Object Detection 101 Continued

RCNN’s and the YOLO Algorithm

What we have seen so far...

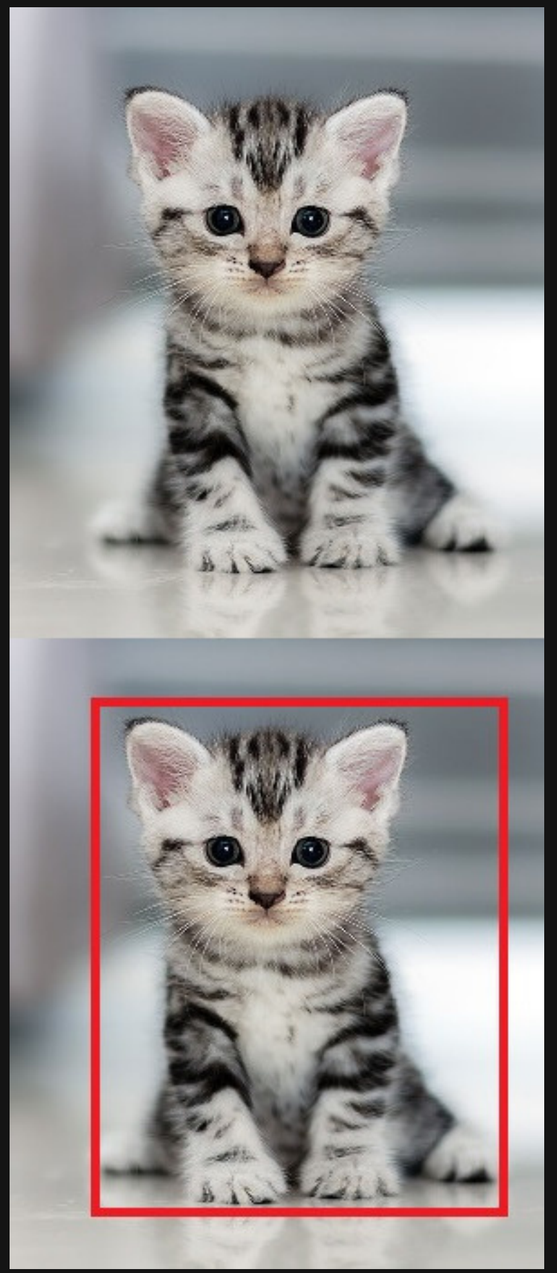
The YOLO Algorithm

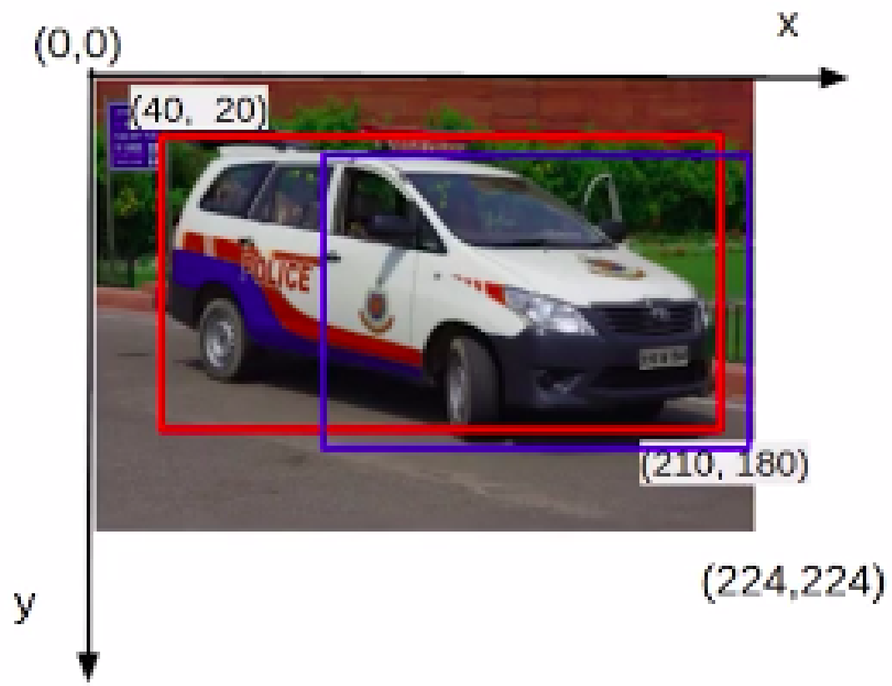
A few Ideas you need to know:

- Bounding box predictions
- The sliding window technique and its convolutional approach
- IoU
- Anchor Boxes
- Non max supression

OBJECT LOCALISATION

The objective is to detect the class and a bounding box of where that object is.





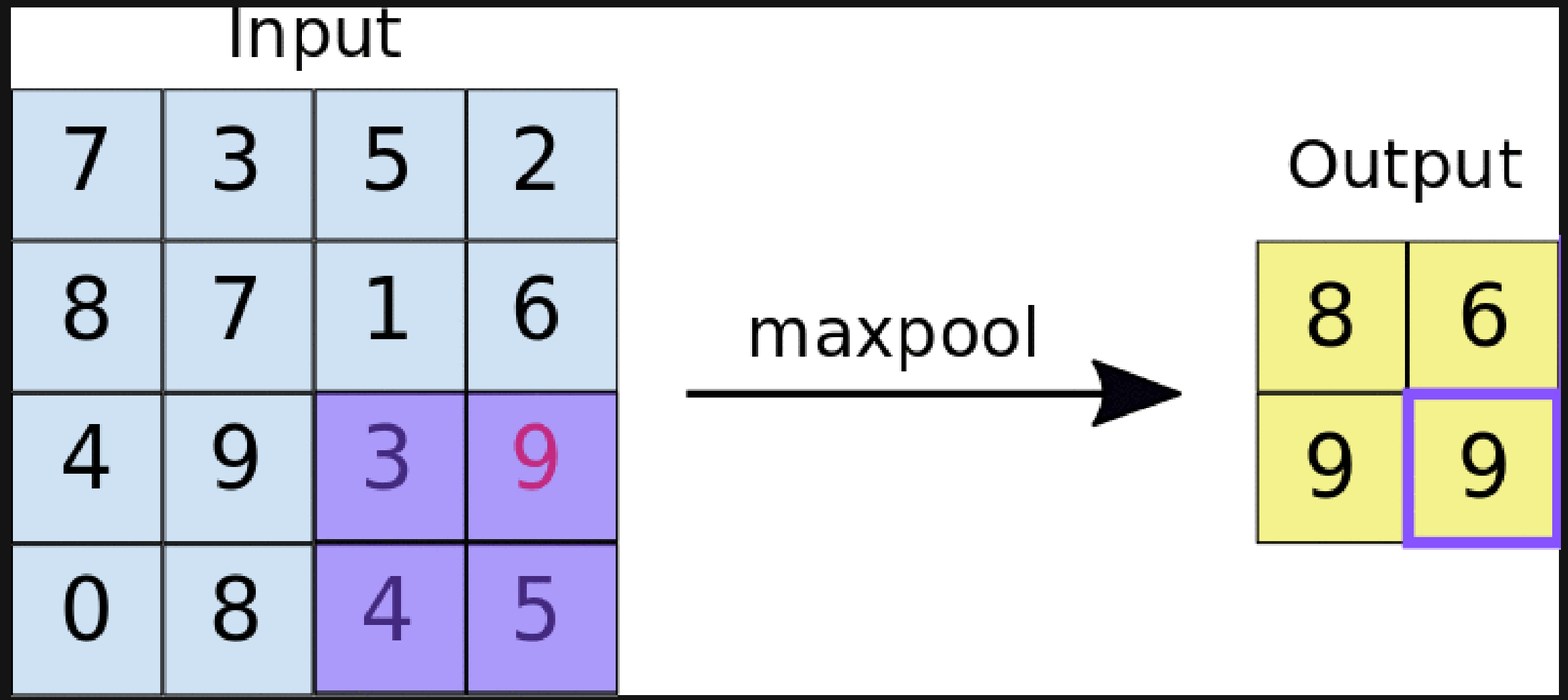
Input Image

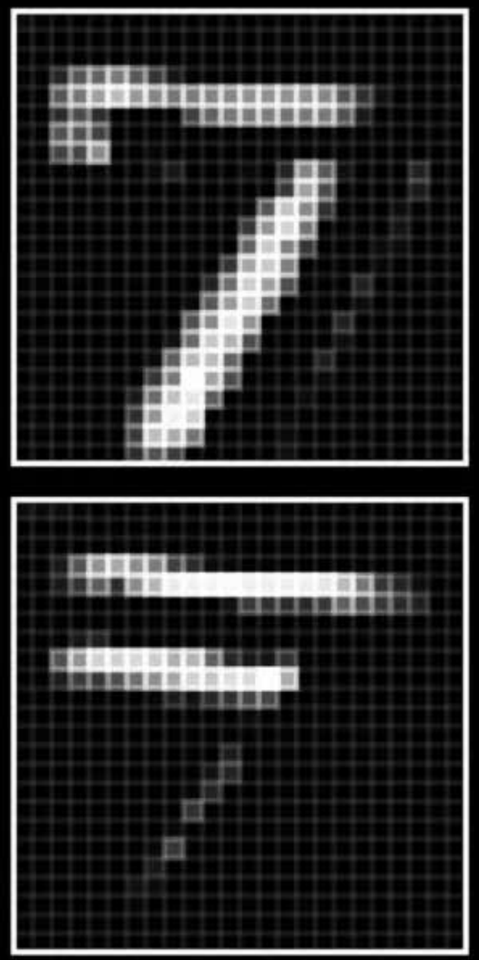
Target Values	Input Target	Predictions
P	1	0.8
x_{min}	40	70
y_{min}	20	30
x_{max}	210	220
y_{max}	180	190
c_1	1	1
c_2	0	0

Target Values | Input Target | Predictions

MaxPooling
A Quick Recap

MaxPooling involves runnning a pooling window over the entire image, outputting the value of the pixel with the maximum value in the sampled window. It helps downsample the image to the required dimensions while retaining the most dominant features.





MaxPooling

Pool Size = (2 x 2), Stride = 2

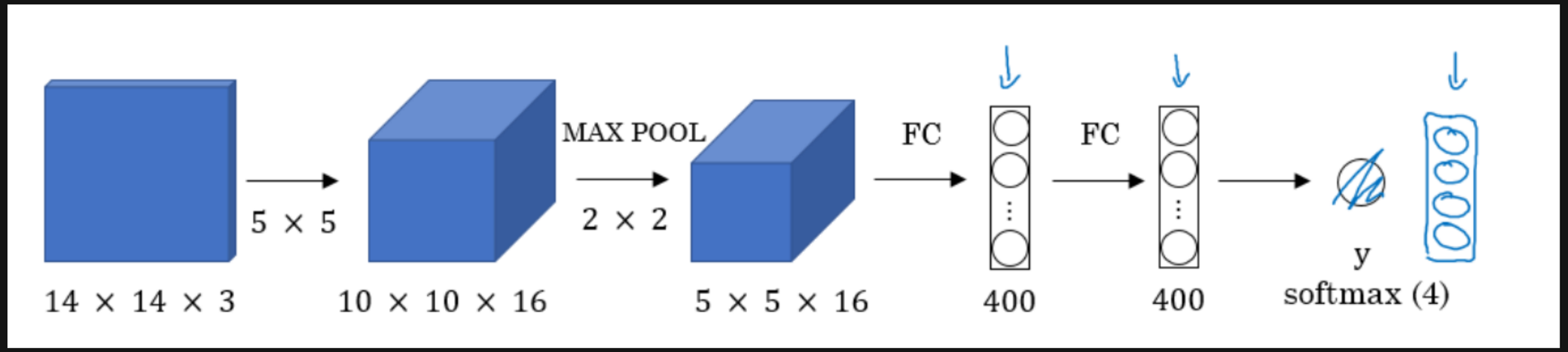
THE SLIDING WINDOW TECHNIQUE

SLIDING WINDOW

The sliding window technique is a highly useful technique in object detection algorithms to get to the target vector.

We first decide a suitable dimension for the sliding window. and then decide a proper stride.

With this stride, we move over the image, each time returning the target vector until the bounding box reaches the end of the image.



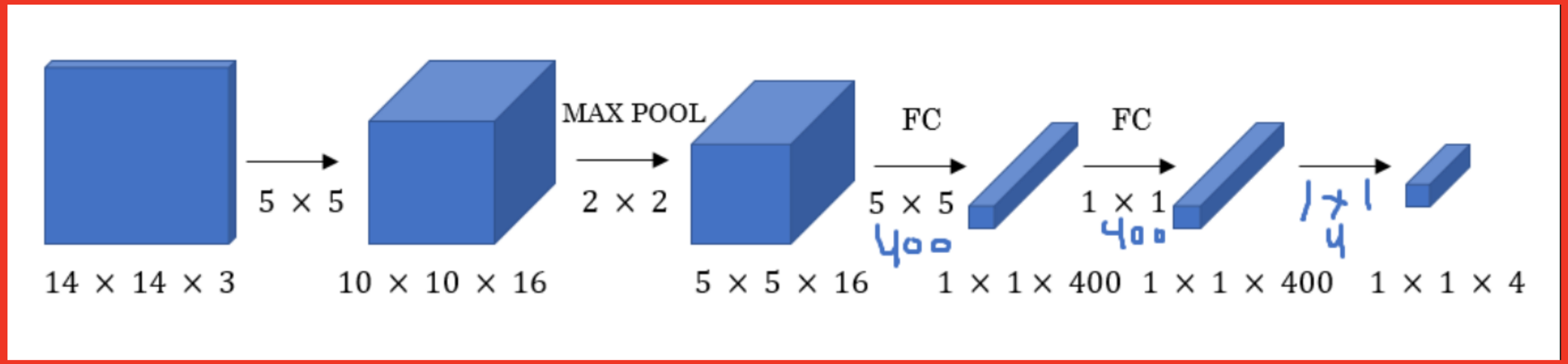
In a normal Convolutional Neural Network:

- Filters are applied to the input, generating feature maps.
- Maxpooling is performed while retaining the number of feature maps.
- The resulting feature maps are then passed through two fully connected layers.
- Finally, a softmax layer predicts the object's class.

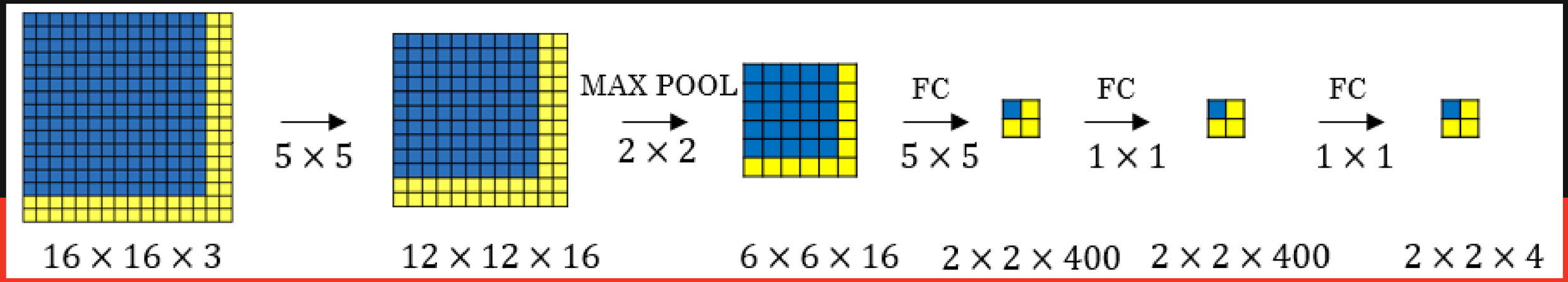
1. The sliding window technique is time-consuming as it processes each area in the image separately and sends it to the pre-trained convolutional network for detection.
2. The convolutional approach improves efficiency by applying filters over the entire image, generating feature maps capturing local patterns at different spatial positions.

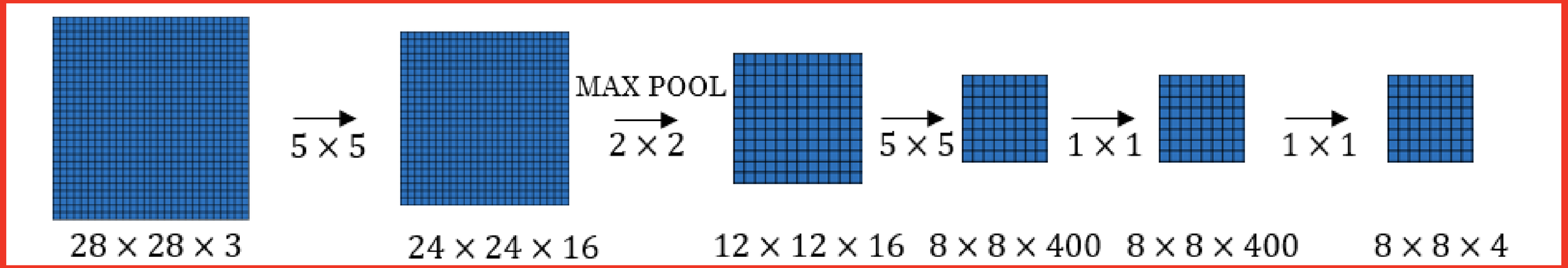
CONVOLUTIONAL IMPLEMENTATION OF THE SLIDING WINDOW

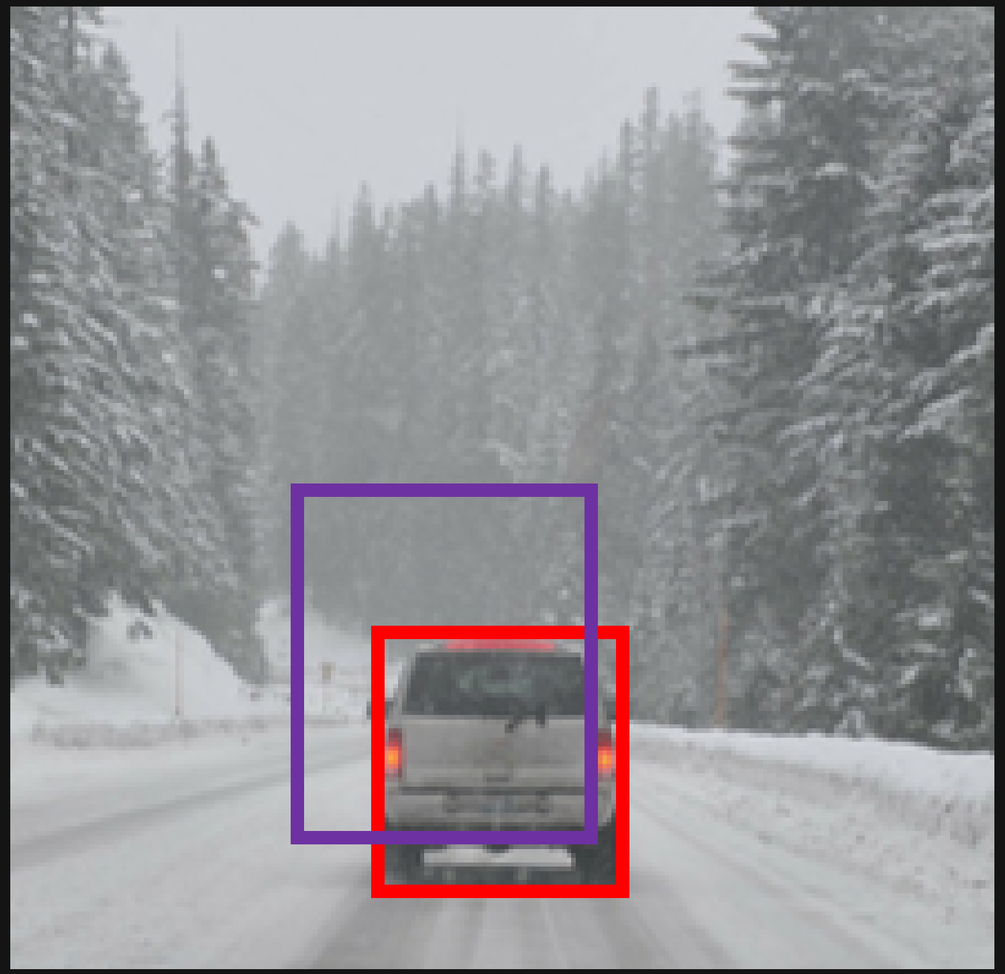
In the convolutional approach for object detection:
- Initial filtering and maxpooling are performed on the input image.
- Instead of using traditional fully connected layers, we apply multiple filters to the output, replicating the FC layers.
- Finally, we apply filters to replicate the softmax operation.



The convolutional approach helps achieve the target vector in a single stride.

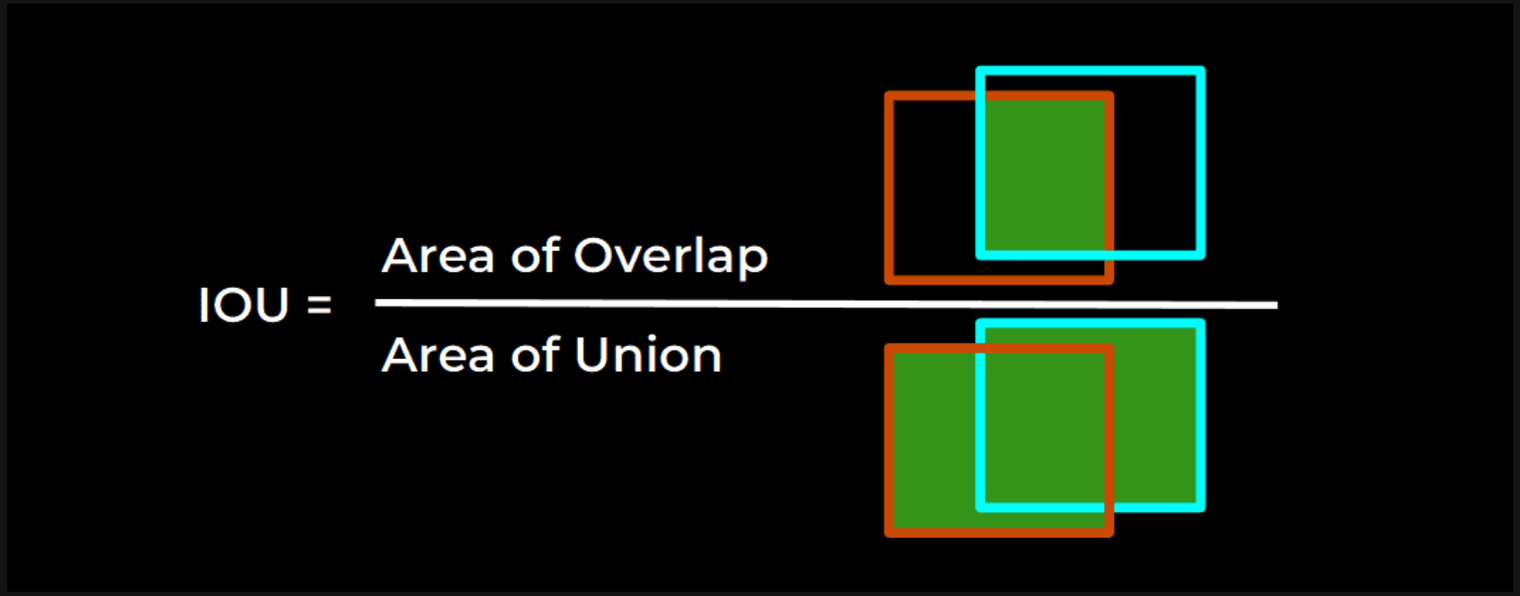


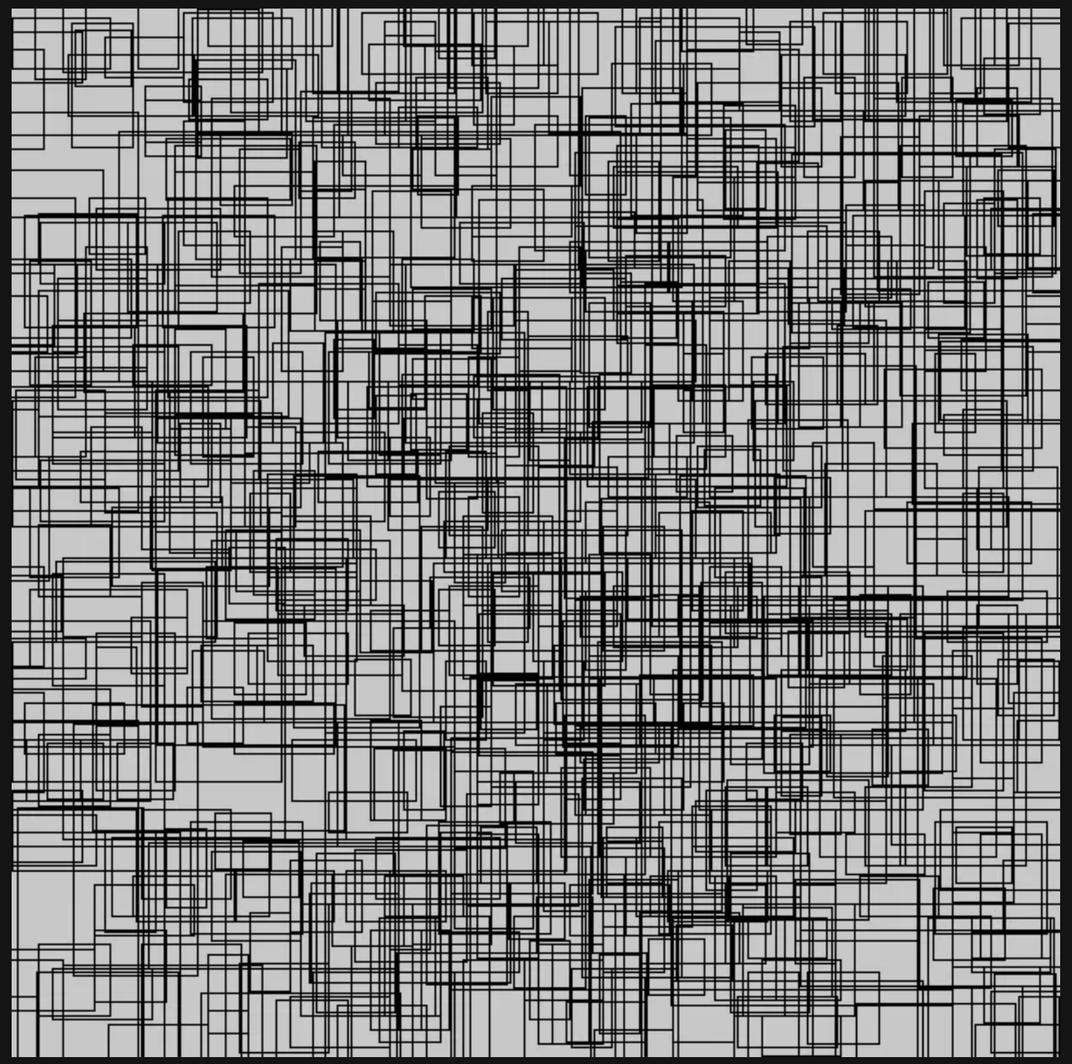




INTERSECTION OVER UNION (IOU)

Intersection over Union (IoU) is basically a way of evaluating the accuracy of the predicted bounding box. More generally, IoU is a measure of the overlap between two bounding boxes. Higher the IoU score, better is the predicted bounding box.



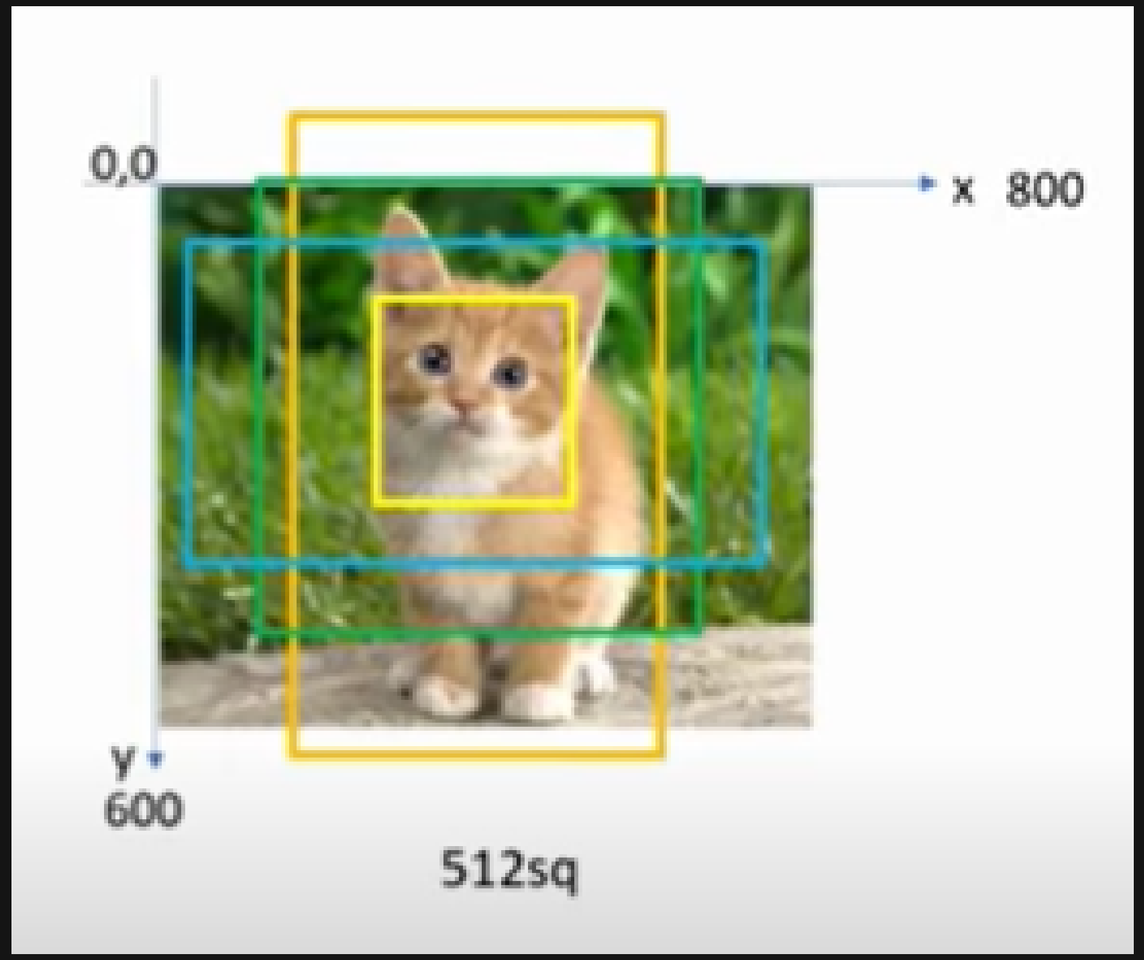


ANCHOR BOXES

WHAT EXACTLY ARE ANCHOR BOXES

Anchor boxes are a set of predefined bounding boxes of a certain height and width.

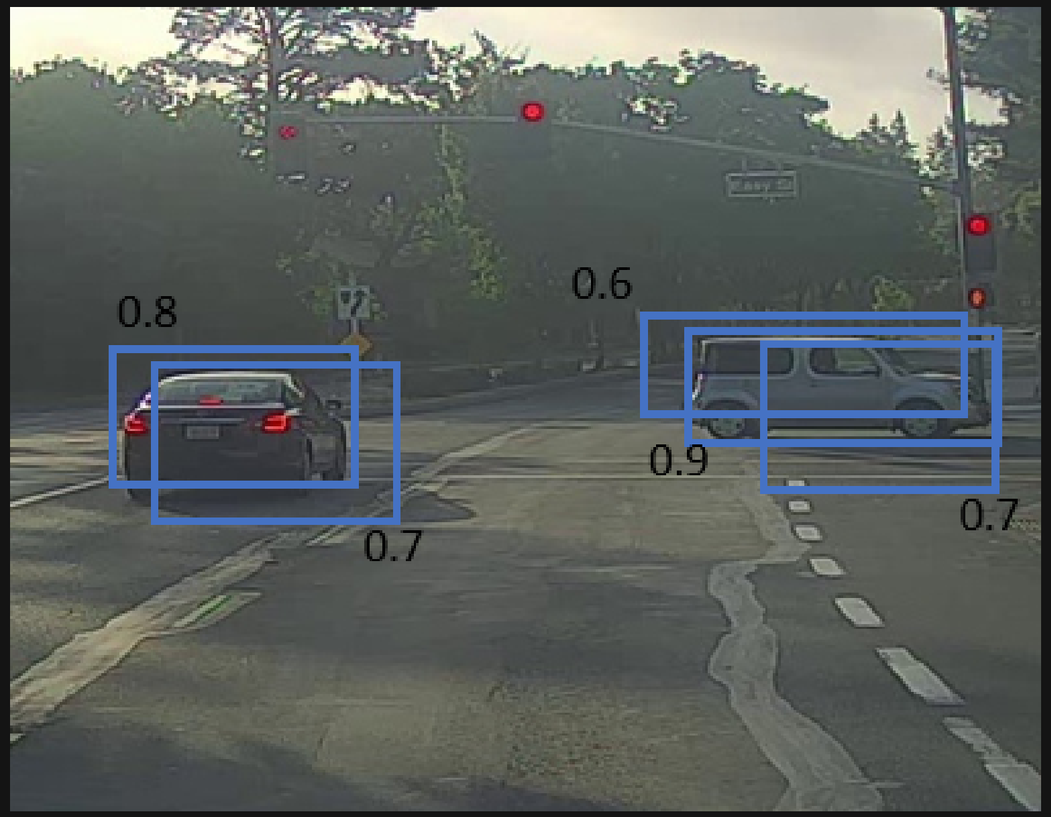
These boxes are defined to capture the scale and aspect ratio of specific object classes



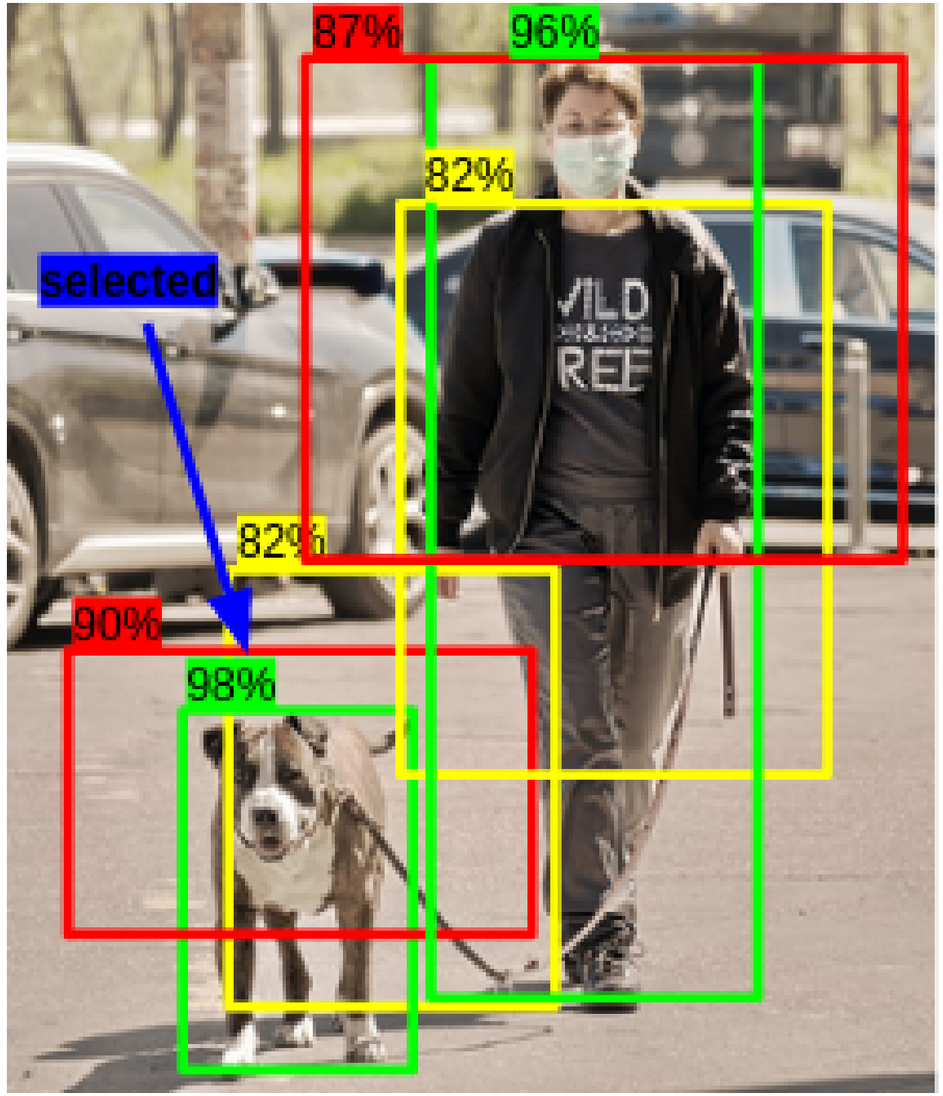
Here is a target vector y that holds the required bounding box predictions for two anchor boxes.

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

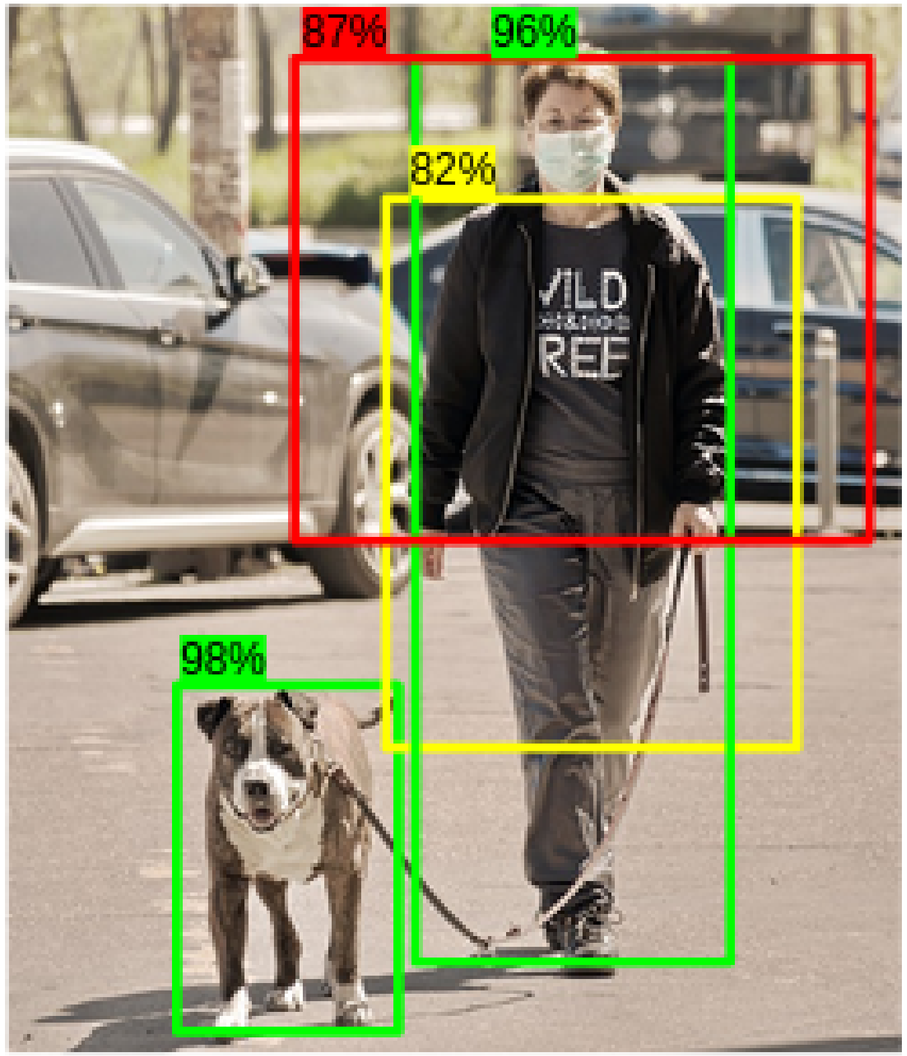
Anchor Boxes also allow us to allot multiple objects to the same grid cell that contains both objects' center by modifying the target vector y



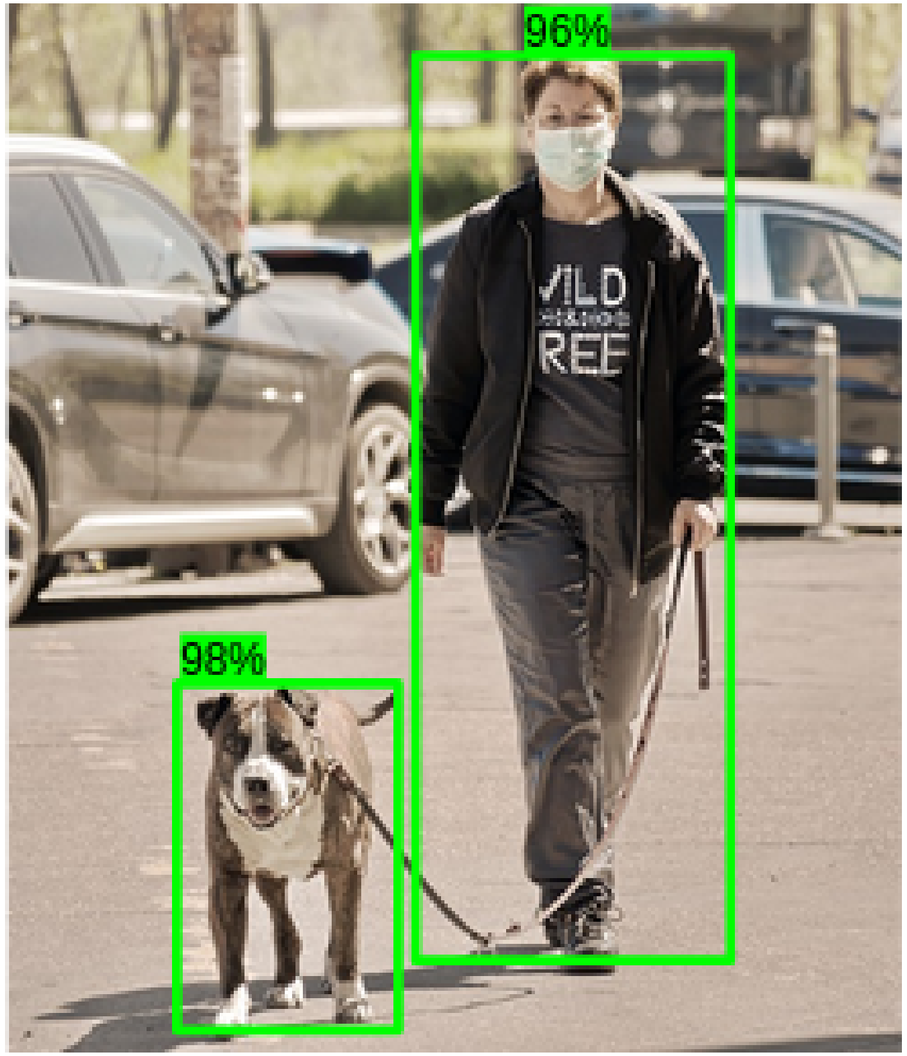
NON MAX SUPRESSION



Step 1: Rremove all bounding boxes that have the pc parameter less than a fixed threshold.



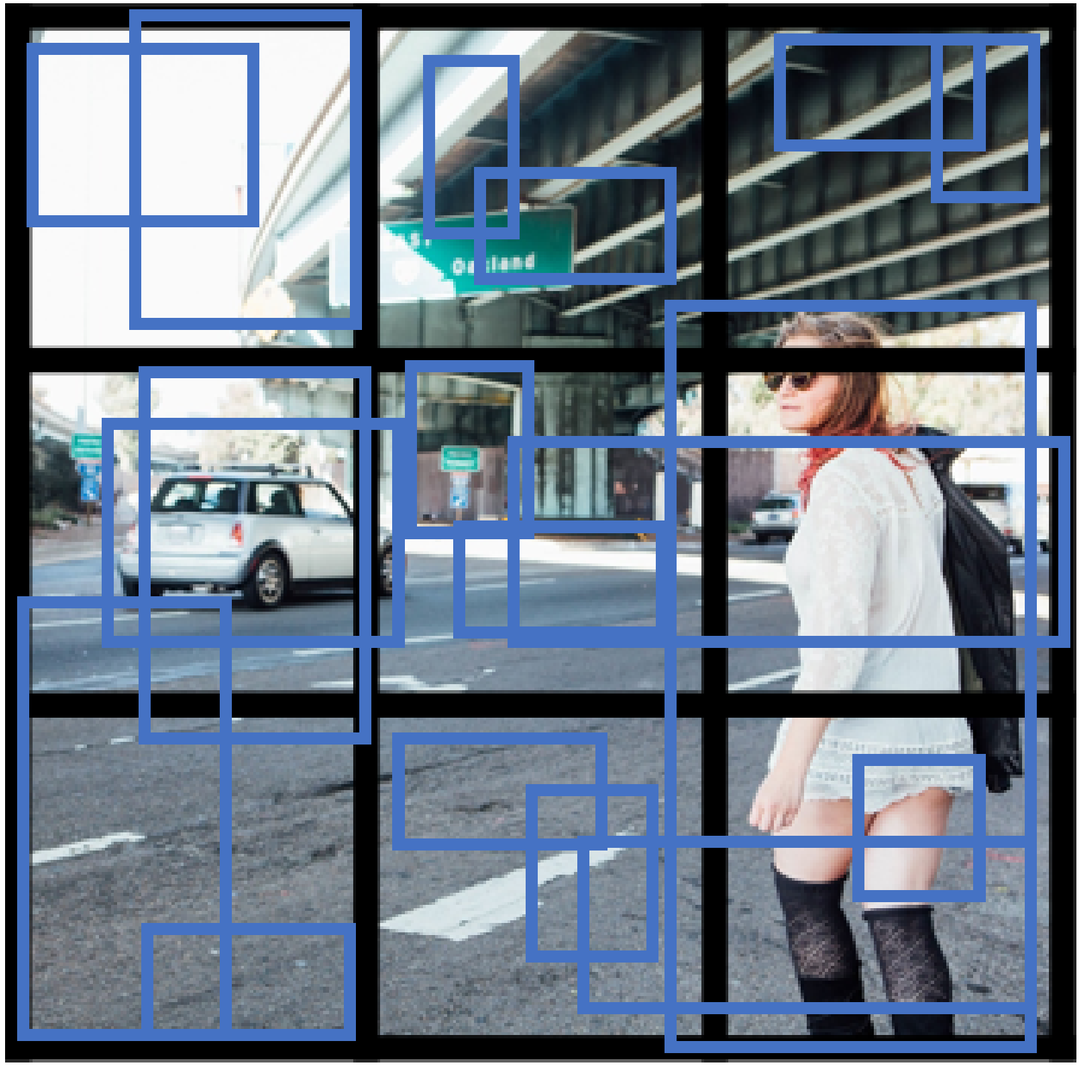
Step 2: First, take the box having the maximum probability, i.e, highest pc value. Then, run the IoU algorithm, removing all bounding boxes that have IoU>0.5.



Step 3: Now, you have the final output. If you have 'c' classes that you want to predict, run the algorithm 'c' times, one for every output class.

Enter YOLO

(finally)



- YOLO divides the input image into a grid of cells.
- For each grid cell, YOLO predicts multiple bounding boxes and their associated confidence scores.

Apply Non-Maximum Suppression (NMS)

YOLO's output is the detected objects in the input image.

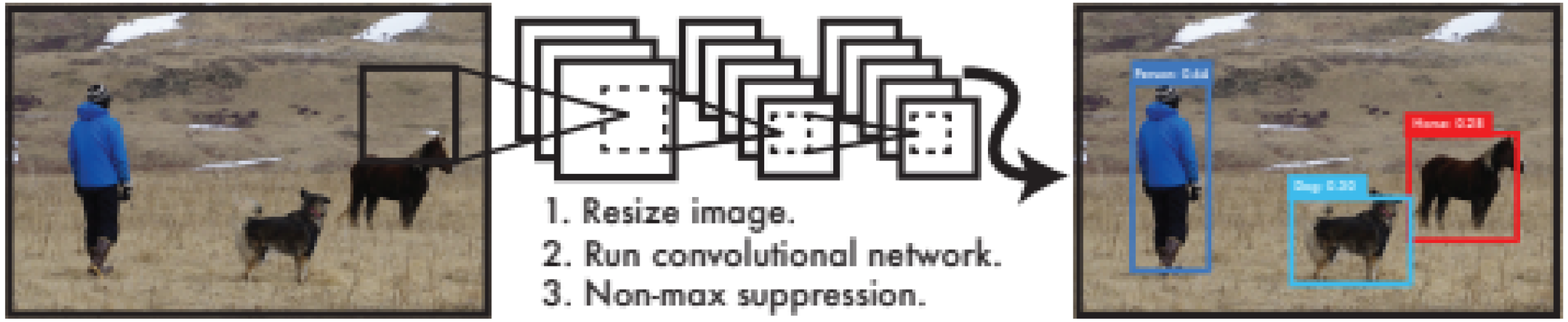


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

Figure 5: Generalization results on Picasso and People-Art datasets.

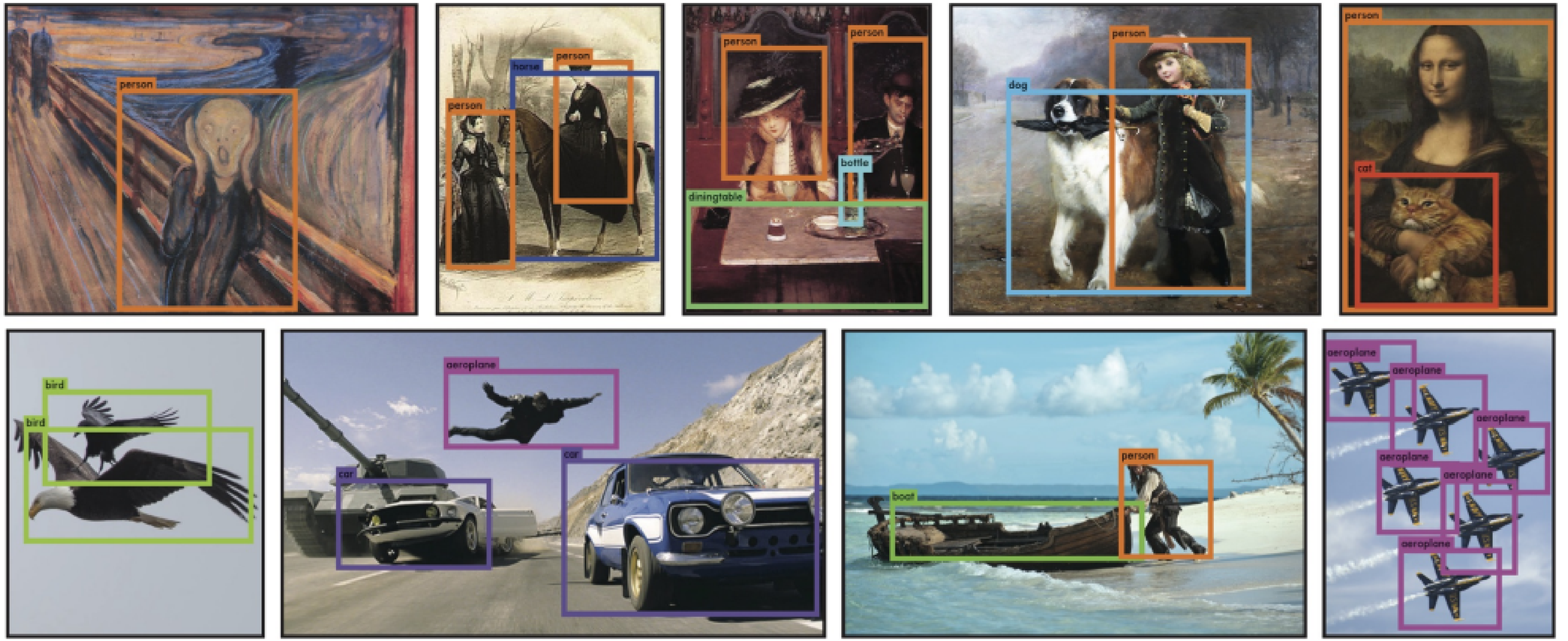


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

R-CNN

(Region based Convolutional Neural Network)

Selective Search and the Greedy Algorithm

A sub-segmentation of input image

Felzenszwalb et al

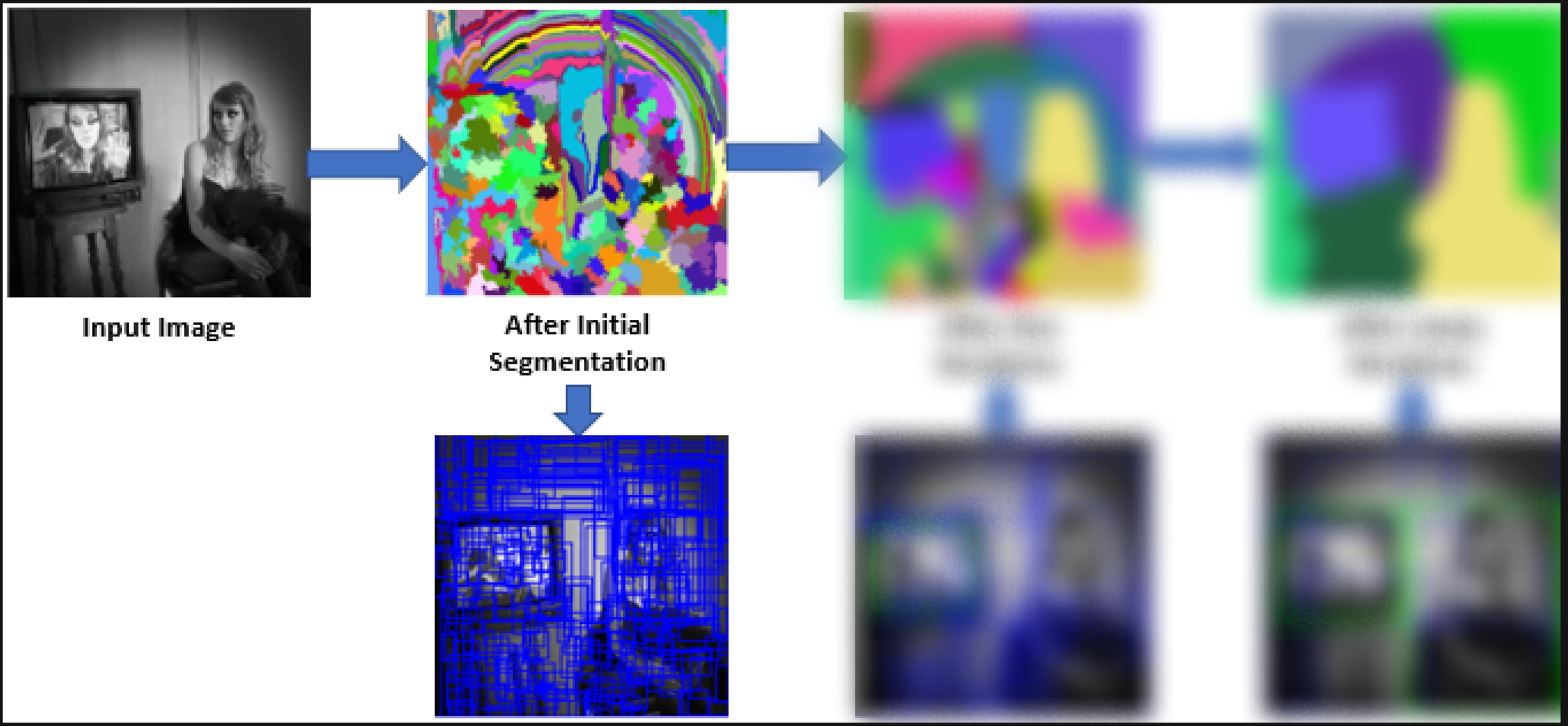
“Efficient Graph-Based Image Segmentation

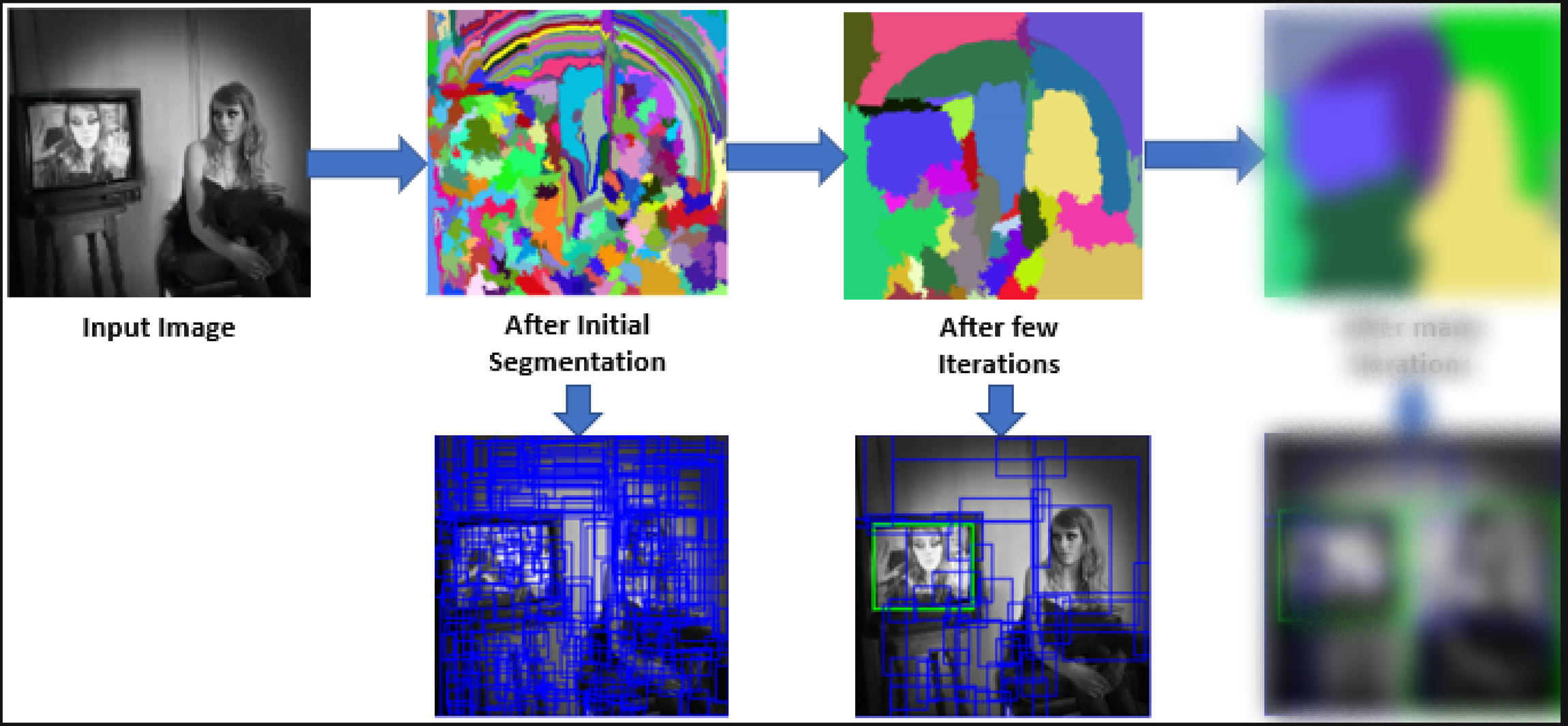
THE GREEDY ALGORITHM

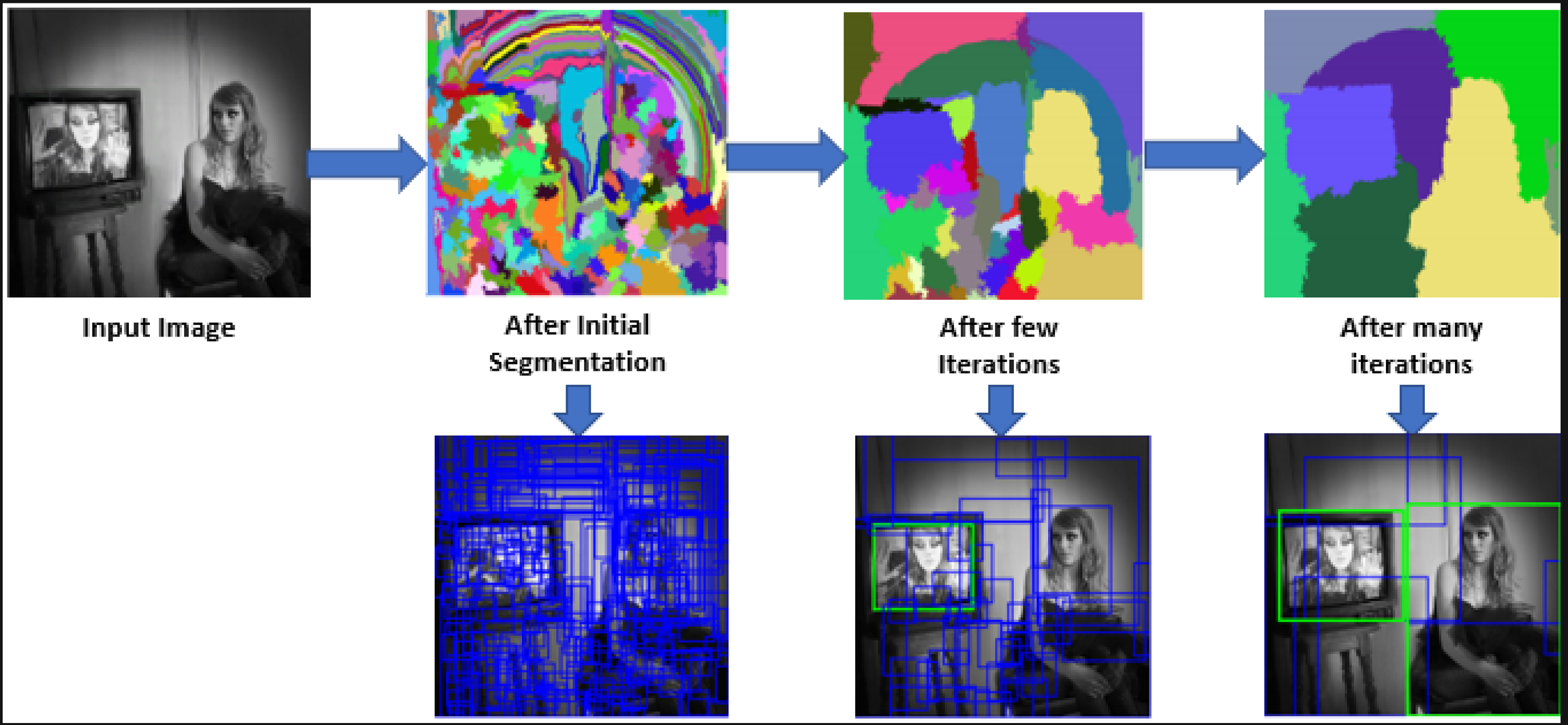
1. From set of regions, choose two that are most similar.

2. Combine them into a single, larger region.

3. Repeat the above steps for multiple iterations.


Input Image
After Initial Segmentation


Input Image
After Initial Segmentation
After few Iterations


Input Image
After Initial Segmentation
After few Iterations
After many iterations

R-CNN: *Regions with CNN features*

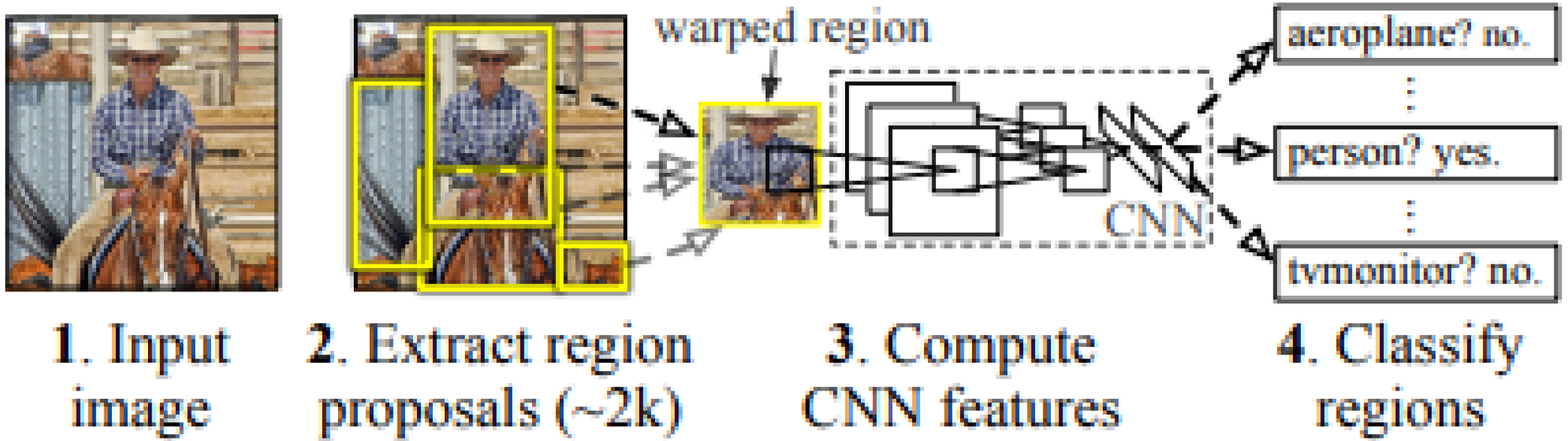


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs.

DRAWBACKS OF THE RCNN ALGORITHM

- Training is a multi-stage pipeline.
- Training is expensive in space and time.
- Object detection is slow.

Fast R-CNN

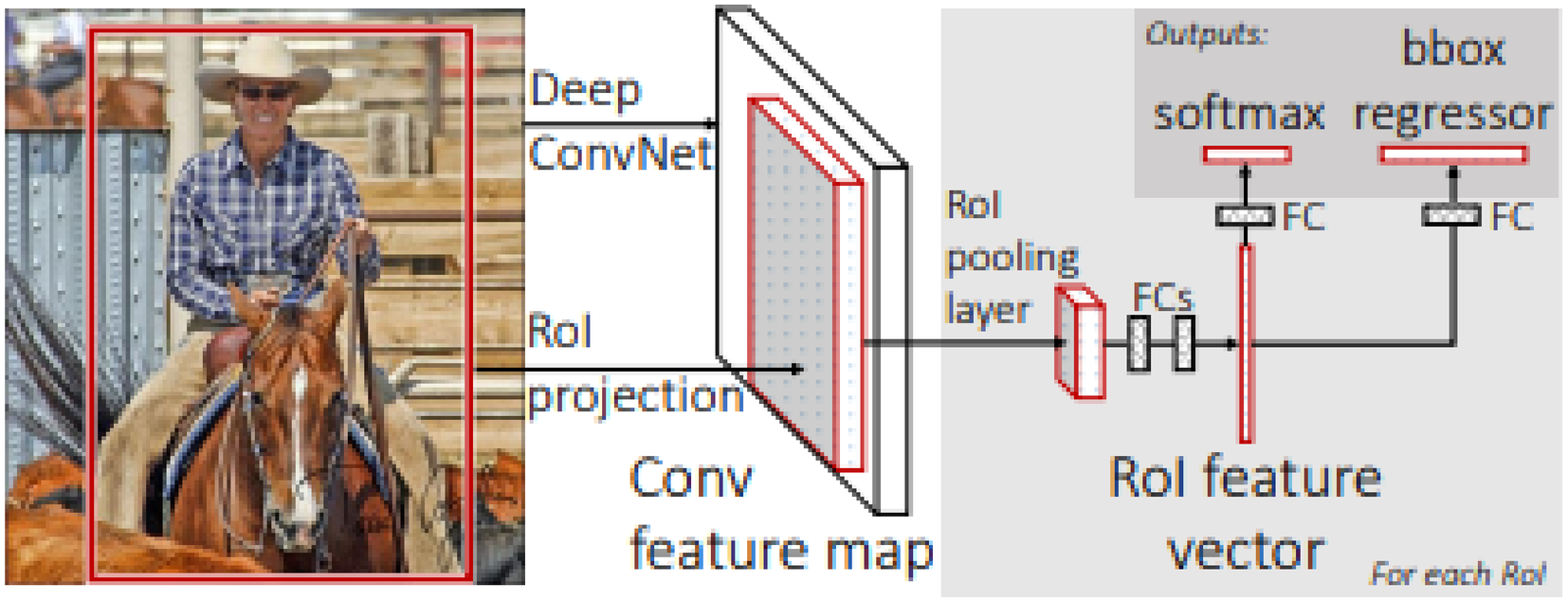


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

DRAWBACKS OF THE FAST RCNN ALGORITHM

- Slower training
- Spatial misalignment
- Single-stage detection

Faster R-CNN

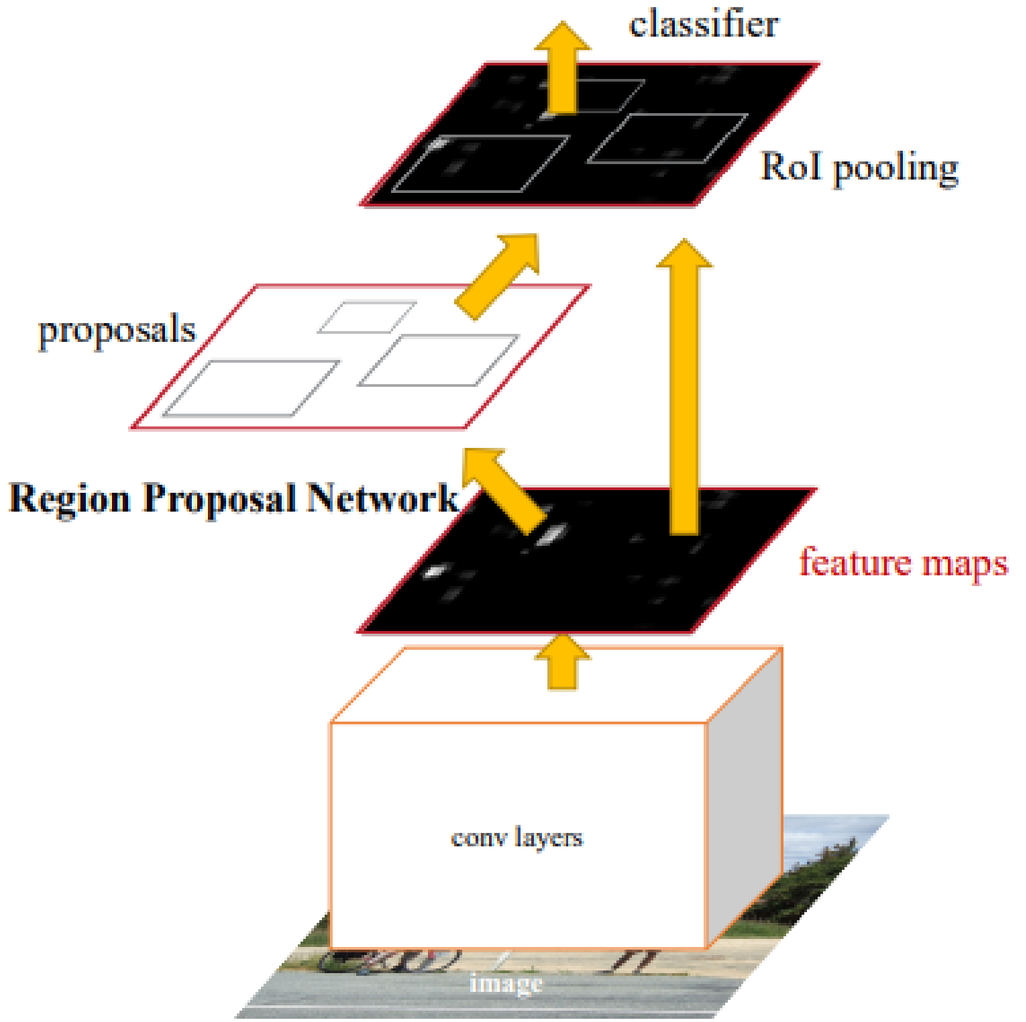


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

"In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features."

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.