
Computer vision

Computer Vision

Lecture 7: Object Detection

Dr. Dina Khattab

dina.khattab@cis.asu.edu.eg

Scientific Computing Department

Instructor:	Dr. Dina Khattab
Email:	dina.khattab@cis.asu.edu.eg
Office:	Main Building – 4^{th} floor – Room 302
Office Hours:	Monday 12:00 - 2:00 PM Thursday 11:00 AM to 12:00 PM

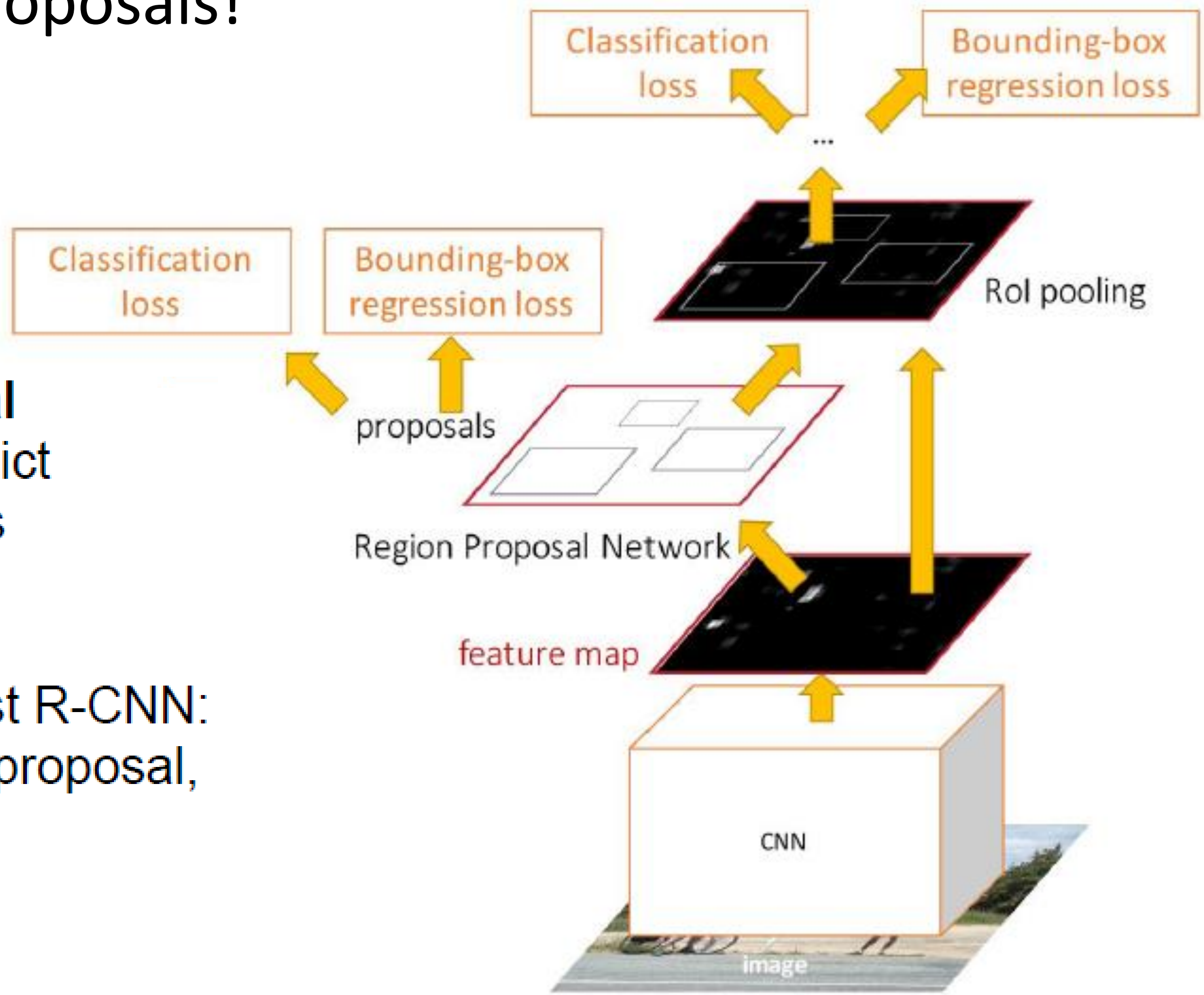
Agenda

- Region-based Detectors
  - Faster R-CNN
- Single Shot Detectors
  - YOLO & SSD

Faster R-CNN:
Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

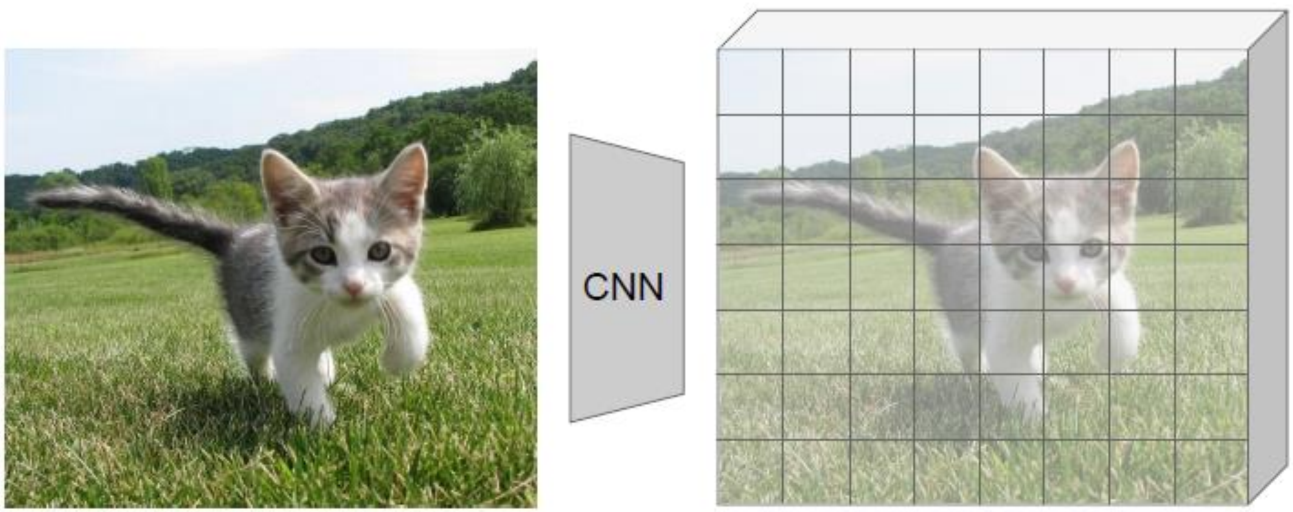
Otherwise same as Fast R-CNN: Crop features for each proposal, classify each one



Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN) using Anchors

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

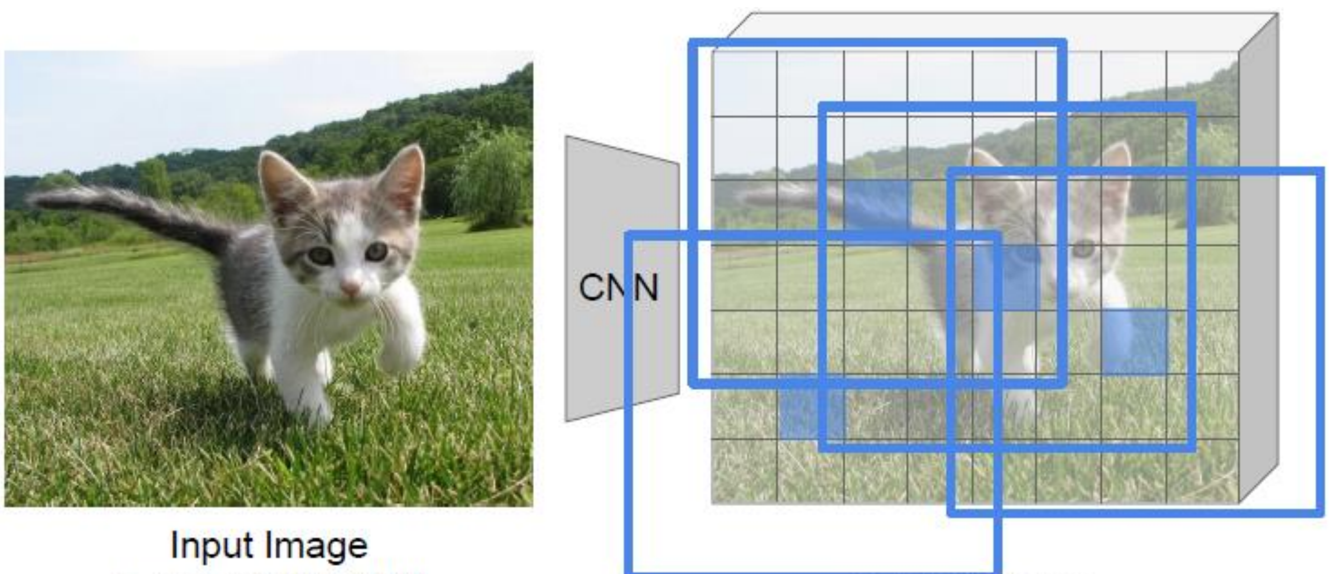


Input Image
(e.g. 3 x 640 x 480)

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

Region Proposal Network

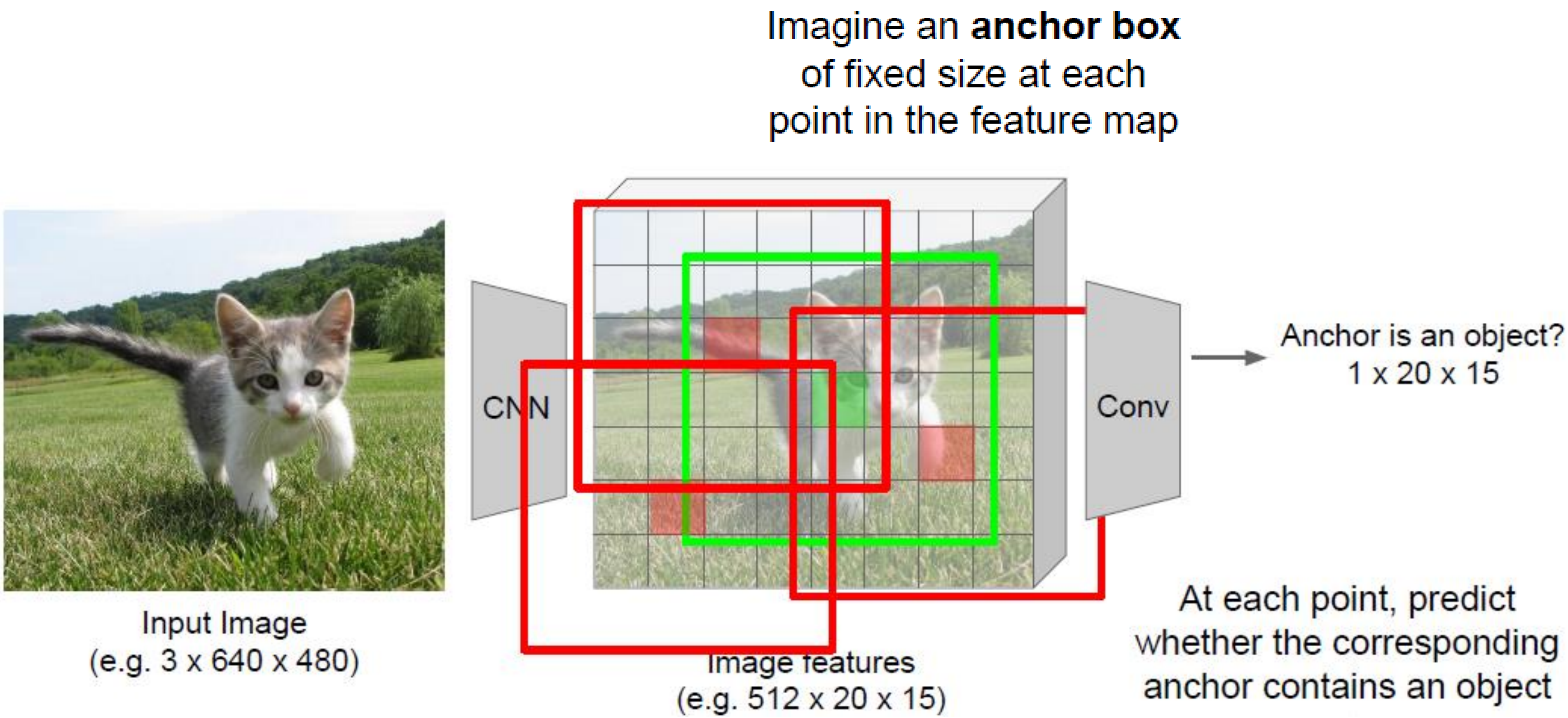
Imagine an **anchor box** of fixed size at each point in the feature map



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

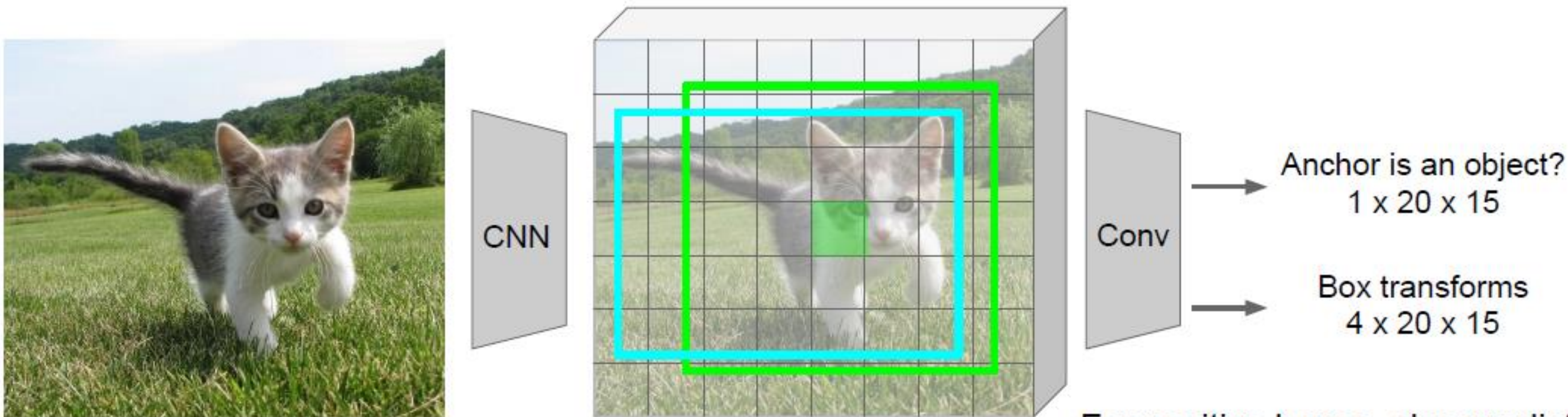
Image features
(e.g. 512 x 20 x 15)

Region Proposal Network



Region Proposal Network

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

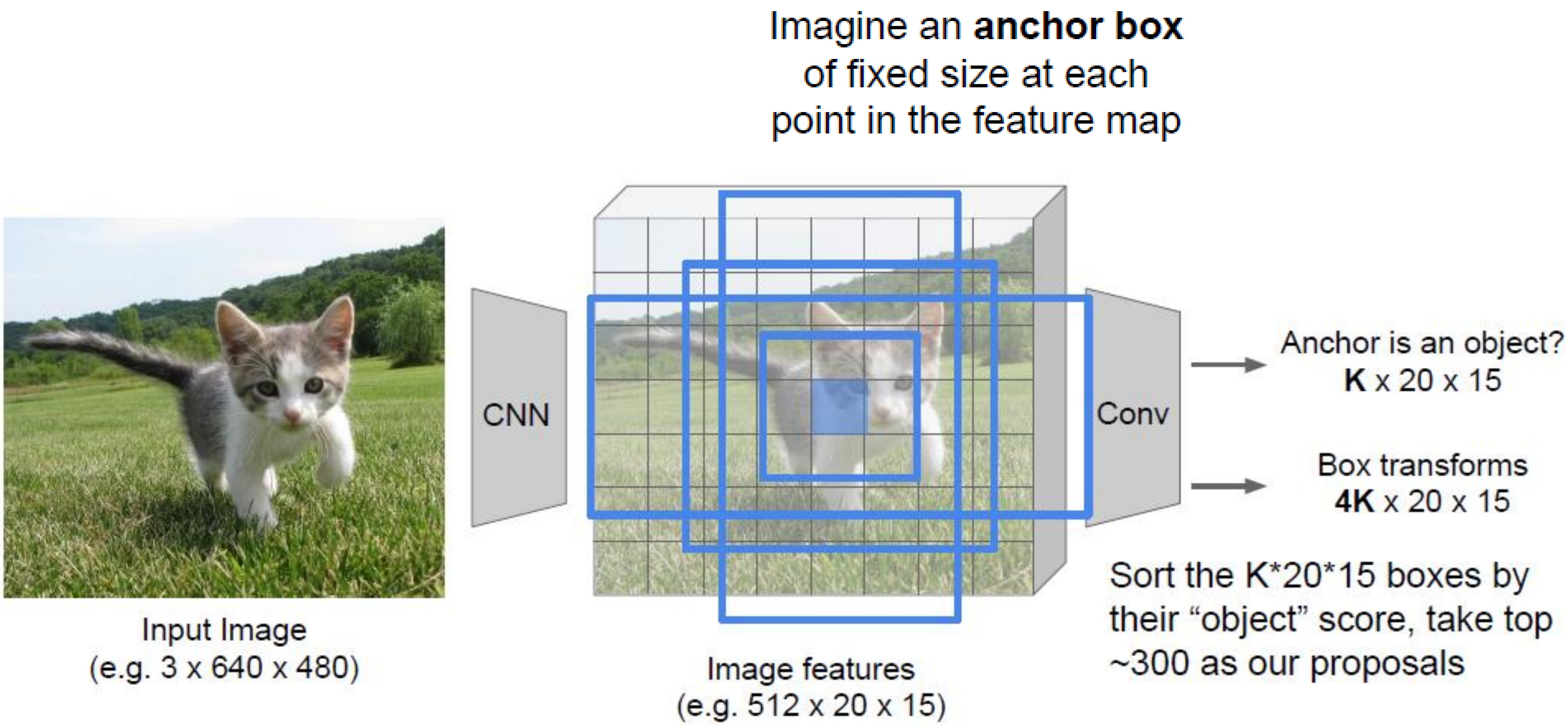


Input Image
(e.g. 3 x 640 x 480)

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

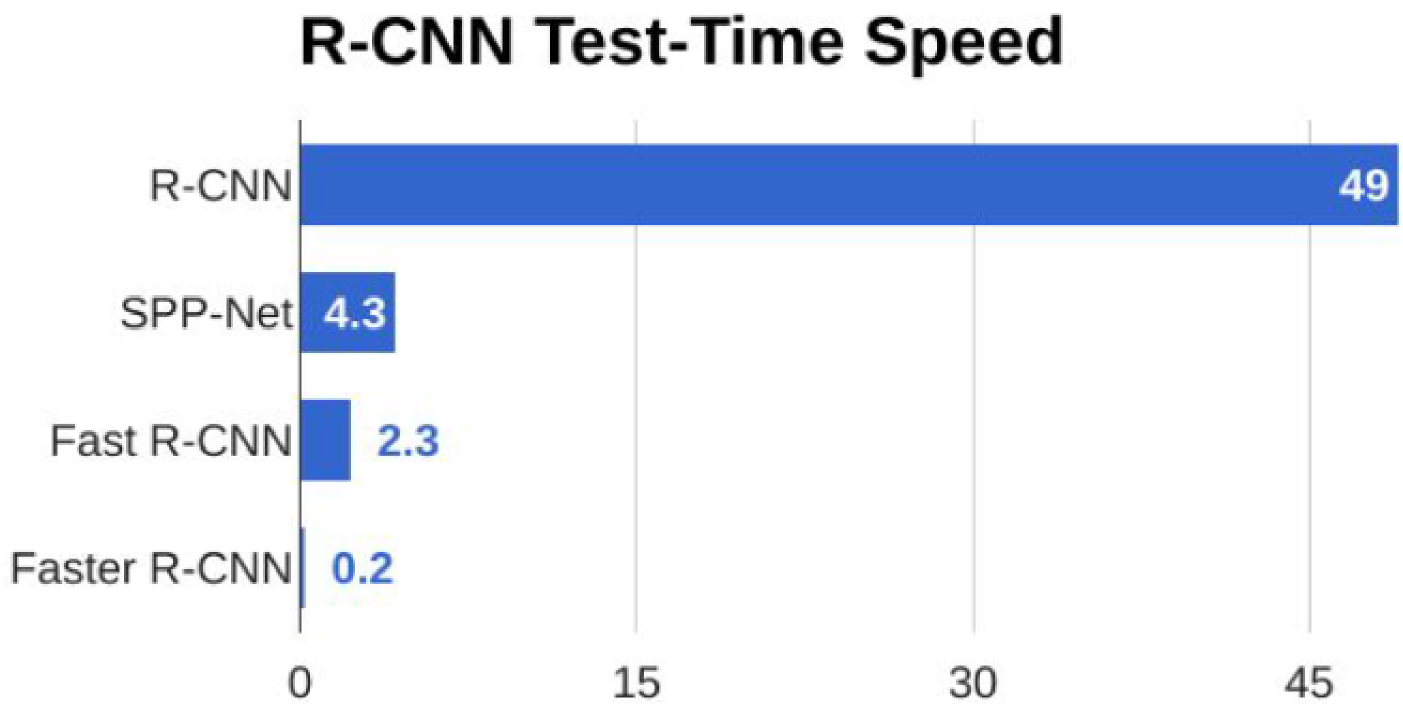
For positive boxes, also predict a transformation from the anchor to the ground-truth box

Region Proposal Network



Faster R-CNN:

Make CNN do proposals!



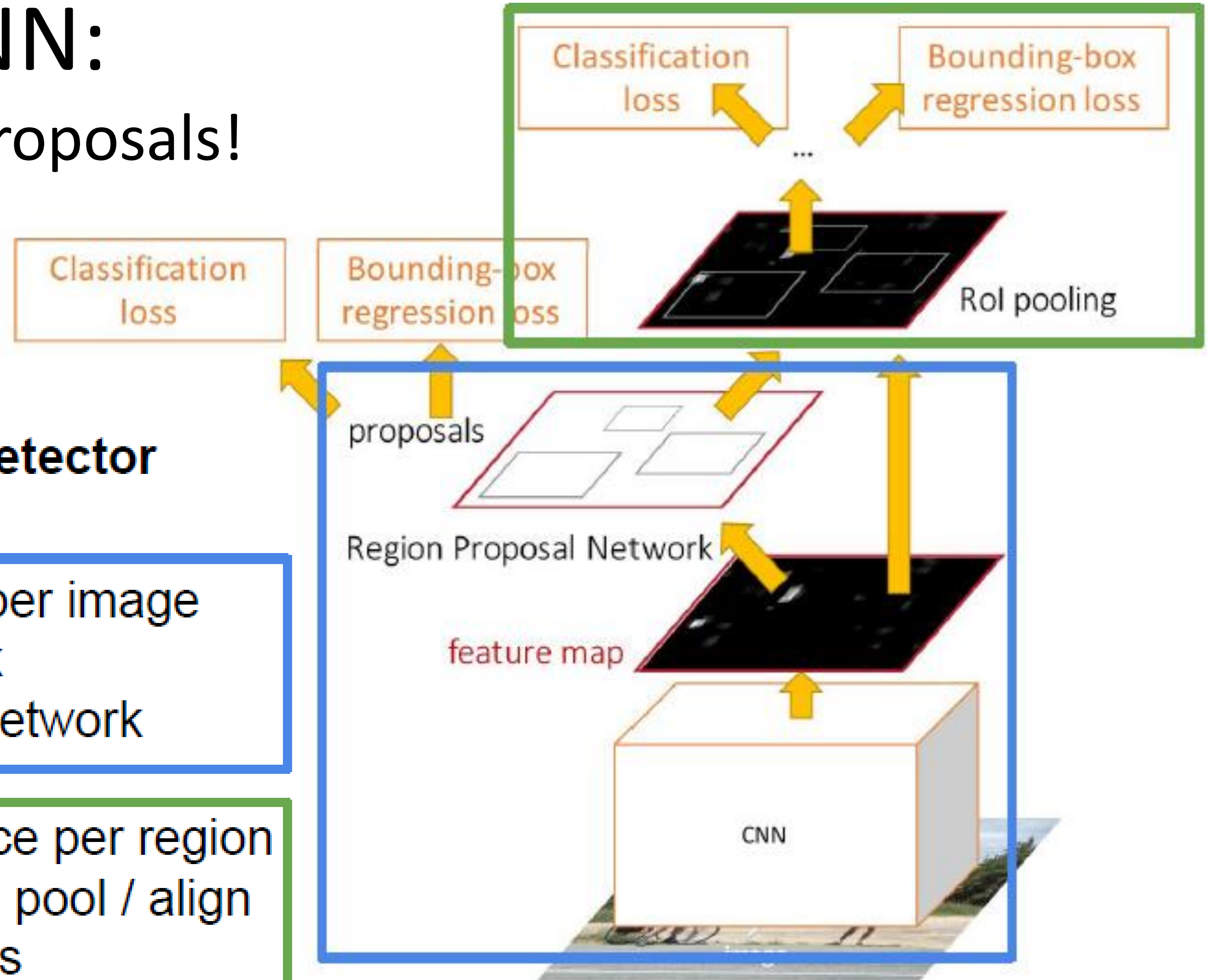
Faster R-CNN:

Make CNN do proposals!

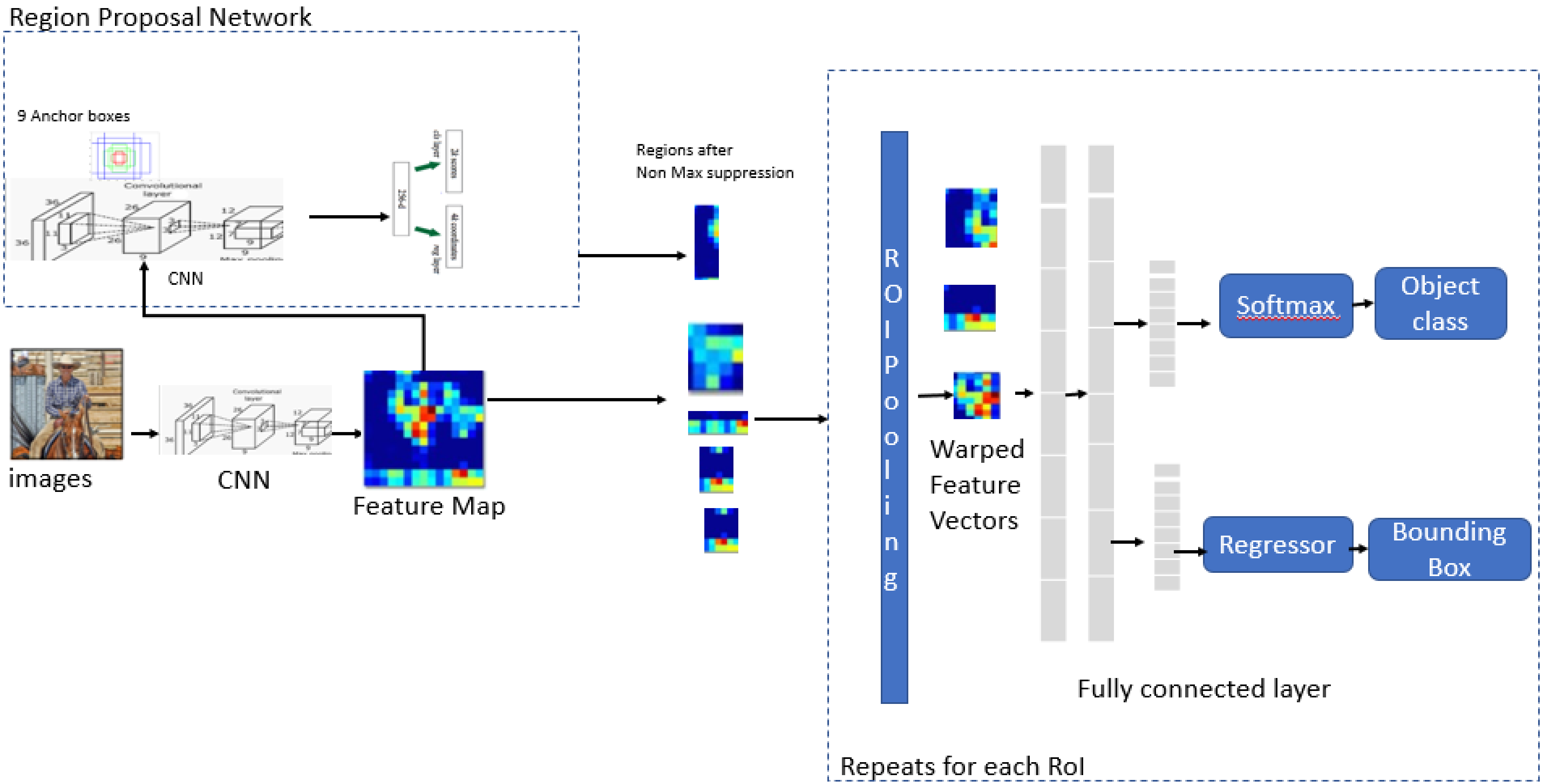
Faster R-CNN is a
Two-stage object detector

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



Faster RCNN

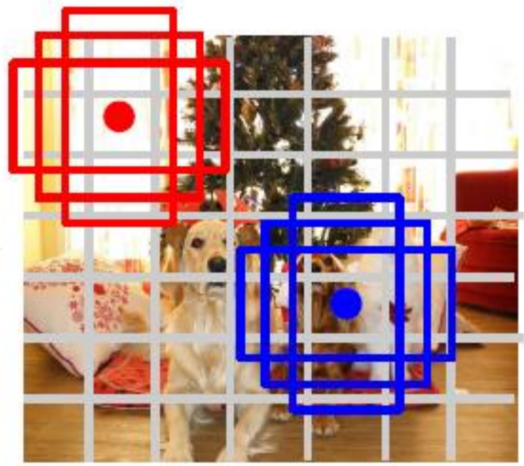


Region Proposal Network
9 Anchor boxes
CNN
Regions after
Non Max suppression
images
CNN
Feature Map
R
O
I
P
o
o
l
i
n
g
Warped
Feature
Vectors
Softmax
Object
class
Regressor
Bounding
Box
Fully connected layer
Repeats for each RoI

SINGLE SHOT DETECTORS

Single-Stage Object Detectors: YOLO / SSD

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:

- Regress from each of the B base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

Features of YOLO

- Sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearances, unlike the sliding window or region-based techniques. Thus making less than half the number of background errors compared to Fast R-CNN.

- It predicts all bounding boxes across all classes for an image simultaneously.

- Extremely fast and accurate (Speed: 45 frames per second)

Limitations of YOLO

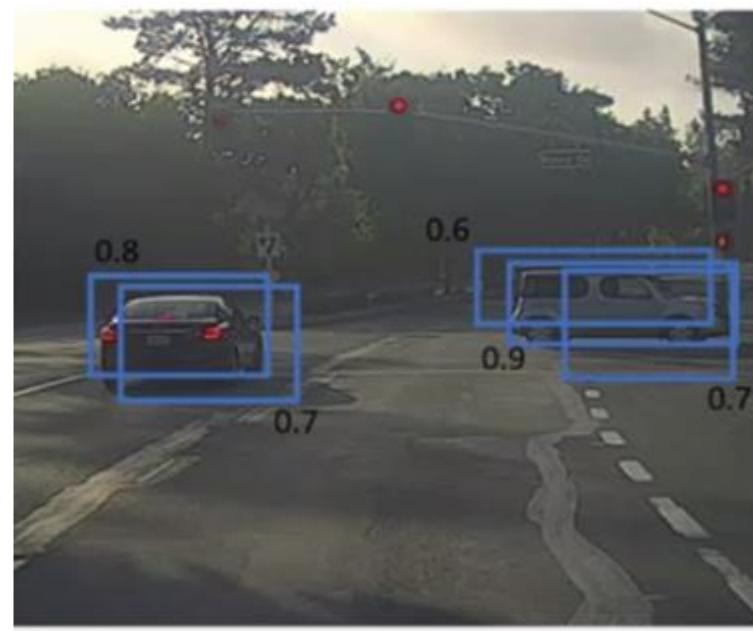
- Imposes strong spatial constraints on bounding box predictions since each grid cell can only have one class and this limits the number of nearby objects that the model can predict.
- Struggles with small objects that appear in groups, such as flocks of birds.
- Struggles to generalize to objects in new or unusual aspect ratios or configurations

Non-Max suppression for object detection

- Our objective is to detect an object just once with one bounding box. However, with object detection, we may find multiple detections for the same objects

Non-Max Suppression

1. Remove all bounding boxes where confidance ≤ 0.5
2. Pick the bounding box with the highest value for confidence and suppress other bounding boxes for identifying the same object.



Object Detection: Lots of variables ...

Backbone Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception
ResNet
MobileNet

"Meta-Architecture"
Two-stage: Faster R-CNN
Single-stage: YOLO / SSD
Hybrid: R-FCN

Image Size
 # Region Proposals
...

Takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Bigger / Deeper backbones work better

Huang et al, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR 2017
Zou et al, "Object Detection in 20 Years: A Survey", arXiv 2019 (today!)

R-FCN: Dai et al, "R-FCN: Object Detection via Region-based Fully Convolutional Networks", NIPS 2016
Inception-V2: Ioffe and Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", ICML 2015
Inception V3: Szegedy et al, "Rethinking the Inception Architecture for Computer Vision", arXiv 2016
Inception ResNet: Szegedy et al, "Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning", arXiv 2016
MobileNet: Howard et al, "Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv 2017

Open Source Frameworks

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)
- Finetune on your own dataset with pre-trained models

Further Readings

- Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 1 https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04

- Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 2 https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-part-2-b0b9e67762b1

Credit for

CS131 "Computer Vision: Foundations and Applications" by University of Stanford (Fall 2019)

CS231n "Convolutional Neural Networks for Visual Recognition" by University of Stanford

(Lecture 11)