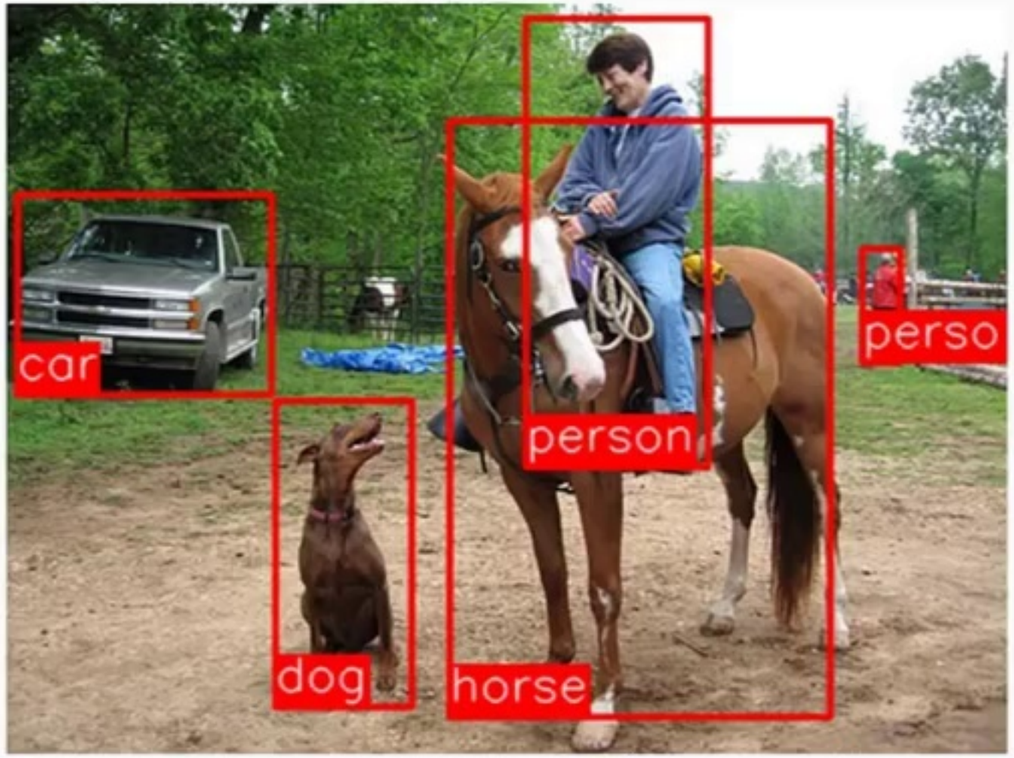
Object Detection – The Holy Grail of Computer Vision

- Classification of images is great and has wide ranging applications
- But can it do this



Object Detection – The Holy Grail of Computer Vision

- It is a mix of Object Classification and Localization

Object Detection – The Holy Grail of Computer Vision

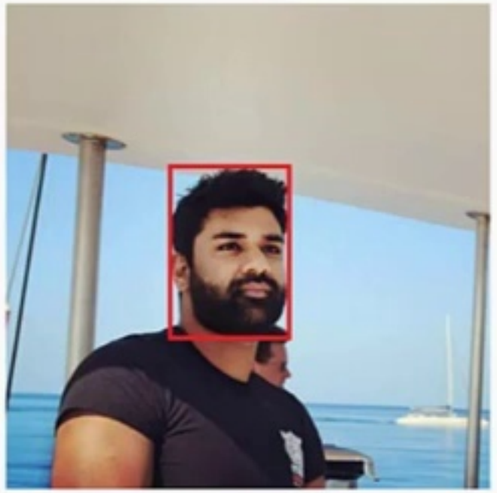
- It is a mix of Object Classification and Localization
- Object Detection is the identification of a bounding box, outlining the object we wish to detect

Object Detection – The Holy Grail of Computer Vision

- It is a mix of Object Classification and Localization
- Object Detection is the identification of a bounding box, outlining the object we wish to detect
- Face Detection as seen in almost all camera apps is a perfect example of object detection

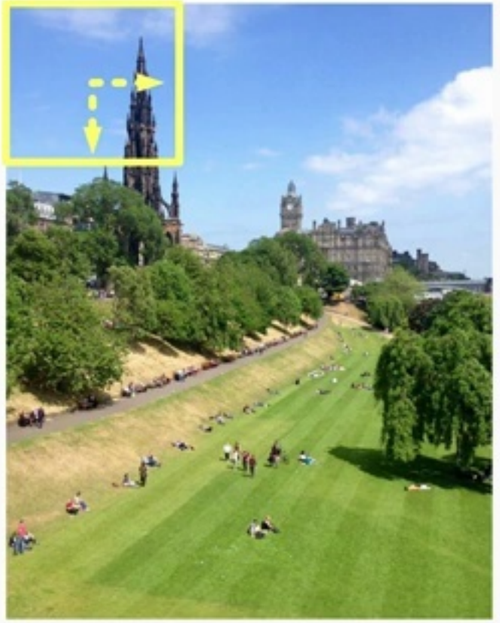
Object Detection – The Holy Grail of Computer Vision

- It is a mix of Object Classification and Localization
- Object Detection is the identification of a bounding box, outlining the object we wish to detect
- Face Detection as seen in almost all camera apps is a perfect example of object detection
- Instead of stating whether an image is a cat or not, Object Detection answers the question, “Where is the cat?”



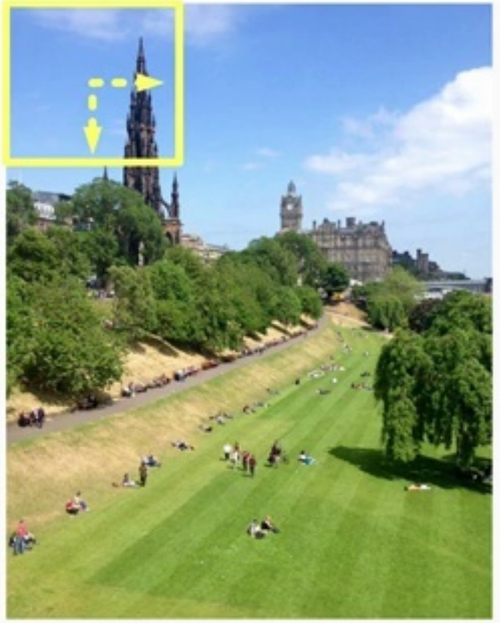
Non-Deep Learning Object Detection

- Before deep learning was used in Object Detection, the following was widely used:
    - Haar Cascade Classifiers



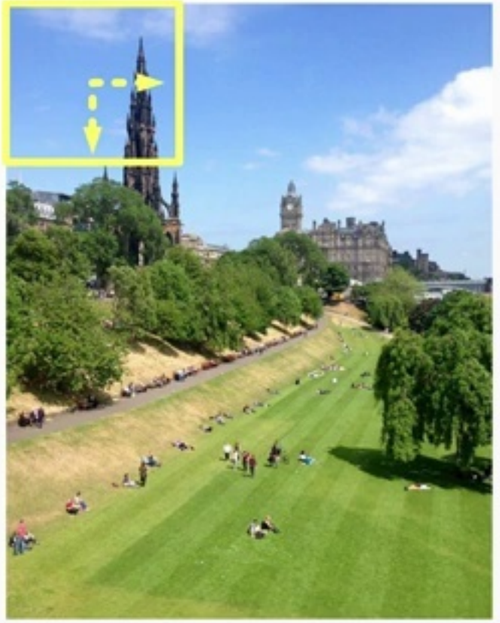
Non-Deep Learning Object Detection

- Before deep learning was used in Object Detection, the following was widely used:
  - Haar Cascade Classifiers
  - Histogram of Gradients (HOG) with Linear SVM



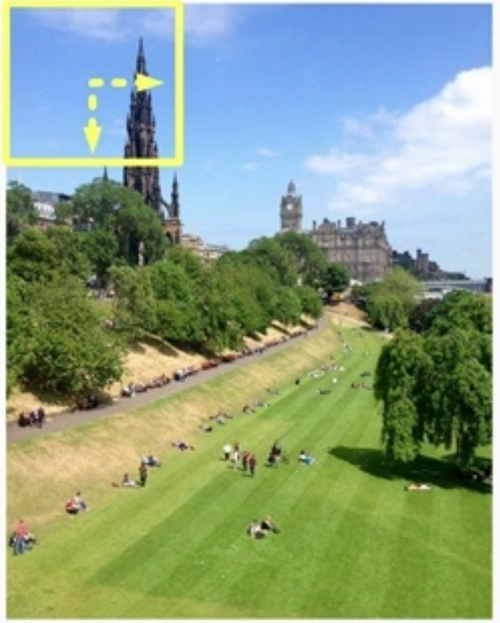
Non-Deep Learning Object Detection

- Before deep learning was used in Object Detection, the following was widely used:
    - Haar Cascade Classifiers
    - Histogram of Gradients (HOG) with Linear SVM
- They utilized slide window approaches where we extracted features and used classifiers such as Linear SVM to 'vote' for detections:
    - Using a sliding window of across an image results in hundreds, even thousands of classifications done for a single image



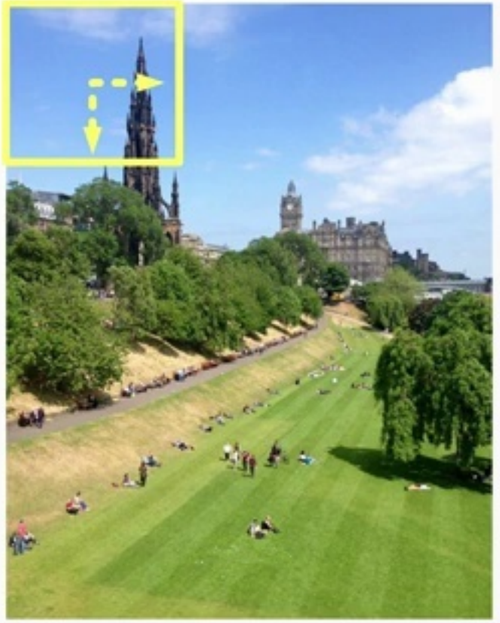
Non-Deep Learning Object Detection (Continued)

- They utilized slide window approaches where we extracted features and used classifiers such as Linear SVM to ‘vote’ for detections:
    - Scaling issues – what size window do we use
    - Is pyramiding robust enough
    - What if the window is not defined as we set it out to be
    - Do we use windows of different ratios



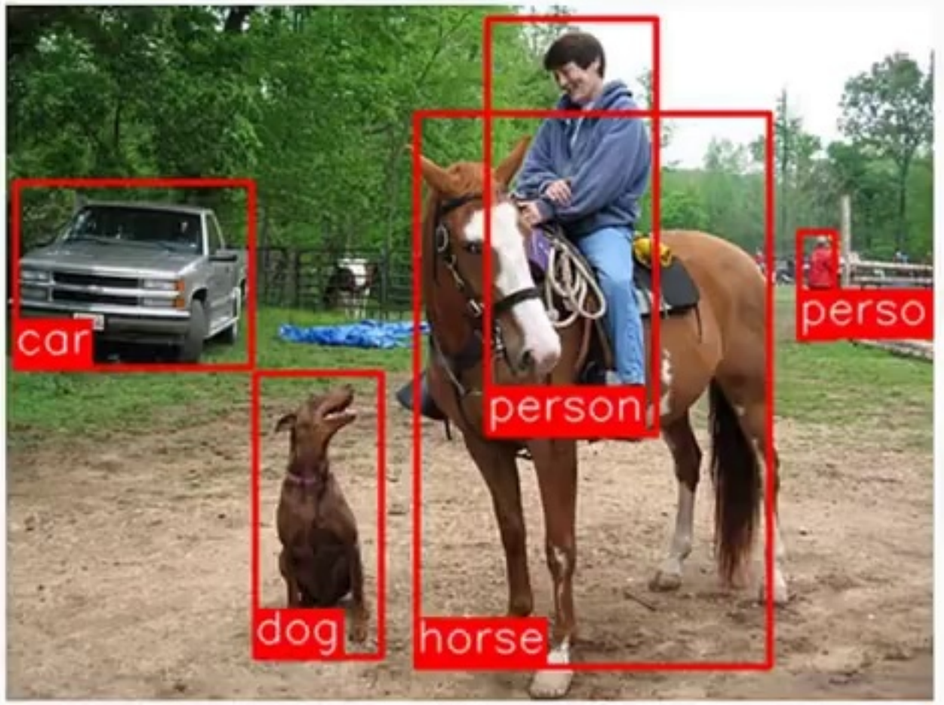
Non-Deep Learning Object Detection (Continued)

- They utilized slide window approaches where we extracted features and used classifiers such as Linear SVM to ‘vote’ for detections:
    - You can easily see how this can blow up into millions of classifications needed for a large image



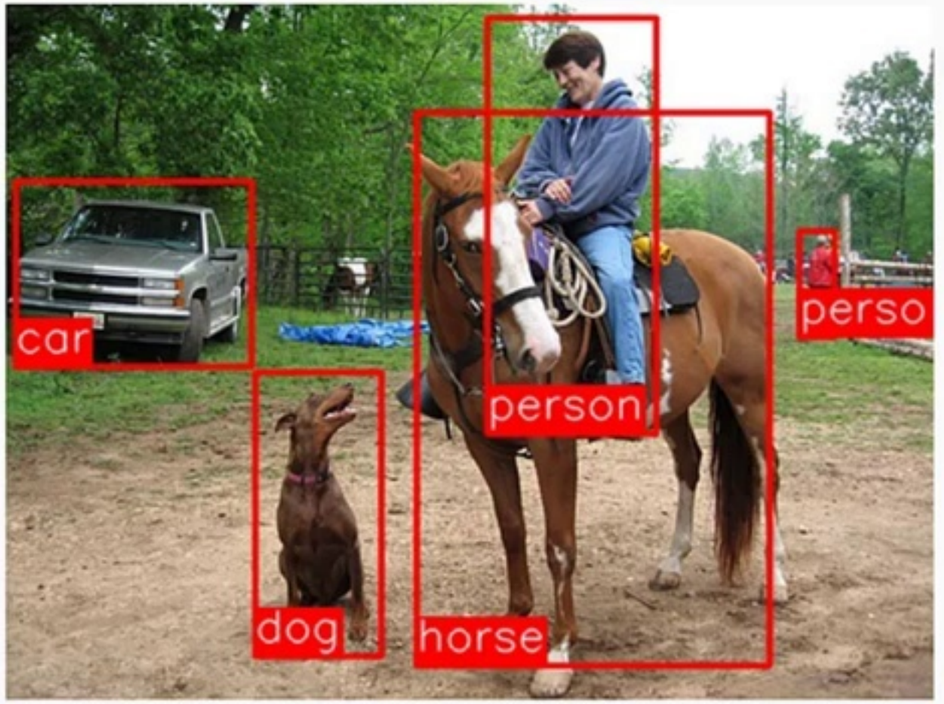
Deep Learning Object Detectors – Regions with CNNs or R-CNNs

- Introduced in 2014 by researchers at University College of Berkeley



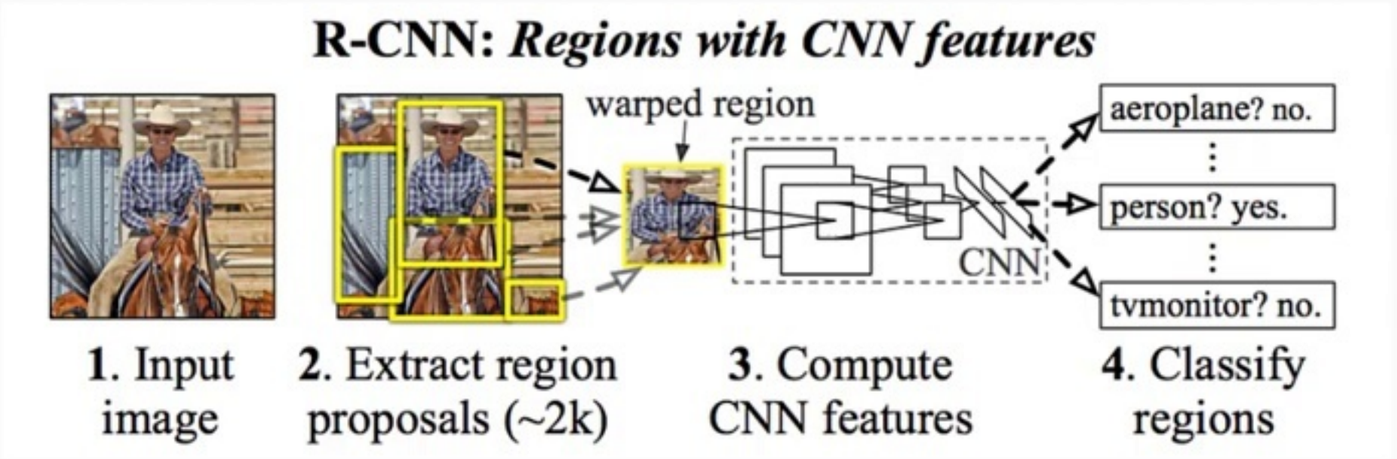
Deep Learning Object Detectors – Regions with CNNs or R-CNNs

- Introduced in 2014 by researchers at University College of Berkeley
- R-CNNs obtained dramatically higher performance in the PASCAL VOC Challenge (ImageNet for Object Detection testing)
- https://arxiv.org/abs/1311.2524



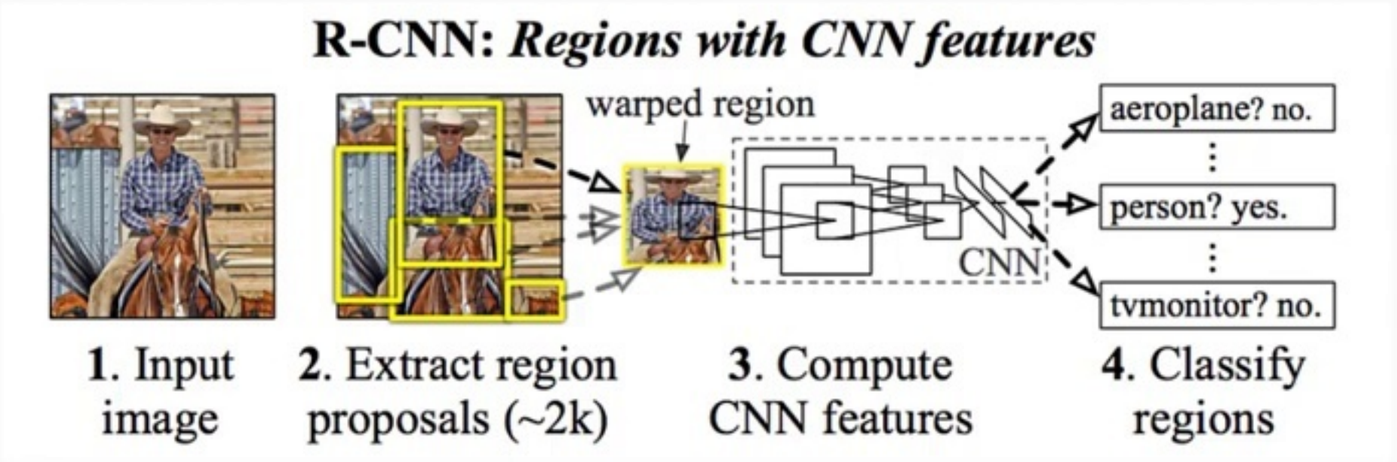
R-CNNs Principles

- R-CNNs attempted to solve the exhaustive search previously performed by sliding windows, by proposing bounding boxes, and passing these extracted boxes to an image classifier



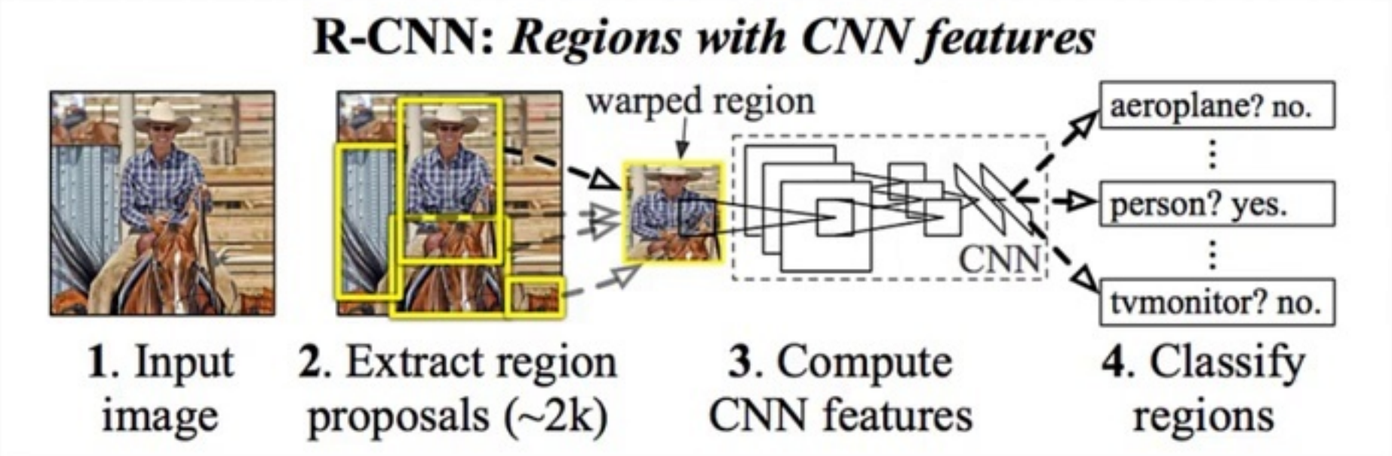
R-CNNs Principles

- R-CNNs attempted to solve the exhaustive search previously performed by sliding windows, by proposing bounding boxes, and passing these extracted boxes to an image classifier
- How do we do these bounding box proposals?



R-CNNs Principles

- R-CNNs attempted to solve the exhaustive search previously performed by sliding windows, by proposing bounding boxes, and passing these extracted boxes to an image classifier
- How do we do these bounding box proposals?
- By using the Selective Search algorithm:



Selective Search

- Selective Search attempts to segment the image into groups, by combining similar areas such as colors/textures and propose these regions as “interesting” bounding boxes

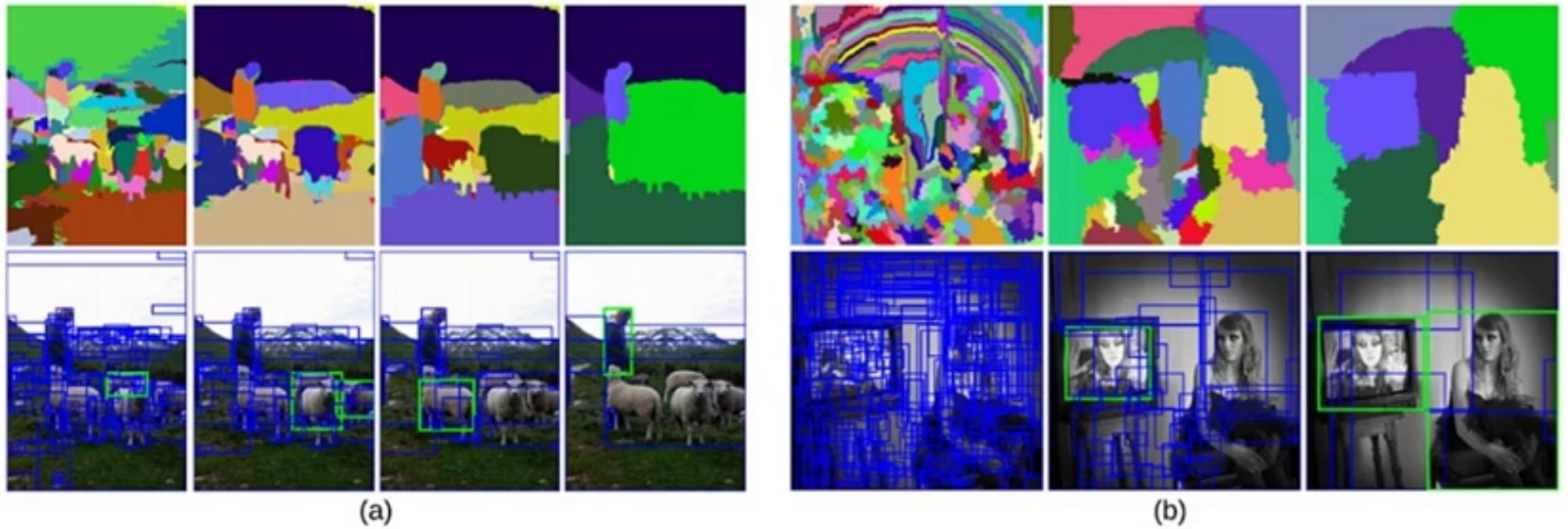
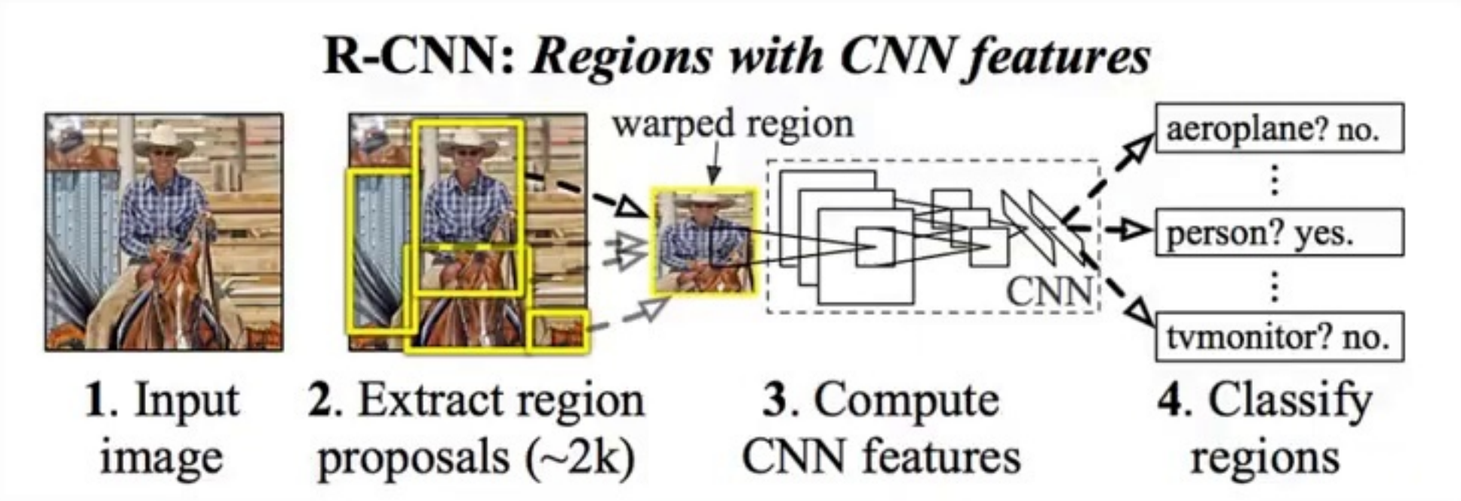


Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

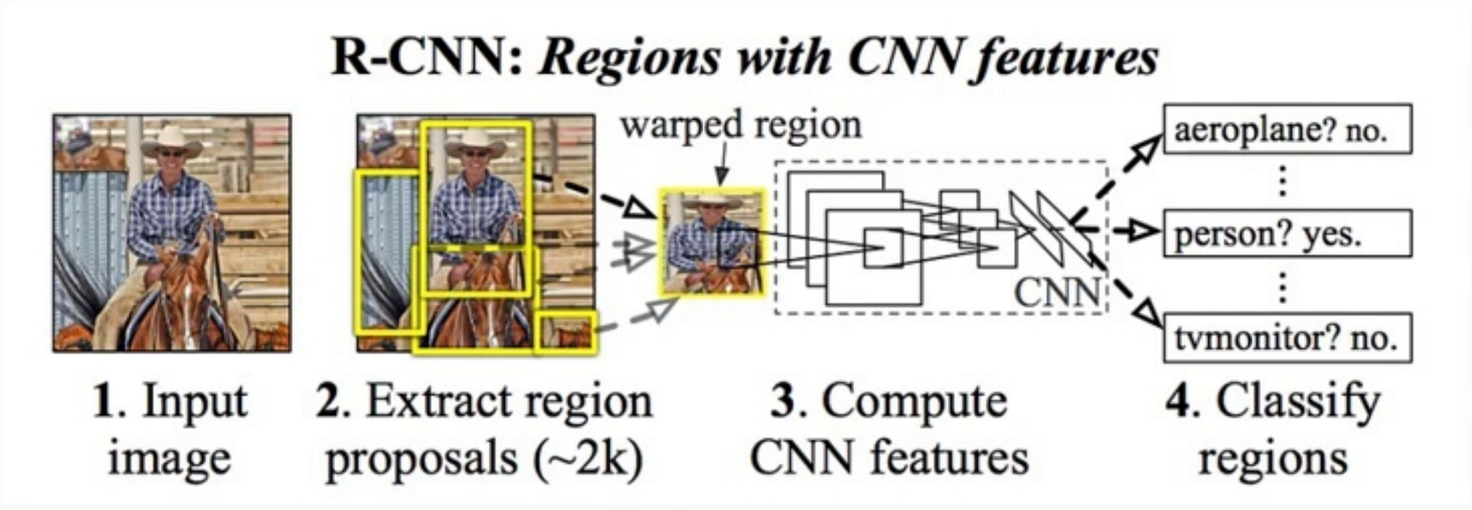
Selective Search

- When selective search has identified these regions/boxes, it passes this extracted image to our CNN (for example, one trained on ImageNet) for classification



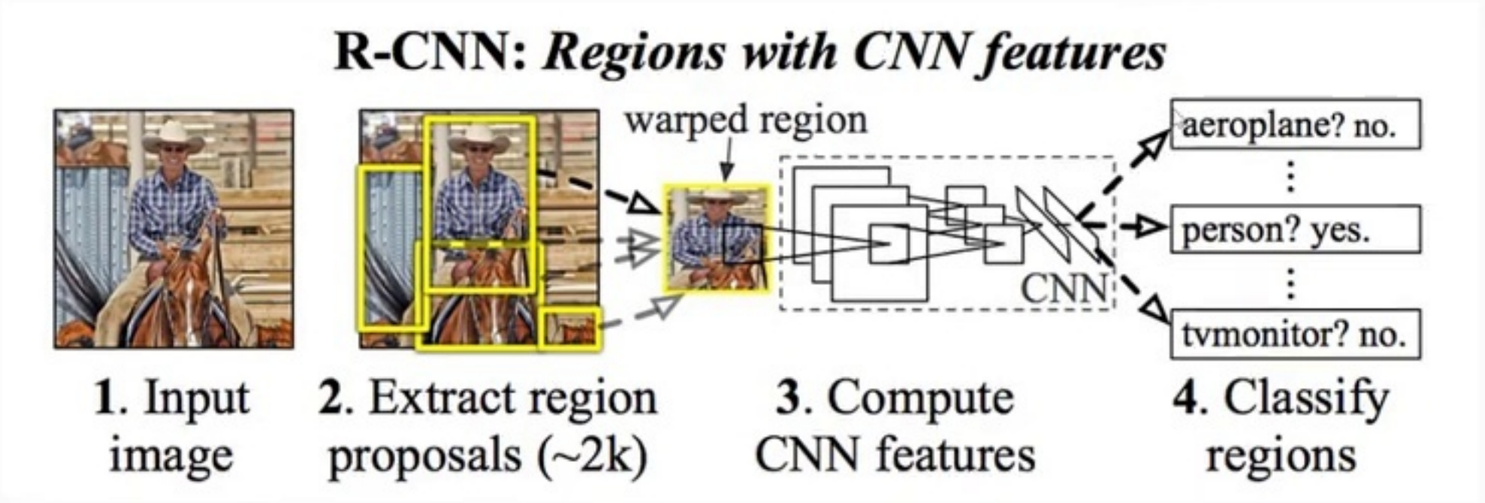
Selective Search

- When selective search has identified these regions/boxes, it passes this extracted image to our CNN (for example, one trained on ImageNet) for classification
- We do not use the CNN directly for classification; we use an SVM to classify the CNN extracted features

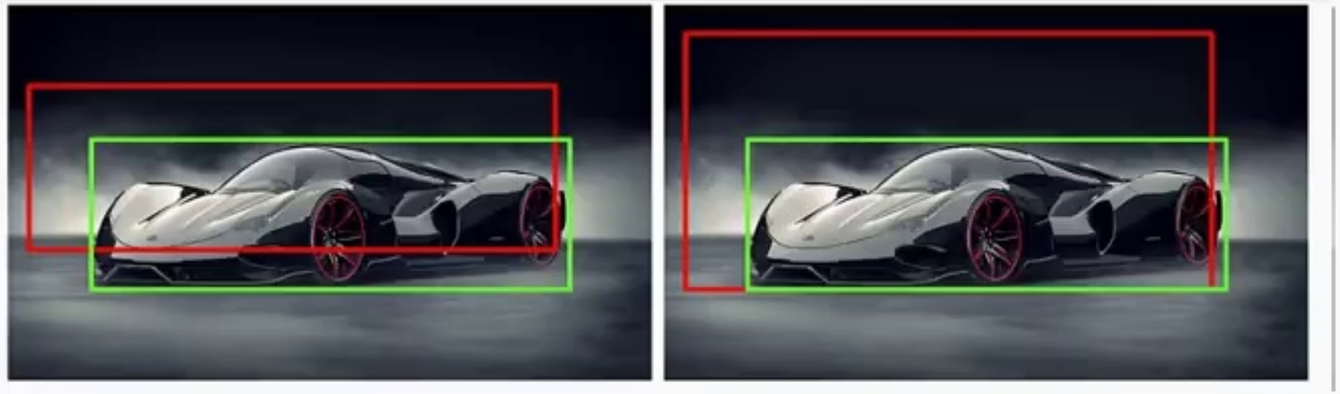


R-CNN – Tightening of Boxes

- After the first region proposal has been classified, we then use a simple linear regression to generate a tighter bounding box
- But how do we know what a good box is



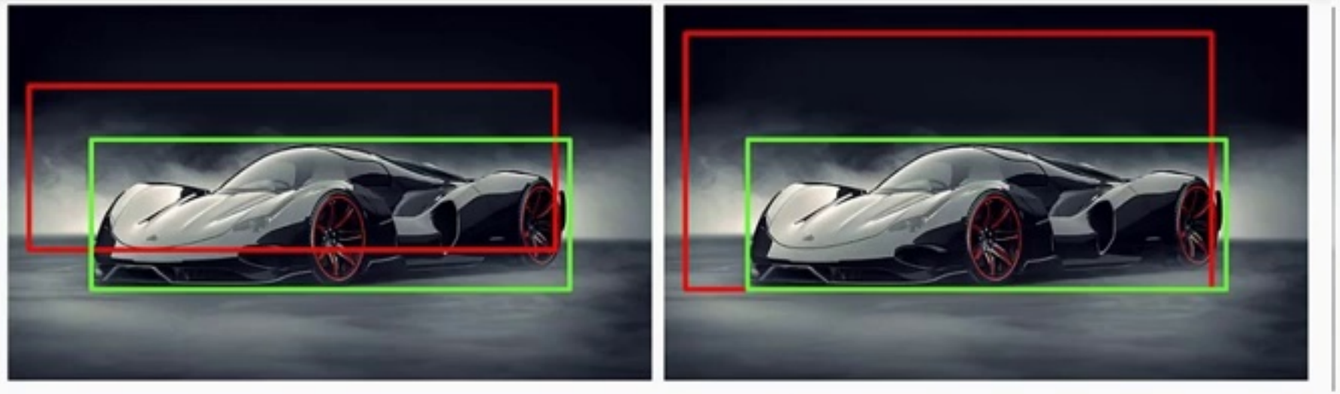
The Intersection over Union (IOU) Metric



- **Green** is our **true** bounding box
- **Red** is our **predicted** bounding box
- Honestly it seems close to 80%. However, is this really a good metric?
- In the second image, the predicted box still covers 95% of our original ground truth.

The Intersection over Union (IOU) Metric

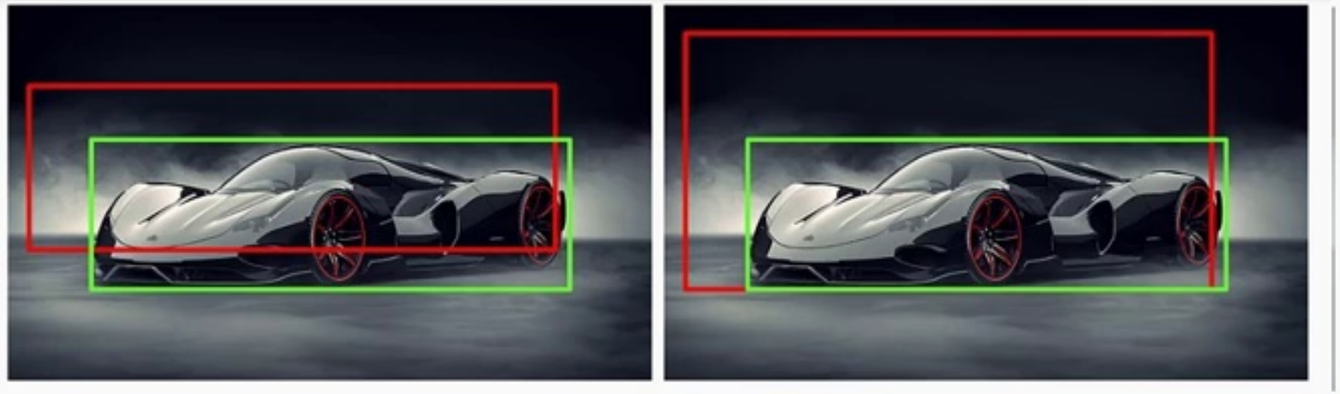
- $$\text{IoU} = \frac{\text{Size of Union}}{\text{Size of Prediction Box}}$$



- **Green** is our **true** bounding box
- **Red** is our **predicted** bounding box
- Honestly it seems close to 80%. However, is this really a good metric?
- In the second image, the predicted box still covers 95% of our original ground truth.

The Intersection over Union (IOU) Metric

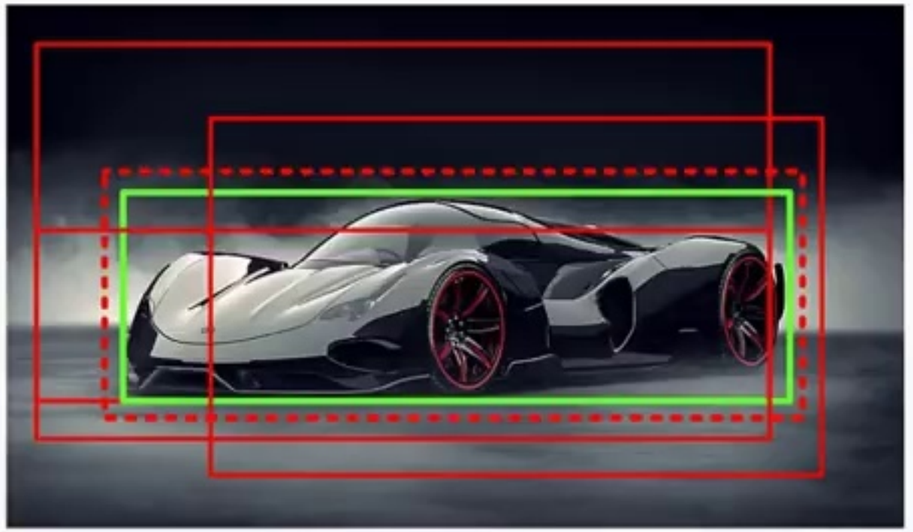
- $\text{IoU} = \frac{\text{Size of Union}}{\text{Size of Prediction Box}}$
- Typically an IoU over 0.5 is considered acceptable
- The higher the IoU, the better the prediction
- IoU is essentially a measure of overlap



- **Green** is our **true** bounding box
- **Red** is our **predicted** bounding box
- Honestly it seems close to 80%. However, is this really a good metric?
- In the second image, the predicted box still covers 95% of our original ground truth.

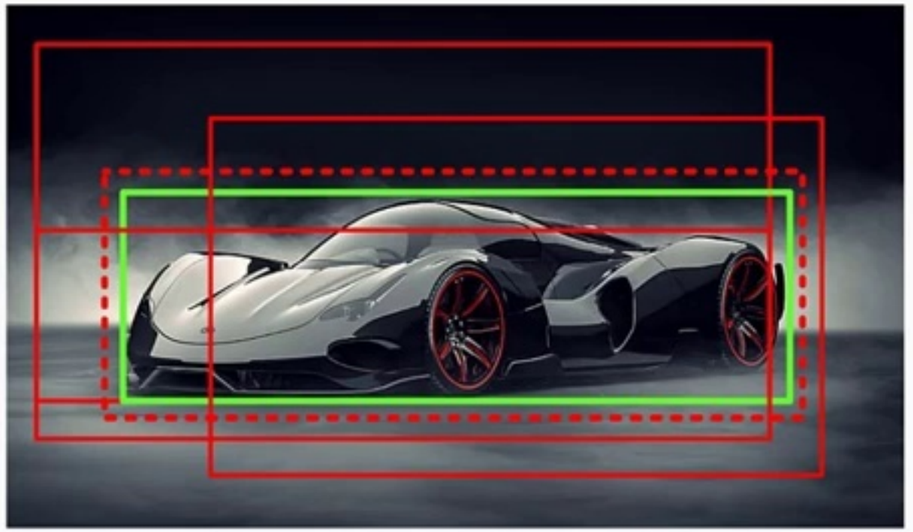
Introducing Mean Average Precision (mAP)

- mAP is a common metric used to measure the accuracy of Object Detectors
- An IoU of 0.5 isn't too bad, but how do we decide what's best when we have multiple boxes predicted for the same object (see right)
- This is a common problem with Object Detectors, when they are overlapping boxes on the same object



Introducing Mean Average Precision (mAP) (Continued)

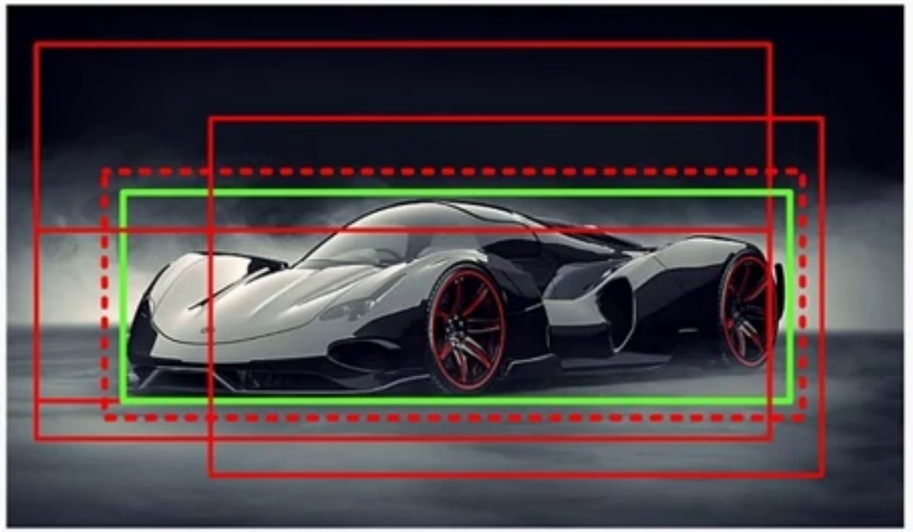
- In the figure below, we have four predicted boxes (in red) with one being the best (dotted red line)
- As such, we have one True-Positive and three False-Positives



Introducing Mean Average Precision (mAP) (Continued)

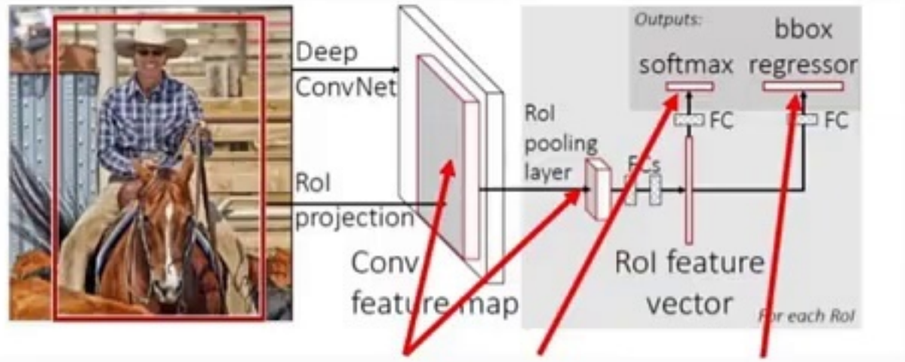
$$Ave\ Precision_{Class} = \frac{\sum True\ Postives_{Class}}{\sum True\ Postives_{Class} + \sum False\ Postives_{Class}}$$

$$mAP = \frac{1}{Number\ of\ classes} \sum_{All\ Classes} \frac{\sum True\ Postives_{Class}}{\sum True\ Postives_{Class} + \sum False\ Postives_{Class}}$$



Introducing Fast R-CNN – 2015

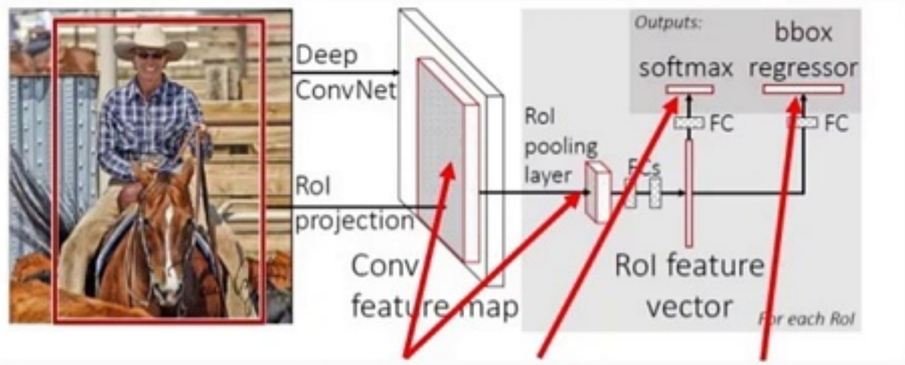
- R-CNN was effective but extremely slow as each proposed bounding box had to be classified by our CNN, as a result of which doing real-time Object Detection was far from possible



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015

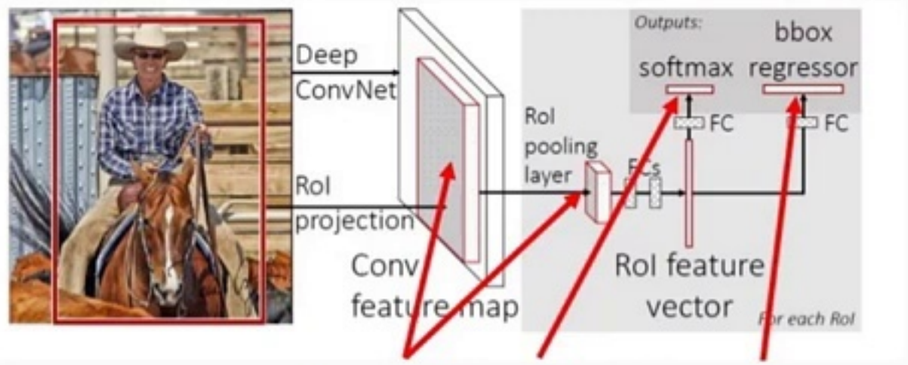
- R-CNN was effective but extremely slow as each proposed bounding box had to be classified by our CNN, as a result of which doing real-time Object Detection was far from possible
- It required 3 models be trained separately:



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015

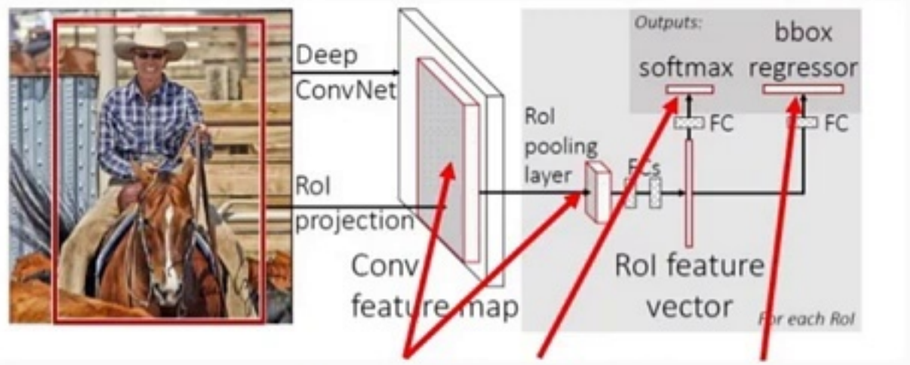
- R-CNN was effective but extremely slow as each proposed bounding box had to be classified by our CNN, as a result of which doing real-time Object Detection was far from possible
- It required 3 models be trained separately:
    - Feature extraction CNN



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015

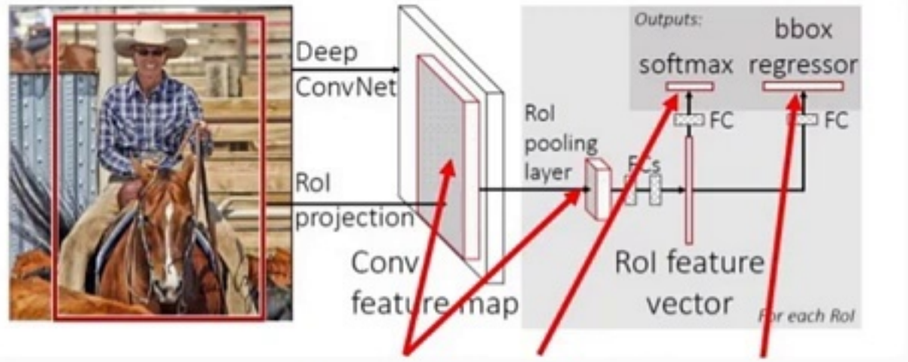
- R-CNN was effective but extremely slow as each proposed bounding box had to be classified by our CNN, as a result of which doing real-time Object Detection was far from possible
- It required 3 models be trained separately:
    - Feature extraction CNN
    - SVM to predict the class



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015

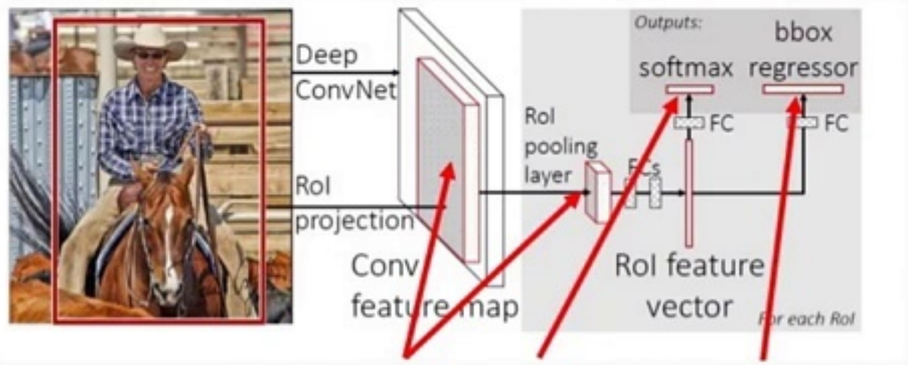
- R-CNN was effective but extremely slow as each proposed bounding box had to be classified by our CNN, as a result of which doing real-time Object Detection was far from possible
- It required 3 models be trained separately:
    - Feature extraction CNN
    - SVM to predict the class
    - Linear Regression model to tighten the bounding box



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015 (Continued)

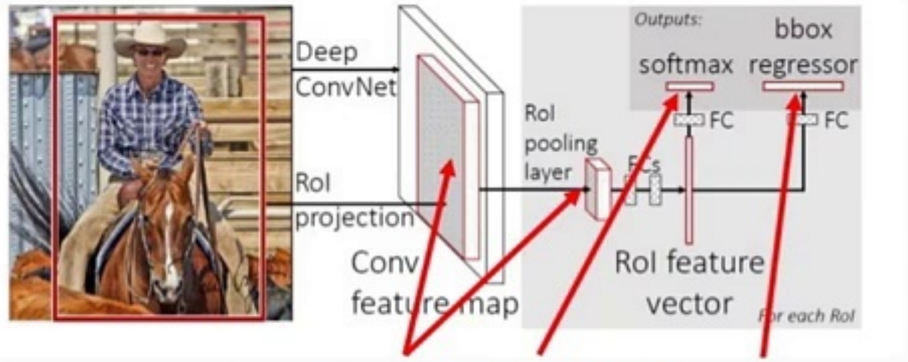
- Fast R-CNNs solve this problem by removing the overlap generated



https://arxiv.org/abs/1504.08083

Introducing Fast R-CNN – 2015 (Continued)

- Fast R-CNNs solve this problem by removing the overlap generated
- How:
    - We run the CNN across the image just once, using a technique called Region of Interest Pooling (RoIPool)



https://arxiv.org/abs/1504.08083

Faster R-CNN – 2016

- Fast R-CNNs made significant speed increases, however, region proposal remained relatively slow, as it still relied on Selective Search

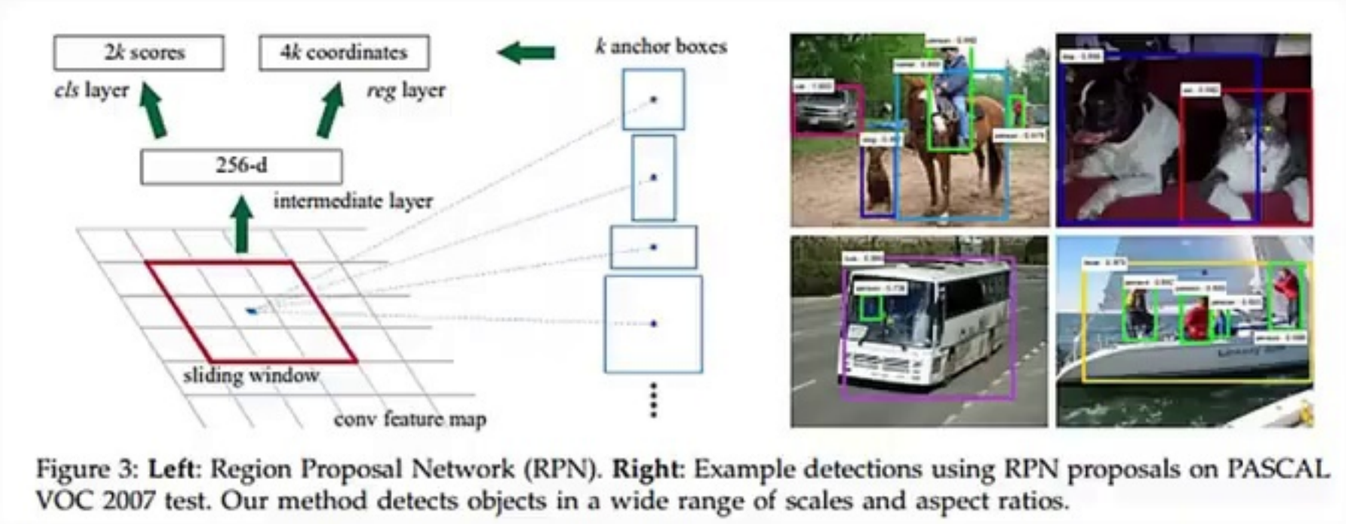


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

Faster R-CNN – 2016

- Fast R-CNNs made significant speed increases, however, region proposal remained relatively slow, as it still relied on Selective Search
- Fortunately, a Microsoft research team figured out how to eliminate this bottleneck using Selective Search, to speed up Region Proposal
- https://arxiv.org/abs/1506.01497

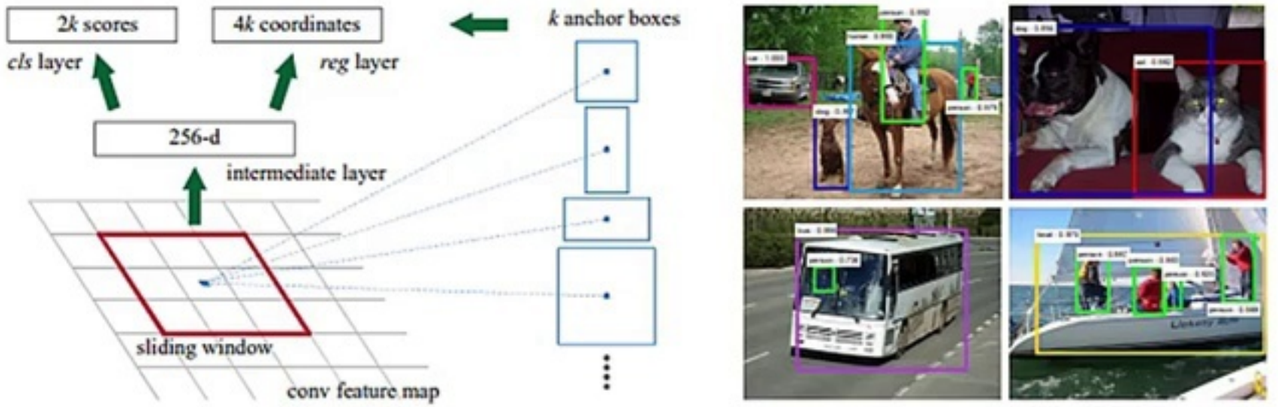


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

Single Shot Detectors (SSDs)

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Single Shot Detectors (SSDs)

- We have just discussed the R-CNN family and seen how successful they can be

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Single Shot Detectors (SSDs)

- We have just discussed the R-CNN family and seen how successful they can be
- However, their performance on video is still not optimal, running typically at 7 fps

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Single Shot Detectors (SSDs)

- We have just discussed the R-CNN family and seen how successful they can be
- However, their performance on video is still not optimal, running typically at 7 fps
- SSDs aim to improve this speed by eliminating the need for the Region Proposal Network
- https://arxiv.org/pdf/1512.02325.pdf

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

How do SSDs Improve Speed

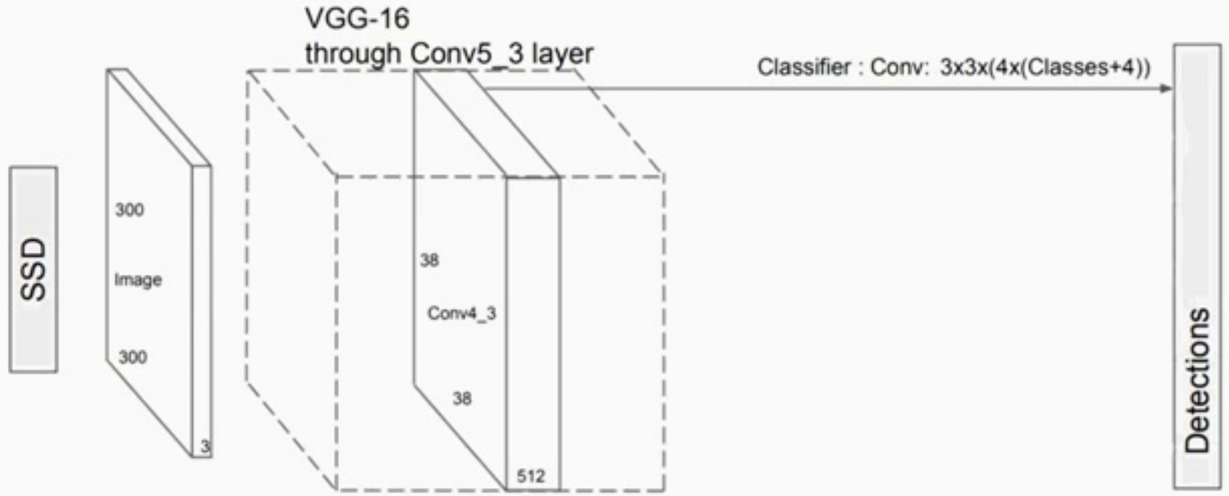
- SSDs use multi-scale features and default boxes, as well as, dropping the resolution images to improve speed

How do SSDs Improve Speed

- SSDs use multi-scale features and default boxes, as well as, dropping the resolution images to improve speed
- This allows SSDs to achieve real-time speed with almost no drop (sometimes even improved) accuracy

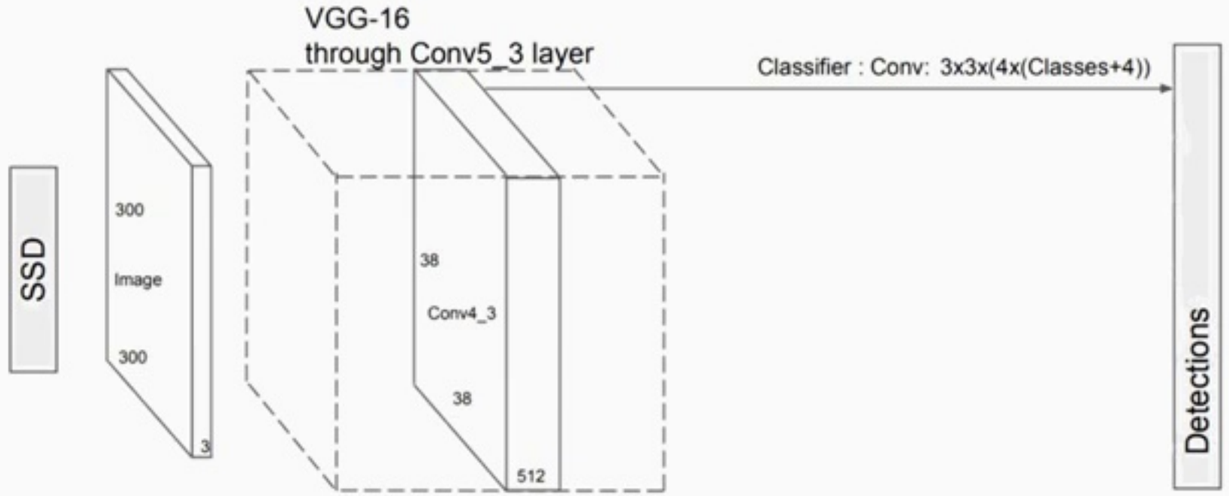
SSD Structure

- SSDs are composed of two main parts:
    - Feature Map Extractor (VGG16 was used in the published paper, but ResNet or DenseNet may provide better results)



SSD Structure

- SSDs are composed of two main parts:
    - Feature Map Extractor (VGG16 was used in the published paper, but ResNet or DenseNet may provide better results)
    - Convolution Filter for Object Detection
- https://arxiv.org/pdf/1512.02325.pdf

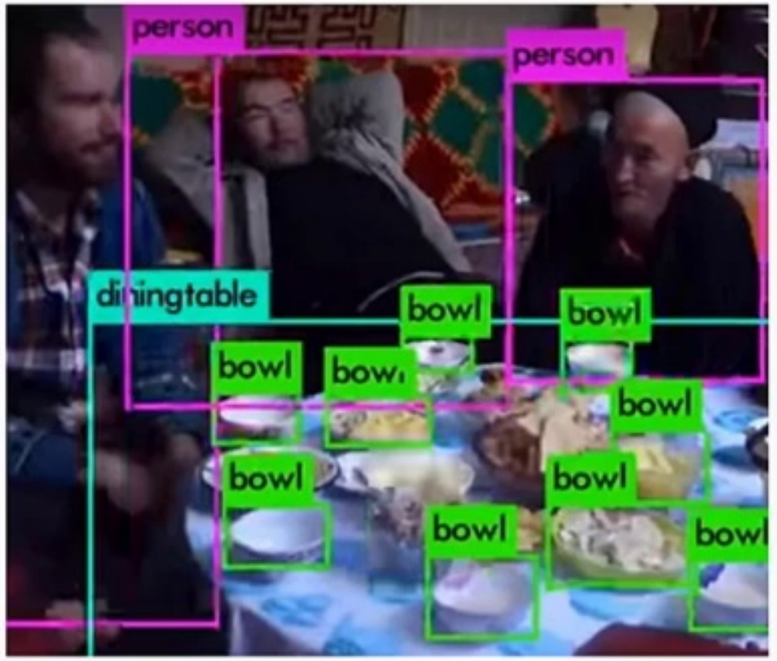


SSD Summary

- SSDs are faster than R-CNN, but less accurate in detecting small objects
- Accuracy increases if we increase the number of default boxes, as well as, have better designed boxes
- Multi-scale feature maps improve detection at varying scales

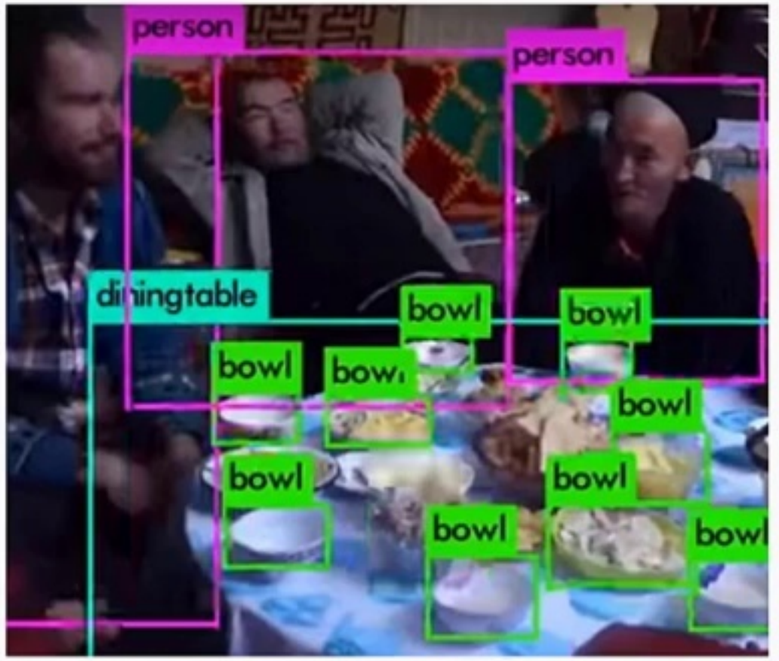
Introducing YOLO – YOLO or You Only Look Once

- The idea behind YOLO is that a single neural network is applied to full image



Introducing YOLO – YOLO or You Only Look Once

- The idea behind YOLO is that a single neural network is applied to full image
- This allows YOLO to reason globally about the image when generating predictions



Introducing YOLO – YOLO or You Only Look Once

- The idea behind YOLO is that a single neural network is applied to full image
- This allows YOLO to reason globally about the image when generating predictions
- It is a direct development of MultiBox; but it turns MultiBox from region proposal into an objection recognition method, by adding a softmax layer in parallel with a box regressor and box classifier layer

Introducing YOLO – YOLO or You Only Look Once (Continued)

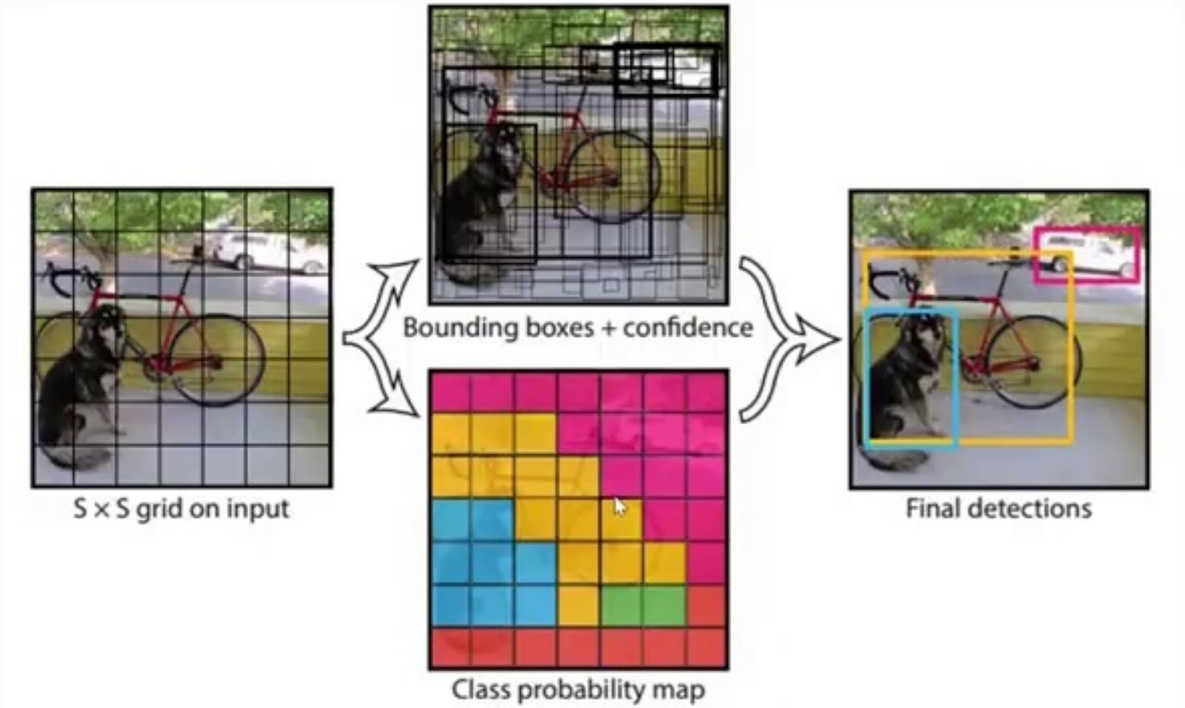
- It divides the image into regions, and predicts bounding boxes and probabilities for each region

Introducing YOLO – YOLO or You Only Look Once (Continued)

- It divides the image into regions, and predicts bounding boxes and probabilities for each region
- YOLO uses a Fully Convolutional Neural Network allowing for input of various image sizes
- https://arxiv.org/abs/1506.02640

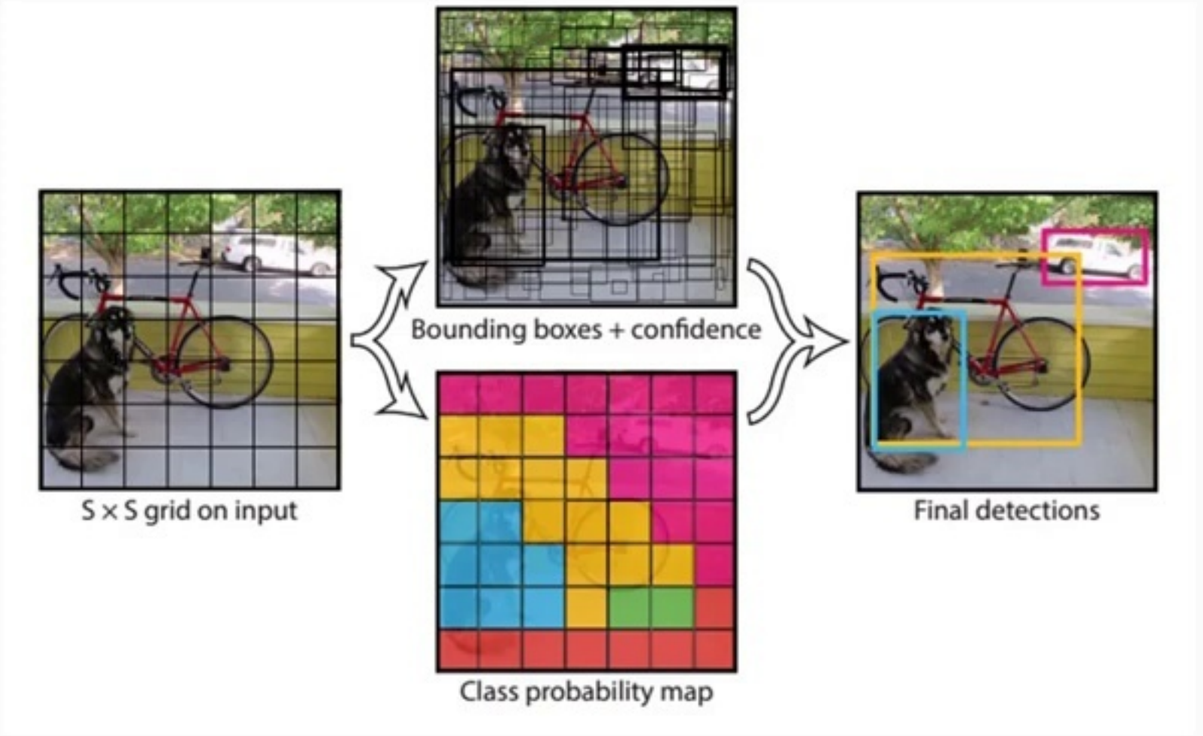
YOLO – How Does it Work

- The input image is divided into an S x S grid



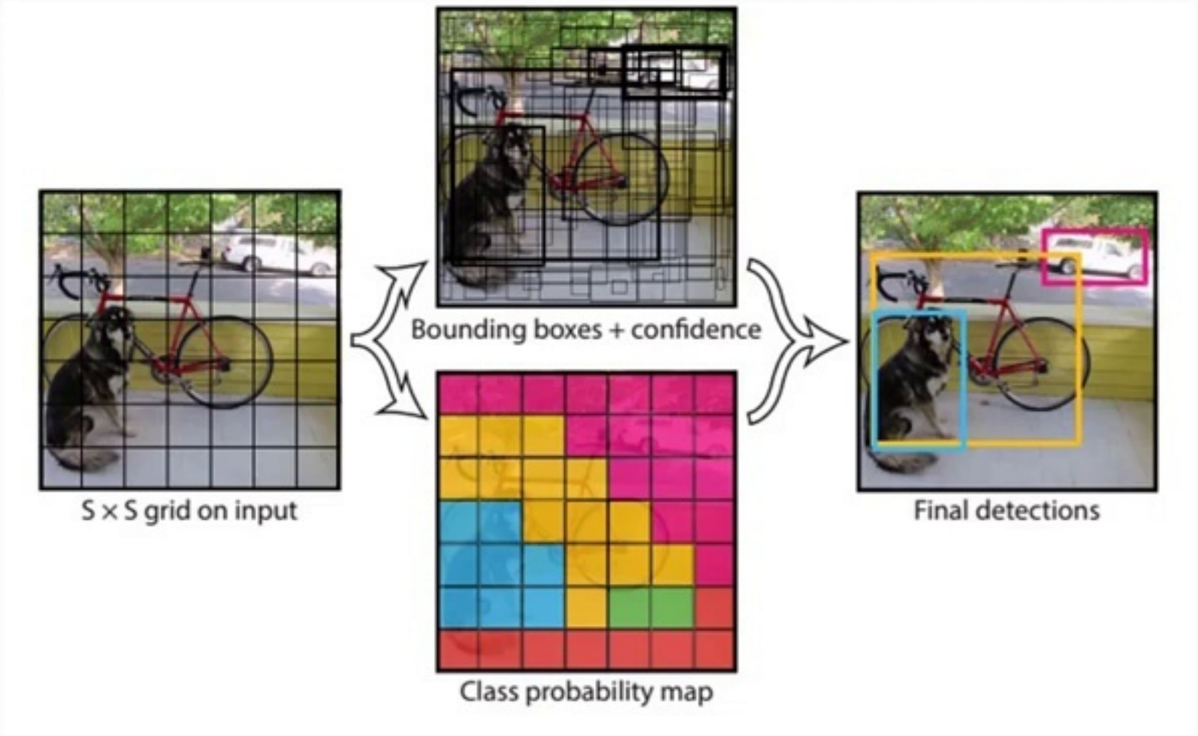
YOLO – How Does it Work

- The input image is divided into an S x S grid
- Each grid predicts several bounding boxes and confidence scores for those boxes



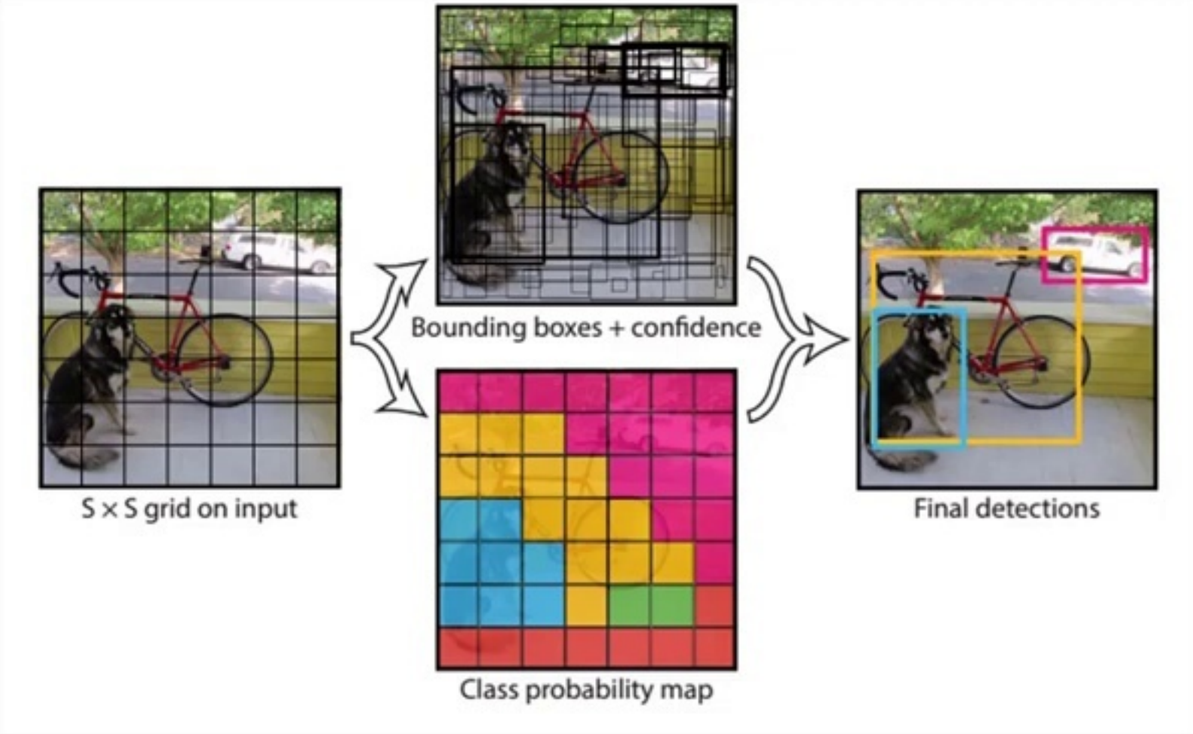
YOLO – How Does it Work

- The input image is divided into an S x S grid
- Each grid predicts several bounding boxes and confidence scores for those boxes
- Confidence here is defined as the probability of an object multiplied by the threshold IoU score

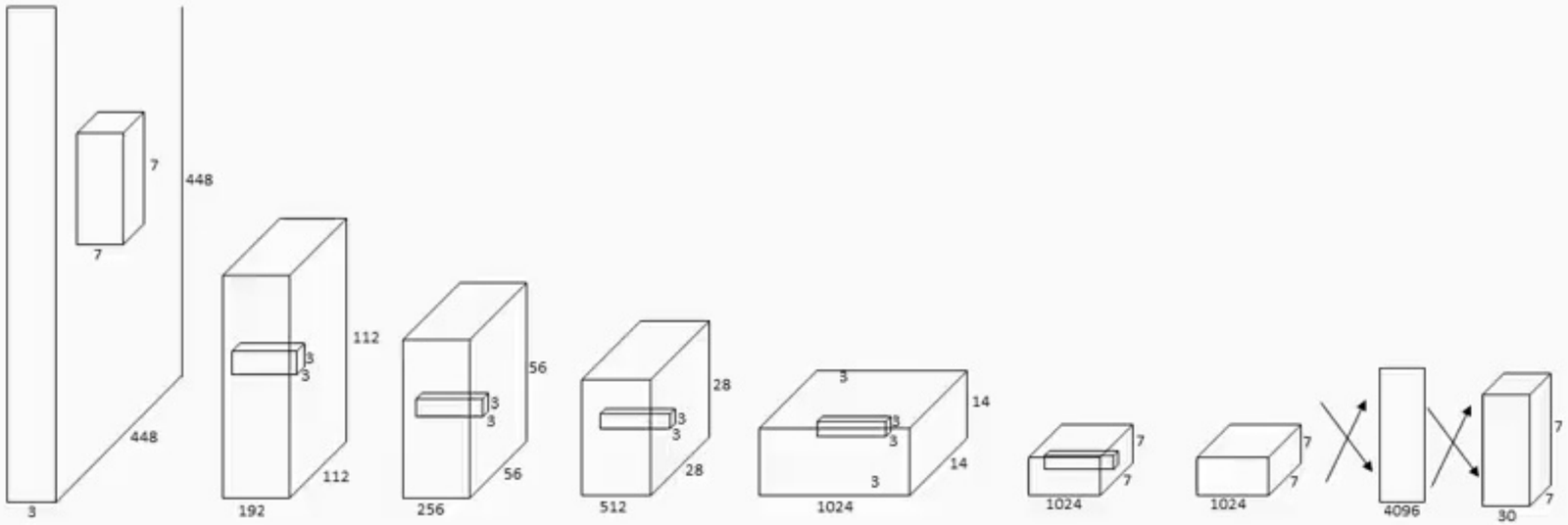


YOLO – How Does it Work (Continued)

- Then, by multiplying the conditional class probability and the individual box confidence predictions, we get the class-specific confidence score for each box
- $\text{Prob}(\text{Class} \mid \text{Object}) \times \text{Prob}(\text{Object}) \times \text{IoU} = \text{Prob}(\text{Class}) \times \text{IoU}$



YOLO Architecture



Conv. Layer
7x7x64/2
Maxpool Layer
2x2/2

Conv. Layer
3x3x192
Maxpool Layer
2x2/2

Conv. Layers
1x1x128
3x3x256
1x1x256
3x3x152
Maxpool Layer
2x2/2

Conv. Layers
1x1x128
3x3x512
1x1x512
3x3x1024
Maxpool Layer
2x2/2

Conv. Layers
1x1x512
3x3x1024
3x3x1024
3x3x1024/2

Conv. Layers
3x3x1024
3x3x1024

Conv.
Layers

Conv.
Layers

YOLO Evolution

- YOLO first appeared in 2016 and was voted the People Choice Award at CVPR

YOLO Evolution

- YOLO first appeared in 2016 and was voted the People Choice Award at CVPR
- YOLOv2 was later released where Batch Normalization was added, which resulted in mAP improvements of 2%; which was later fined tuned to work at higher resolution (of 448 x 448), giving a 4% increase in its mAP

YOLO Evolution

- YOLO first appeared in 2016 and was voted the People Choice Award at CVPR
- YOLOv2 was later released where Batch Normalization was added, which resulted in mAP improvements of 2%; which was later fined tuned to work at higher resolution (of 448 x 448), giving a 4% increase in its mAP
- YOLO3 was fined tuned even further and introduced multi-scale training to better help detect smaller objects