Deep learning- computer vision

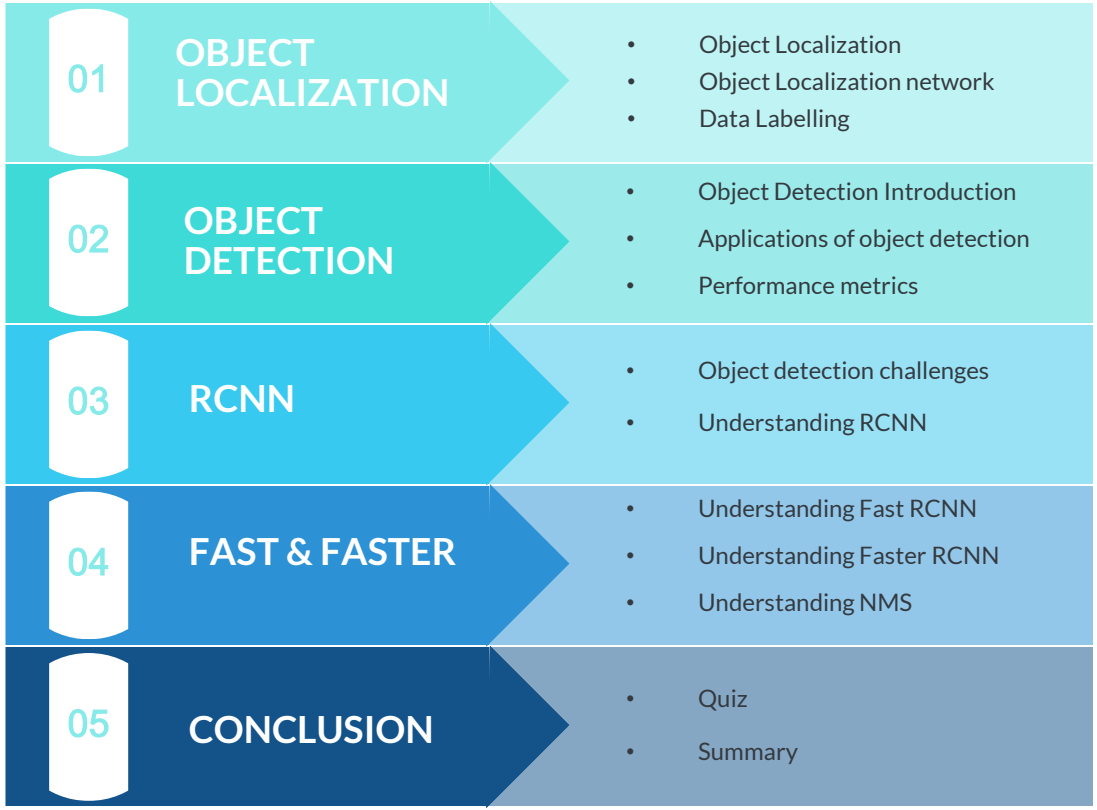
Object localization & object detection

Learning Objectives

At the end of this module, you will be able to:

- Understand the concept of object localization.
- Learn about data labelling, IoU and mAP.
- Understand various concepts like object detection, RCNN, fast RCNN, faster RCNN and its implementation.

Agenda


01
OBJECT LOCALIZATION
Object Localization
Object Localization network
Data Labelling
02
OBJECT DETECTION
Object Detection Introduction
Applications of object detection
Performance metrics
03
RCNN
Object detection challenges
Understanding RCNN
04
FAST & FASTER
Understanding Fast RCNN
Understanding Faster RCNN
Understanding NMS
05
CONCLUSION
Quiz
Summary

01

OBJECT LOCALIZATION

- Object Localization
- Object Localization network
- Data Labelling

Object localization

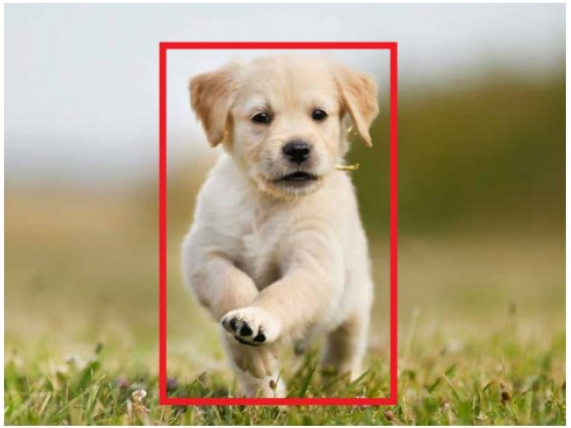
- Single object in the image
- Involves both classification and regression
- Use labellmg to build training data
- Use ImgAug for augmentation

What is object localization?

- Object localization predicts the object in an image as well as its boundaries.
- Performed by drawing a bounding box around the location of the object in image.

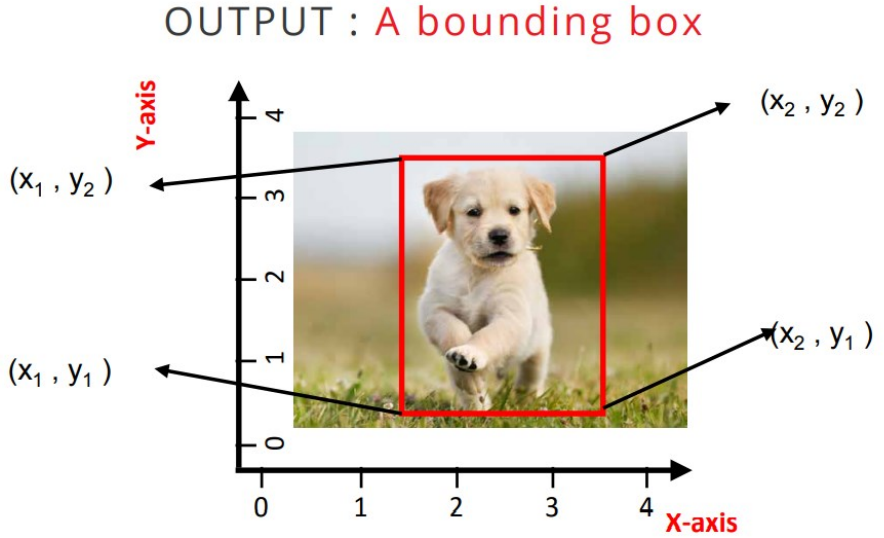
INPUT : An Image

OUTPUT : A bounding box



How to find bounding box for object localization?

- Object bounding boxes are found using x-axis and y-axis coordinates based on the size of the image
- Bounding box helps to separate most prominent object from background



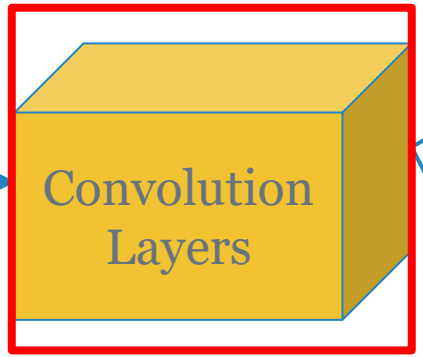
Object Localization

Object localization using transfer learning

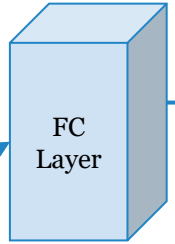
- Localization means entailing an object to a bounding box or precisely to a rectangle for localizing the concerned object after identifying it.
- There are single-object localization, multiple object localization and semantic segmentation for doing things of similar means with different forms of doing the purpose.

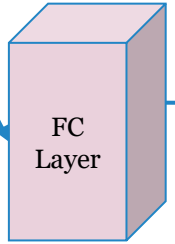
- Pre-trained or Transfer Learning models have gained huge popularity for its implementation in Object localization-based applications. Here are some of the very popularly used pre-trained object detection models
- R-CNN
- Resnet50
- FPN
- Retinanet
- Yolo V3/V2
- Faster R-CNN
- SSD

Object localization network



Transfer
Learning





Classification

Class Probability

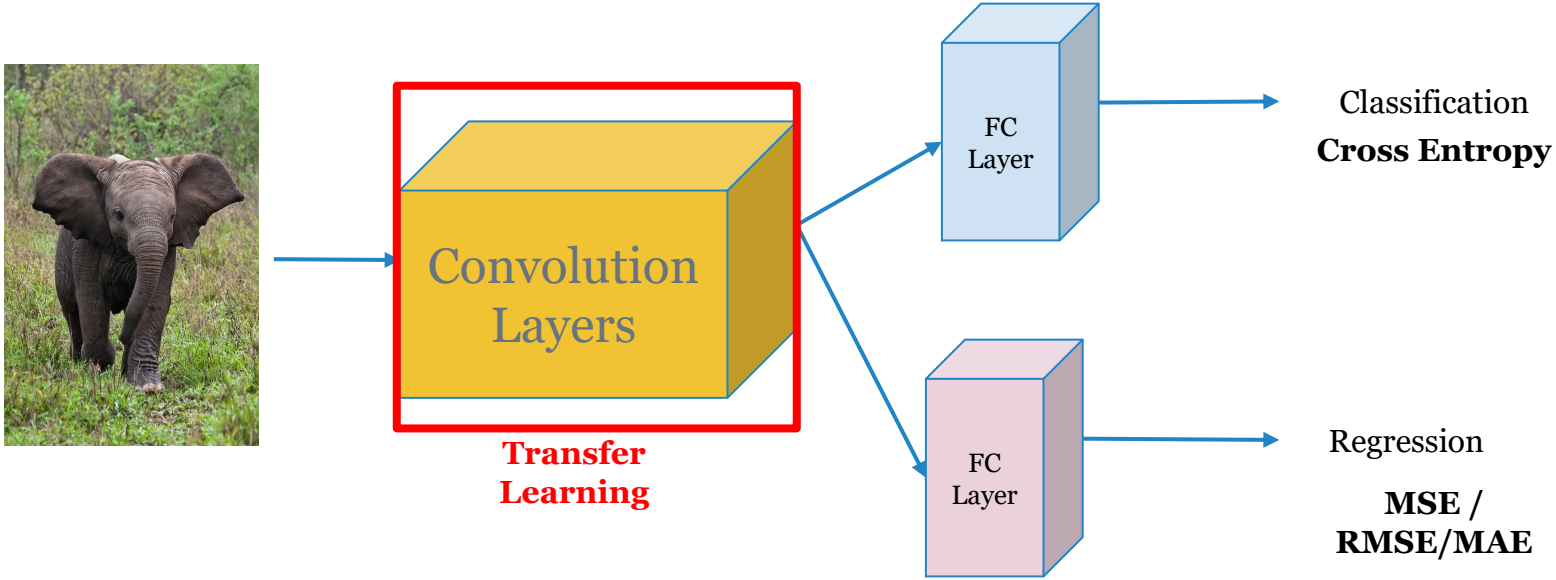
Box Coordinates

(x, y, W, h)

Regression

Localization requires both Classification and Regression in the same model

Loss calculation for object localization



Loss in Object Localization

Data labelling

- There are various labeling approaches. depending on the problem statement, the time frame of the project, and the number of people who are associated with the work.
- While internal labeling and crowdsourcing are very common, the terminology can also extend to include novel forms of labeling and annotation that make use of AI and active learning for the task.

The most common approaches for annotation of data are listed below.

- In-house data labelling
- Crowdsourcing
- Outsourcing
- Machine-based annotation

Common types of data labeling

From what we have seen till now, data labeling is all about the task we want a machine-learning algorithm to perform with our data.

For example

If we want a machine learning algorithm for the task of defect inspection, we feed it data such as images of rust or cracks. The corresponding annotation would be polygons for localization of those cracks or corrosion, and tags for naming them.

2 types are:

- Preparation of data with bounding boxes using tool labelling.
- Labelling Images with Labelling.

Dataset for object localization

PASCAL VOC

http://host.robots.ox.ac.uk/pascal/VOC/

COCO
(Common objects in Context)

http://cocodataset.org

IIIT Oxford Pets Dataset

http://www.robots.ox.ac.uk/~vgg/data/pets/

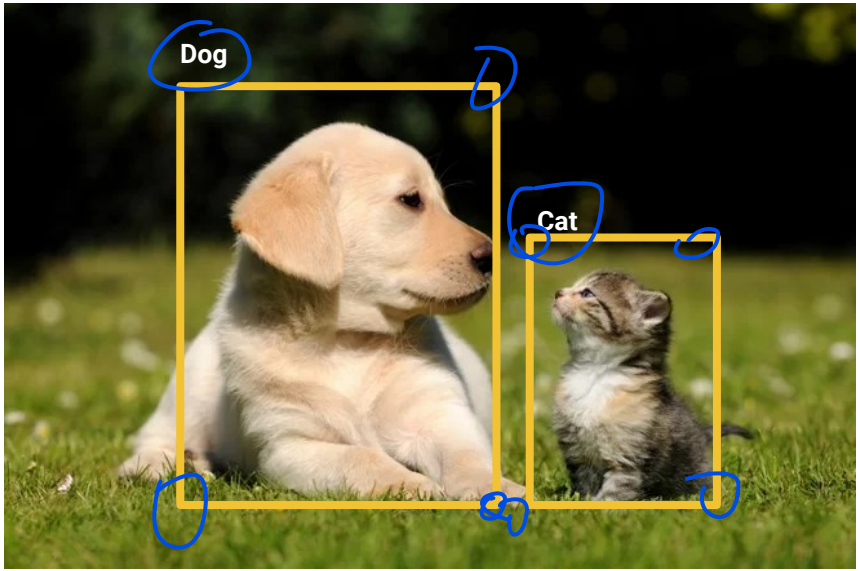
Some examples of datasets for object localization

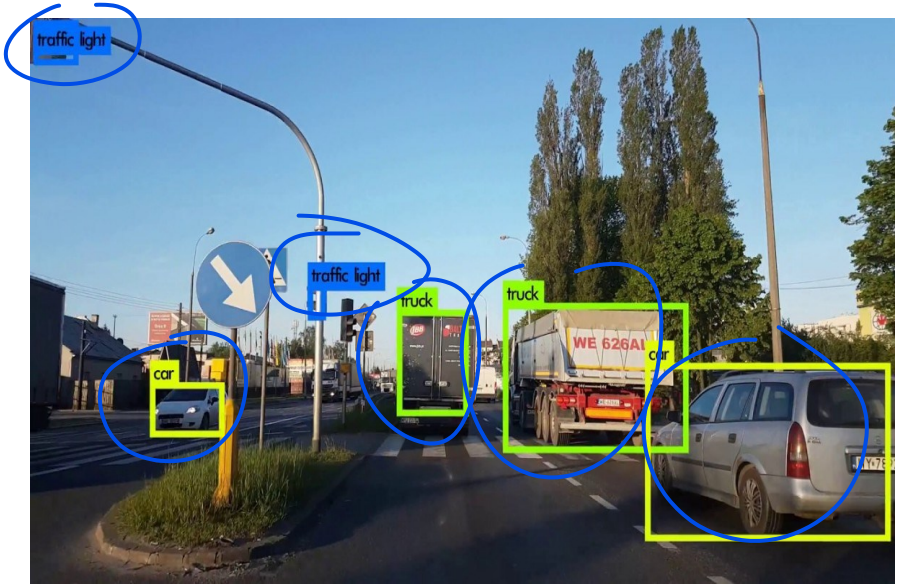
02

OBJECT DETECTION

- Object Detection Introduction
- Applications of object detection
- Performance metrics

Object detection



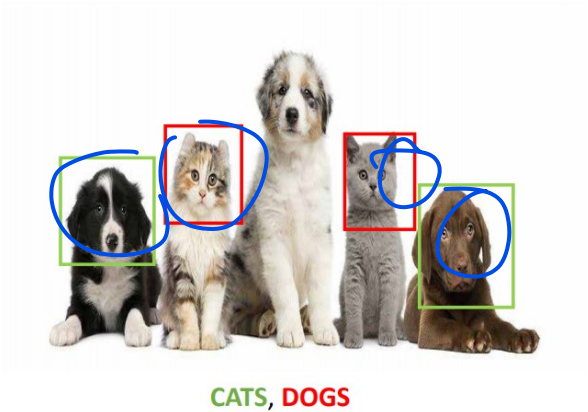


Can involve a very complex scenery

Object detection

Input : An Image

Output : Class labels + *Bounding boxes for all the objects*



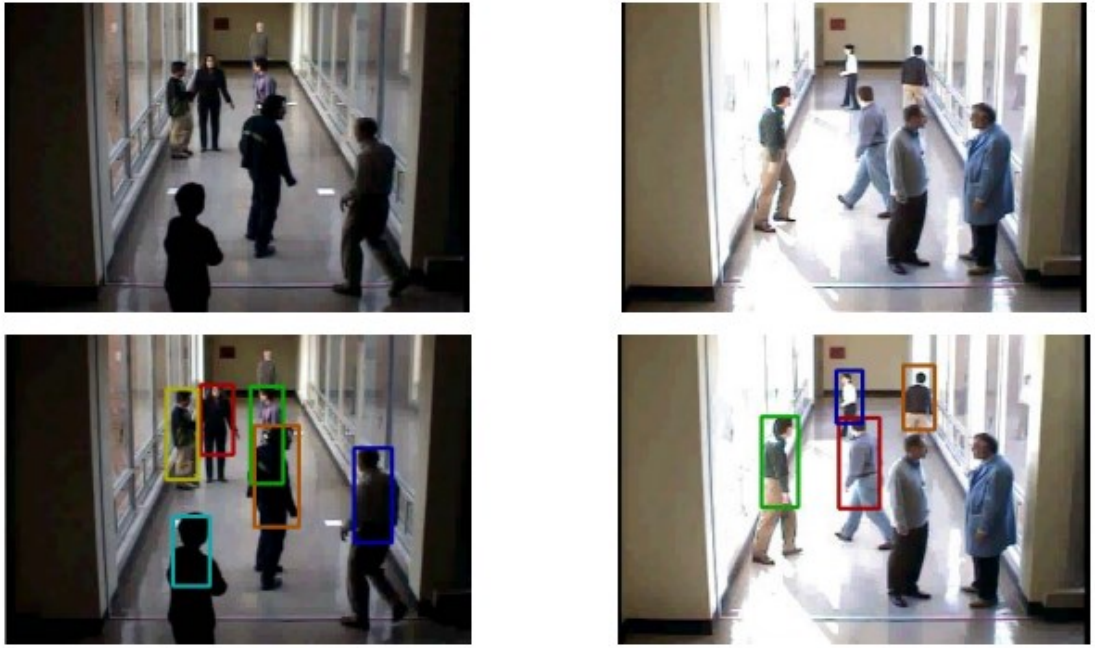
CATS, DOGS

Applications of object detection

Optical Character Recognition (OCR)

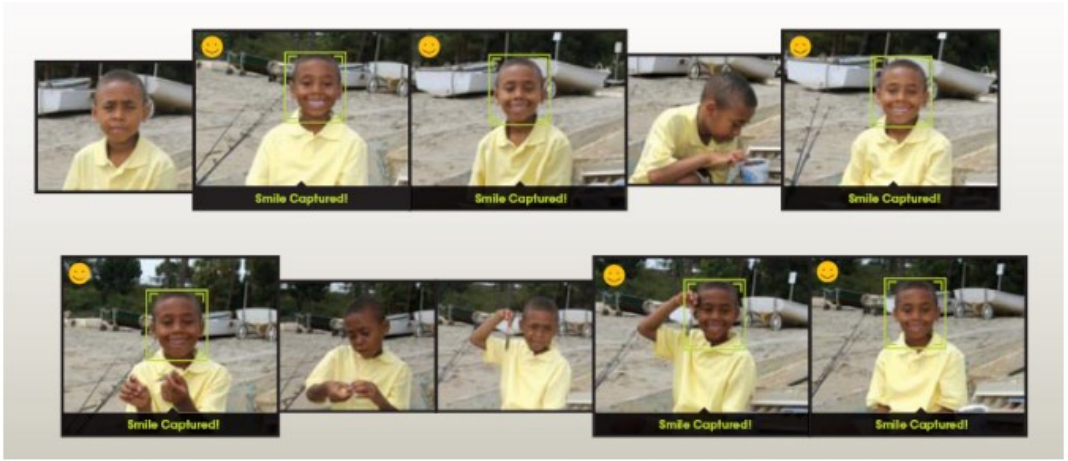
Applications of object detection

Surveillance and Security



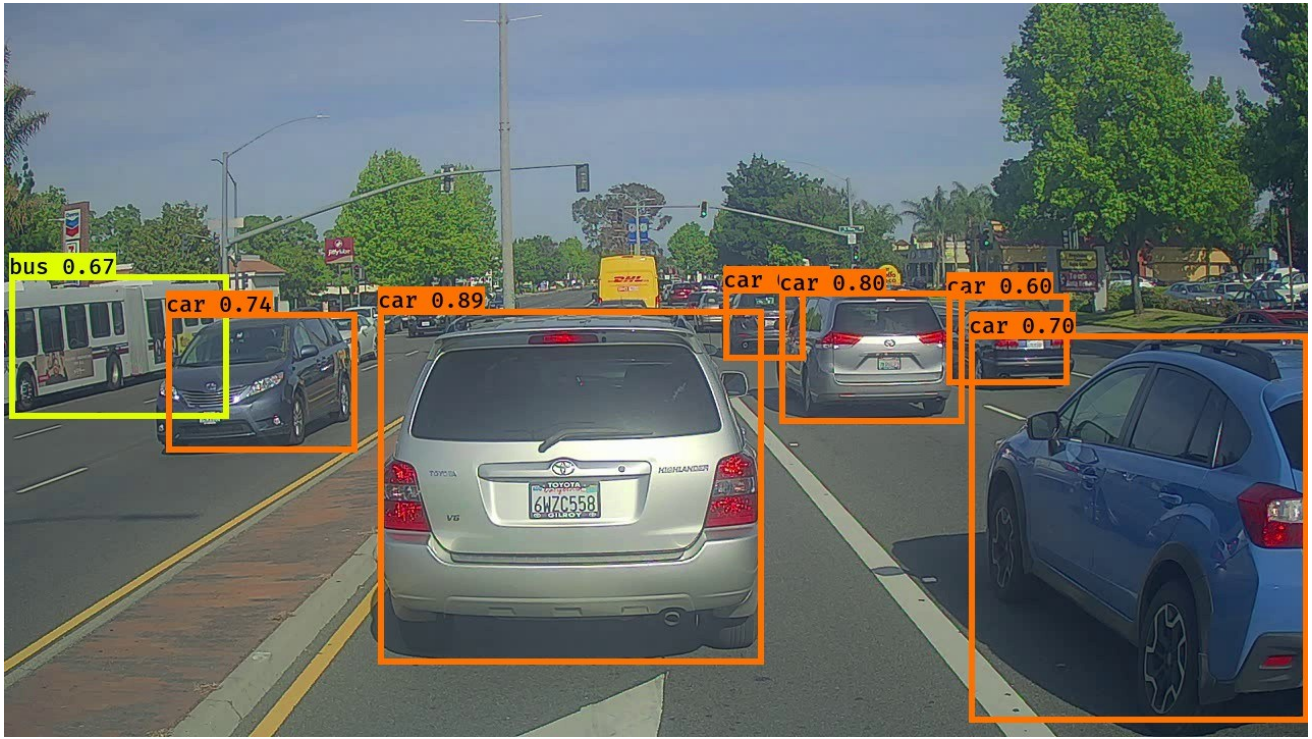
Applications of object detection

Pedestrian Tracking



Emotion Detection

Applications of object detection

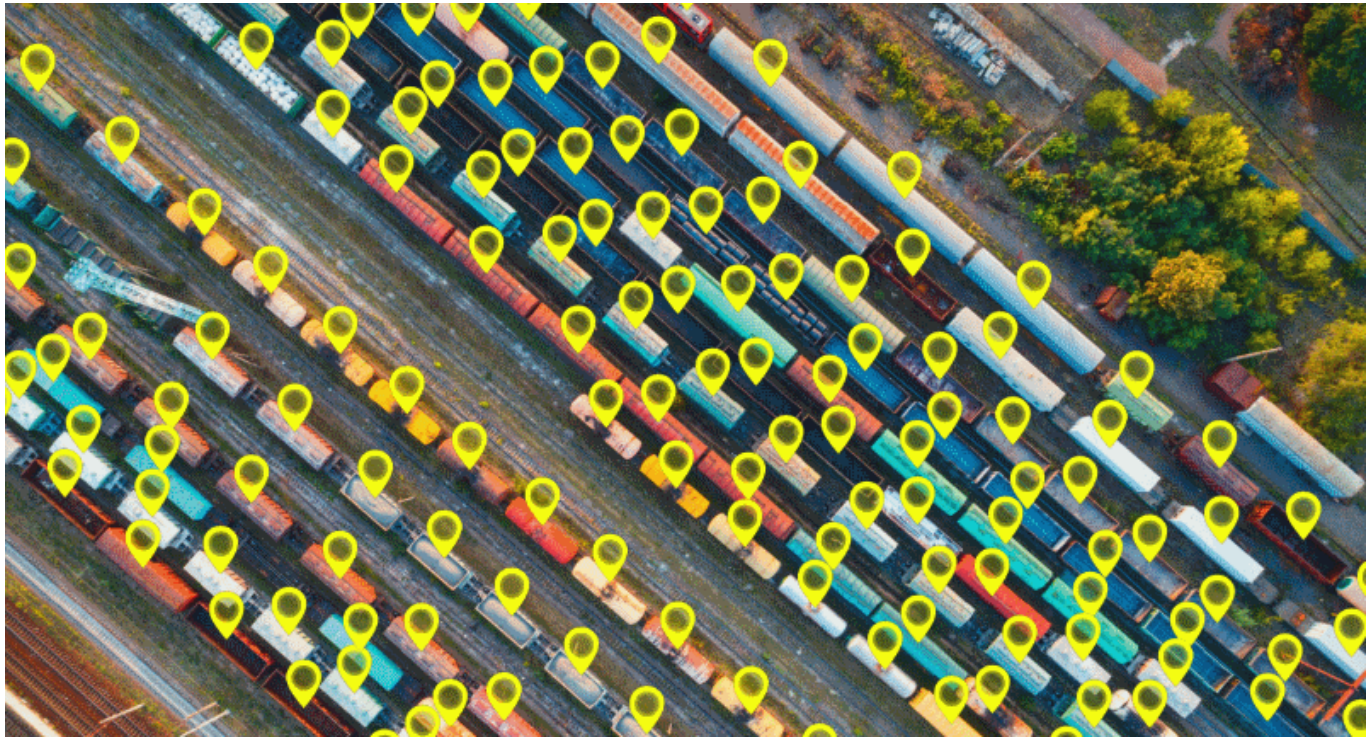


Autonomous Driving

Applications of object detection

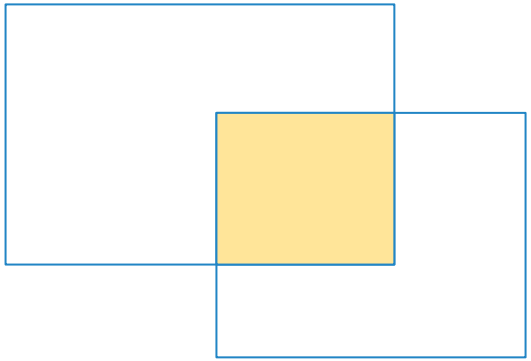
Counting Objects

Applications of object detection



Container Tracking

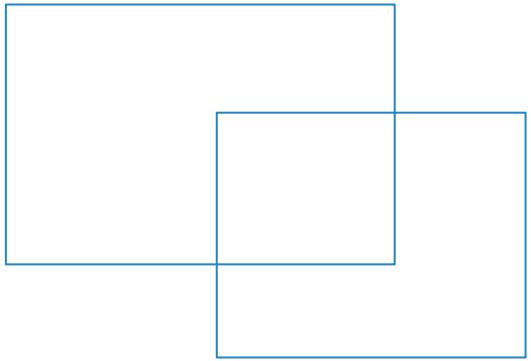
IoU



Intersection over Union (IOU)

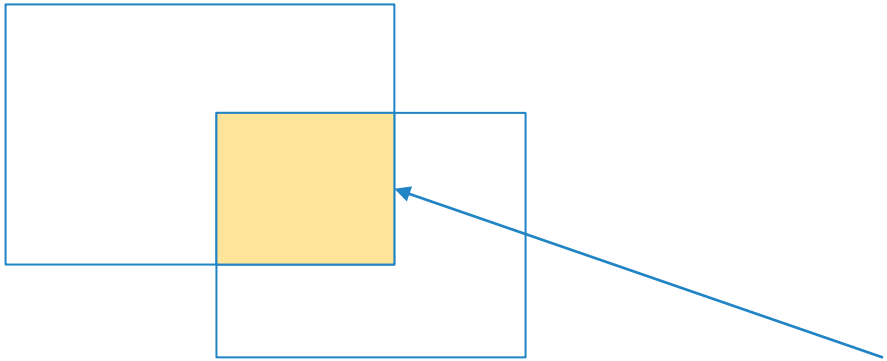
An approach to measure localization accuracy

IoU



Actual bounding box in the data

IoU

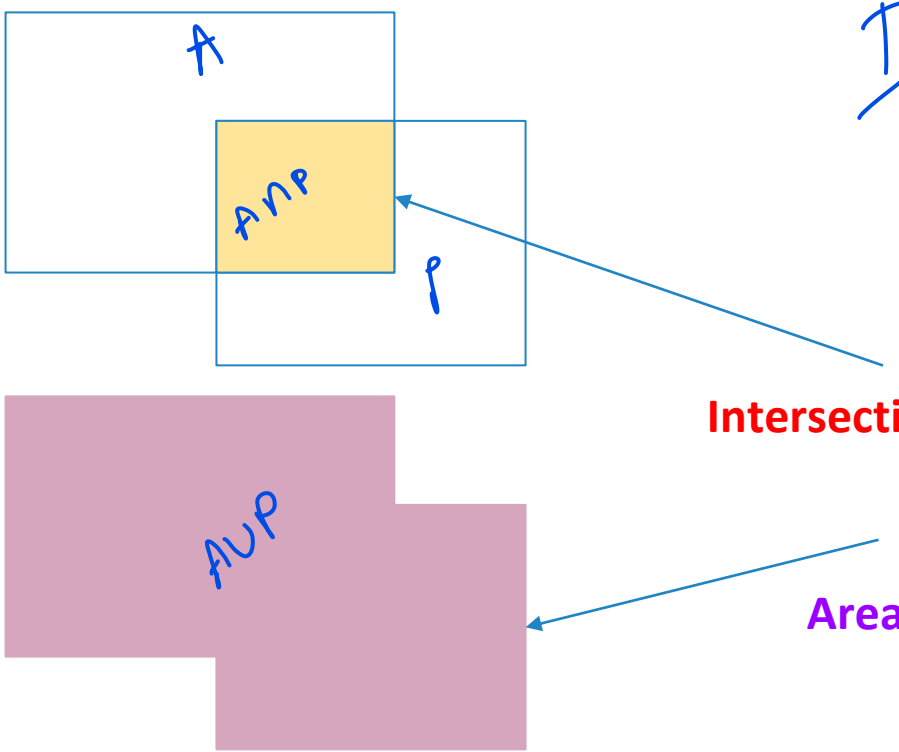


Intersection area

Area common between both anchor box and actual box

IoU

$$IoU = \frac{A \cap P}{A \cup P}$$

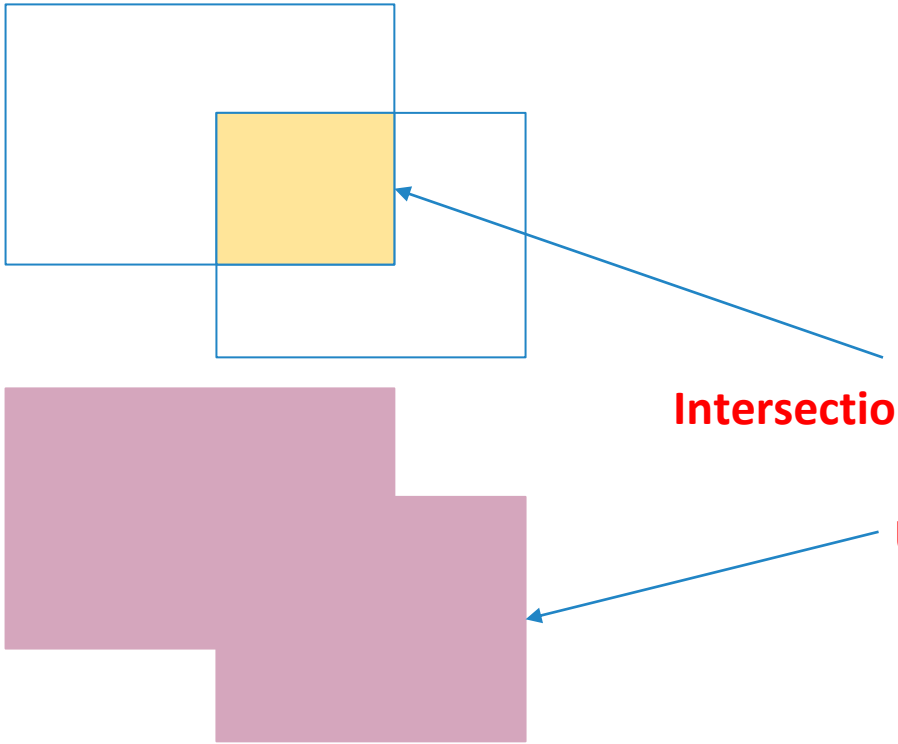


Intersection area

Union area

Area covered by either box

IoU

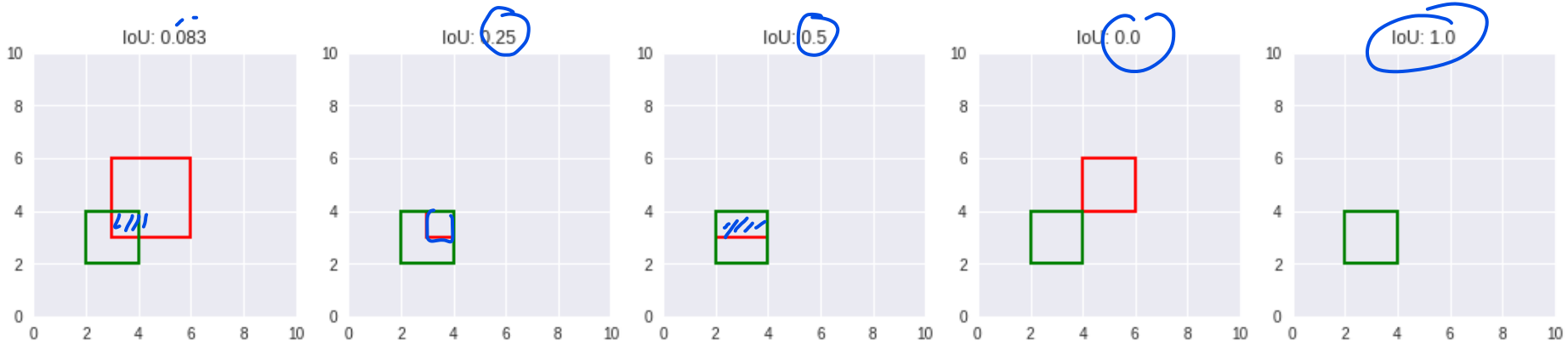


IOU = a / b

Intersection area (a)

Union area (b)

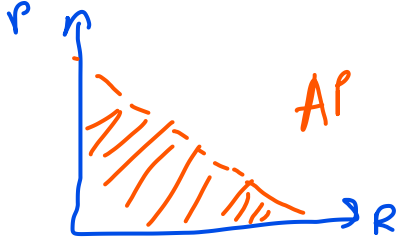
IoU

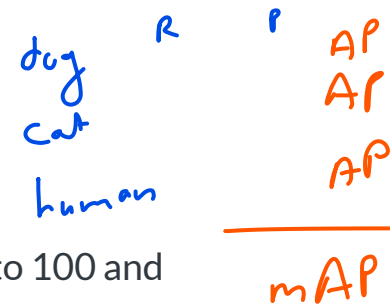


IoU provides a simple way to understand overlapping between two boxes

mAP

mean Average Precision





The evaluation metric used in object recognition tasks is ‘mAP’. It is a number from 0 to 100 and higher values are typically better.

Each bounding box will have a score associated (likelihood of the box containing an object).

Based on the predictions, a precision-recall curve (PR curve) is computed for each class by varying the score threshold.

The average precision (AP) is the area under the PR curve. First the AP is computed for each class, and then averaged over the different classes. The end result is the mAP.

mAP

The mAP of the object detection model is calculated according to the below equation.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

$$AP_k = \text{the AP of class } k$$

$$n = \text{the number of classes}$$

Variations Among mAP

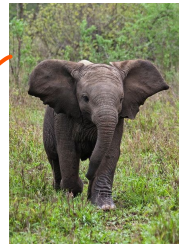
- mAP iou=0.5 represents the model has used 0.5 threshold value to remove unnecessary bounding boxes, it is the standard threshold value for most of the models.
- mAP iou=0.75 represents the model has used 0.75 threshold value, By using this we can get accurate results by removing bounding boxes with less than 25% of the intersection with ground truth image.
- mAP small represents the model has given mAP score based on smaller objects in the data.
- mAP large represents the model has given mAP score based on larger objects in the data.

03

RCNN

- Object detection challenges
- Understanding RCNN

What should the model predict for Object detection?



CNN

Elephant: (x, y, w, h)

CNN

Dog: (x, y, w, h)
Cat : (x, y, w, h)

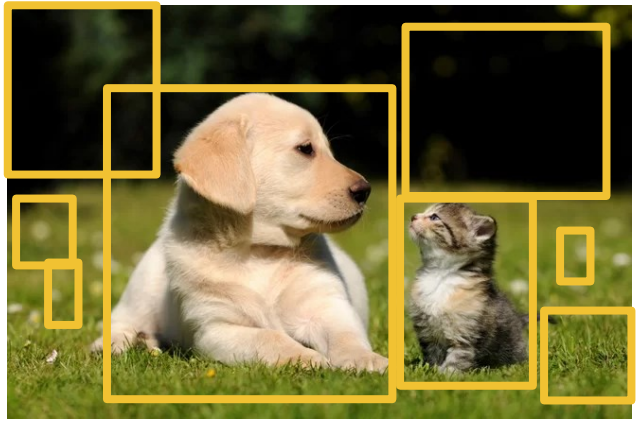
CNN

Person 1: (x, y, w, h)
Person 2 : (x, y, w, h)
. . . .
Bike 1: (x, y, w, h)
. . . .
Car 1: (x, y, w, h)

Each image may need different number of outputs

How do we build an object detector

Break down original picture into multiple pictures

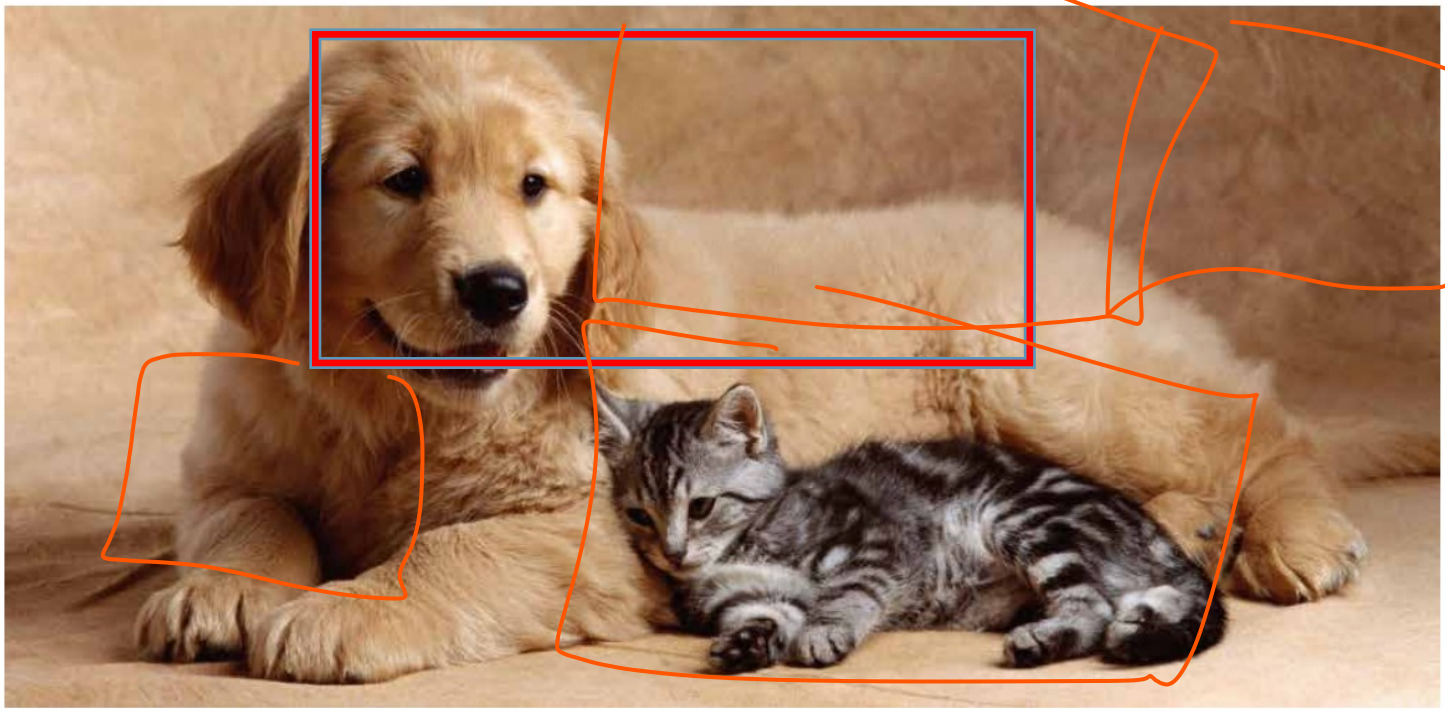


NAIVE WAY - SLIDING WINDOW FOR OBJECT CLASSIFICATION

1. First, we take an image as input:

NAIVE WAY - SLIDING WINDOW FOR OBJECT CLASSIFICATION

2. We decide a window size and slide it across an image to obtain corresponding regions.



Naive way - sliding window for object classification

- 3. Pass all these regions (images) to the CNN and classify them into various classes.
- 4. Once we have divided each region into its corresponding class, we can combine all these regions to get the original image with the detected objects

Problem with computational complexity

- A huge number of regions are obtained as the image gets larger.
- Results in large amount of computational time.

Problem with scale: what should be the window size

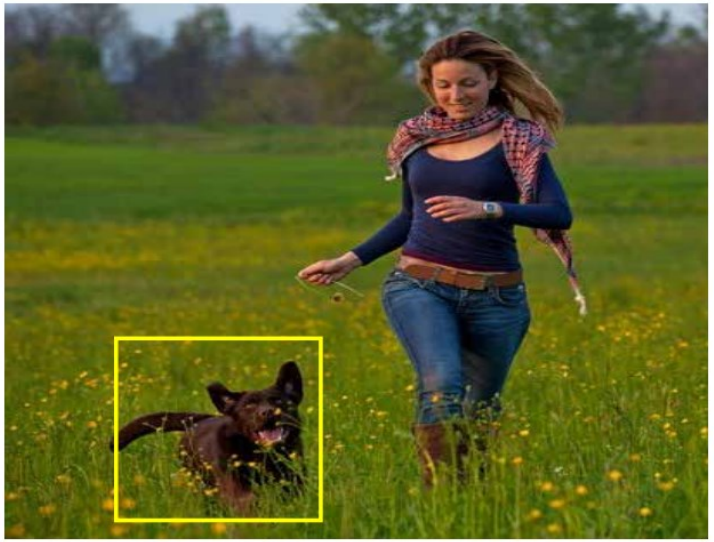
- A fixed window size cannot capture all objects in an image.
- Objects in an image have different scales.
- The bird is detected but the fish is not due to its smaller size.
- All sizes of objects must be captured by the object detector.

Problem with scale: what should be the window size

- Objects in an image can have different aspect ratios and spatial locations.
- Here the person is tall and thin whereas the horse is broad.

Problem with object coverage in image

- Some cases the object might be covering most of the image
- While in others , the object might only be covering a small percentage of the image.



Object detection overview

1. Understand Object Localization - detect single object in an image

2. Object detection involves detecting multiple objects in an image.

3. In Object detection, we break down the original image into multiple images and perform Object Localization on each part.

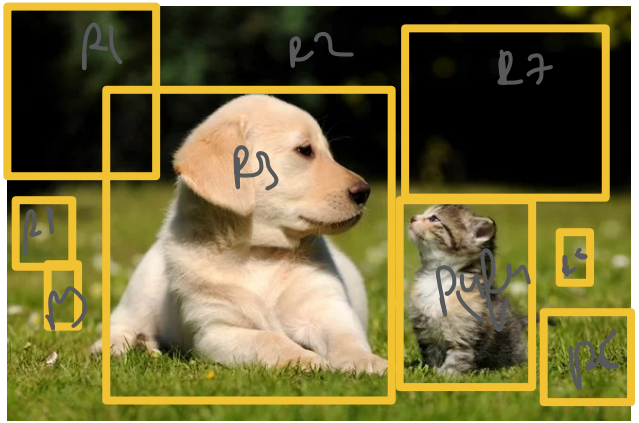
4. Extracted image parts are called RoI / Anchor Boxes/ Priors

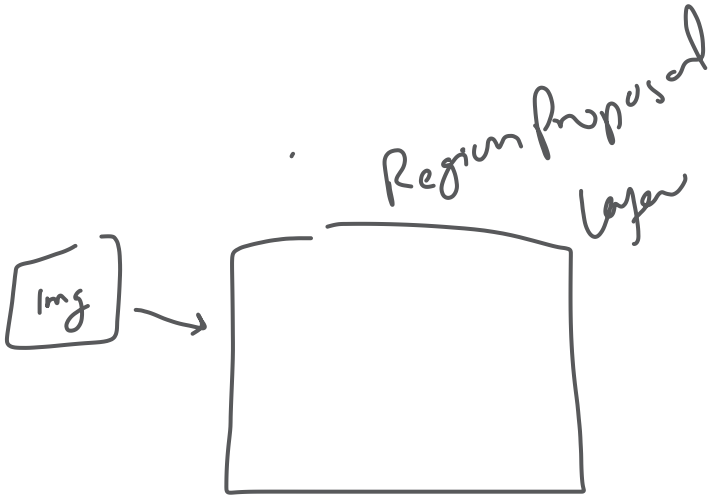
- R-CNN uses ‘Selective Search’ to extract RoIs
- Fast-RCNN also uses Selective Search but extract RoIs from Feature Map
- Faster R-CNN uses Region Proposal Network (RPN) to extract RoIs
- In SSD, we (humans) provide the RoIs using multiple grids and anchor boxes (or priors)

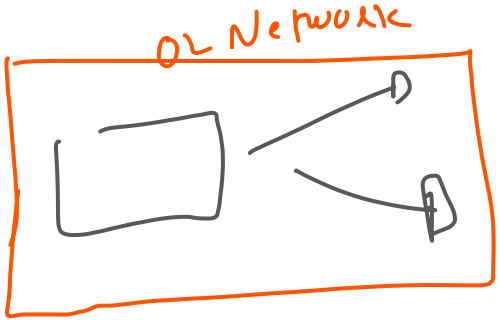
5. Allow CNN to look at different RoIs to see if there is an object in the RoI and what is their bounding box.

Region based CNN (RCNN)

- Used Selective Search to identify RoIs in an image
- Created 2000 RoIs from each picture
- Region based CNN or RCNN proposes a set of regions as bounding boxes and sees if any of them actually correspond to an object.







WHO PROPOSED R-CNN

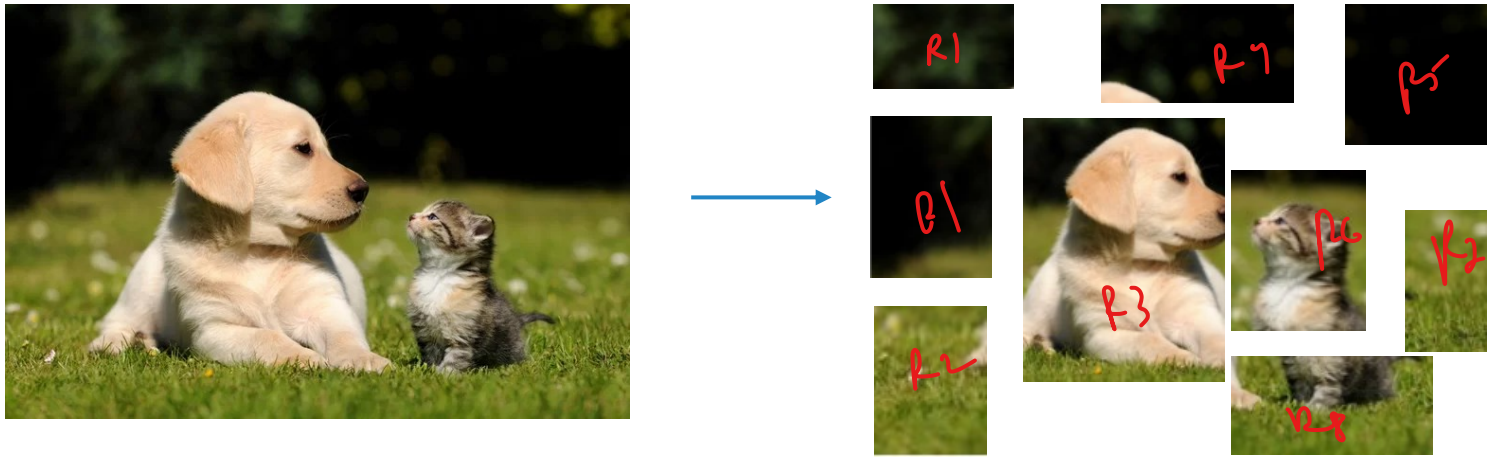
Inspired by the research of Hinton's lab at the University of Toronto, a small team at UC Berkeley, led by

- Professor Jitendra Malik, comprised of
- Ross Girshick,
- Jeff Donahue, and
- Trevor Darrel

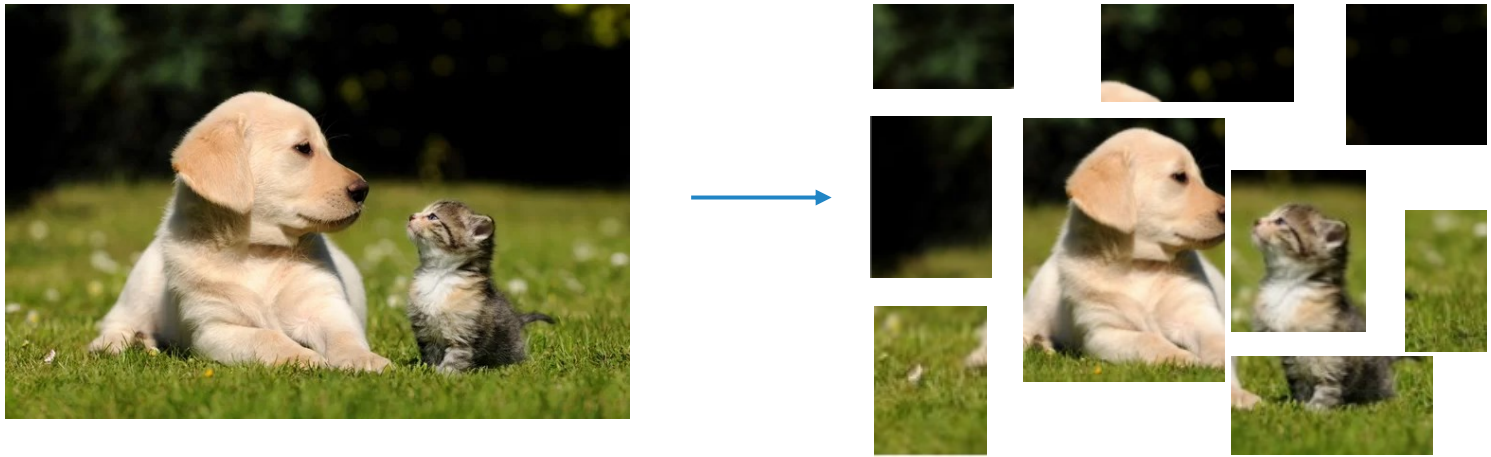
proposed R-CNN by testing on the PASCAL VOC challenge.

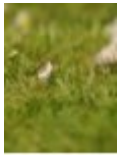
RoI
Region Interest

From one original image, we will create lots of image's e.g. 1000 images



Assume each new image has exactly one object in it





Model

Model

Model

Dog: (x, y, w, h)

Cat: (x, y, w, h)

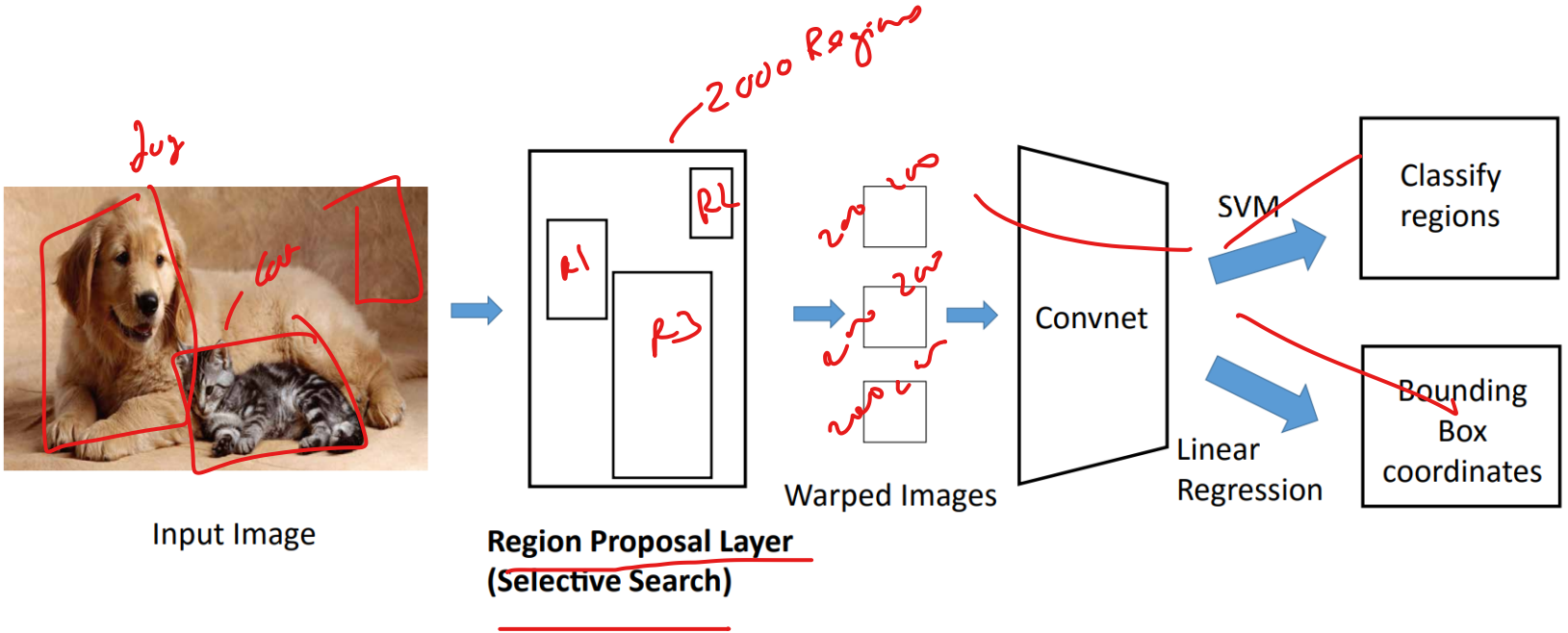
Background: (x, y, w, h)

Perform Object Localization for each new image

Each new image is also called Region of Interest (ROI)

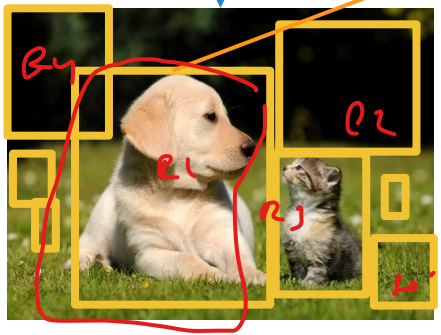
Majority of new images will have Background class object in them

Architecture of r-cnn



RCNN

Apply Selective Search



Take out one region

CNN

FC Softmax

Dog

Cat

Background

FC

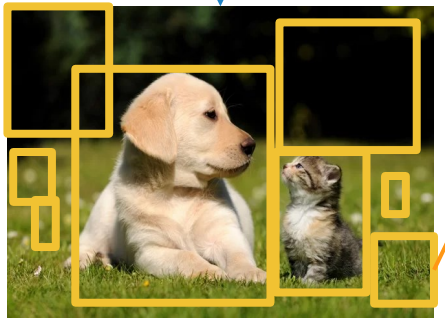
x, y, w, h

Classification

Regression

RCNN

Apply Selective Search



Take out another region

CNN

FC Softmax

Dog

Cat

Background

FC

x, y, w, h

Classification

Regression

How does r-cnn work?

1. An image is input to the network.
2. Generates 2000 regions of interest using Selective search.
3. Convert regions into fixed sized images
4. Each region proposal is passed into a CNN for feature extraction
5. Linear SVM classifier helps classify objects
6. Regression model finds bounding box coordinates

What is selective search and how it identifies different regions?

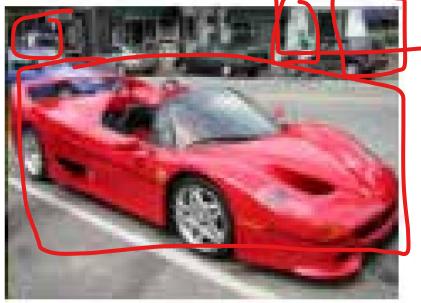
- Selective search identifies patterns in the image and combines similar patterns based on the following characteristics, to propose various regions:

Varying scales

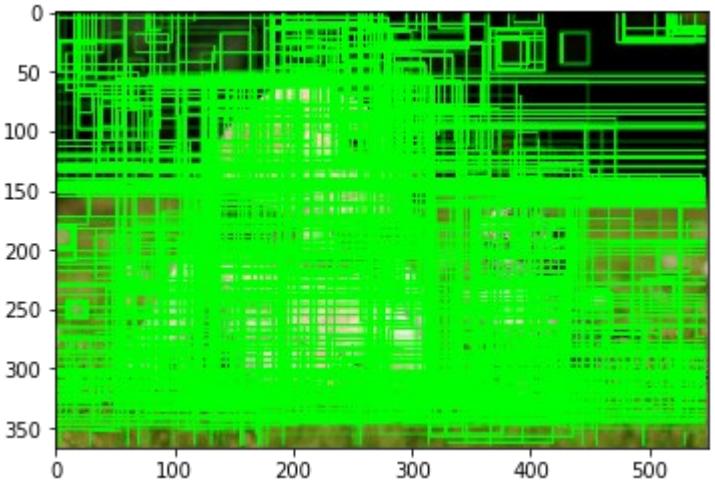
Colour

Texture

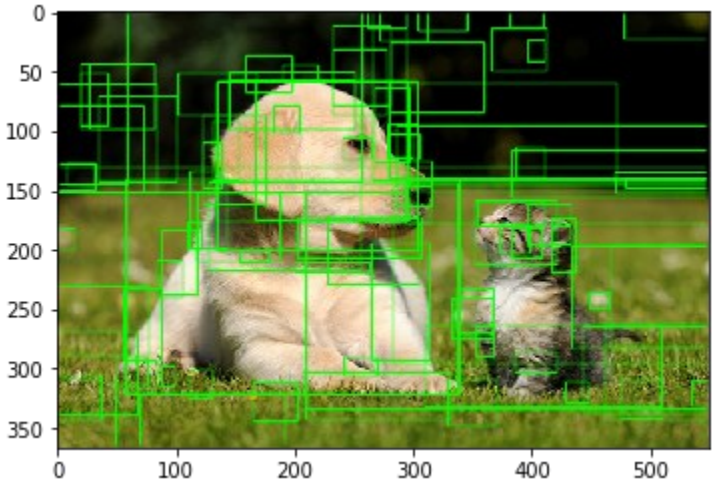
Enclosed objects



Selective search provides 1000s of RoIs for each image



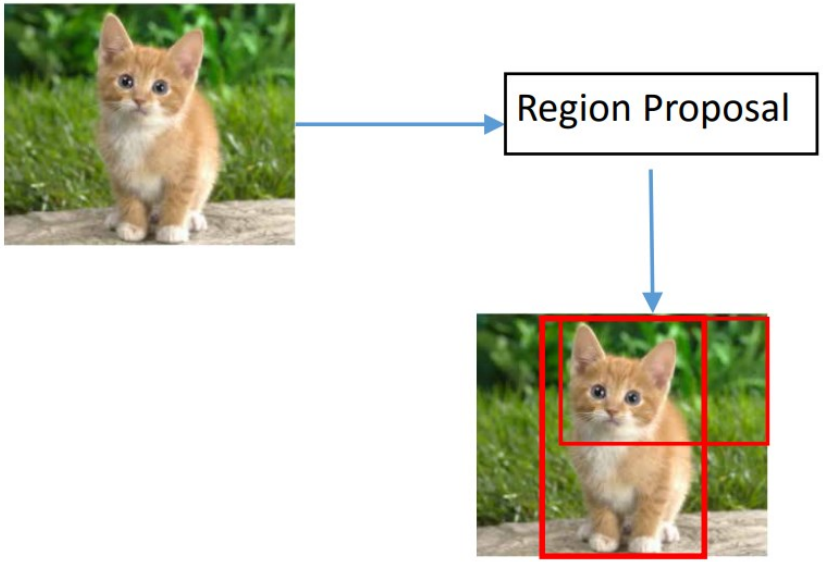
We can limit number of RoI results based on ‘objectiness’ score



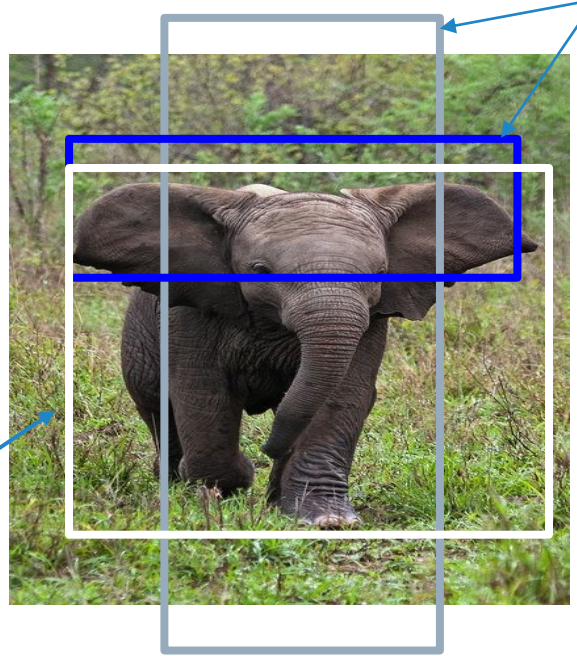
How do we find where these bounding boxes are?

Solution: Region Proposal Method

- RCNN uses selective search to extract boxes from an image.
- It tries to find boxes of different sizes where it feels there might be an object.
- RCNN algorithm proposes a bunch of 2000 boxes called regions in an image.



Preparing label for each ROI



Building Labels for each RoI

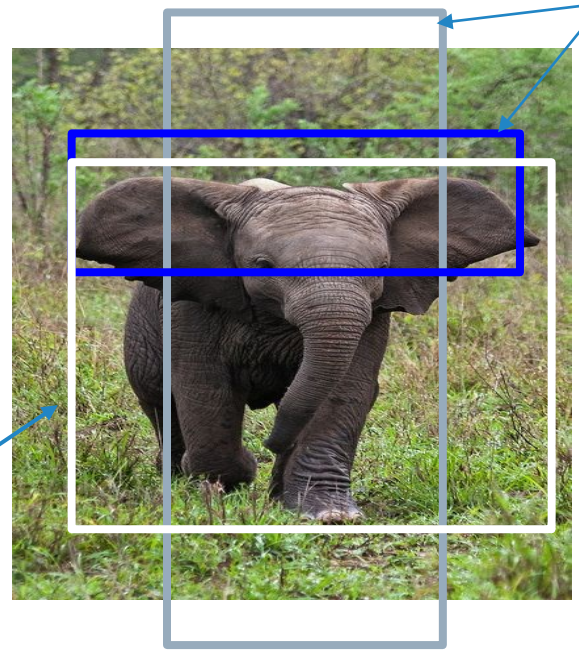
To prepare label for each RoI (region of interest)

1. Calculate overlap (IOU) between RoI and actual object boxes

2. If IOU is above a threshold e.g 0.5
   - Class Label: Actual class
   - Regression Label: Actual Box coordinates

3. If IOU < threshold:
   - Class Label: Background
   - Regression Label: Set to [0,0,0,0], it will be ignored during loss calculation

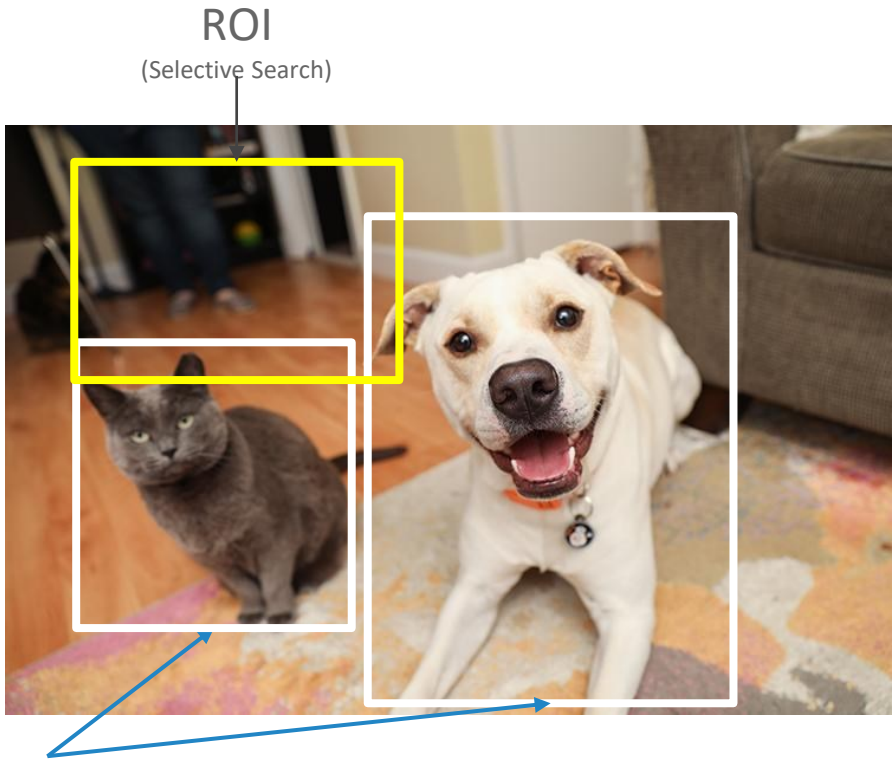
Preparing label for each ROI



Building Labels for each RoI

1. Let's assume we have 4 actual classes -
    - Dog (1), Cat(2), Elephant (3), Horse(4).
    - Background class (0)
2. IOU Calculations
    - IOU for Yellow RoI and Actual Box (White) = 0.6
    - IOU for Blue RoI and Actual Box (White) = 0.1
    - IOU Threshold = 0.5 (we have to decide)
3. Label for Yellow RoI (IOU - 0.6 above threshold)
    - Class (Elephant): [0, 0, 0, 1, 0]
    - Regression Labels: [x,y,w,h] of White actual box
4. Label for Blue RoI (IOU - 0.1 below threshold)
    - Class (Background): [1, 0, 0,0, 0]
    - Regression Labels: [0,0,0,0]

Preparing label for each ROI



1. IOU Calculations

- IOU for Yellow RoI and Actual box of Dog : 0.05

- IOU for Yellow RoI and Actual box of Cat : 0.07

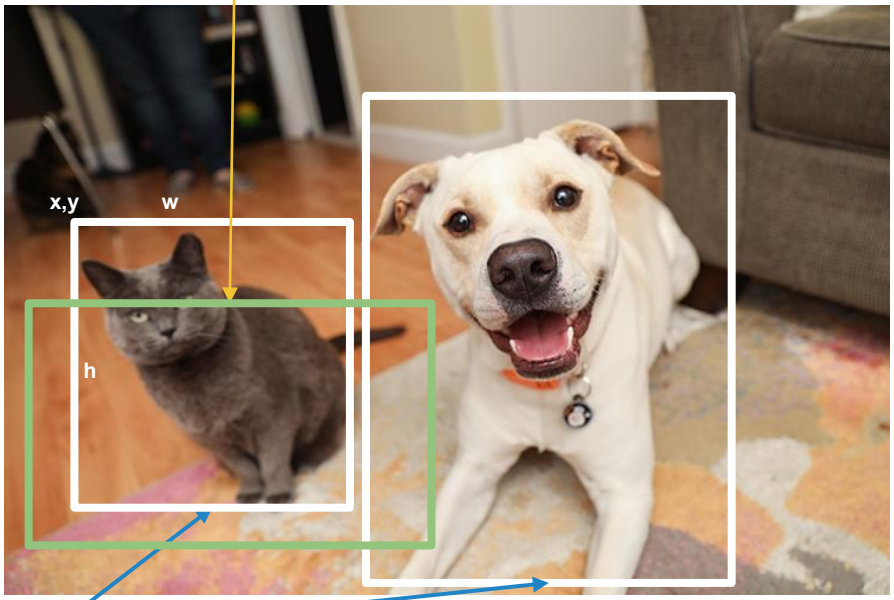
2. Labels for Yellow RoI

Classification (Background): [1, 0, 0, 0, 0]

Regression: [0, 0, 0, 0]

Preparing label for each ROI

ROI
(Selective Search)



Actual Object boxes
(from labelImg)

1. IOU Calculations

- IOU for Green RoI and Actual box of Dog : 0.08

- IOU for Green RoI and Actual box of Cat : 0.62

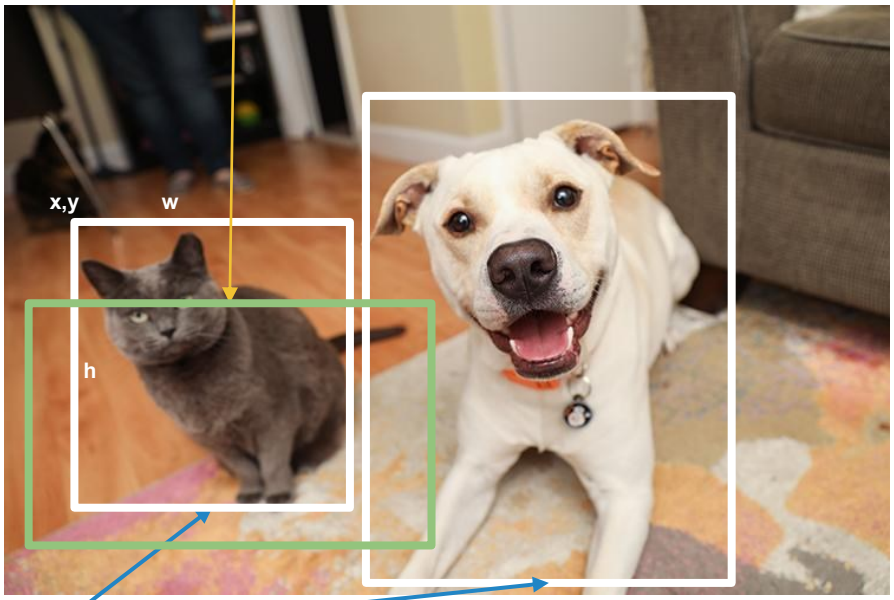
2. Labels for Green RoI

Classification (Cat): [0, 0, 1, 0, 0]

Regression: [x, y, w, h] - actual cat box coordinates

Preparing label for each ROI

ROI
(Selective Search)



Actual Object boxes
(from labelImg)

1. An RoI will have only one object

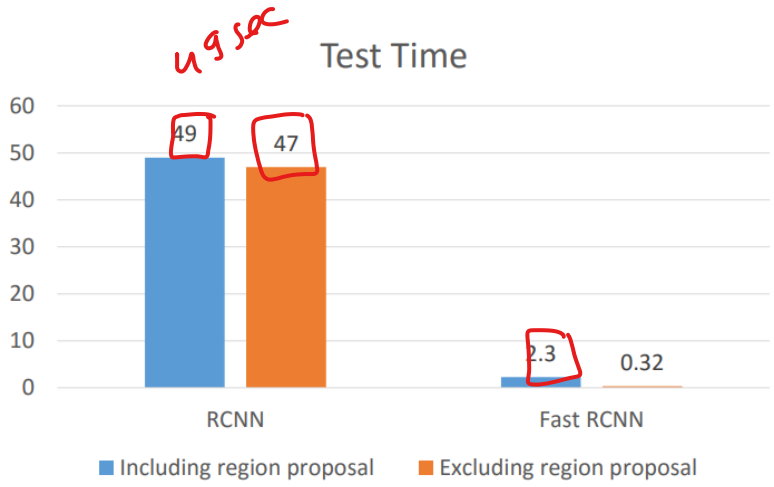
2. If IOU with multiple actual boxes is above threshold, then highest IOU object will be considered.

Limitations of r-cnn

- Training an RCNN model is expensive and slow
- Takes a huge amount of time to train the network as there are 2000 region proposals per image.
- Cannot be implemented real time as it takes around 47 seconds for each test image.

Performance of fast rcnn in test

- Fast RCNN is really fast in prediction.
- The only bottleneck is region proposal which takes almost 2 seconds which is a lot of time.



04

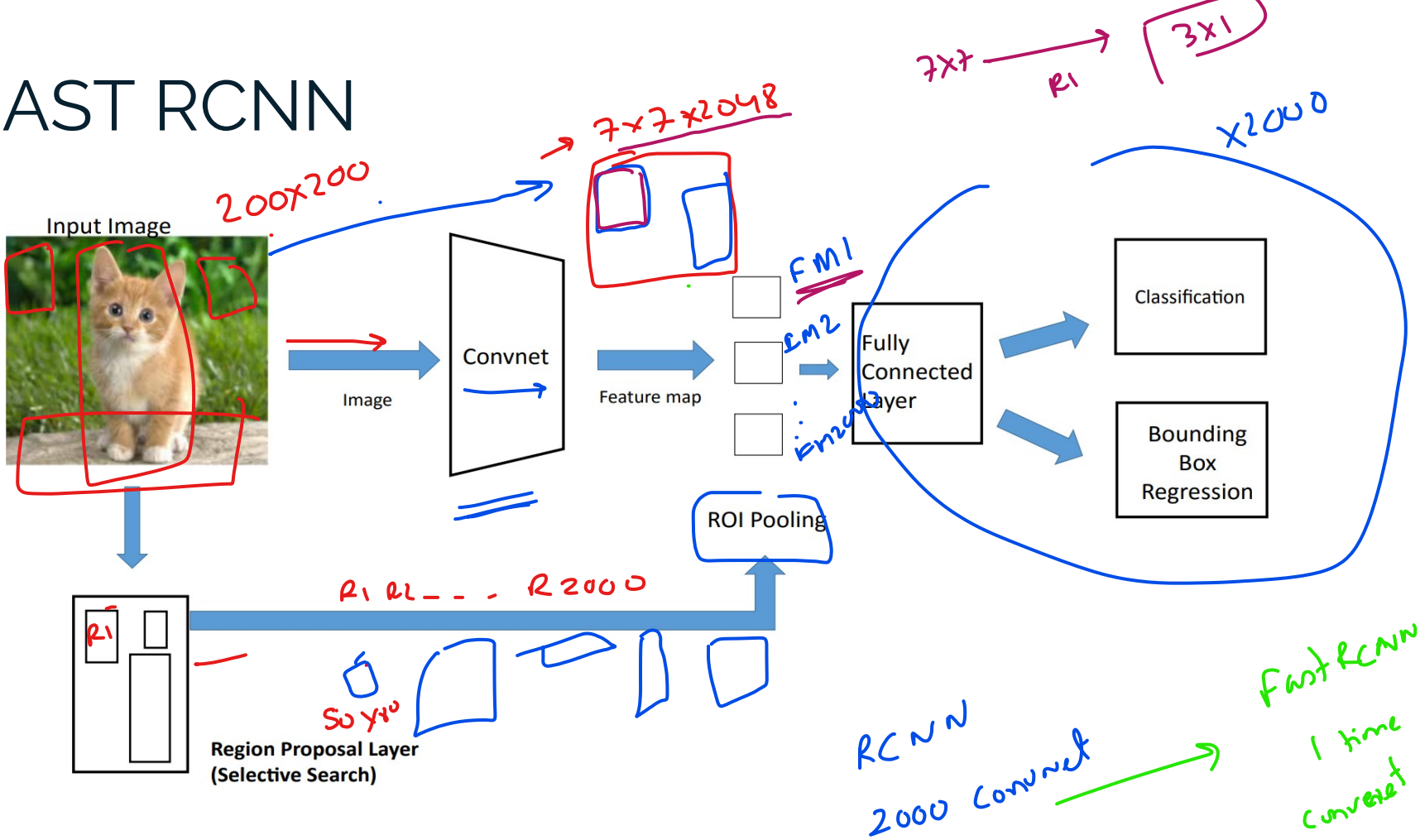
FAST & FASTER

- Understanding Fast RCNN
- Understanding Faster RCNN
- Understanding NMS

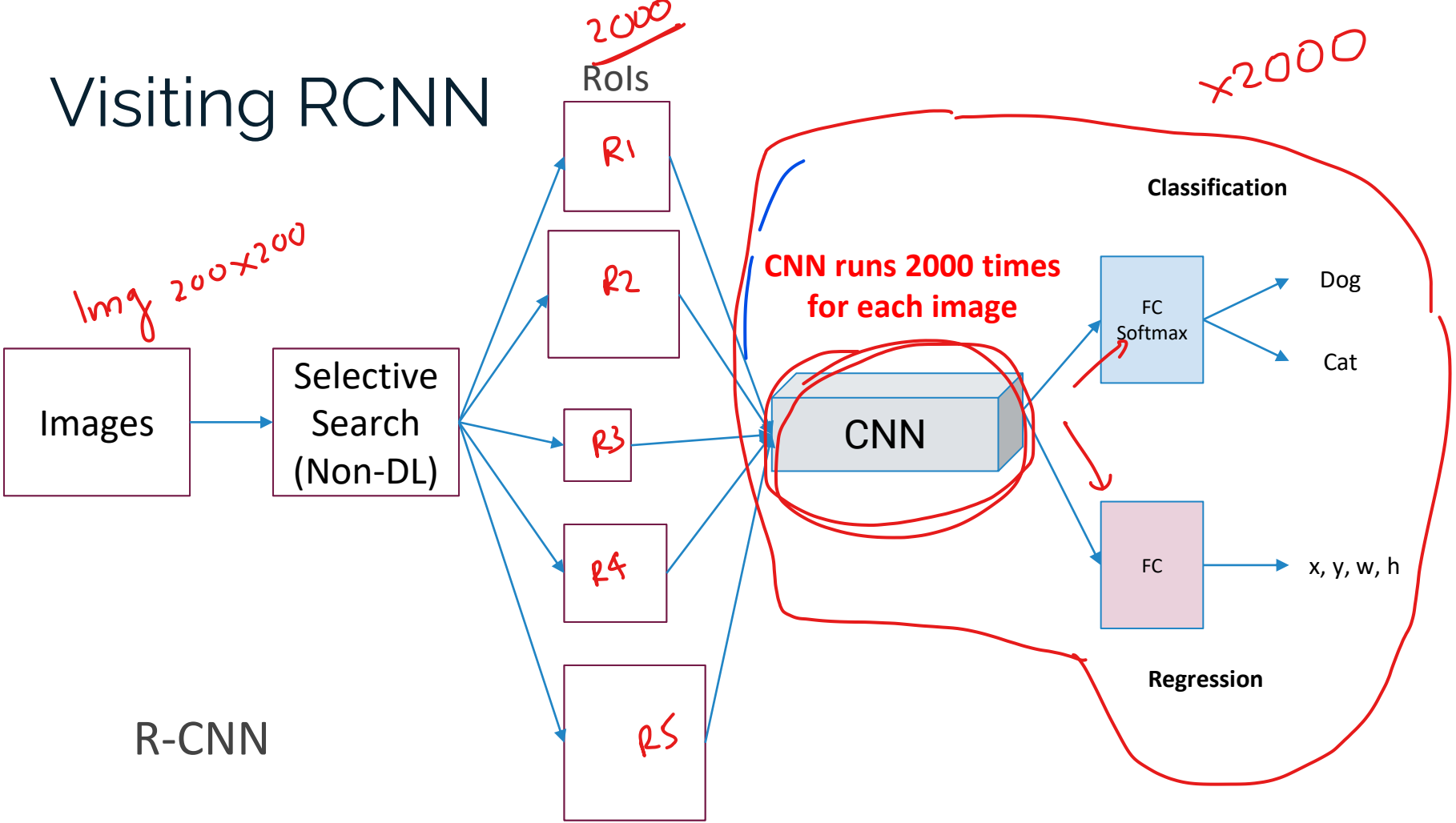
FAST RCNN

- Fast RCNN inputs an image and a group of region proposals. The convolutional network is trained on the whole image instead of individual regions.
- Hence the computation time is reduced from 2000 convolutions to 1 convolution

FAST RCNN

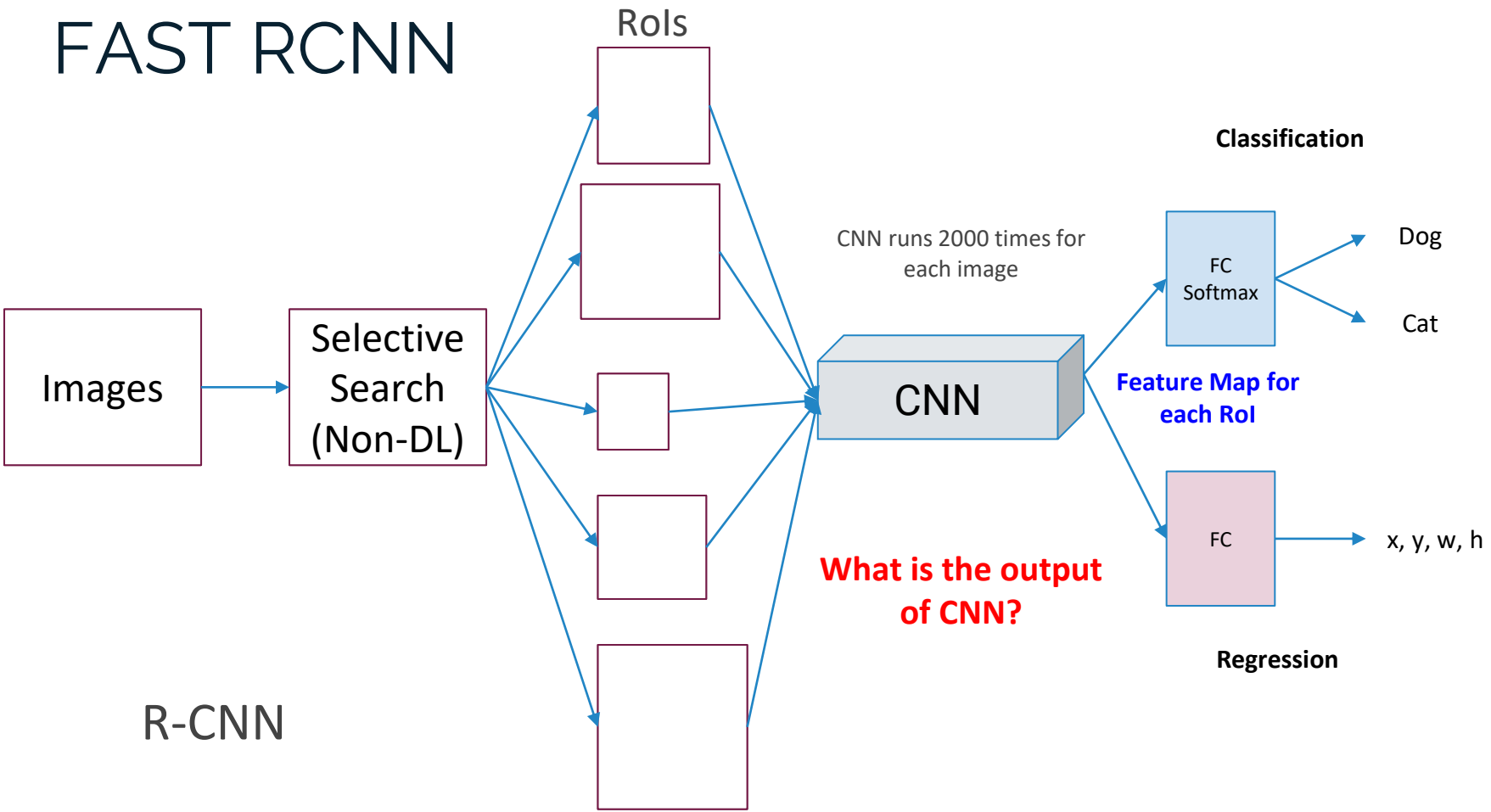

Input Image
200x200
7x7x2048
Convnet
Image
Feature map
FM1
FM2
Fully Connected Layer
Classification
Bounding Box Regression
ROI Pooling
R1 R2 . . . R2000
R1
Region Proposal Layer
(Selective Search)
So you
7x7
R1
3x1
x2000
RCNN
2000 Convnet
Fast RCNN
1 time Convnet

Visiting RCNN



R-CNN

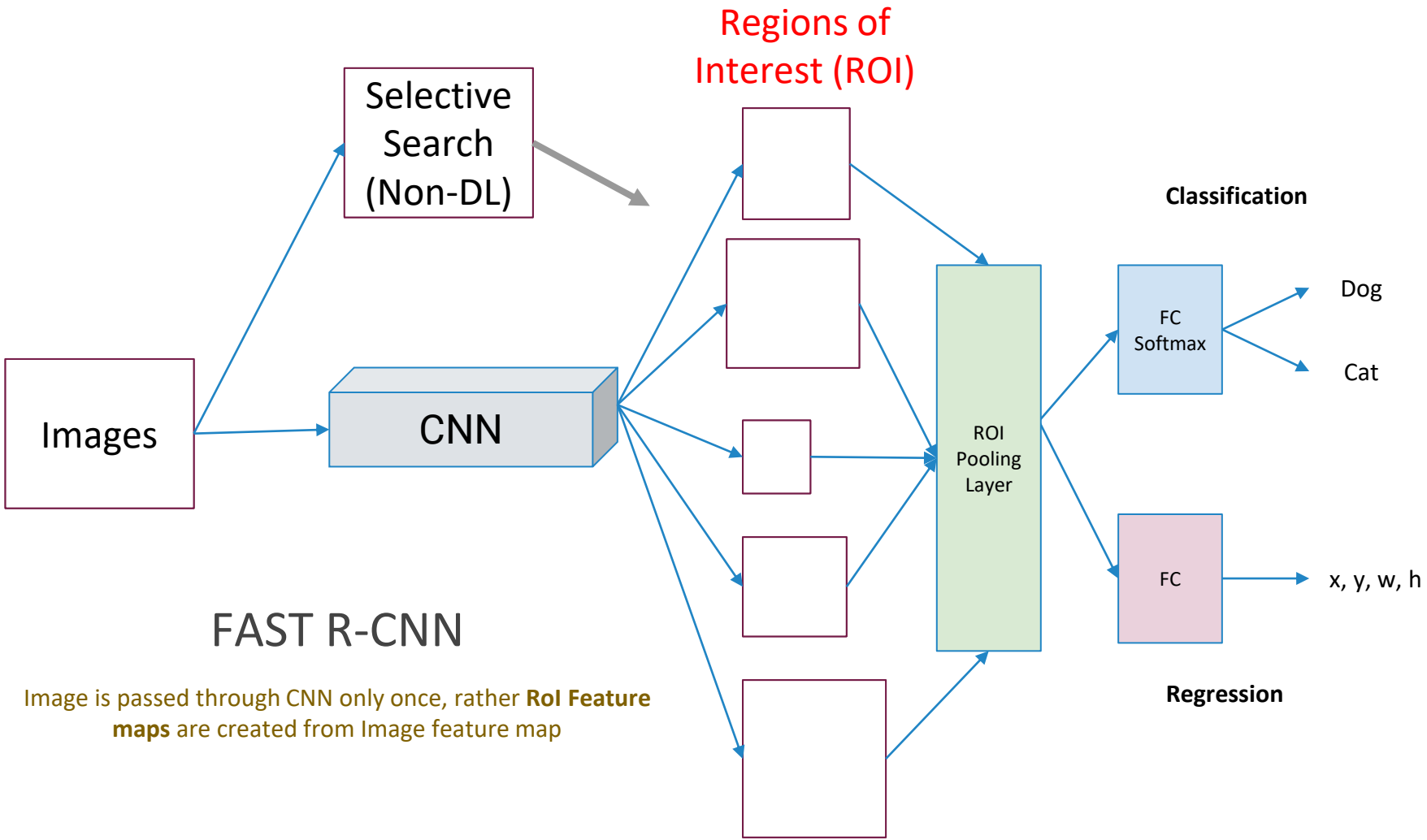
FAST RCNN



R-CNN

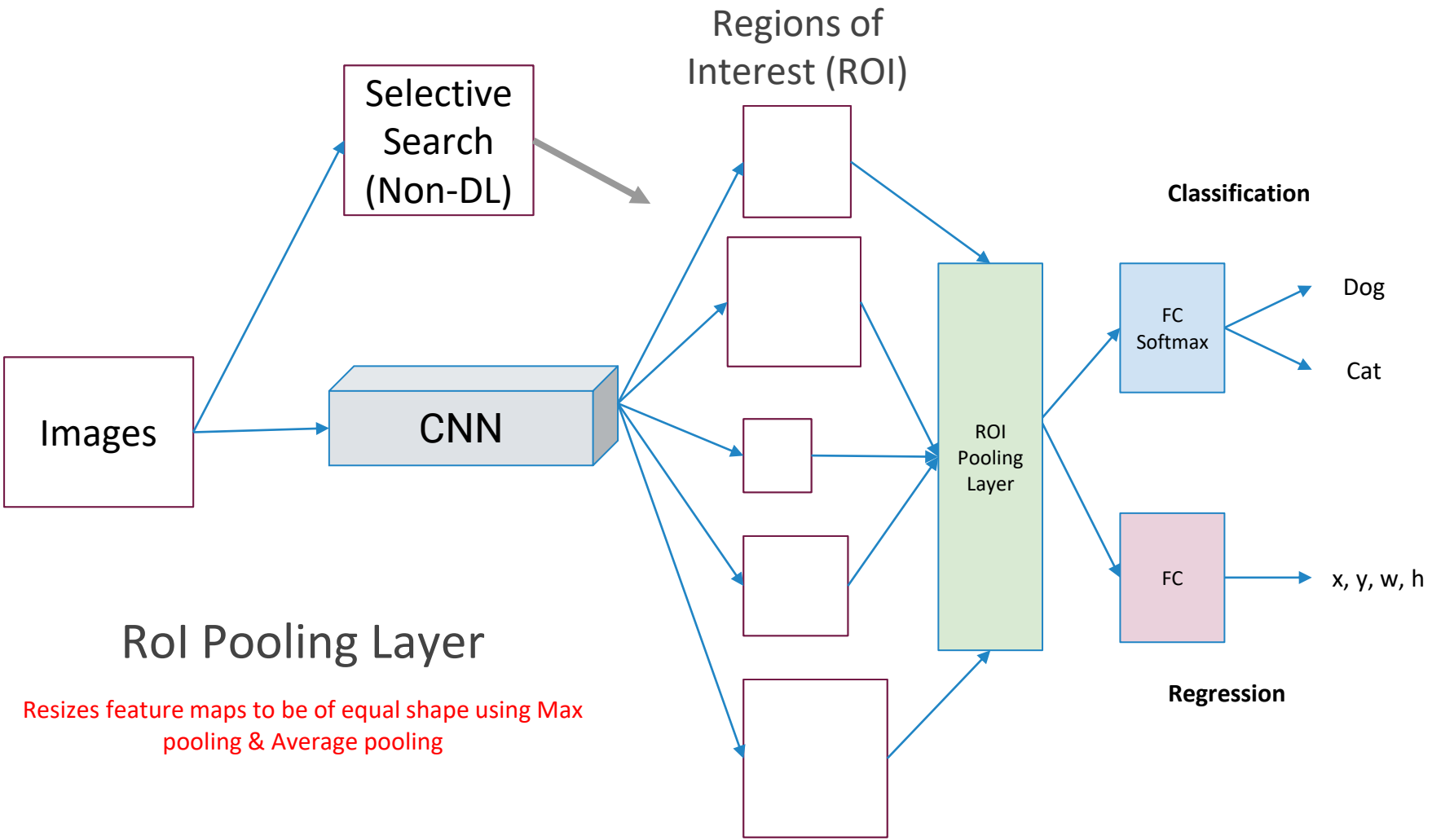
How Fast R-CNN is faster than R-CNN?

- It does not break original image in 2000 RoIs directly.
- Rather it breaks original image feature map into 2000 RoI feature maps.



FAST R-CNN

Image is passed through CNN only once, rather **RoI Feature maps** are created from Image feature map

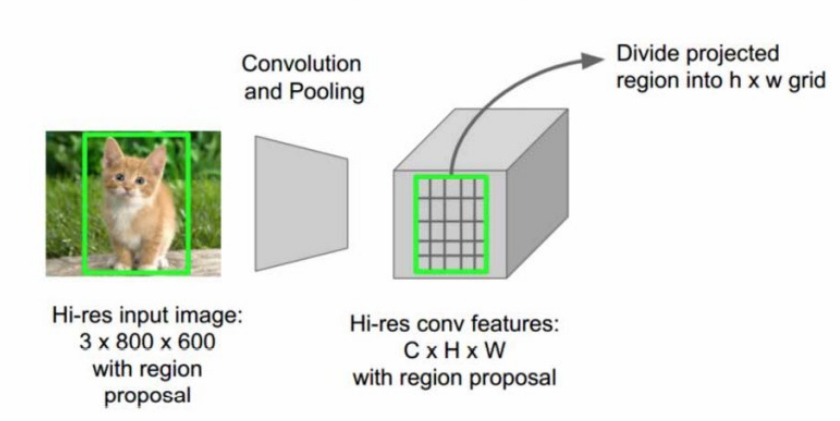


RoI Pooling Layer

Resizes feature maps to be of equal shape using Max pooling & Average pooling

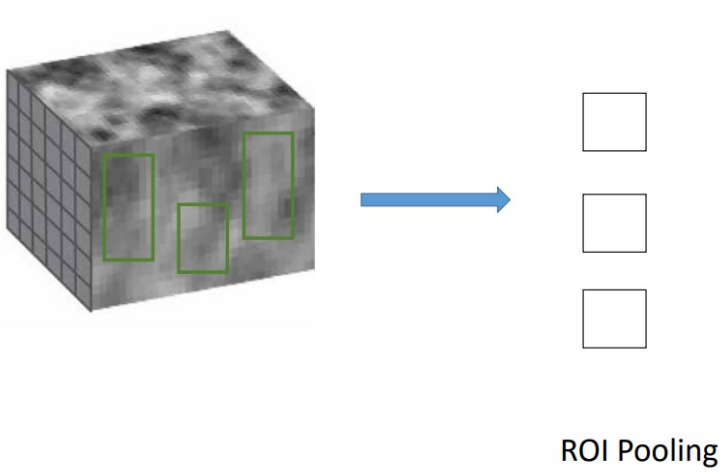
Use of convolutional layers

- CNNs extract high level features from image like lines, edges, shapes.
- The Fast R-CNN detector processes the entire image. Hence we did the work of 2000 convolutions in a single step.
- Fast R-CNN chooses CNN features corresponding to each region proposal.

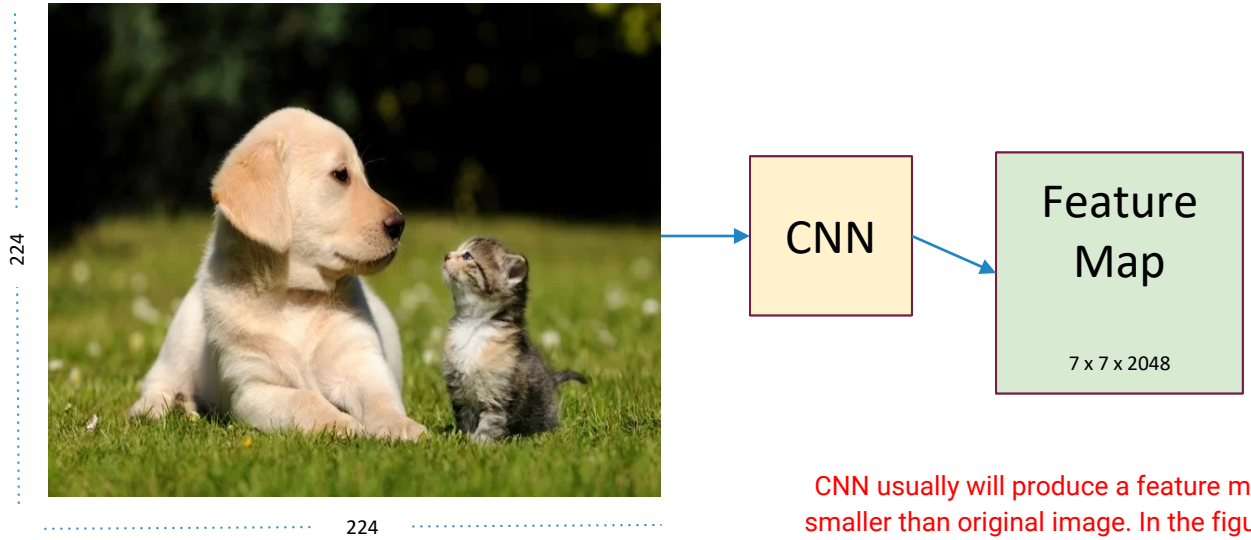


Use of region proposal layer and roi pool

- ROI pool layer convert these regions to a smaller feature map of fixed size before feeding to fully connected neural network.
- This is done using max pooling or average pooling.



What is RoI Pooling Layer?

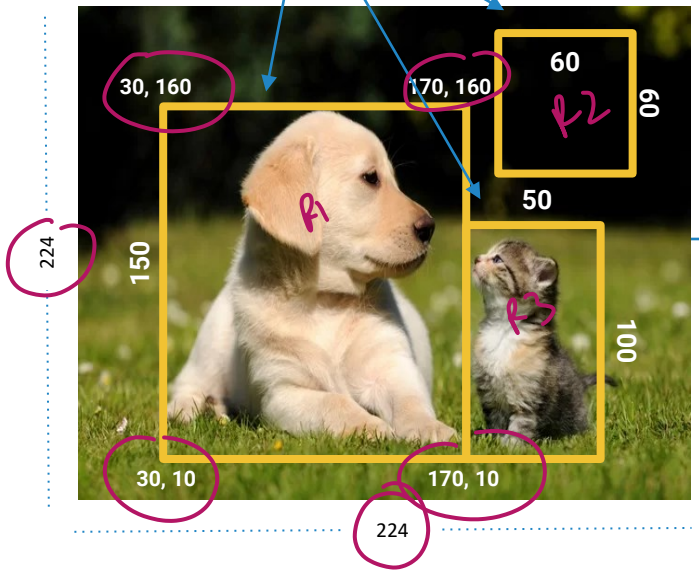


CNN usually will produce a feature map which spatially is much smaller than original image. In the figure above, we started with a 224x224 image and got a feature map of 7x7 *i.e.* a reduction by a factor of 32 (224/7 = 32).

RoIs

(Selective Search)

RoI coordinates from Selective Search will be in original image scale e.g 224x224



CNN

Feature Map

7 x 7 x 2048

224 X224

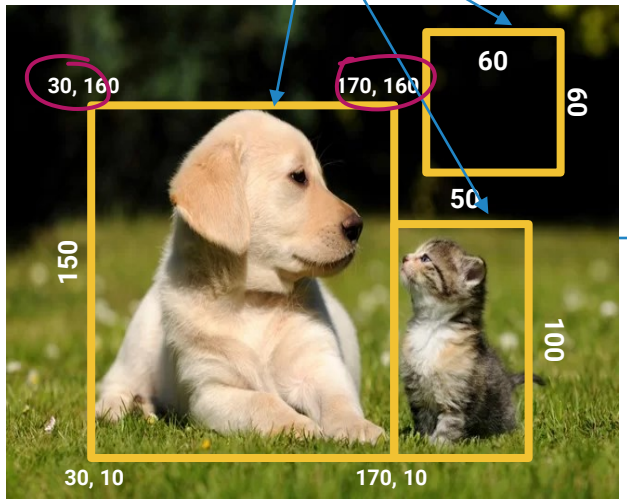
convnet

7X7

$\frac{224}{32} = 7$

RoIs

(Selective Search)



CNN

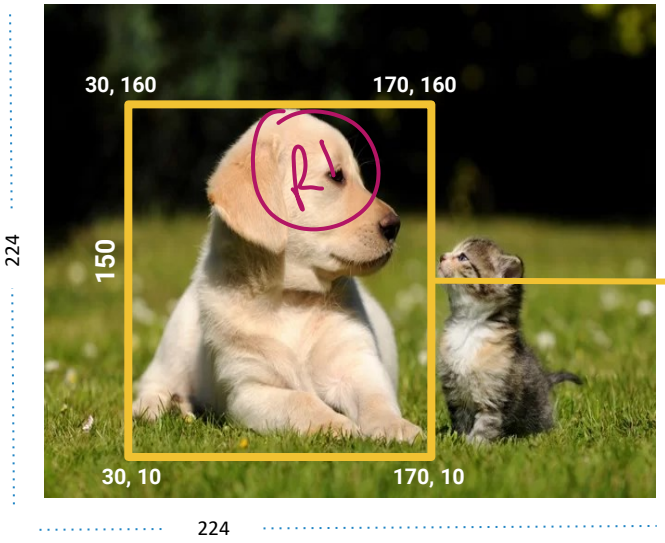
Original image Feature Map

0.06	0.05	0.1	0.03	0.04	0.0	0.01
0.1	0.9	0.3	0.07	0.01	0.0	0.0
0.3	0.01	0.4	0.2	0.05	0.05	0.1
0.2	0.9	0.6	0.3	0.7	0.6	0.2
0.66	0.7	0.3	0.1	0.12	0.36	0.03
0.01	0.5	0.45	0.05	0.6	0.5	0.06
0.0	0.1	0.7	0.1	0.3	0.92	0.05

Feature map of whole image - 7x7x2048

How do we get RoI feature maps?

Building RoI Feature Map



R1 R1/32 fm1

RoI Coordinates	Divide coords by 32	Convert to Int (Floor)
30,160	0.9, 5	0, 5
170, 160	5.3, 5	5, 5
30,10	0.9, 0.3	0, 0
170,10	5.3, 0.3	5, 0

Let's take one RoI to understand the steps

Convert RoI coordinates from original image scale (224x224) to Feature map scale (7x7)

RoI Coordinates (224x224 scale)	ROI Coordinates (7x7 Scale)
30.160	0, 5
170, 160	5, 5
30,10	0, 0
170,10	5, 0

0.06	0.05	0.1	0.03	0.04	0.0	0.01
0.1	0.9	0.3	0.07	0.01	0.0	0.0
0.3	0.01	0.4	0.2	0.05	0.05	0.1
0.2	0.9	0.6	0.3	0.7	0.6	0.2
0.66	0.7	0.3	0.1	0.12	0.36	0.03
0.01	0.5	0.45	0.05	0.6	0.5	0.06
0.0	0.1	0.7	0.1	0.3	0.92	0.05

Original image Feature Map - 7x7x2048

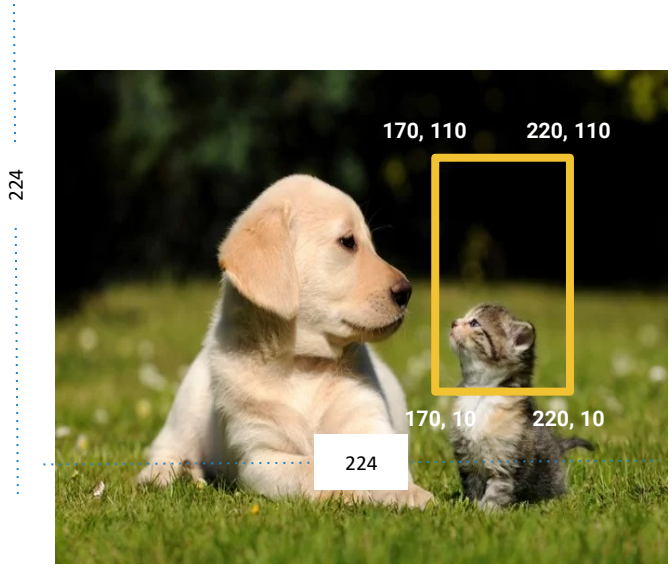
fm1

0.3	0.01	0.4	0.2	0.05
0.2	0.9	0.6	0.3	0.7
0.66	0.7	0.3	0.1	0.12
0.01	0.5	0.45	0.05	0.6
0.0	0.1	0.7	0.1	0.3

ROI Feature Map (5x5x2048)

Feature map for another RoI

Convert RoI coordinates from original image scale (224x224) to Feature map scale (7x7)



RoI Coordinates	Divide coords by 32	Convert to Int (Floor)
170,10	5.3, 0.3	5, 0
220, 10	6.9, 0.3	6, 0
220,110	6.9, 3.4	6, 3
170,110	5.3, 3.4	5, 3

RoI Coordinates (224x224 scale)	ROI Coordinates (7x7 scale)
170,10	5, 0
220, 10	6, 0
220,110	6, 3
170,110	5, 3

0.06	0.05	0.1	0.03	0.04	0.0	0.01
0.1	0.9	0.3	0.07	0.01	0.0	0.0
0.3	0.01	0.4	0.2	0.05	0.05	0.1
0.2	0.9	0.6	0.3	0.7	0.6	0.2
0.66	0.7	0.3	0.1	0.12	0.36	0.03
0.01	0.5	0.45	0.05	0.6	0.5	0.06
0.0	0.1	0.7	0.1	0.3	0.92	0.05

Original image Feature Map

7x7x2048

0.36
0.5
0.92

ROI Feature Map

3x1x2048

Different Feature maps will have different sizes

0.3	0.01	0.4	0.2	0.05
0.2	0.9	0.6	0.3	0.7
0.66	0.7	0.3	0.1	0.12
0.01	0.5	0.45	0.05	0.6
0.0	0.1	0.7	0.1	0.3

ROI Feature Map for One area

0.36
0.5
0.92

ROI Feature Map for another area

Next Layer will expect all ROI feature Maps to be of same size

Resizing ROI Feature Maps

- Let's say we want to resize all RoI feature Maps to 2x2.
- Also let's say we want to use a 3x3 pool size with stride 3.

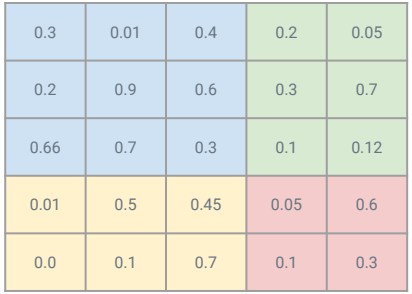
0.3	0.01	0.4	0.2	0.05
0.2	0.9	0.6	0.3	0.7
0.66	0.7	0.3	0.1	0.12
0.01	0.5	0.45	0.05	0.6
0.0	0.1	0.7	0.1	0.3

ROI Feature Map for One area

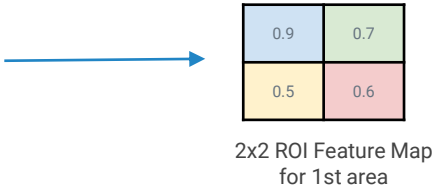
0.36
0.5
0.92

ROI Feature Map for another area

Resizing ROI Feature Maps

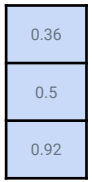


ROI Feature Map for 1st area

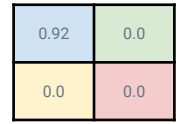


2x2 ROI Feature Map for 1st area

In RoI Pooling, you can have uneven size parts



ROI Feature Map for 2nd area



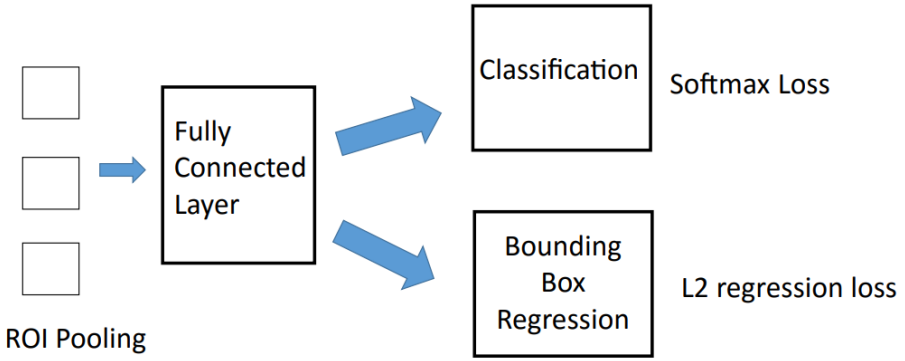
2x2 ROI Feature Map for 2nd area

In RoI Pooling, you can have uneven size parts

Fully connected layer and final prediction

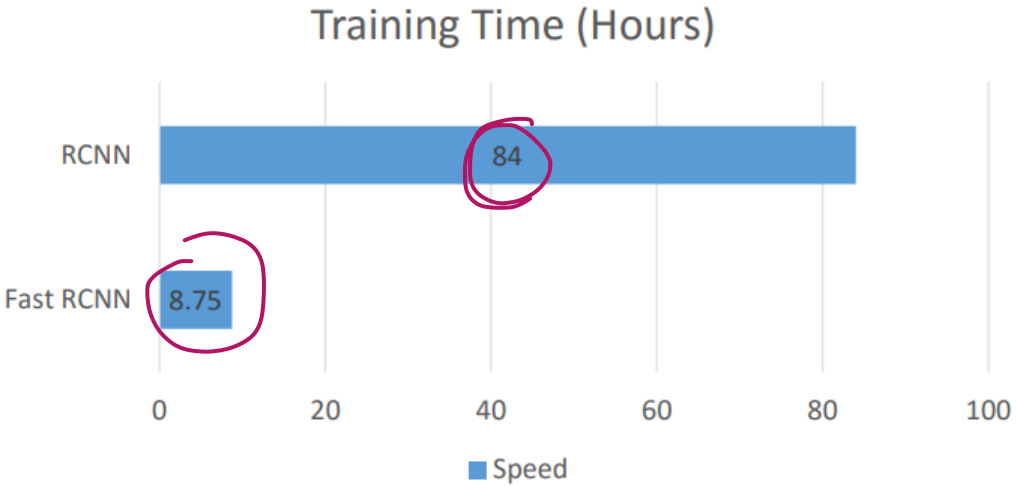
Fully Connected layers are used to produce 2 outputs.

- Predict the class label of the object in ROI
- Regression to find (x, y, w, h) coordinates of Bounding Box
  x, y = X and Y coordinates of the bounding box
  w = width of bounding box
  h = height of bounding box



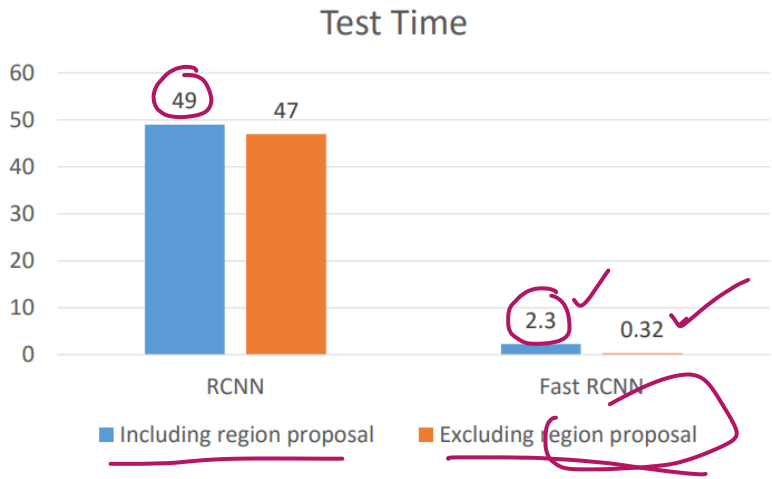
Performance of fast rcnn in training

- Fast RCNN is10x faster compared to RCNN because
- Only a single image is processed by a CNN, instead of 2000 CNNs on images obtained from selective search in RCNN algorithm.



Performance of fast rcnn in test

- Fast RCNN is really fast in prediction.
- The only bottleneck is region proposal which takes almost 2 seconds which is a lot of time.

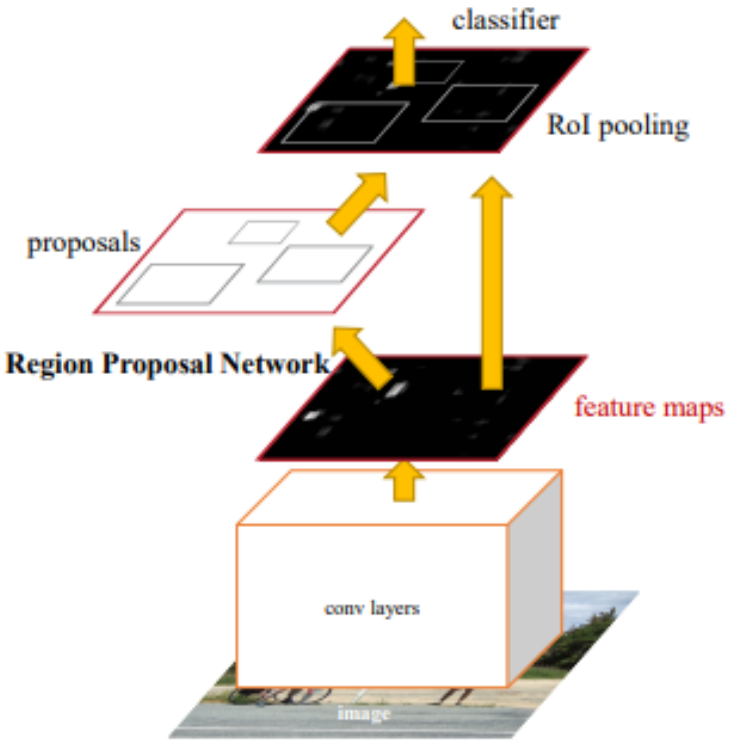


Limitations of fast rcnn

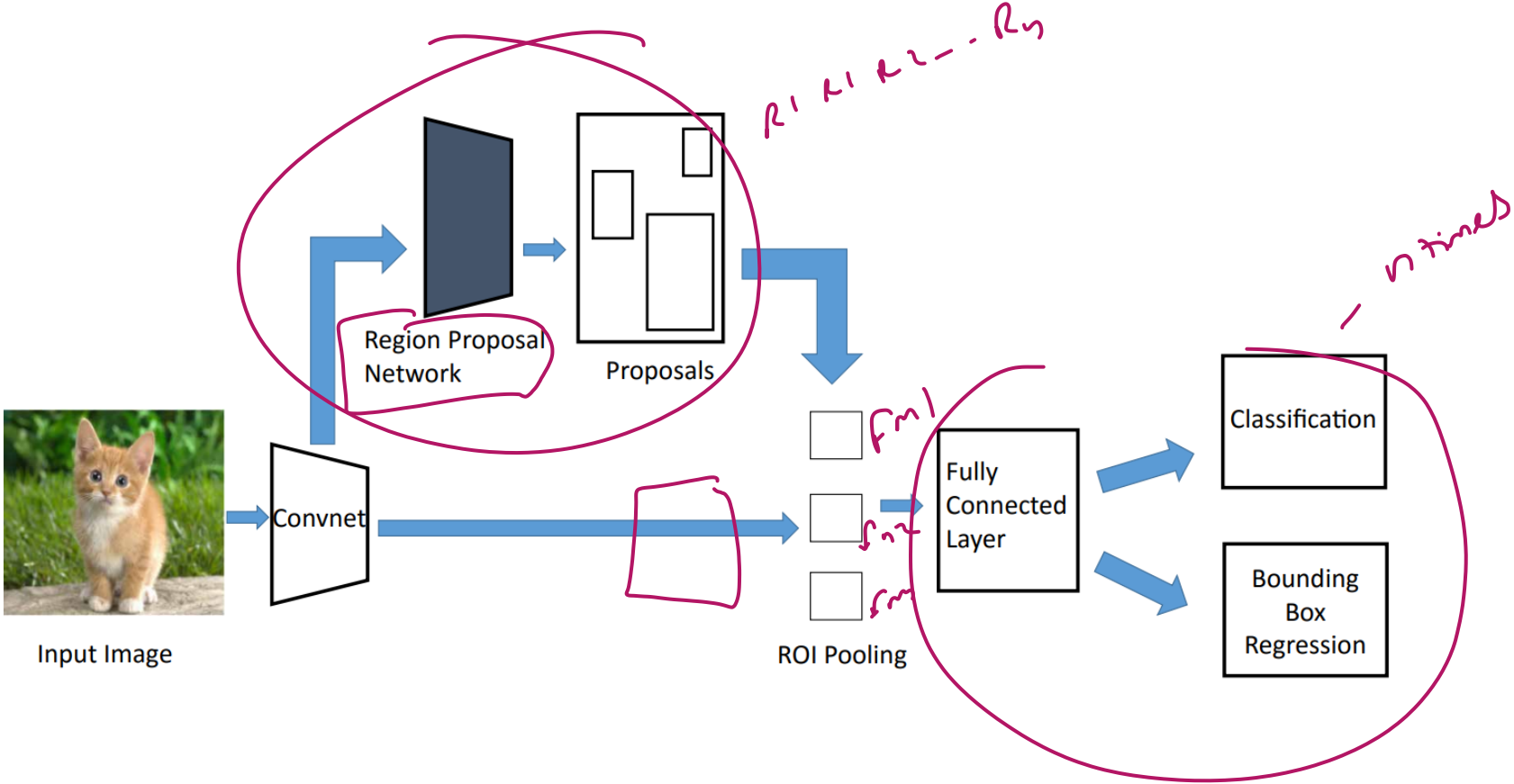
- Region proposal takes a lot of time (2 seconds).
- Selective Search is a slow and time consuming process which is unsuitable for real world, large datasets.
- This issue is resolved with Faster RCNN.

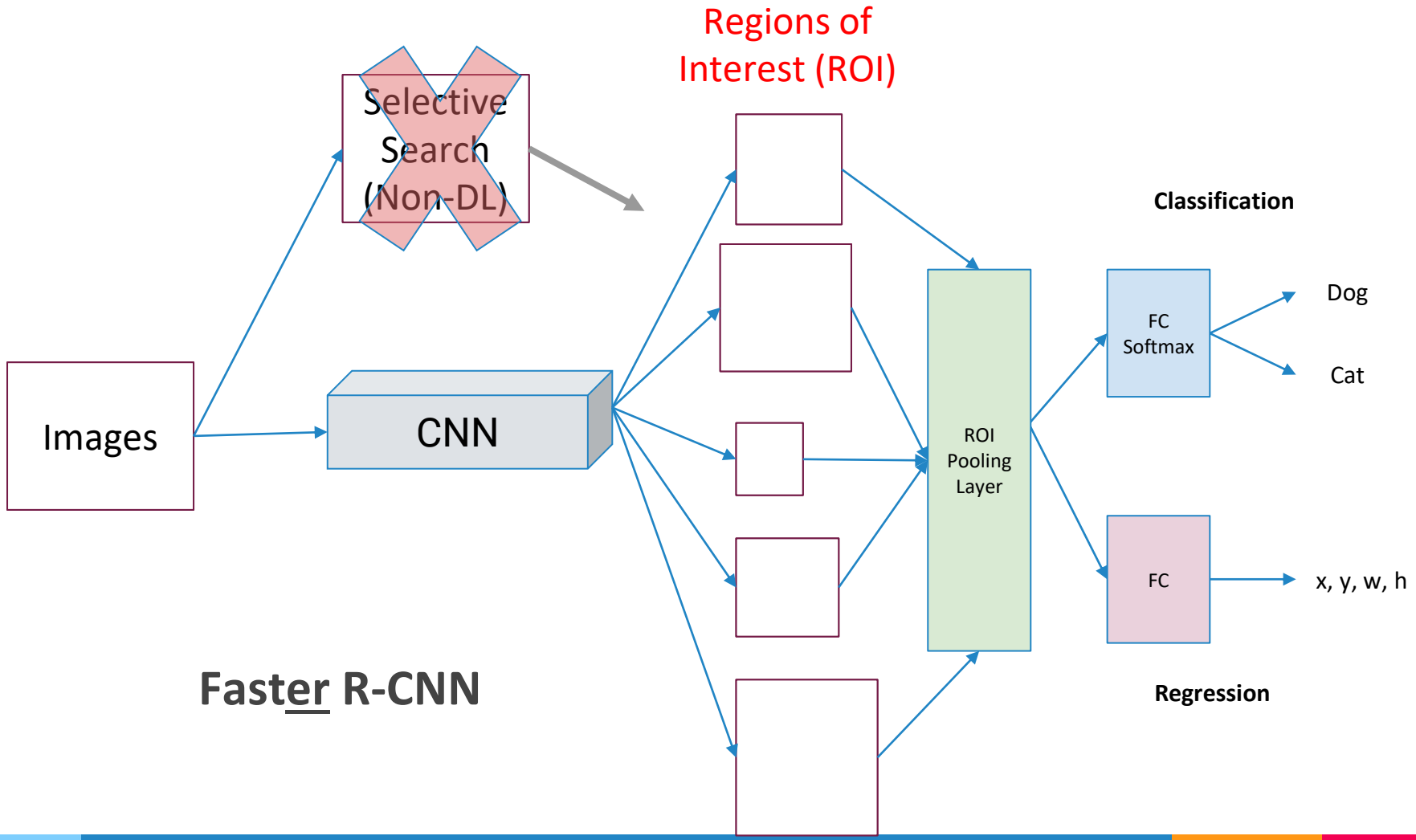
Faster rcnn

- Faster RCNN uses an inbuilt region proposal network (RPN) to generate region proposals directly in the network without using an external algorithm like Selective search.

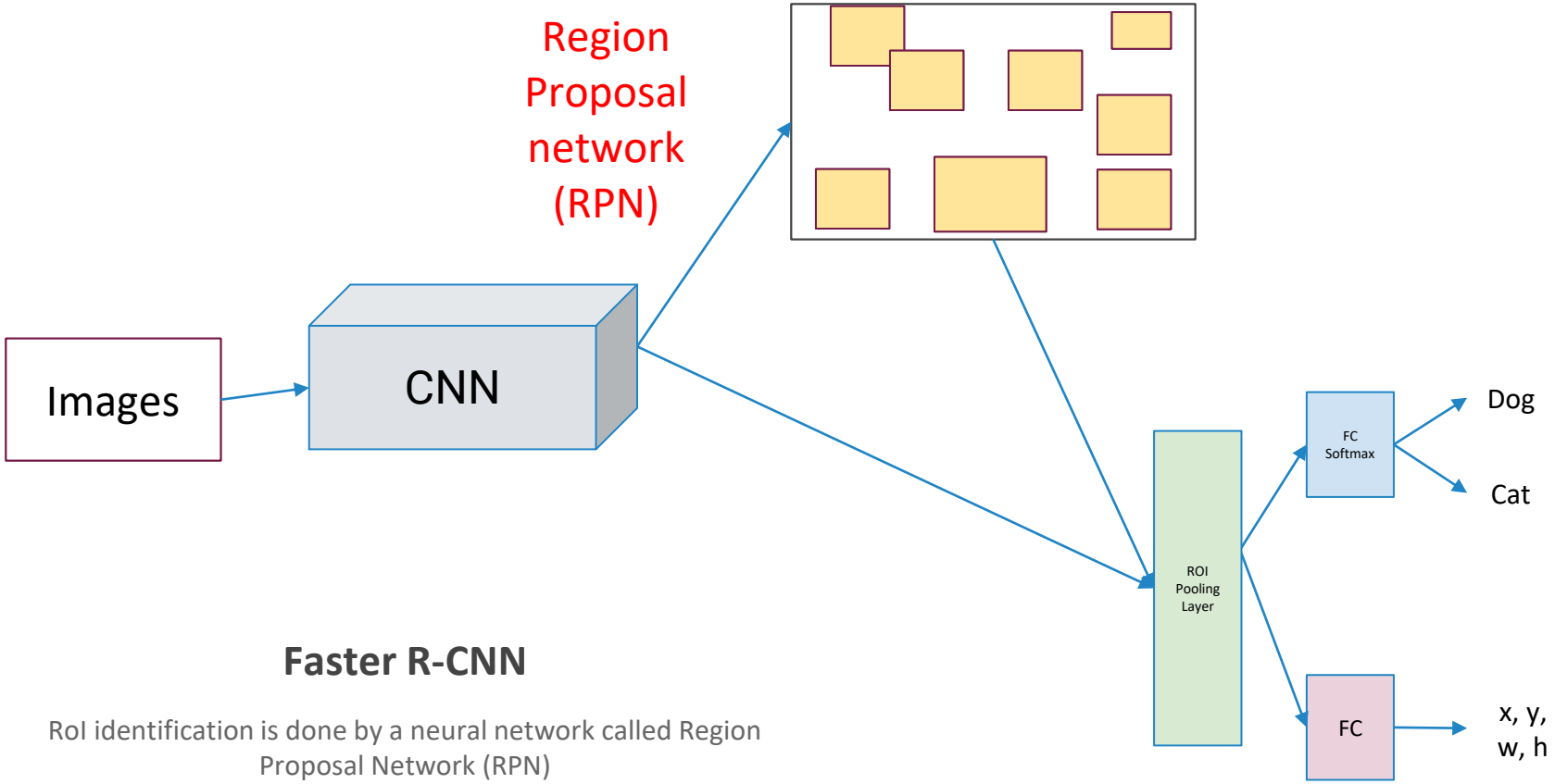


Faster rcnn network




Regions of Interest (ROI)
Selective Search (Non-DL)
Images
CNN
ROI Pooling Layer
Classification
FC Softmax
Dog
Cat
FC
x, y, w, h
Regression
Faster R-CNN

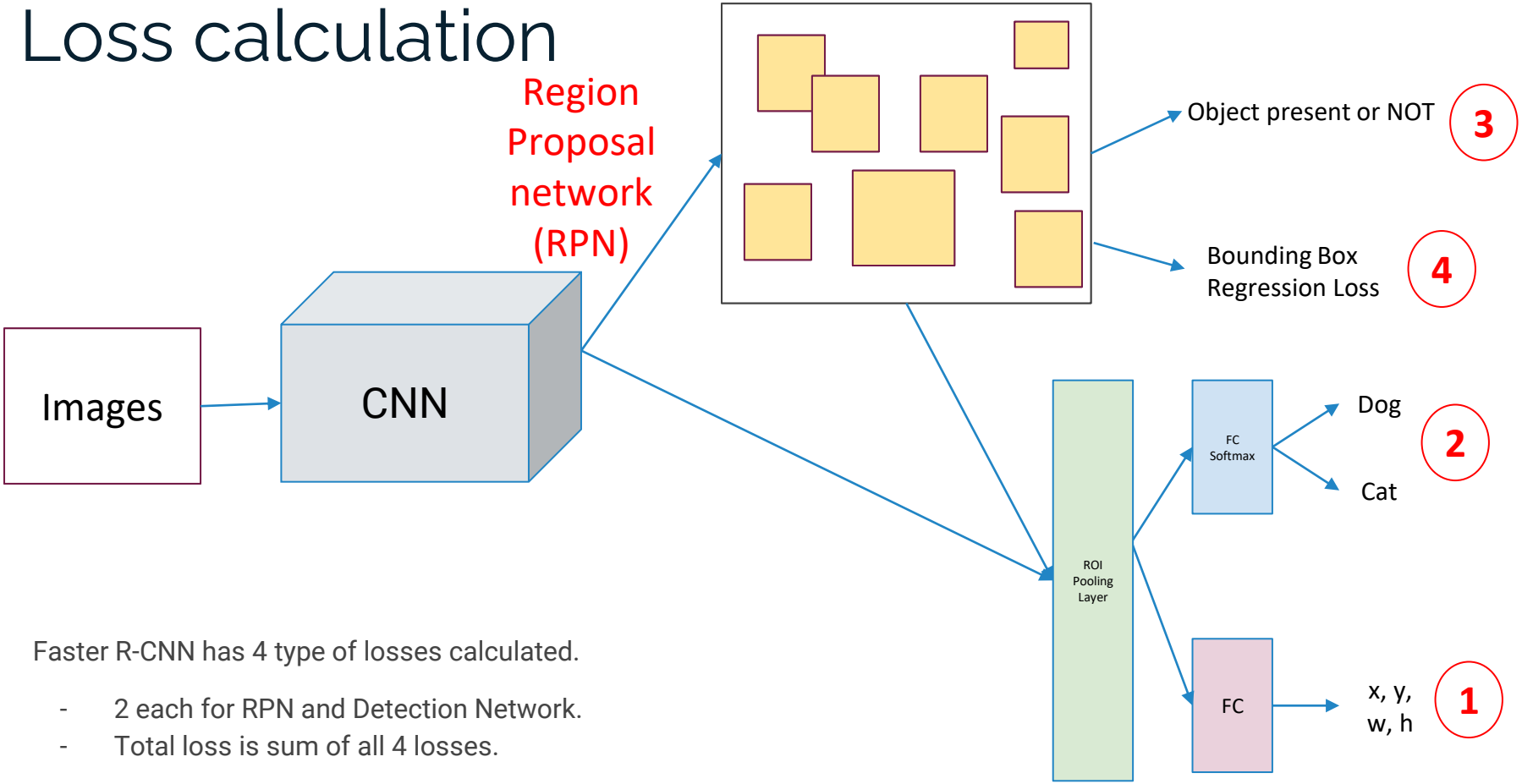
We use a neural network to get RoI



Faster R-CNN

RoI identification is done by a neural network called Region Proposal Network (RPN)

Loss calculation



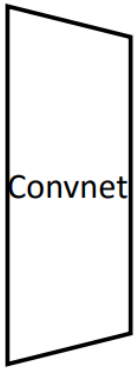
Faster R-CNN has 4 type of losses calculated.

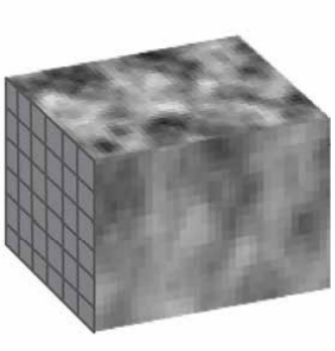
- 2 each for RPN and Detection Network.
- Total loss is sum of all 4 losses.

Region proposal network

- Region Proposal Network creates boxes called anchors which are most likely to contain object.

Original



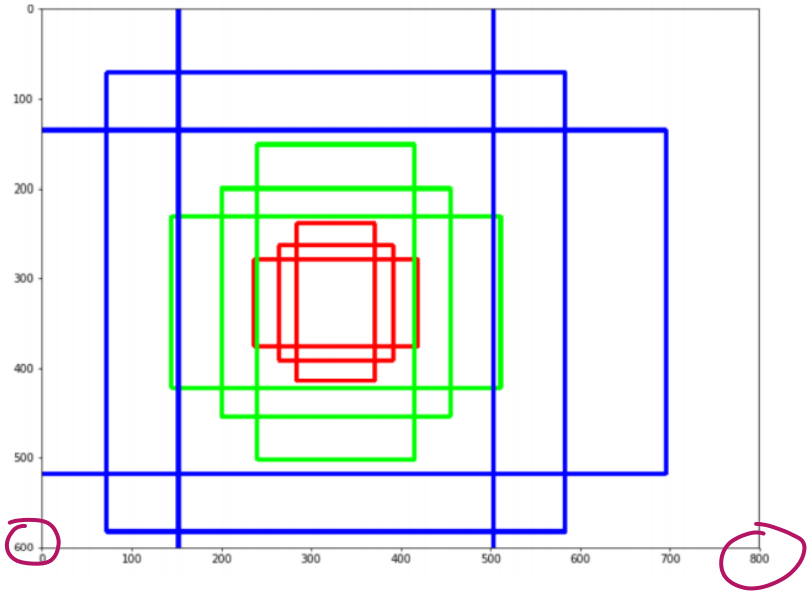


Features

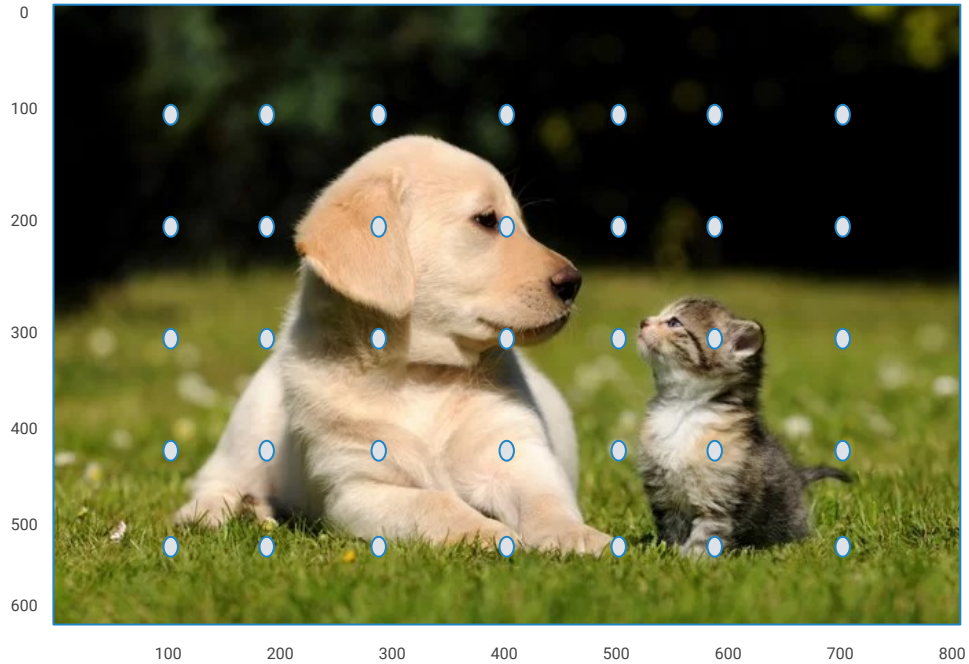
Regions of Interest

Anchors in faster RCNN

- Anchors are fixed boxes placed across the images to find location of object.
- Faster RCNN has 9 anchors of fixed size like 128x128, 256x256 and 512x512 (red, green and blue).
- For each size, the width to height ratio is 1:1, 1:2 and 2:1.



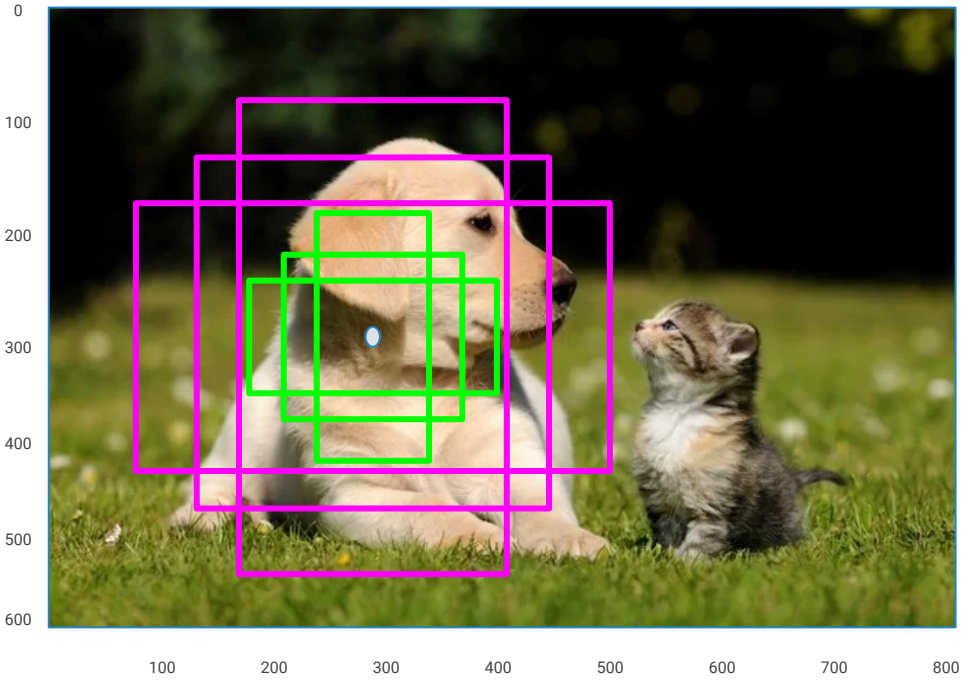
Anchors in faster RCNN



- Identify anchor points for input image. In this example, our image is 600 x 800 x 3
- Usually number of anchor points are same as feature map e.g if feature map is 38x51x d then number of anchor points will be 38 x 51
- Place boxes of different size and aspect ratio at each anchor point.

Understanding Anchor Boxes

Anchors in faster RCNN



1. Place a box of a defined size e.g **128 x 128** at the anchor point.

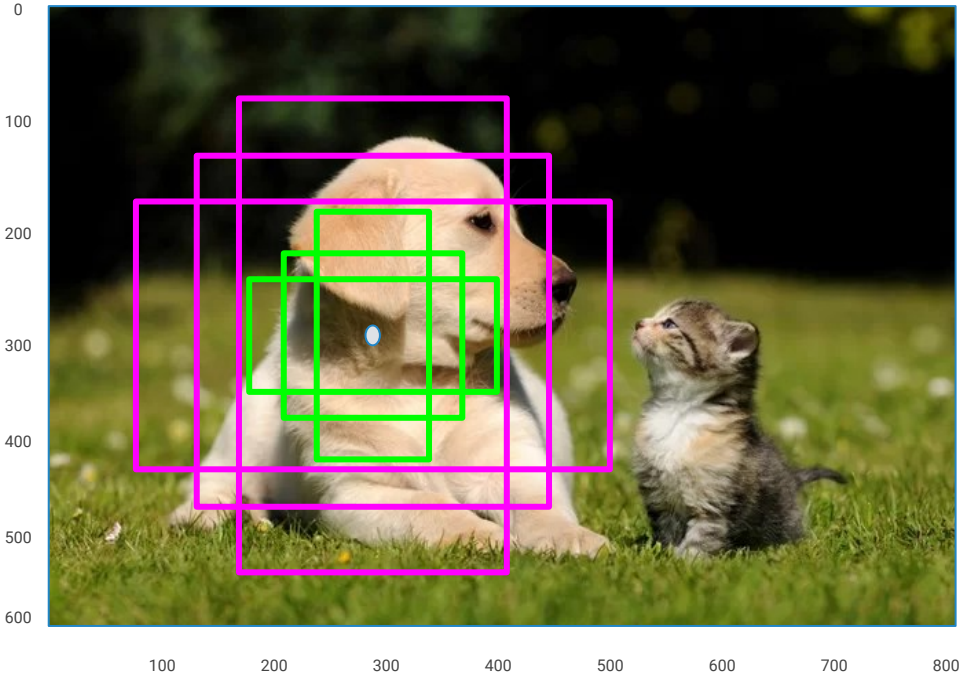
2. The image part covered by this box becomes an RoI which will be looked up by RPN network for objectiness and more precise boundary for the object as usually object size is not exactly same as anchor box size.

3. We place other anchor boxes at this point with different **aspect ratios** e.g 1:2 or 2:1 ratio between width and height.

4. We also place anchor boxes of **different sizes** e.g 256 x 256 or 512 x 512 to create more anchor boxes for RPN to look at.

Understanding Anchor Boxes

Anchors in faster RCNN



5. Usually we define **10s of thousands** of anchor boxes for an image in Faster R-CNN.

For example:

Number of anchor points : **38 x51**
(each feature map point can be an anchor)

Number of anchor boxes at each point : **9**
(3 sizes and 3 aspect ratios)

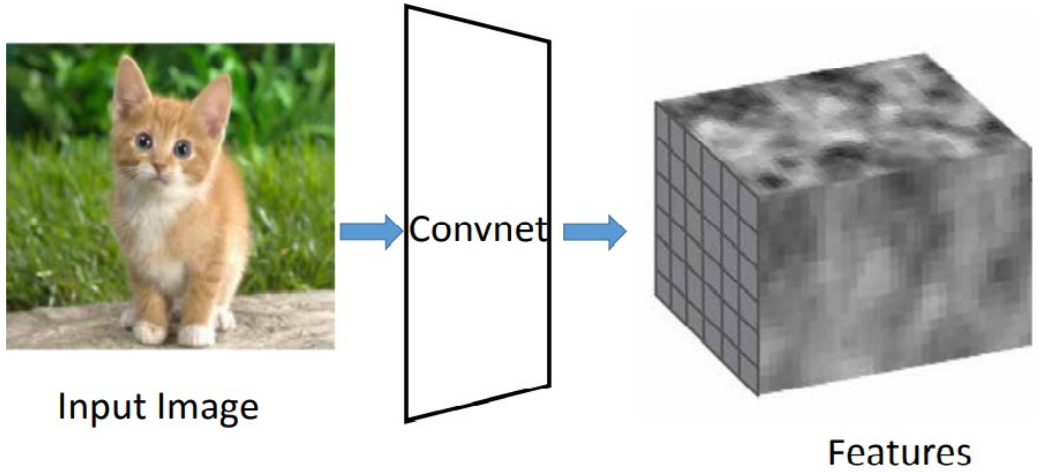
Total # anchor boxes for an image : **17,442**
(38 x 51 x 9)

Number of anchor boxes remain same for every image in a dataset.

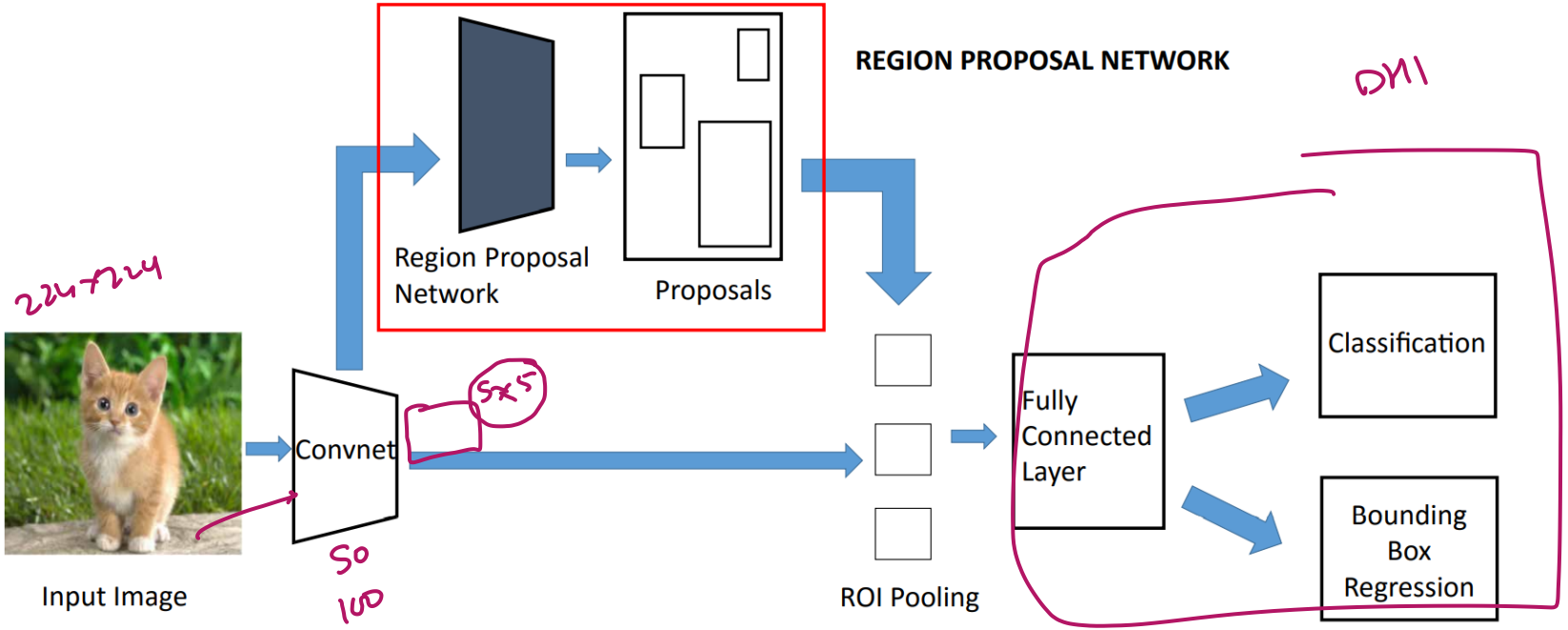
Understanding Anchor Boxes

Step1 : convolutional layer

- First a picture goes through convolutional layer and feature maps are extracted.

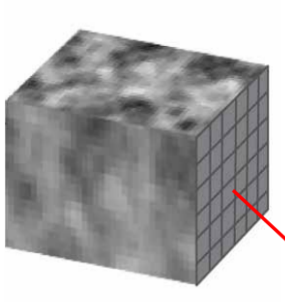


Step2 : region proposal network

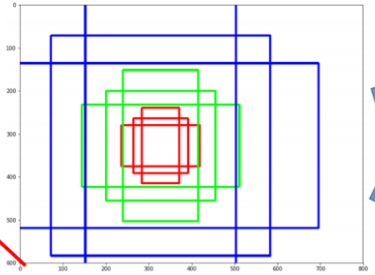


Step2 : region proposal network

- A sliding window is drawn over each point in feature map. At every point, 9 anchor boxes are drawn for generating region proposals

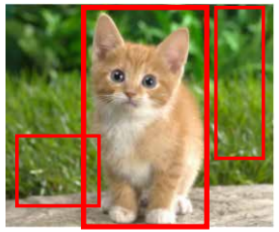


Features



2*9 channels for classification whether it is **object** or **background**

4*9 channels to find **bounding boxes** of these objects

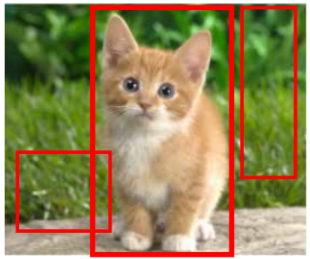


Outputs a set of region proposals on the image

Step 3 : detection network

1. Regions are of different sizes

2. Regions are converted to fixed size before passing through the network



ROI Pooling

Fully Connected Layer

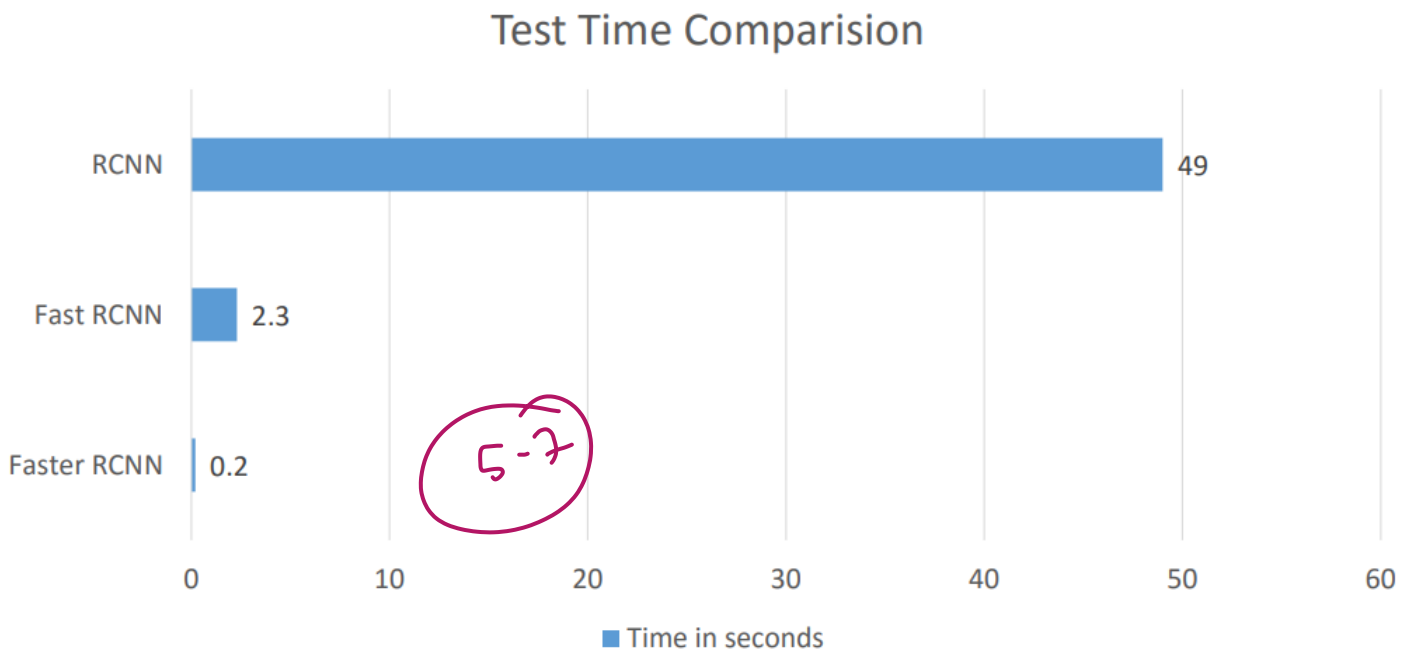
Classification

Bounding Box Regression

3. Classify the objects to their unique classes like cat etc..

4. Fine tune the bounding boxes of each object.

Performance of faster rcnn



Faster RCNN

Uses Deep Learning for RoI

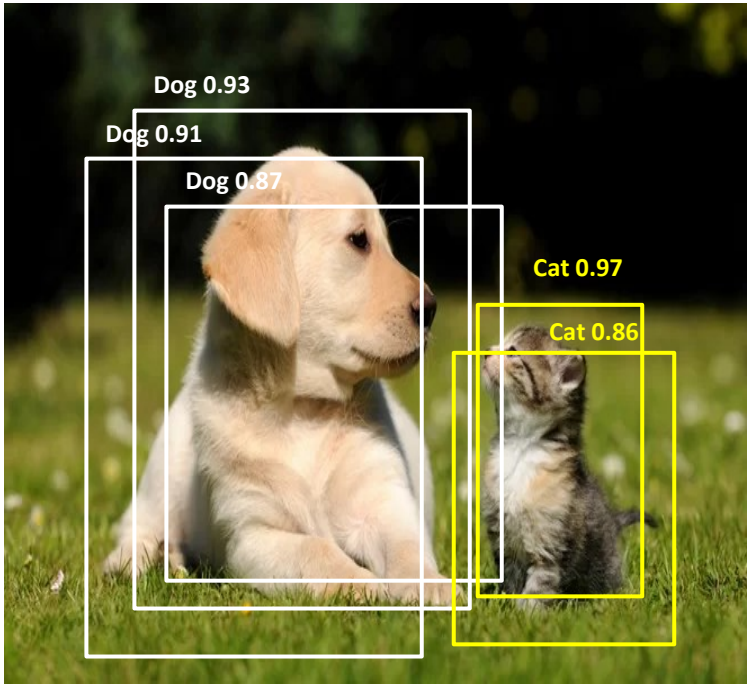
Most accurate architecture for Object detection

Takes **0.2** secs to predict on One Image

i.e. **5 fps**

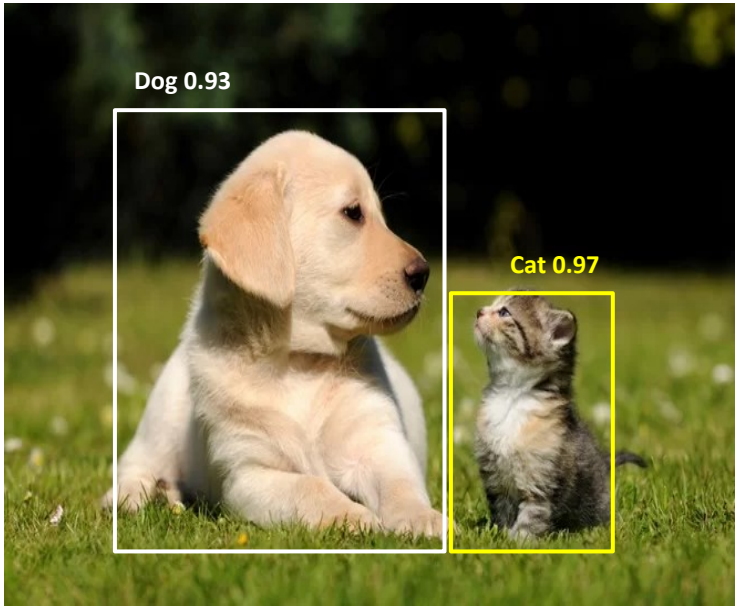
High accuracy, but still not suitable for real time

Non-Maximum Suppression (NMS)



- For a given class, select the prediction with highest confidence.
- Calculate IOU between predicted box with highest confidence and other predicted boxes of the same class.
- Remove all the lower confidence predicted boxes where IOU is above a threshold (let's say 0.7). These are duplicate predictions.
- Take the next highest confidence predicted of the same class and repeat the steps 1 to 3.
- Repeat steps 1 to 4 for all the predicted classes

Final prediction after NMS



Model Predictions

Predicts 1000s of objects

Remove Background Predictions

...10-100s predictions

Remove Duplicate Predictions (NMS)

...0-10s predictions

TensorFlow object detection API

TensorFlow Object Detection API

Reduces coding effort and allows us to build production ready Object detections models

This API also supports **Faster R-CNN**

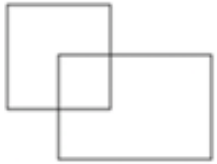
05

CONCLUSION

- Quiz
- Summary

Quiz

What is the IoU between these two boxes? The upper-left box is 2x2, and the lower-right box is 2x3. The overlapping region is 1x1.



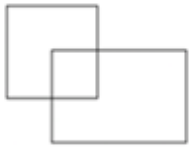
A. 1/9

B. 1/6

C. 1/10

Quiz

What is the IoU between these two boxes? The upper-left box is 2x2, and the lower-right box is 2x3. The overlapping region is 1x1.



A. 1/9

B. 1/6

C. 1/10

Answer - A

Quiz

Which of the following is/are one-stage methods for object detection?

A. Faster RCNN
B. RCNN
C. YOLO
D. All of above

Quiz

Which of the following is/are one-stage methods for object detection?

A. Faster RCNN
B. RCNN
C. YOLO
D. All of above

Answer - D

Summary

In a nutshell we learnt

- Object localization is about predicting name of object and bounding box coordinates assuming there is only one object in the image
- Object detection is about predicting class and bounding box coordinates of multiple objects in an image.
- RCNN, Faster RCNN and Faster RCNN are some of the techniques for object detection.

Thank You

Happy Learning!