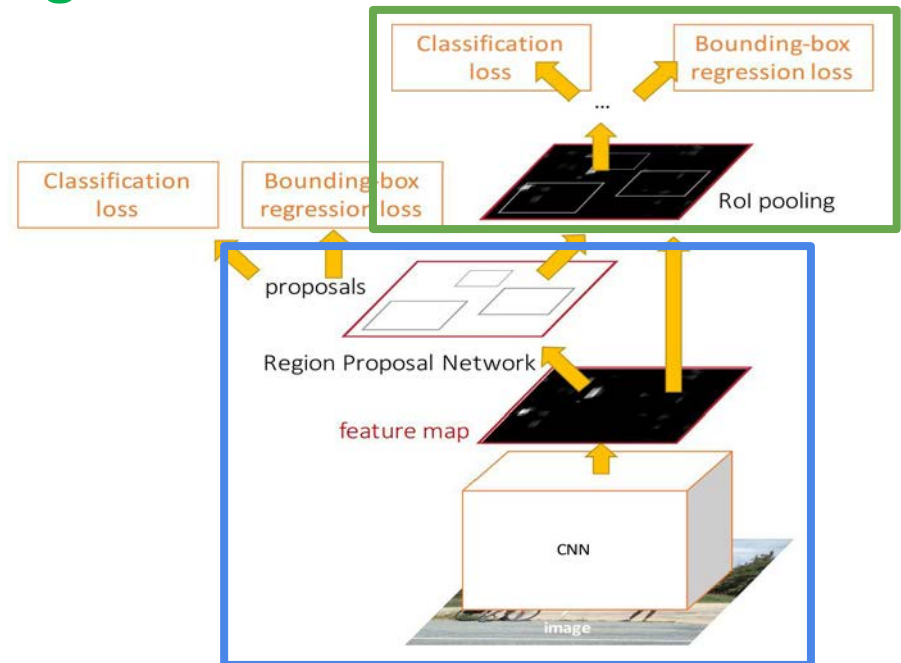


Faster R-CNN



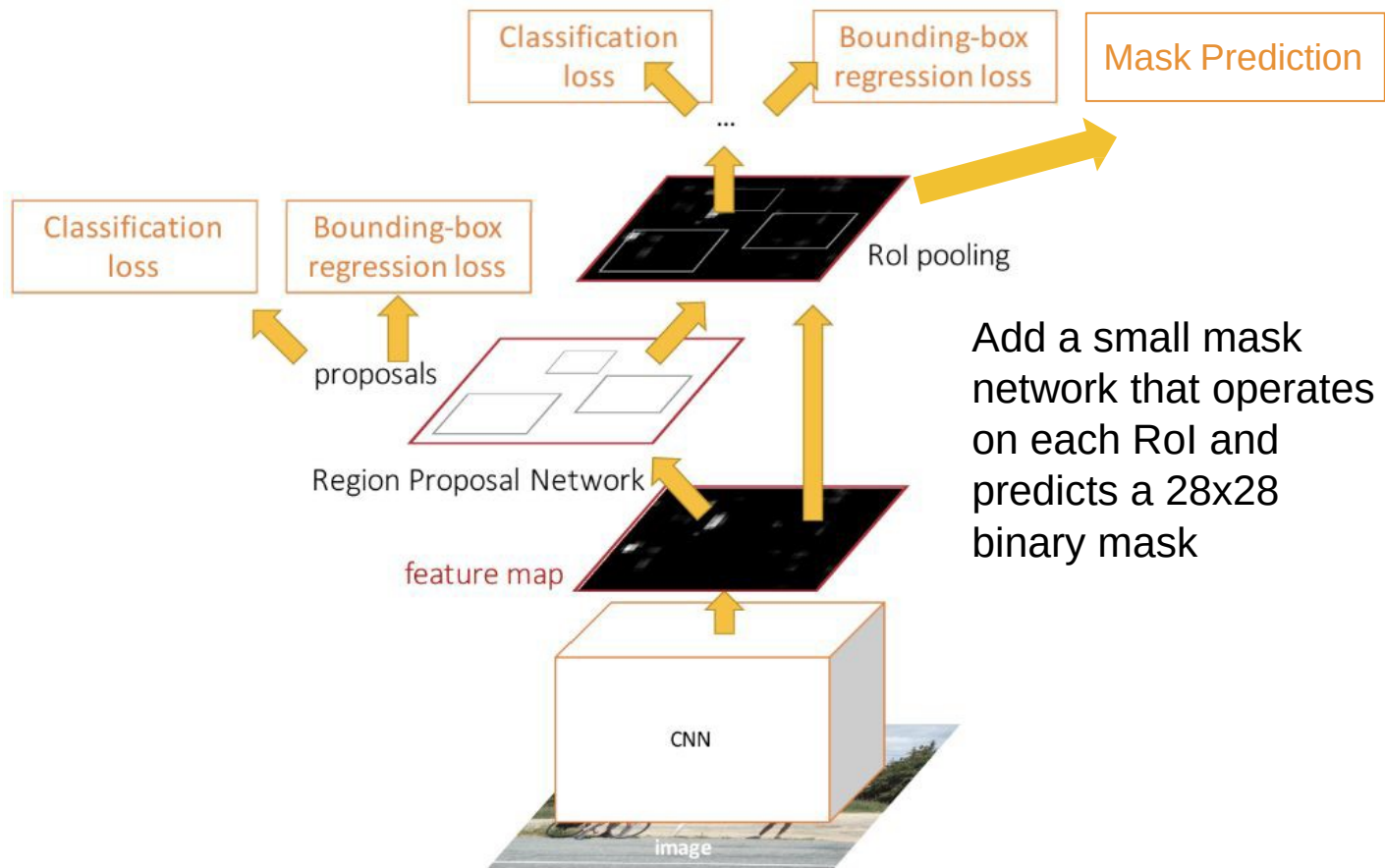
Faster R-CNN

- Two-stage object detector
- First stage: Run once per image
 - Backbone network
 - Region proposal network
- Second stage: Run once per region
 - Crop features: RoI pool / align
 - Predict object class
 - Prediction bbox offset

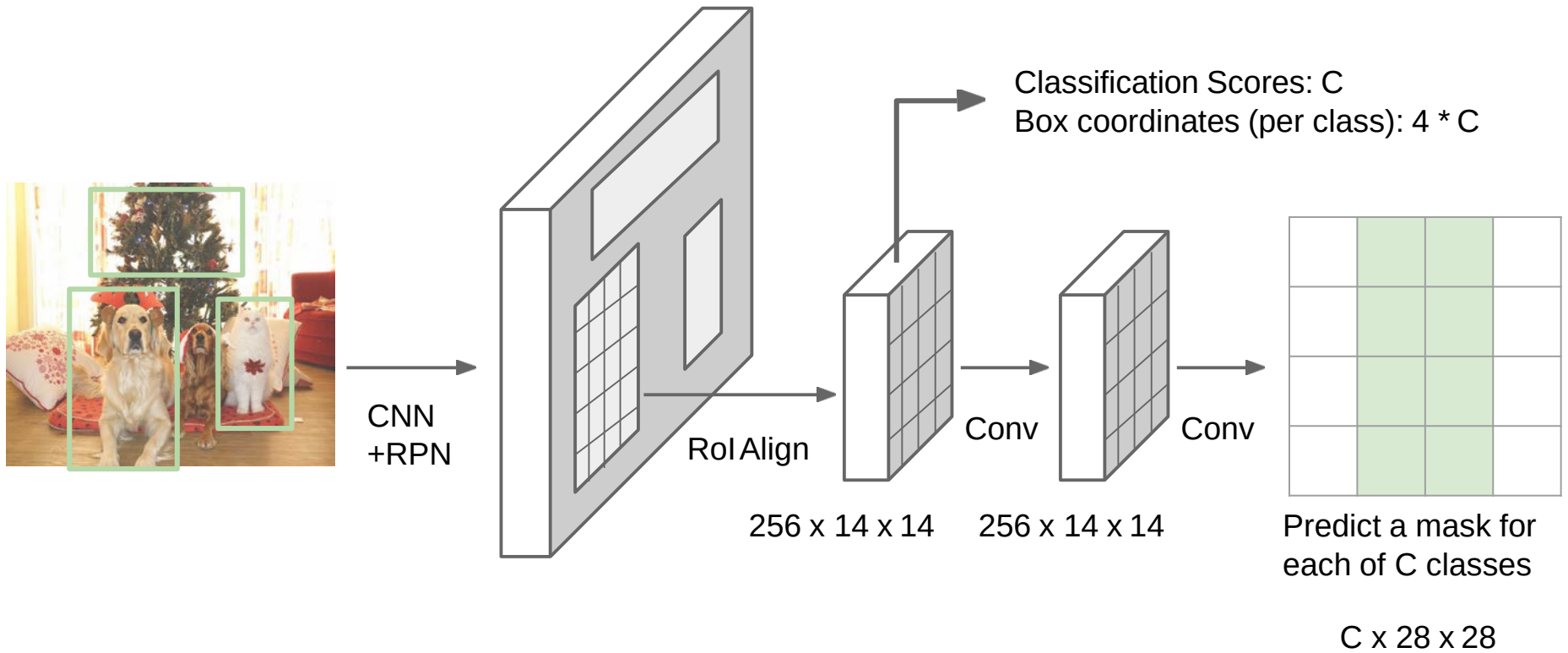


Mask R-CNN

- Instance Segmentation

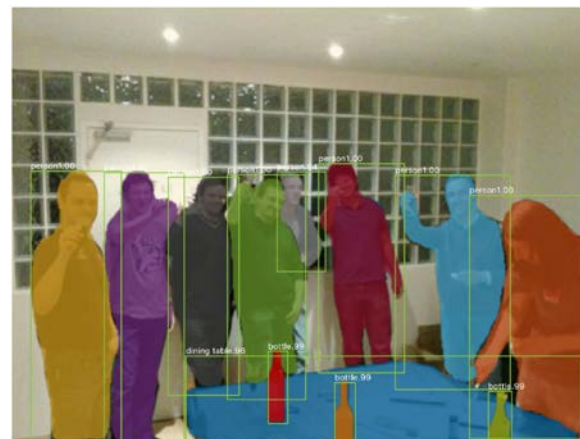
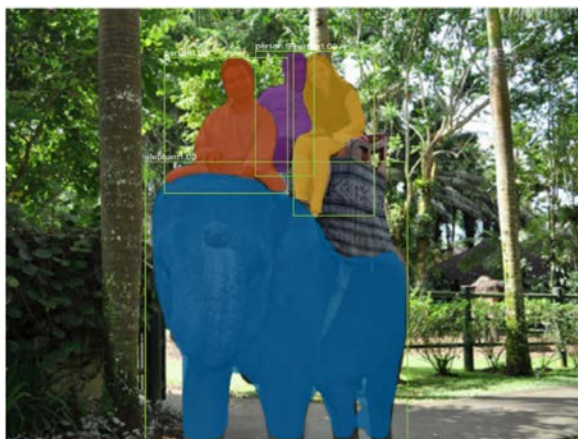


Mask R-CNN



Mask R-CNN

- Very Good Results



Mask R-CNN

- Lots of good implementations on GitHub!
- TensorFlow Detection API:
 - https://github.com/tensorflow/models/tree/master/research/object_detection
 - Faster RCNN, SSD, RFCN, Mask R-CNN
- Caffe2 Detectron:
 - <https://github.com/facebookresearch/Detectron>
 - Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN
- CUHK MMDetection framework
 - <https://github.com/open-mmlab/mmdetection>

Evaluation

- Project presentation (8%)
 - 10 mins presentation+5 mins Q&A
 - Evaluated by me and other students (poll)
- Reports: (8%)
 - Use standard latex template for a conference
 - <http://ras.papercept.net/conferences/support/tex.php>
 - **Overleaf** is good way to write a paper with latex
 - Around 6 pages
 - Including abstract, introduction, method, results and conclusion
 - Submit before Dec. 10th by email per group with zip file, including all materials for the paper
- Algorithm Performance (4%)
- All the codes should be open (a link)

Presentation arrangement

SID	NAME	GROUP	TASK
55228648	HE Guan Cheng	1	Segmentation
55234839	LAM King Hei	1	Segmentation
55224293	LEE Hung Hin	1	Segmentation
55216557	TSOI Hing Chung Sunny	1	Segmentation
55236833	WONG Yat Chau	1	Segmentation
55236476	LAM Hau Sze	2	Classification
55211875	LI Cheuk Yin	2	Classification
55235166	SIU Wai Fung	2	Classification
55099428	YU Tsz Ho	2	Classification
40132316	BARRIOS VIDAL Lucia Alejandra	2	Classification
54814615	CHAN Chung Hong	3	Classification
54398836	CHENG Kai Ming	3	Classification
55299045	LEE Man Lok	3	Classification
54845886	TANG Chun Ki	3	Classification
55297200	WONG Kai Chun	3	Classification
55237079	CHAN Cheuk Fung	4	Segmentation
55239915	HO Shing Fung	4	Segmentation
55238746	LEE Wing Yan	4	Segmentation
54850792	LUI Yu Yuet	4	Segmentation
55241843	NG Chak Ka	4	Segmentation
54777375	FANG Yucheng	5	Classification
55202608	HAO Yining	5	Classification
55203003	HE Jinwei	5	Classification
55200351	XU Tianxiao	5	Classification
55202909	YANG Qihan	5	Classification
55228071	CHAN Ding Wan	6	Segmentation
55234090	KWOK Wah Chung	6	Segmentation
55218956	LAW Yick Ming	6	Segmentation
55240232	TSANG Yiu Wai	6	Segmentation
54816829	WONG Ka Ho	6	Segmentation
55218354	AU Cheuk Ming	7	Classification
54845745	TSE Siu Ki	7	Classification
55218551	BUTT Ka Kiu	7	Classification
55240293	CHAN Ka Hung	7	Classification
55210290	SHAM Pui Hei	7	Classification
54818720	SIN Joy Leong	8	Segmentation
55293157	TANG Ho Yan Sophia	8	Segmentation
54400770	WONG Chak Kong	8	Segmentation
55231192	YAU Hiu Kwan	8	Segmentation

