Deep Learning and Convolutional Neural Network (42028)

Object Detection- 2

Frameworks

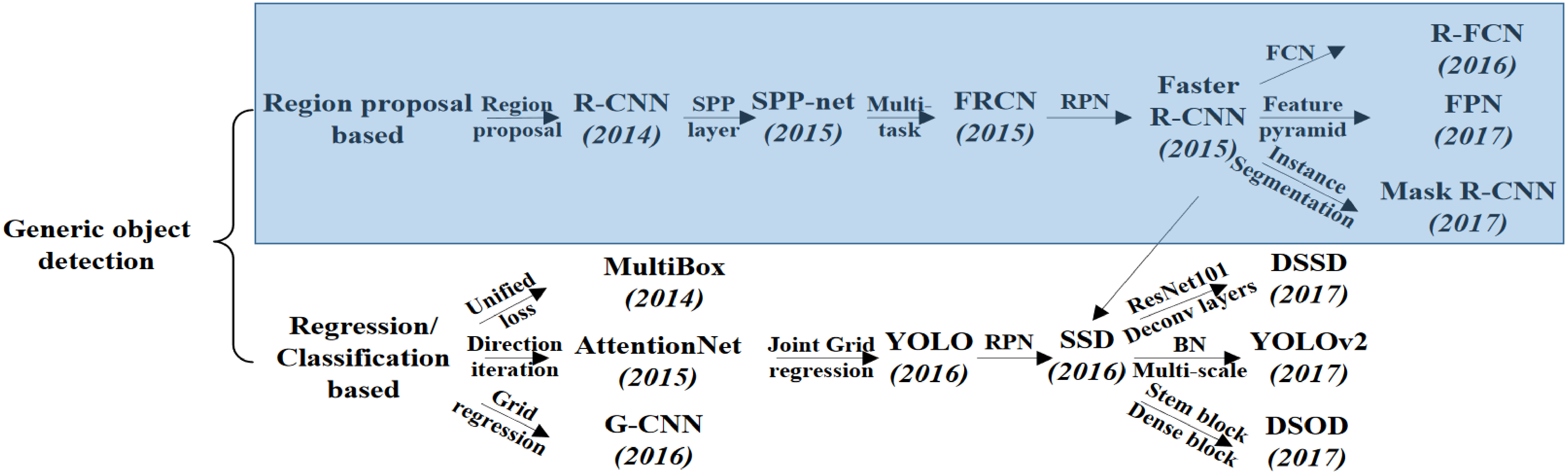
Object Detection

Region Proposal Based

Regression/Classification Based

Images source: https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad

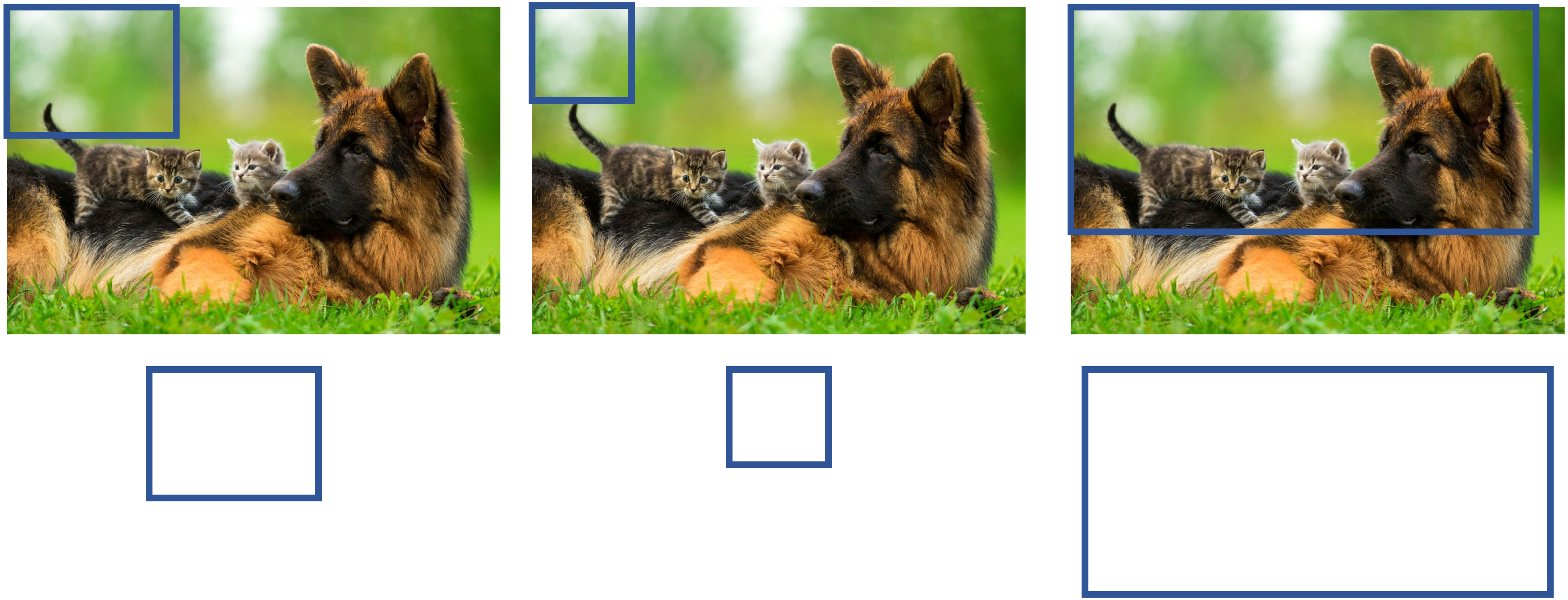
Object Detection Techniques History



Images source: https://arxiv.org/pdf/1807.05511.pdf

Object Detection Techniques Recap

Sliding Window technique



Source and Reference: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf
Image source: http://www.kamloopspropertyforsale.com/blog/does-owning-a-cat-depreciate-your-home.html

Object Detection Techniques Recap

Sliding Window technique

- Crop images and classify using CNN
- Try different sizes of the sliding window

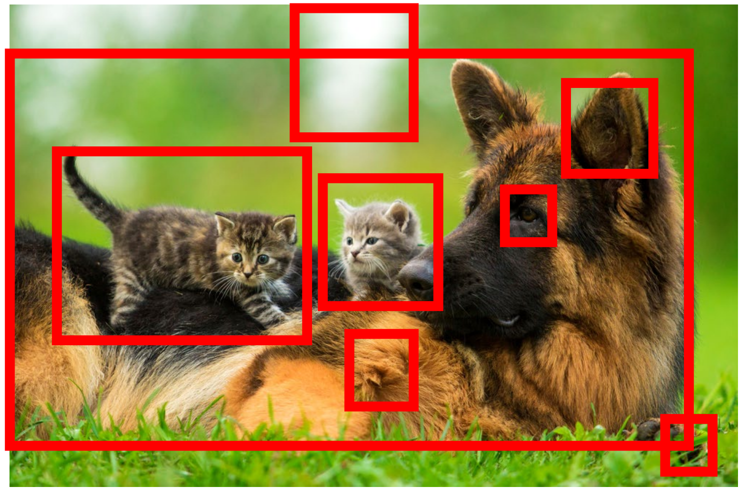
Issues:

- Slow
- Computationally very expensive
- Less accurate

Images source: https://arxiv.org/pdf/1807.05511.pdf

Object Detection Techniques Recap

Region Proposals



Source and Reference: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf
Image source: http://www.kamloopspropertyforsale.com/blog/does-owning-a-cat-depreciate-your-home.html

Predicting Bounding Boxes

Currently:
- Sliding Window
- Selective Search
- Region Proposals

Task:
- Predict Bounding boxes from CNN

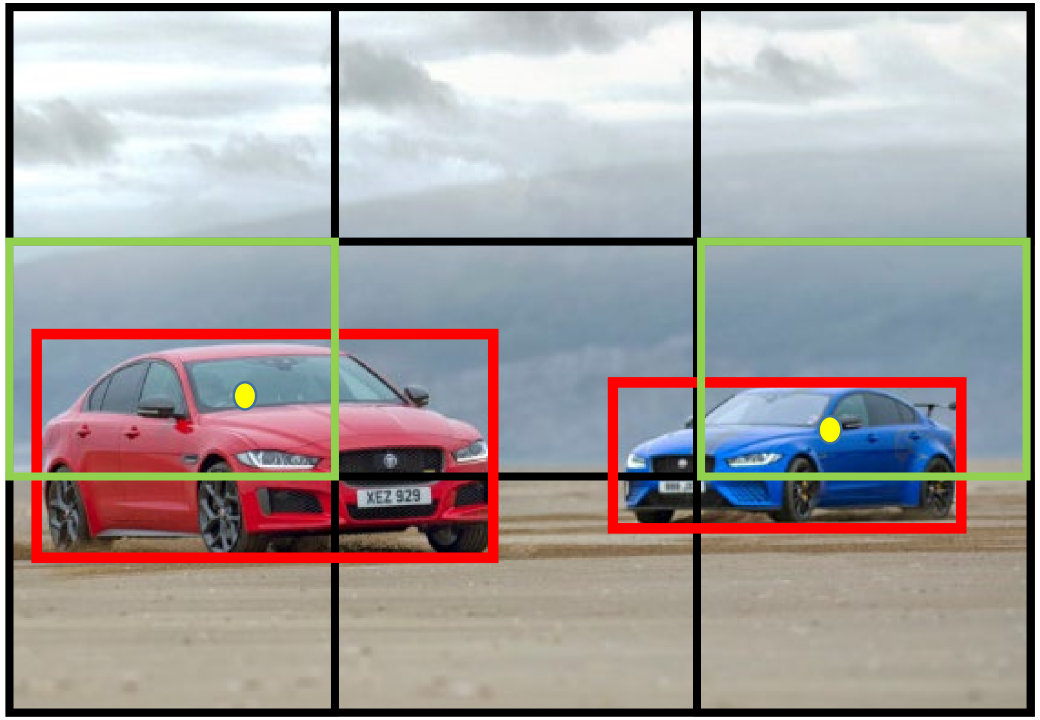
Source and Reference: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf

Predicting Bounding Boxes

- Place a grid over the image
- Apply image classification and localization to each of the grid cells

Images source: https://yallacompare.com/car-deals/uae/en/two-cars-one-dnajaguar-xe-300-sport-and-xe-sv-project-8/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Predicting Bounding Boxes



Class : {car, bike}

Training Strategy:

Input:

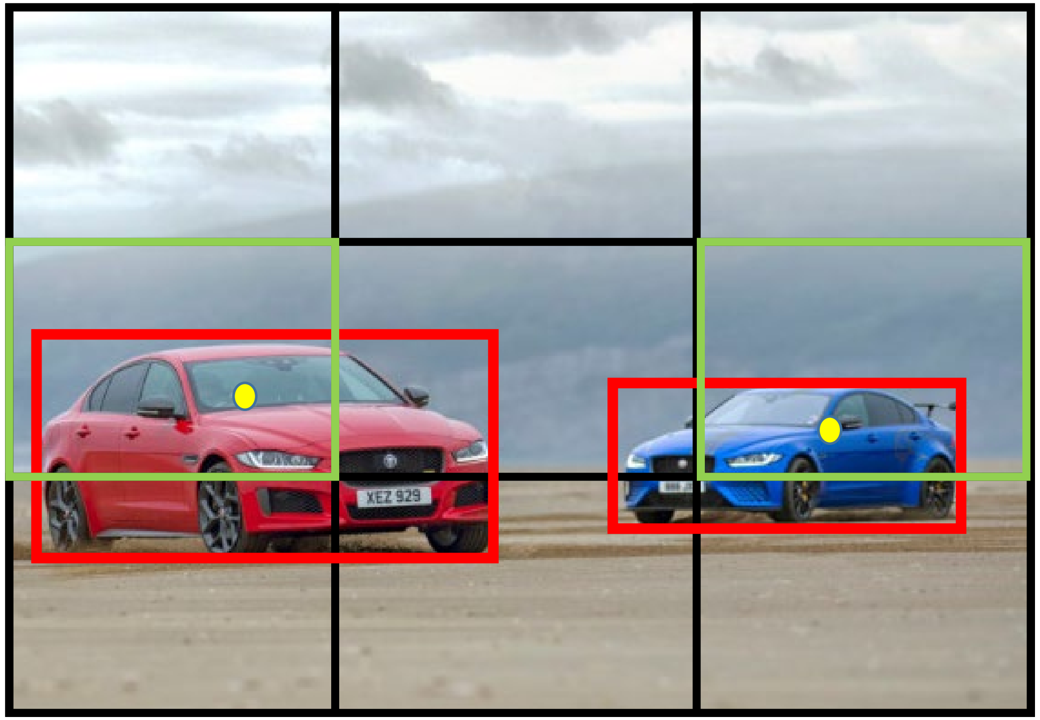
- Image: (ht x wd x 3)

Target:

- Bounding box information for each object
- Class for each object

Images source: https://yallacompare.com/car-deals/uae/en/two-cars-one-dnajaguar-xe-300-sport-and-xe-sv-project-8/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Predicting Bounding Boxes



Class : {car, bike}

Idea: Take the mid-point of the object and Assign it to a grid cell based on its location

Training Strategy:

Target:

$Y = \{p_o, x, y, h, w, c_1, c_2\}$ for each cell

e.g:

Cell(1,1) = {0, ?, ?, ?, ?, ?, ?}

:

Cell(2,1) = {1, x, y, h, t, 1, 0}

Cell(2,2) = {0, ?, ?, ?, ?, ?, ?}

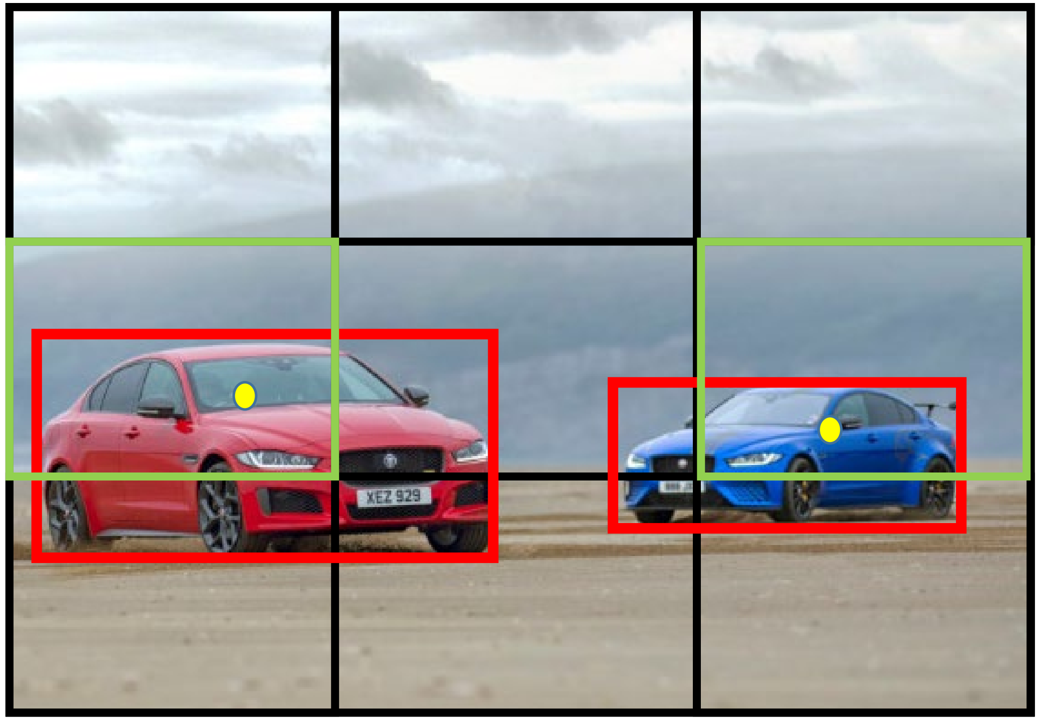
Cell(2,3) = {1, x, y, h, t, 1, 0}

:

Cell(3,3) = {0, ?, ?, ?, ?, ?, ?}

Images source: https://yallacompare.com/car-deals/uae/en/two-cars-one-dnajaguar-xe-300-sport-and-xe-sv-project-8/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Predicting Bounding Boxes



Class : {car, bike}

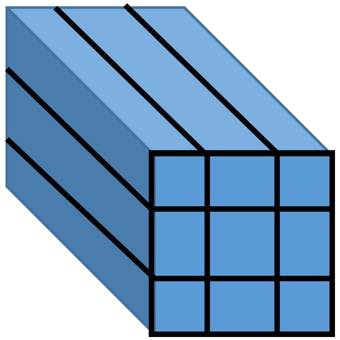
Training Strategy:

Target output vector:

3 X 3 X 7

3 X 3: Grid size

7: (5 + Number-of-Classes)



3 X 3 X 7

Images source: https://yallacompare.com/car-deals/uae/en/two-cars-one-dnajaguar-xe-300-sport-and-xe-sv-project-8/

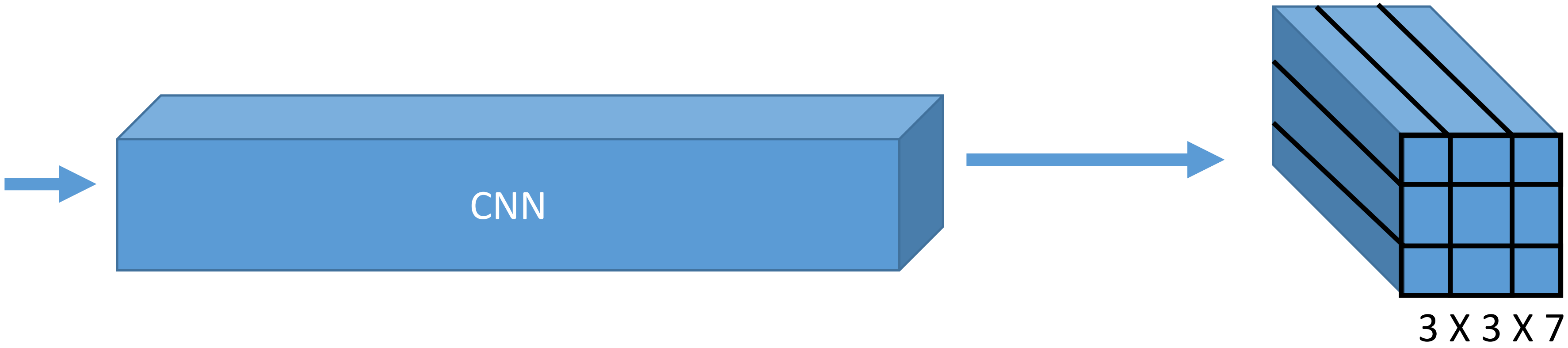
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Predicting Bounding Boxes

Training Strategy:

Input: X

Target: Y



Class : {car, bike}

In practice: The grid is finer, 19 X 19 instead of 3 X 3

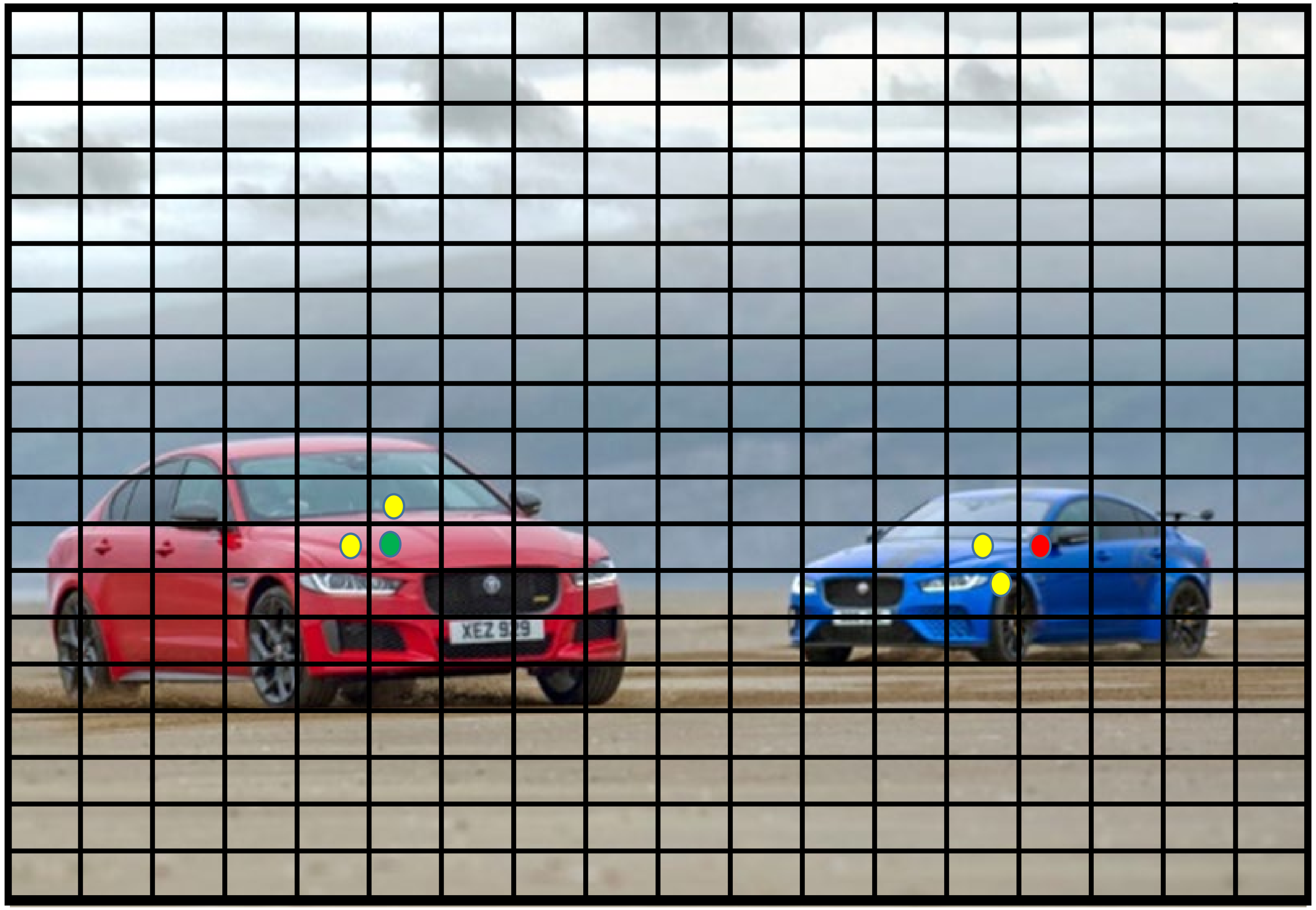
So, Target will be of size: 19 X 19 X 7

Works well for non-overlapping objects

Images source: https://yallacompare.com/car-deals/uae/en/two-cars-one-dnajaguar-xe-300-sport-and-xe-sv-project-8/

Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Non Maxima Suppression (NMS)

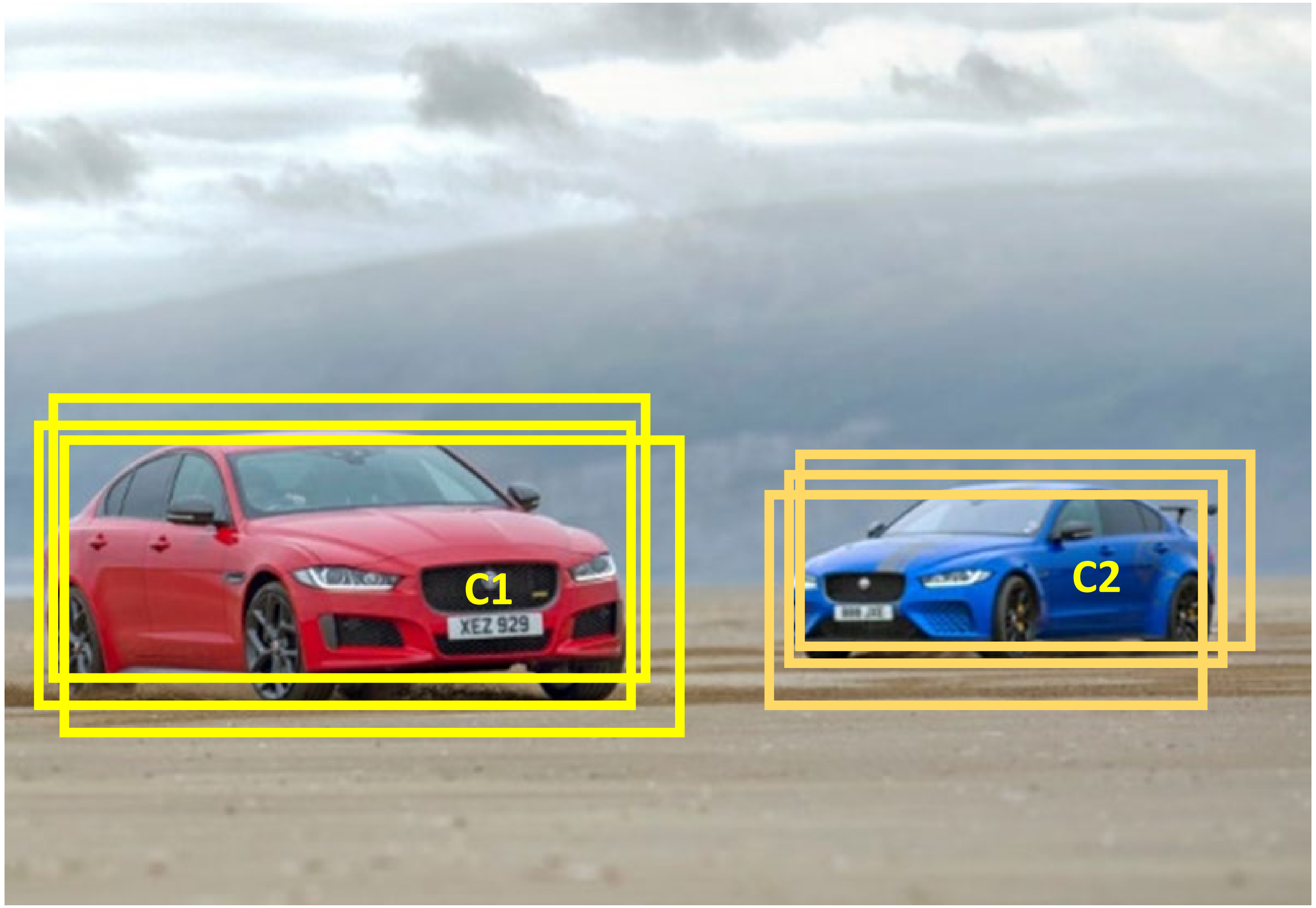


Issues with Object Detection:

1. **Each object has one mid-point**
2. **As each cells are subjected to object+localization classification**
3. **Hence, neighbouring cells might assume that it has the mid-point**
4. **Hence, Multiple detection bounding box**

Images source: https://arxiv.org/pdf/1807.05511.pdf

Non Maxima Suppression (NMS)



Sample prediction:

<u>For C1:</u>

Box1: 0.9 (Confidence Score)

Box2: 0.79

Box3: 0.82

<u>For C2:</u>

Box1: 0.92

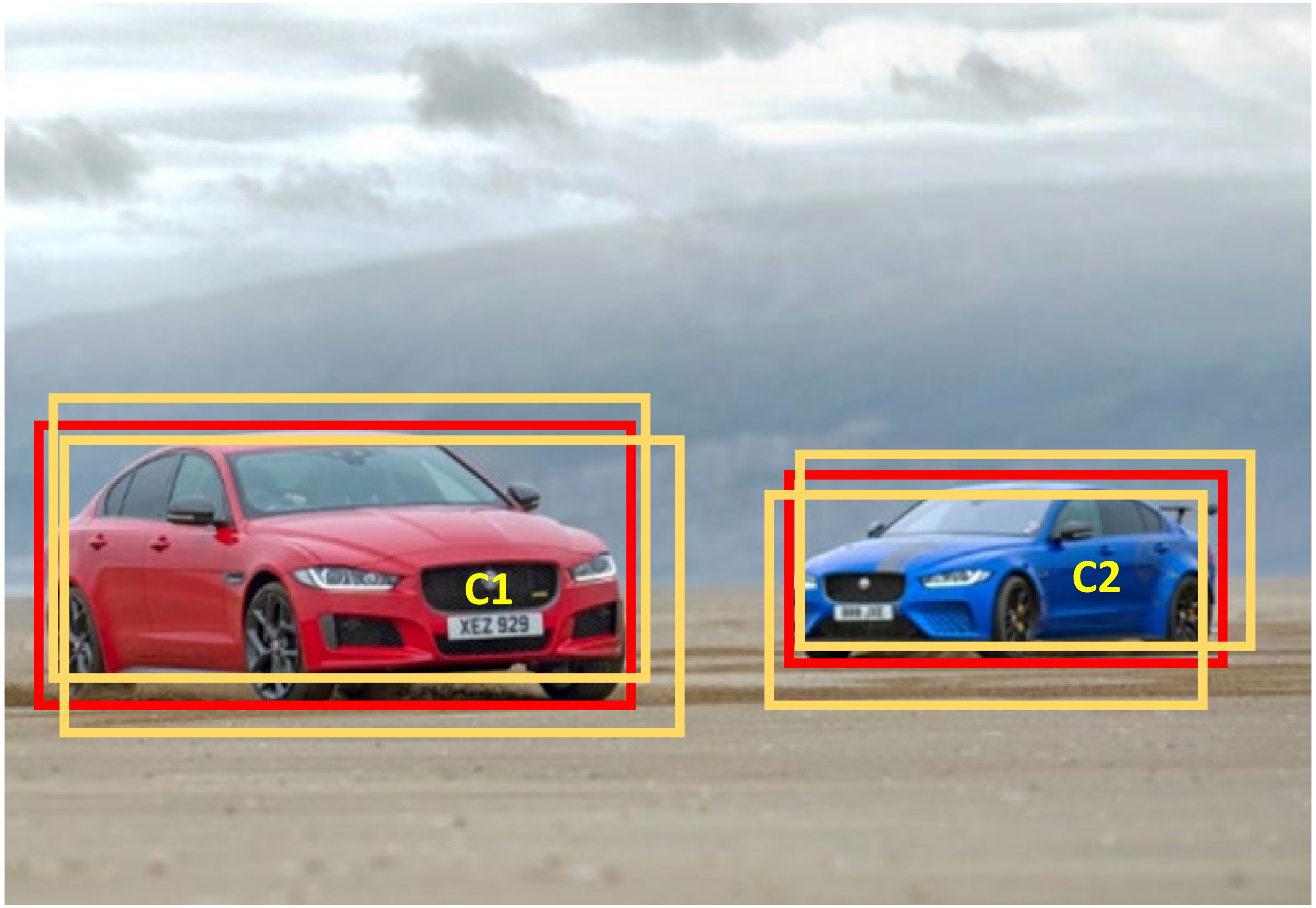
Box2: 0.85

Box3: 0.7

NMS cleans/removes the multiple detection and only keeps the one with very high confidence

Images source: https://arxiv.org/pdf/1807.05511.pdf

Non Maxima Suppression (NMS)

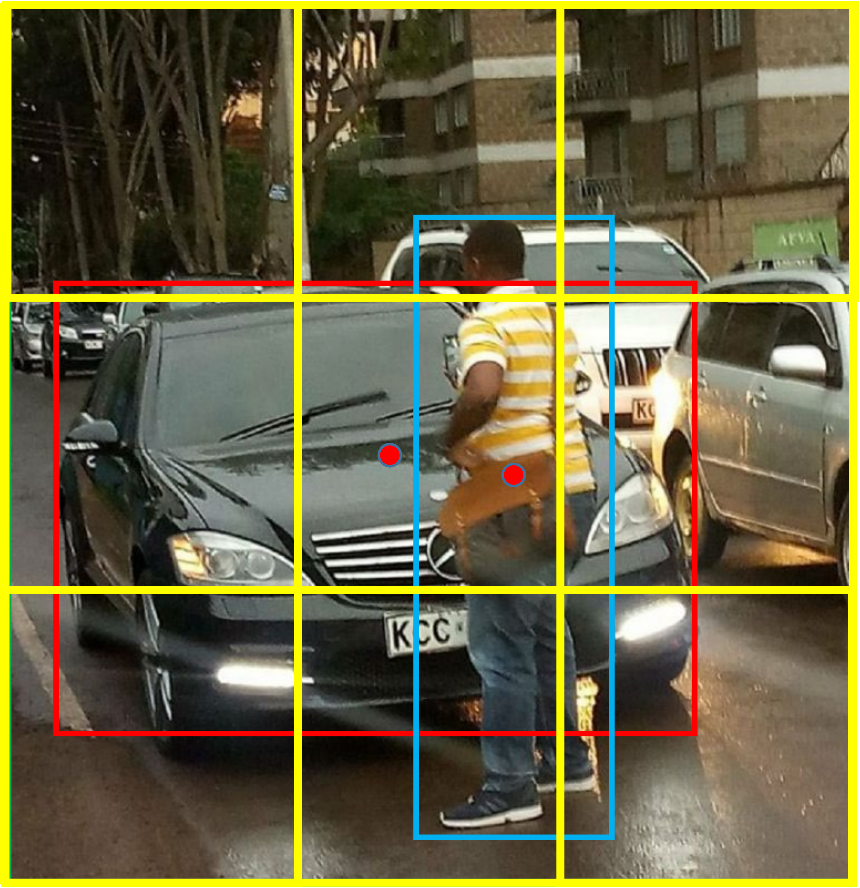


1. **Check the probabilities of each detection and keep ones with *score > Threshold (0.7)***

2. **For remaining boxes:**
   - Box with highest score is the detection results.
   - Discard any remaining boxes with IoU > 0.5 with final detected box,
   i.e: overlap with the box with highest score.

Images source: https://arxiv.org/pdf/1807.05511.pdf

YOLO: You Only Look Once Algorithm

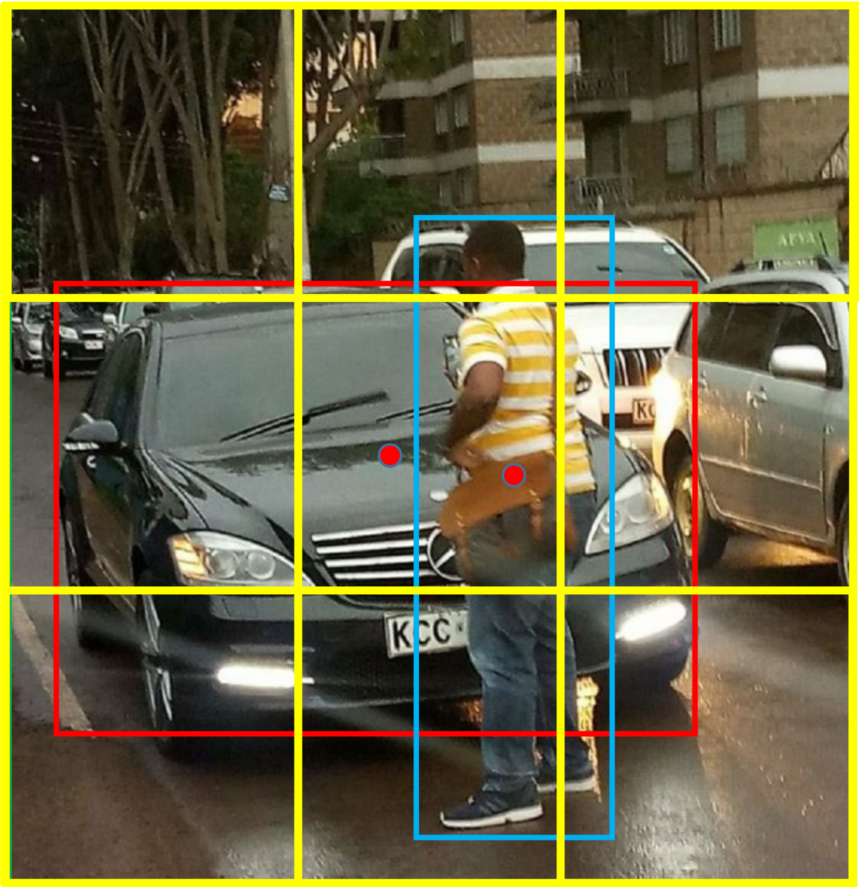


Challenges with overlapping objects

- Each grid cell detect only one object
- For multiple overlapping objects, Mid point are on the same grid cell

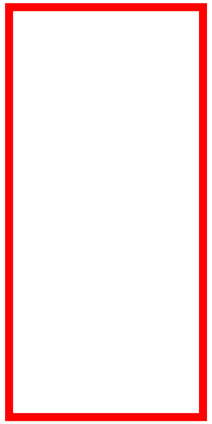
Images source: https://businesstoday.co.ke/brave-kenyan-blocks-mps-overlapping-car/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Anchor Boxes



So,   Currently the Target Y = {1, x, y, h, w, C1, C2},
As the mid-points for both the objects are on the same grid cell, only one of the objects will be associated

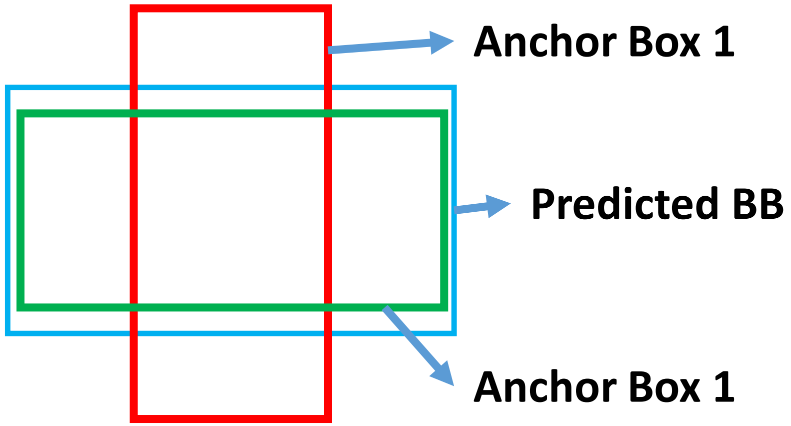
Anchor Box 1



Anchor Box 2

Images source: https://businesstoday.co.ke/brave-kenyan-blocks-mps-overlapping-car/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Anchor Boxes



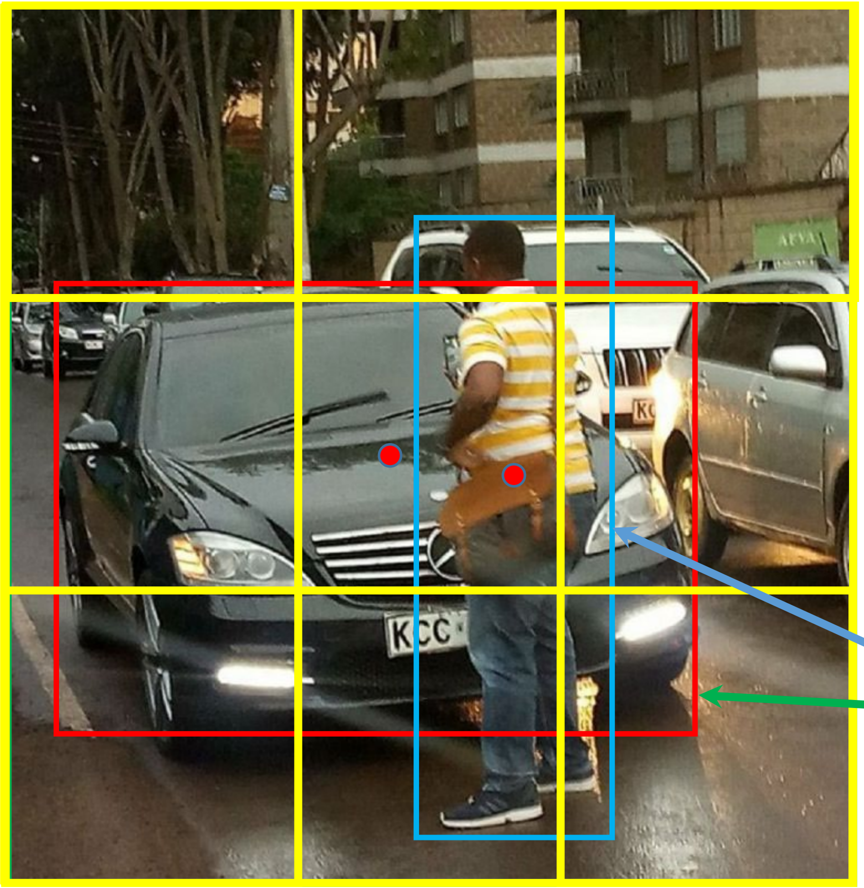
Associate each object to:

1. **A cell which contains its mid-point and**
2. **Anchor box for the cell with highest IoU**

Calculate the IoU of
Anchor boxes and predicted BB

Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

Anchor Boxes



Anchor Box 1

Anchor Box 2

Similar Shape

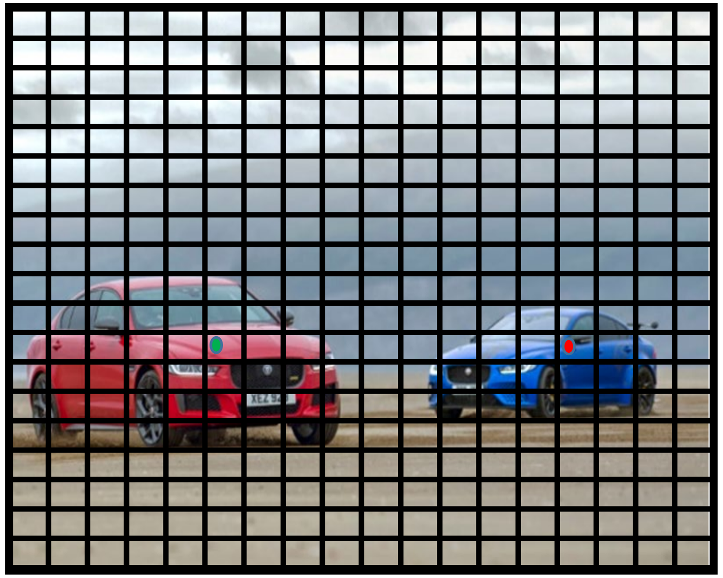
So, with Anchor boxes:

Target Y = $\{\underbrace{P_o, x, y, h, w, C1, C2}_{\text{Anchor Box 1}}, \underbrace{P_o, x, y, h, w, C1, C2}_{\text{Anchor Box 2}}\},$

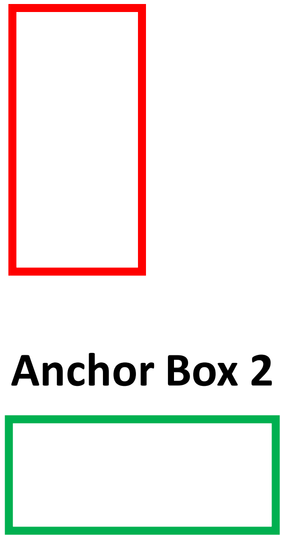
Images source: https://arxiv.org/pdf/1807.05511.pdf

YOLO: You Only Look Once Algorithm

Input X



Anchor Box 1



Anchor Box 2

Class : {1:car, 2:bike}

Y size : ($\boxed{19 \text{ X } 19}$ X 2 X 7)

Grid Size

#Anchor Box

5(P_o, x,y,h,w) + #Classes(2)

Training Set

Y = {P_o, x, y, h, w, C1, C2, P_o, x, y, h, w, C1, C2}

Cell(1,1) = {0, ?, ?, ?, ?, ?, ?, 0, ?, ?, ?, ?, ?, ?}

:

Cell(12,6)= {0, ?, ?, ?, ?, ?, ?, 1, x, y, h, w, 1, 0}

:

Cell(12,15)= {0, ?, ?, ?, ?, ?, ?, 1, x, y, h, w, 1, 0}

:

Cell(19,19)= {0, ?, ?, ?, ?, ?, ?, 0, ?, ?, ?, ?, ?, ?}

Images source: https://arxiv.org/pdf/1807.05511.pdf

YOLO: You Only Look Once Algorithm

Training:

Input: X

Target: Y

19 X 19 X 2 X 7

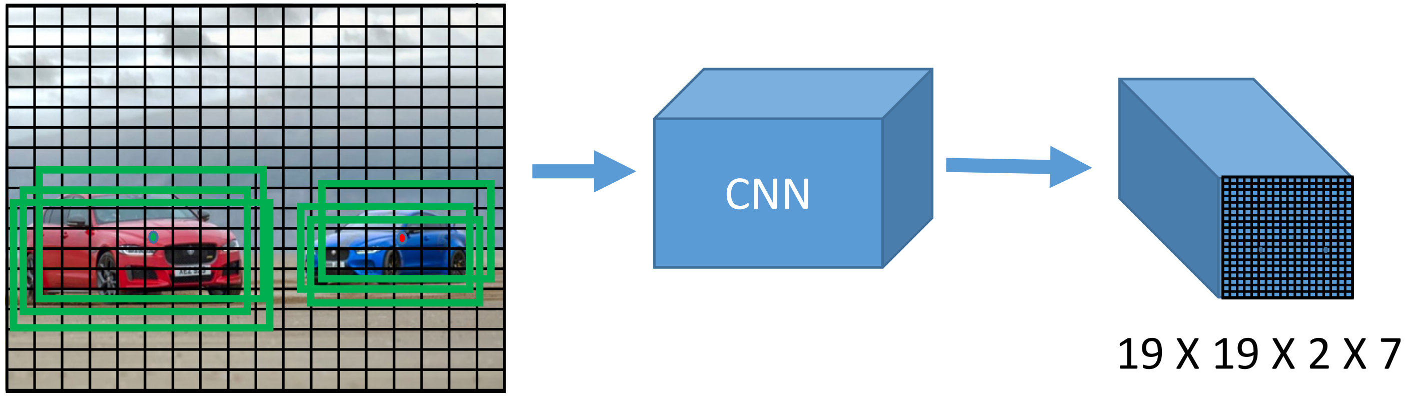
Class : {car, bike}

Images source: https://businesstoday.co.ke/brave-kenyan-blocks-mps-overlapping-car/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

YOLO: You Only Look Once Algorithm

Testing:

Input: X



Class : {car, bike}

Y = {P_o, x, y, h, w, C1, C2, P_o, x, y, h, w, C1, C2}

{0, ?, ?, ?, ?, ?, ?, 0, ?, ?, ?, ?, ?, ?}

:

{0, ?, ?, ?, ?, ?, ?, 1, x, y, h, w, 1, 0}

:

{0, ?, ?, ?, ?, ?, ?, 1, x, y, h, w, 1, 0}

:

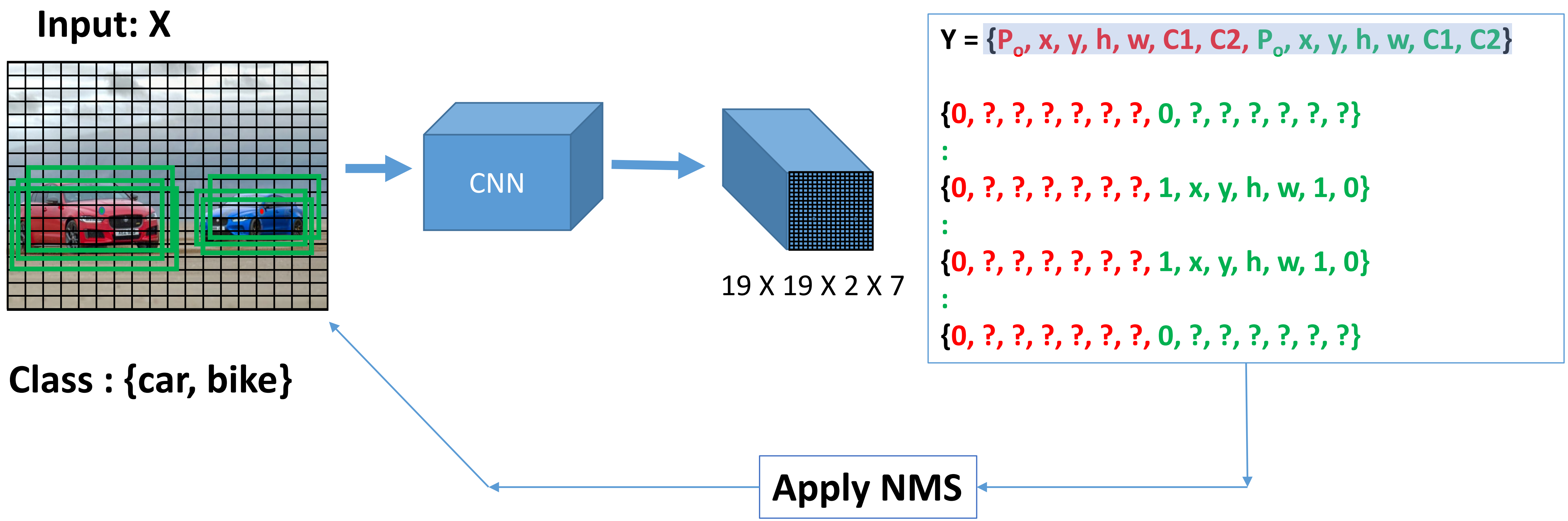
{0, ?, ?, ?, ?, ?, ?, 0, ?, ?, ?, ?, ?, ?}

Images source: https://businesstoday.co.ke/brave-kenyan-blocks-mps-overlapping-car/

Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

YOLO: You Only Look Once Algorithm

Testing:



Images source: https://businesstoday.co.ke/brave-kenyan-blocks-mps-overlapping-car/
Source and Reference: https://www.youtube.com/watch?v=gKreZOUi-O0

YOLO: You Only Look Once Algorithm

- Real-time performance with 45 frames per sec, 0.02 sec per image
- Not suitable for small objects
- Issues with new or multiple aspect ratios and unable to generalize

Source: https://arxiv.org/pdf/1807.05511.pdf

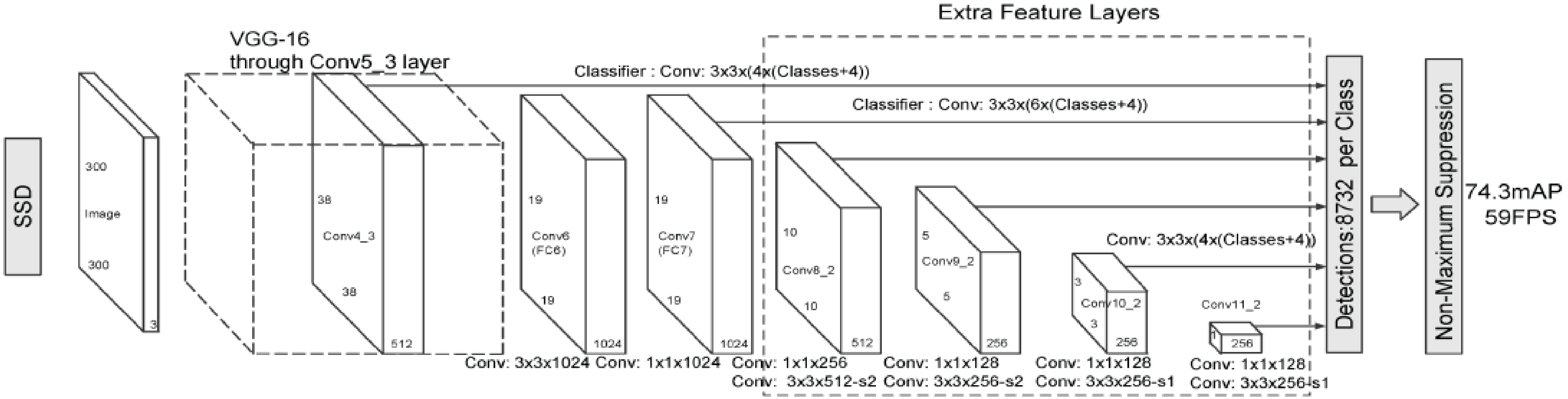
Single Shot Detector(SSD)

- Similar to YOLO
- VGG16 base CONV layers
- Take advantage of Anchor boxes with different aspect ratios
- Large number of anchors boxes are chosen
- Not suitable for small objects
- 3 times faster than Faster-RCNN
- With ResNet101 base SSD may be help in detecting small objects with better features from the CONV base

Source: https://arxiv.org/pdf/1807.05511.pdf

Single Shot Detector(SSD)

SSD300 architecture:



Source: https://arxiv.org/pdf/1807.05511.pdf

Object Detection State-of-the-Art

Dataset: PASCAL VOC 2007 and 2017

Test Dataset : PASCAL VOC 2007

Method	Train Dataset	mAP	Time in sec/image	Time Frame /sec
RCNN (VGG16)	Pascal VOC 2007	66.0	50	-
Fast RCNN	VOC 2007+2012	70.0	2	-
Faster RCNN (VGG16)	VOC 2007+2012	73.2	0.11	9
Faster RCNN (ResNet101)	VOC 2007+2012	**83.8**	2.24	0.4
Yolo	VOC 2007+2012	63.4	0.02	45
SSD300	VOC 2007+2012	74.3	0.02	45
SSD512	VOC 2007+2012	**76.8**	0.05	19

Source: https://arxiv.org/pdf/1807.05511.pdf

Object Detection Summary

Base Networks:

- VGG16
- REsNet101
- Inception V2
- Inception V3
- ResNet
- MobileNet
- Alexnet
- ZFNet

Etc.

Object Detection FrameWorks:

- RCNN Family (RCNN, Fast/Faster RCNN)
- Yolo
- SSD
- F-RCN

Summary:

- Faster-RCNN is more accurate but slower
- Yolo/SSD are faster/real-time but not much accurate

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf