Bocconi
AI Neuroscience
Student
Association

logo

YOLO

title

YOU ONLY LOOK ONCE: UNIFIED, REAL-TIME OBJECT DETECTION

name

Giorgio Micaletto

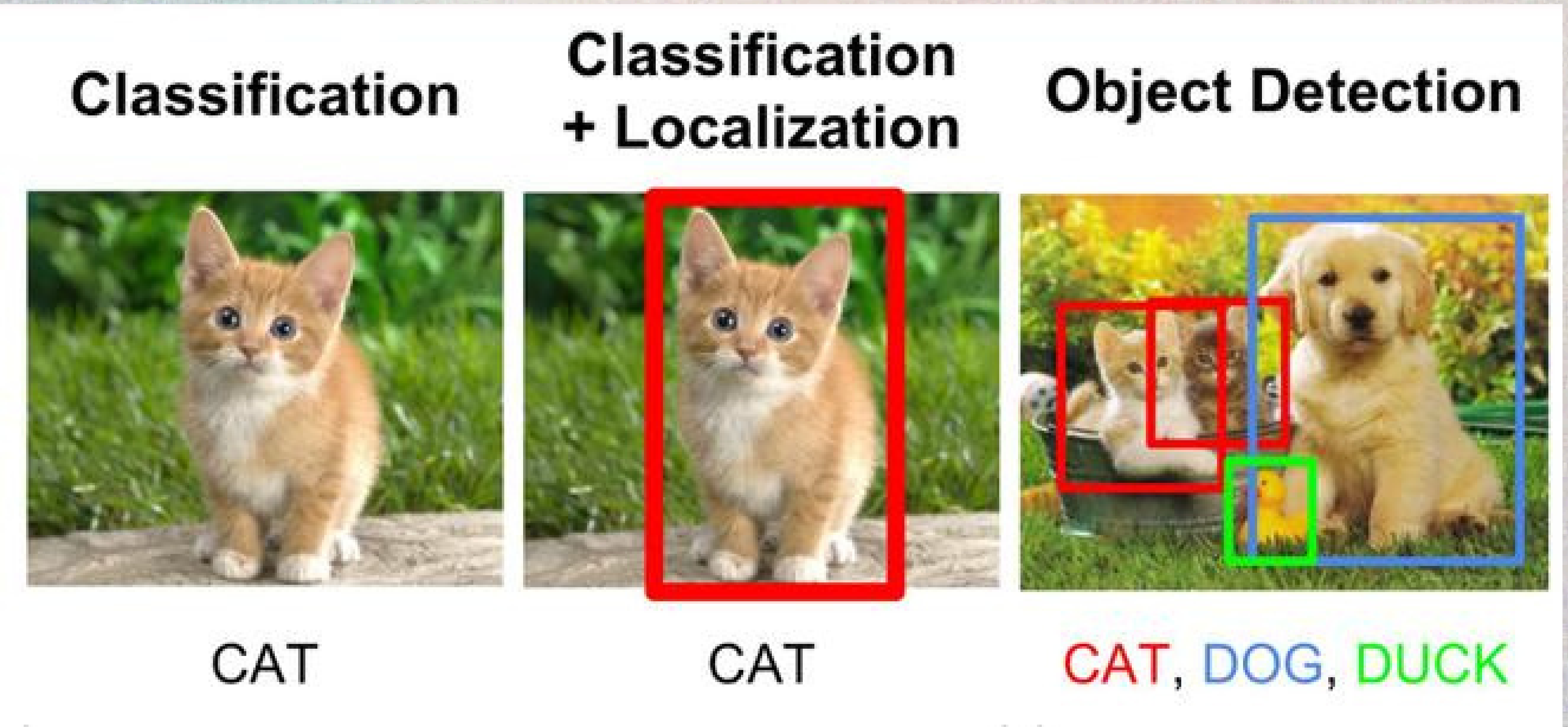
name

Martin Patrikov

name

Alisia Picciano

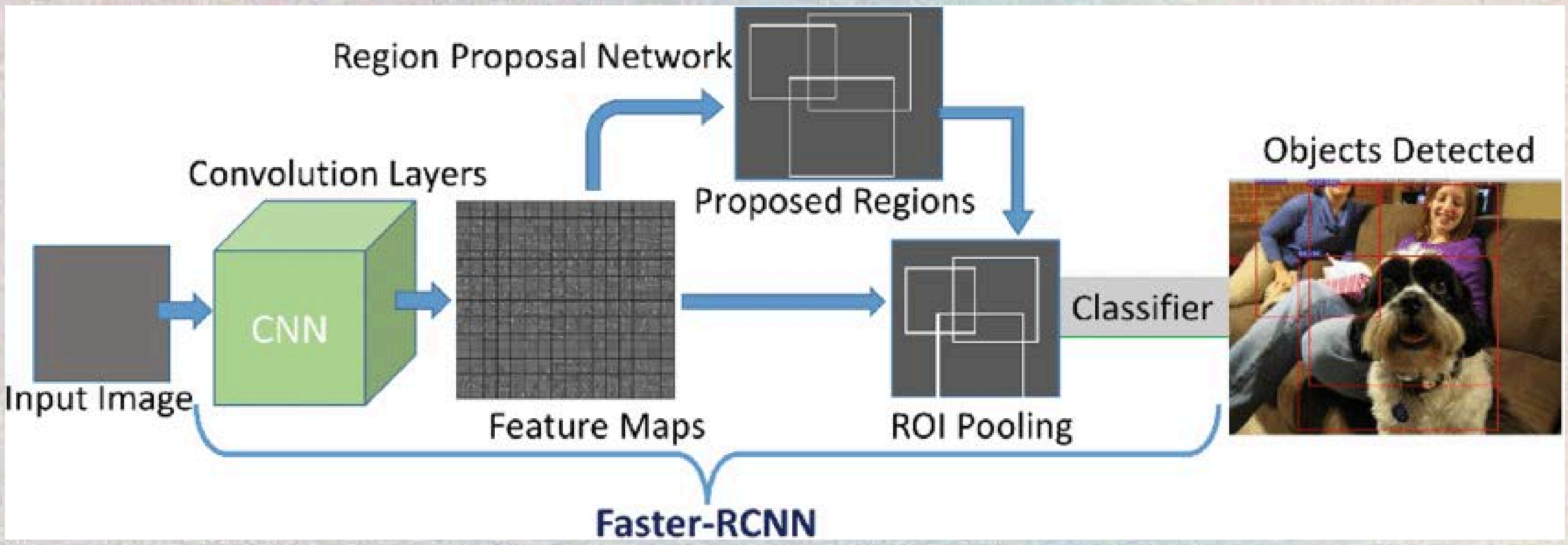
OBJECT DETECTION



Real-life applications: autonomous vehicles, medical imaging, AR etc.

YOLO: A GAME-CHANGER

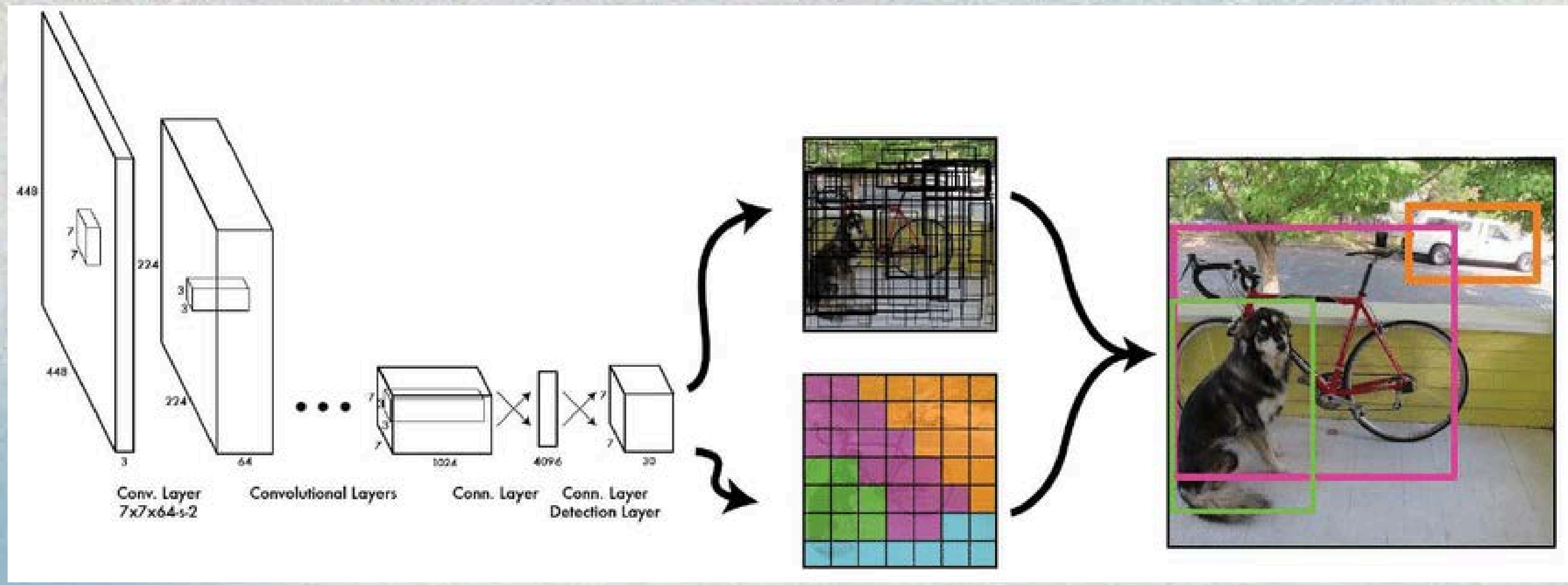
Faster-RCNN pipeline:



Setting before YOLO:

- Repurposing classifiers to perform detection
- Slow and complicated pipelines (multi-stage detection)
- Real-time limitations

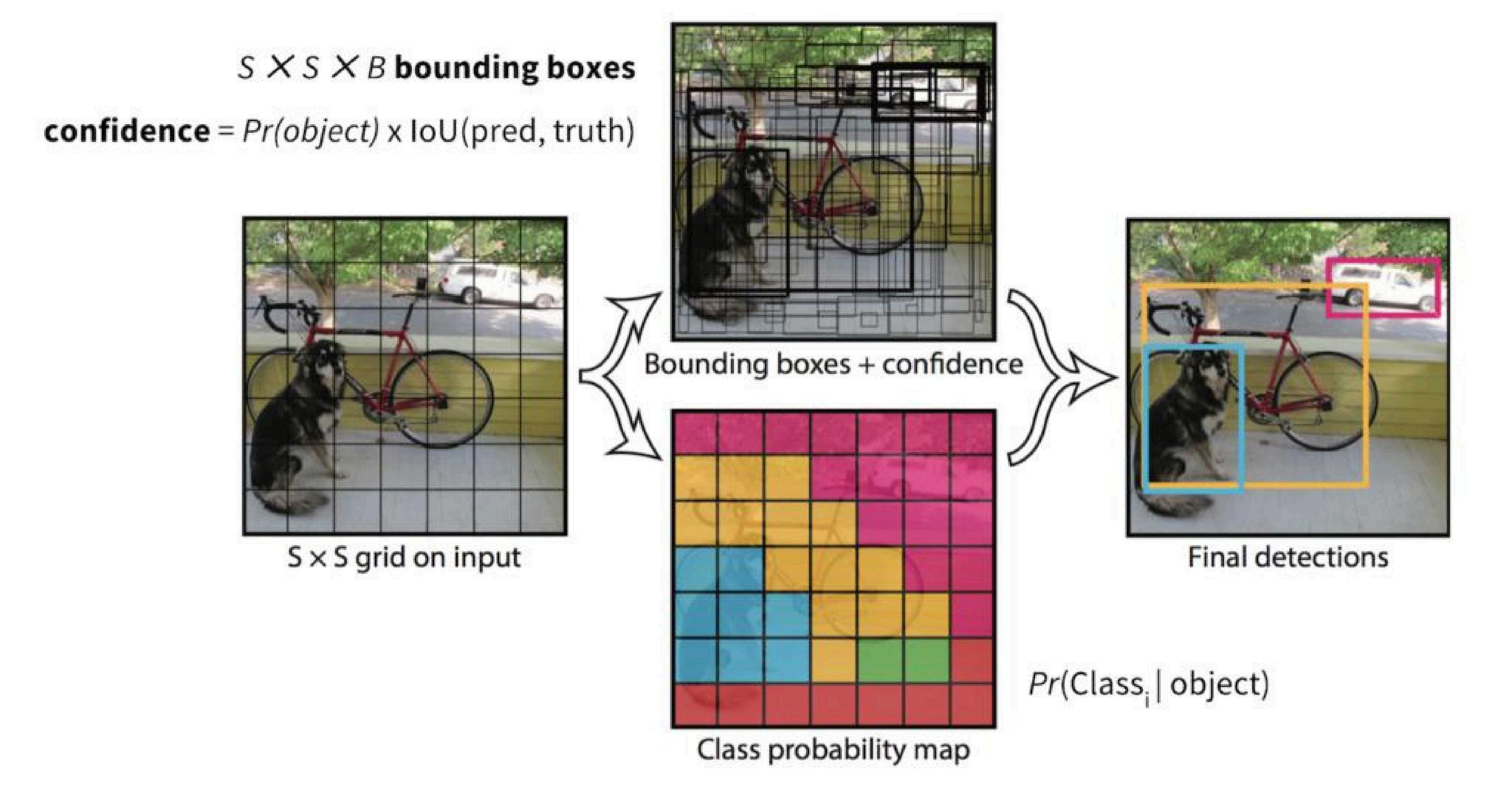
YOLO's pipeline:



Why YOLO was needed:

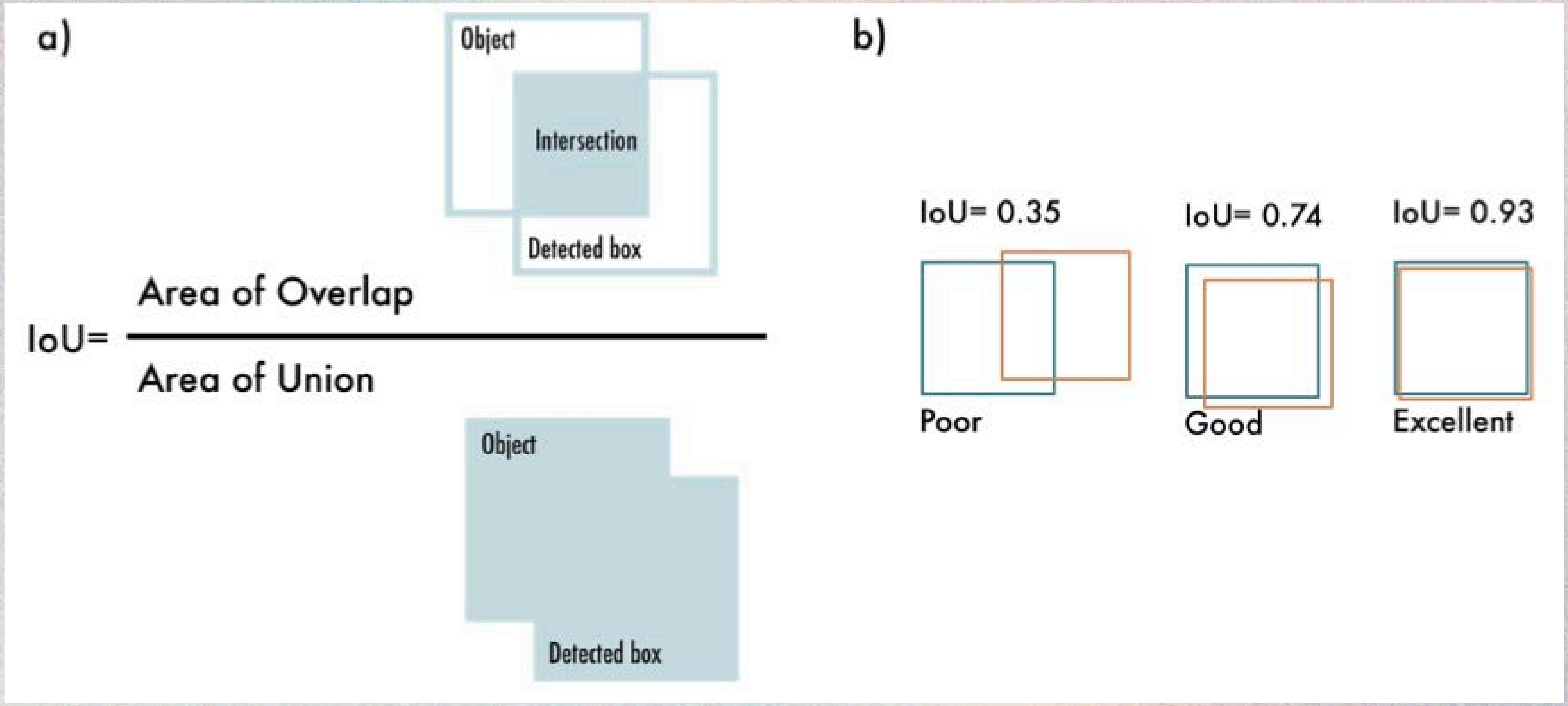
- Faster model was necessary for applications like autonomous driving, real-time surveillance, and robotics
- YOLO introduced a unified approach that treated object detection as a single regression problem rather than a series of stages

DETECTION

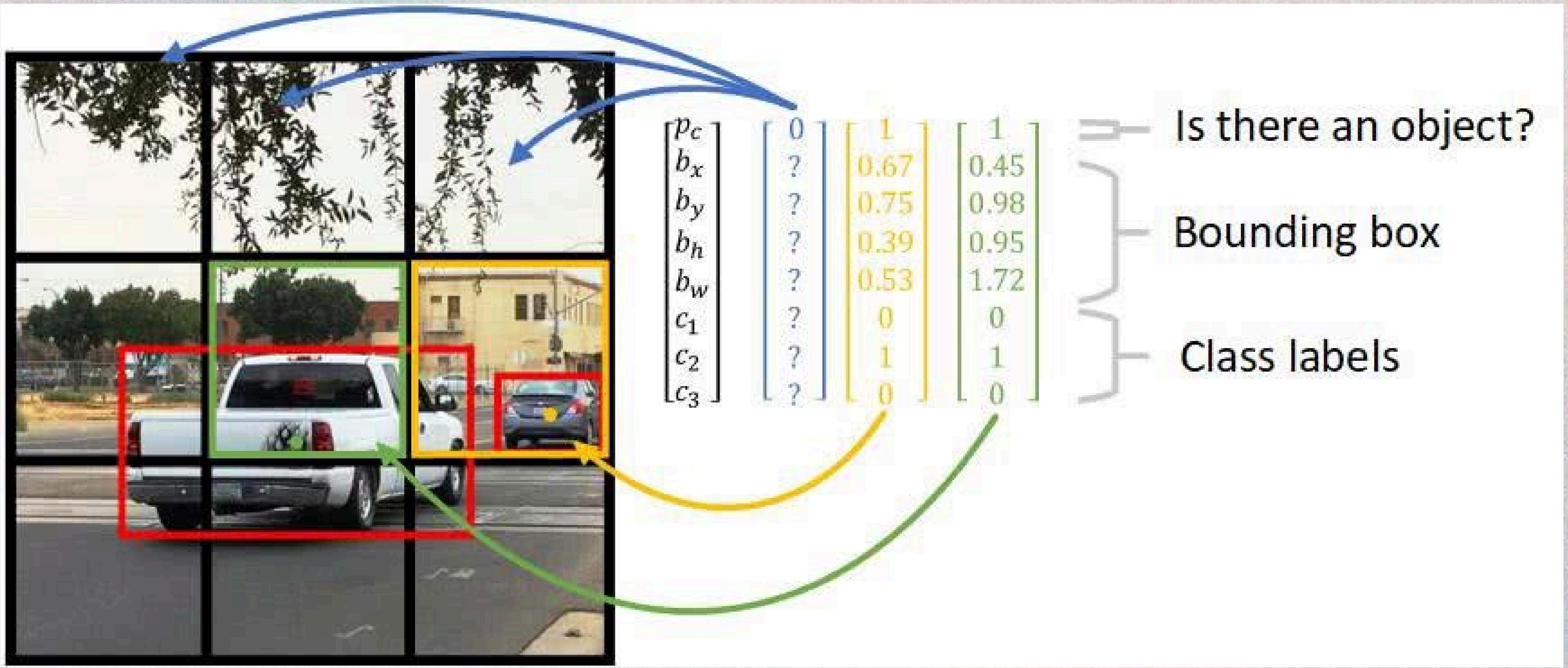


- Divides the input image into a fixed grid **SxS**
- Each grid cell is responsible for detecting object whose **center falls** within that cell
- Each grid cell predict B **bounding boxes** and corresponding **confidence scores**
- Each bounding box includes:
  - Coordinates ***(x,y)*** for the center of the box.
  - Width ***(w)*** and height ***(h)***, relative to the whole image.
  - Confidence score indicating how likely the box contains an object and the likelyhood that the bounding box is correctly localized

INTERSECTION OVER UNION

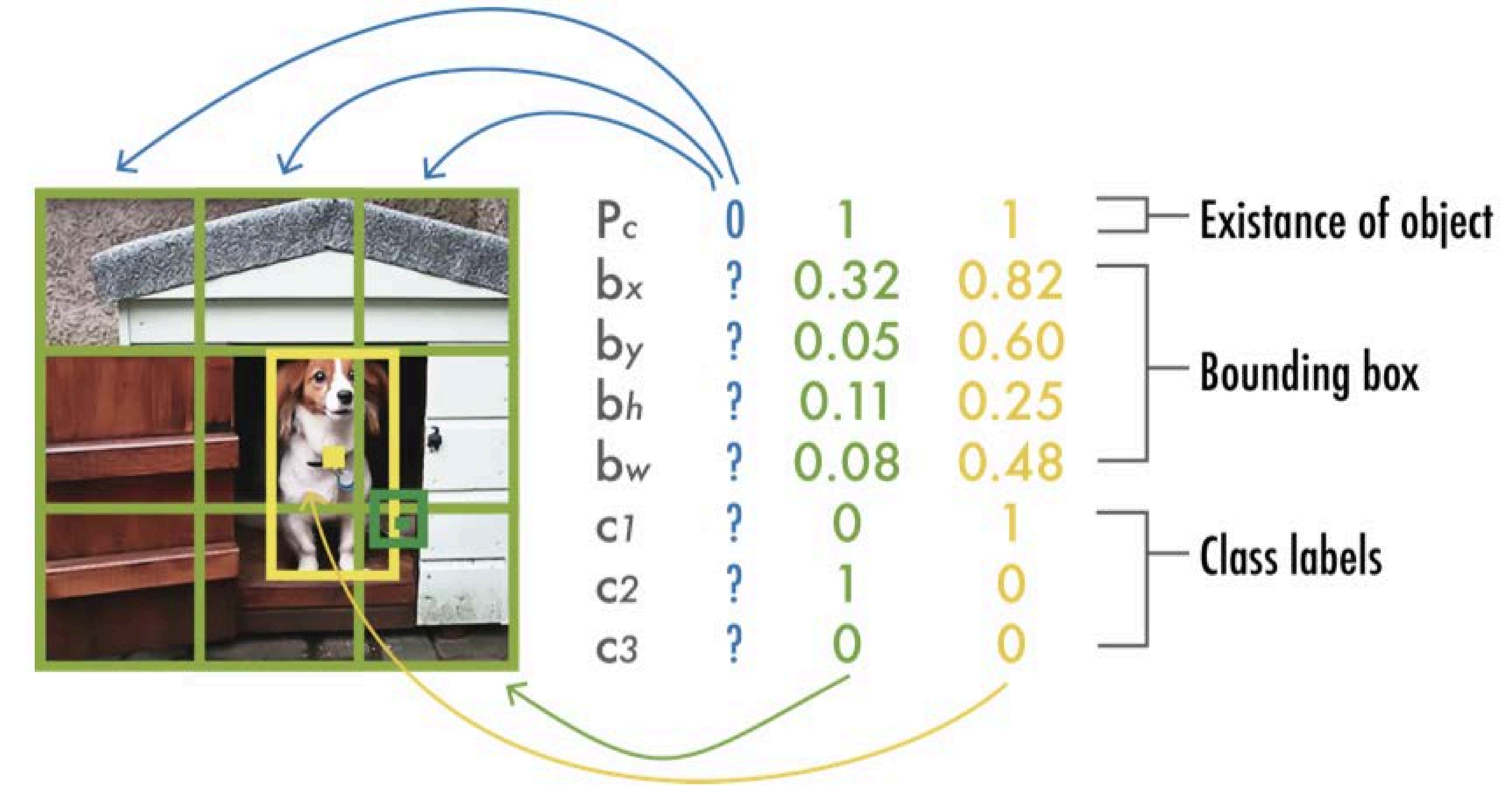


DETECTION



p_c
b_x
b_y
b_h
b_w
c_1
c_2
c_3
0 ? ? ? ? ? ? ?
1 0.67 0.75 0.39 0.53 0 1 0
1 0.45 0.98 0.95 1.72 0 1 0
Is there an object?
Bounding box
Class labels

DETECTION



Class Probabilities:

- Each grid cell also predicts class probabilities for the object present.
- Only one set of class probabilities is predicted per grid cell, even if multiple boxes are predicted.

Final Detections:

- The final predictions are made by multiplying the class probability with the confidence score to produce class-specific confidence scores for each bounding box.

DETECTION

Final Detections:

- Non-max suppression is used to eliminate redundant bounding boxes and select the most accurate ones.

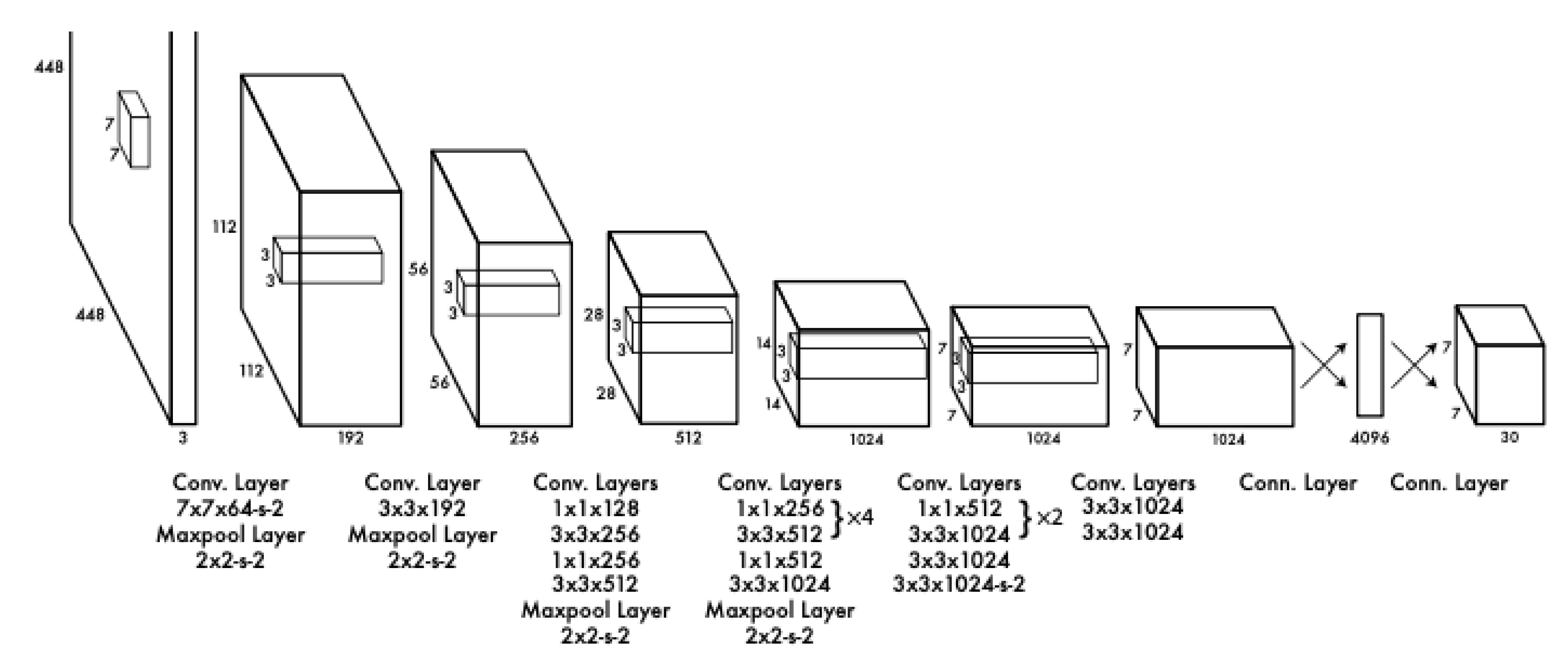
Algorithm 1 Non-Maximum Suppression Algorithm

Require: Set of predicted bounding boxes B, confidence scores S, IoU threshold τ, confidence threshold T
Ensure: Set of filtered bounding boxes F

$F \leftarrow \emptyset$
Filter the boxes: $B \leftarrow \{b \in B \mid S(b) \geq T\}$
Sort the boxes B by their confidence scores in descending order
while $B \neq \emptyset$ **do**
    Select the box b with the highest confidence score
    Add b to the set of final boxes F: $F \leftarrow F \cup \{b\}$
    Remove b from the set of boxes B: $B \leftarrow B - \{b\}$
    for all remaining boxes r in B **do**
        Calculate the IoU between b and r: $iou \leftarrow IoU(b, r)$
        if $iou \geq \tau$ **then**
            Remove r from the set of boxes B: $B \leftarrow B - \{r\}$
        end if
    end for
end while

NETWORK DESIGN

#1 CNN

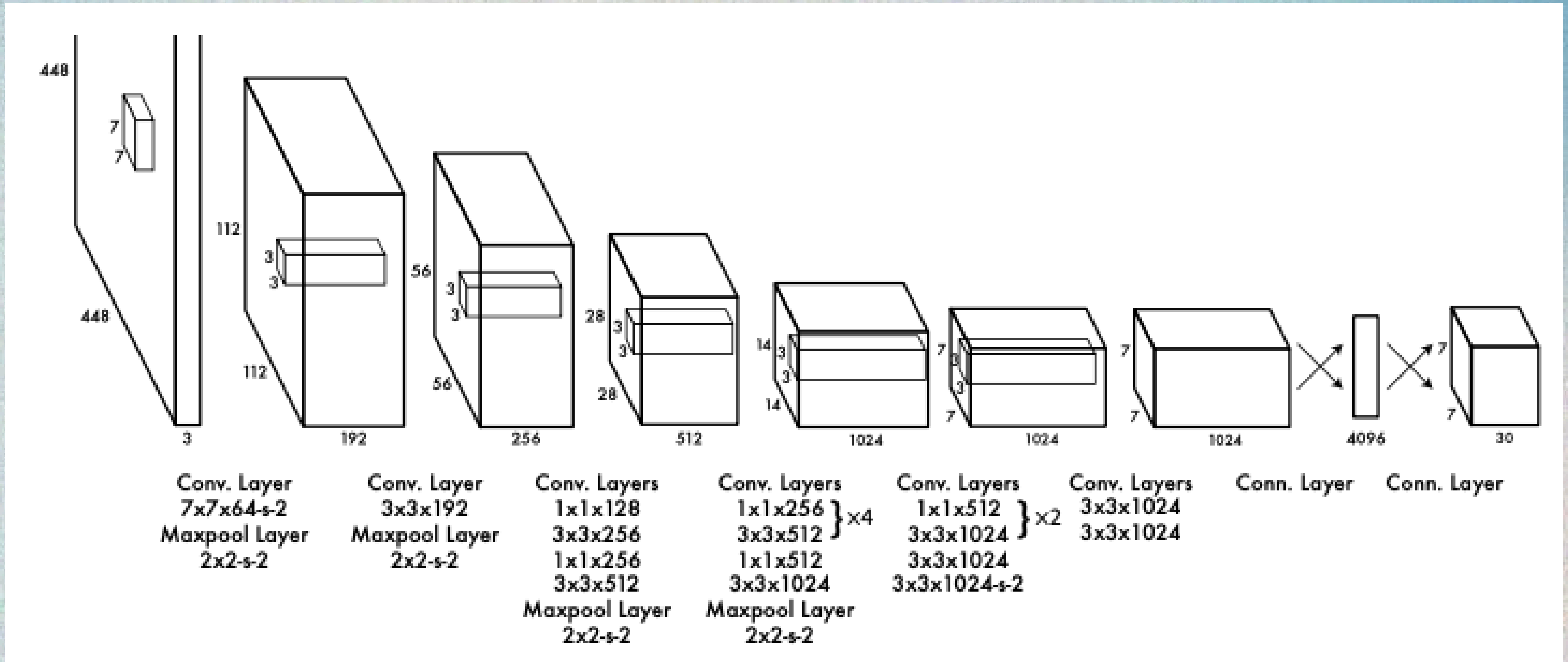


#24 Convolutional Layers

#9 for Fast YOLO

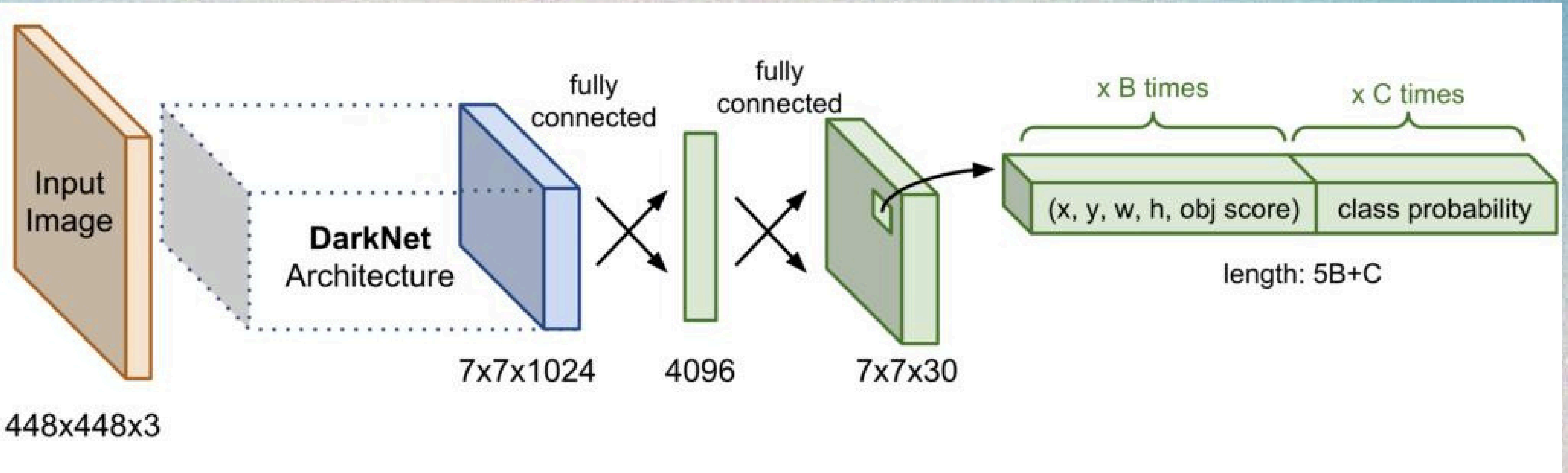
#2 Fully Connected Layers

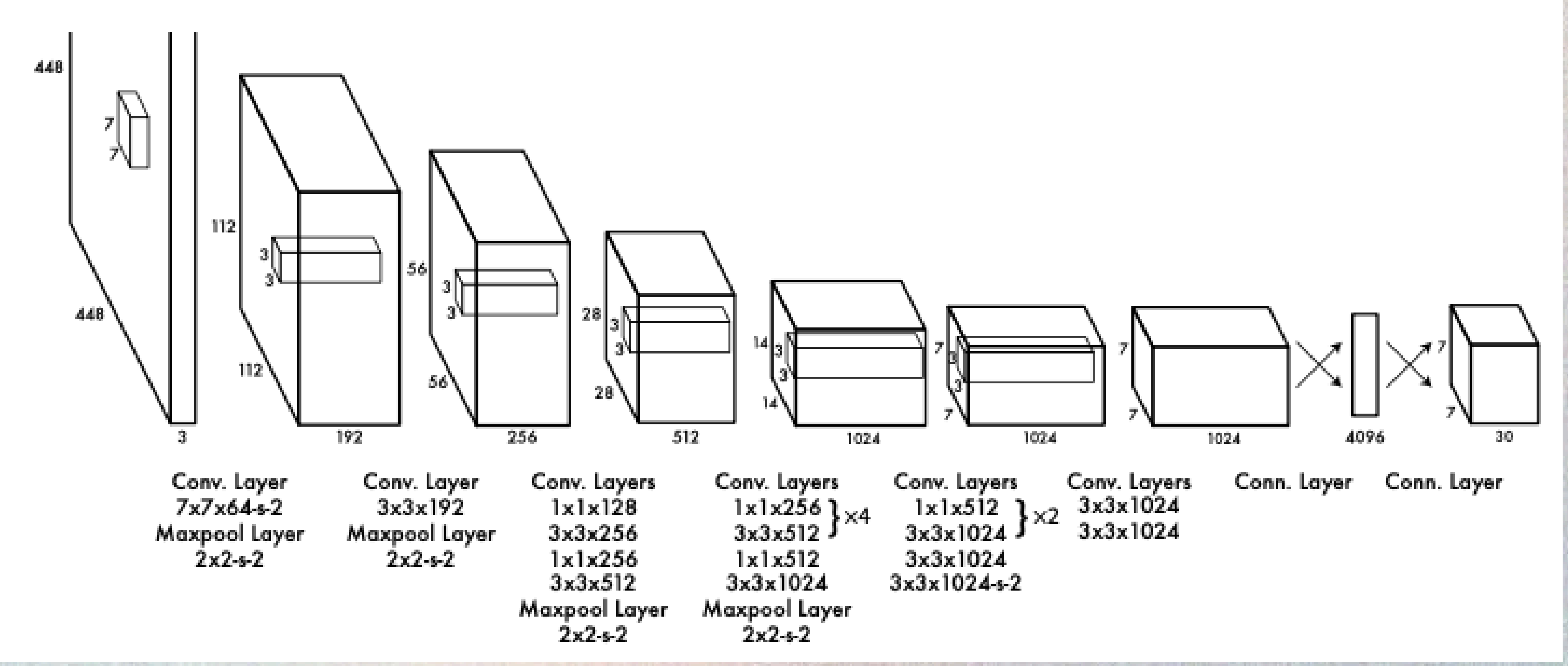
NETWORK DESIGN



- **INPUT:** 448×448×3 (Width x Height x Channels i.e. RGB)
- **Convolutional Layer:** applies filters to learn features (locally)
- **Maxpooling Layer:** downsampling while preserving most important information
- **Fully Connected Layer:** combines features globally and produces a final prediction (the last one)
- as we go forward, the no. of channels grows (more info) and step-by-step the model sees more and more of the whole image

NETWORK DESIGN





- **1 × 1 convolutions = bottlenecks**
- **without them:**
  - computational inefficiency
  - over-parametrisation
  - inability to combine features across channels

- **FINAL OUTPUT:** $7 \times 7 \times 30$ tensor of predictions ($B = 2$, $C = 20$)

TRAINING

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

- **pretraining:** 20 conv. layers + average pooling layer + FC layer, ImageNet, ca. 1 week
    - 88% accuracy on the ImageNet 2012 validation set
- these layers are **fine-tuned for detection tasks** (+4 conv. layers, +2 FC layers with randomly initialised weights)
- for detection: **resolution is increased** from 224x224 to 448x448 for better detail capture
- **leaky ReLU** activation (slope 0.1 for negative inputs i.e. unfavourable inputs) is used throughout the network

LOSS FUNCTION

$$Loss_{yolo} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] + \longrightarrow \text{Bounding Box coord}$$

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(C_i - \hat{C}_i \right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left[\left(C_i - \hat{C}_i \right)^2 + \longrightarrow \text{Confidence}$$

$$\sum_{i=0}^{S^2} 1_{ij}^{noobj} \sum_{C \in classes} \left[\left(p_i(C) - \hat{p}_i(C) \right)^2 \right] \longrightarrow \text{Classification}$$

$1_{ij}^{noobj} \longrightarrow$ 1 if box j and cell i no match,
0 otherwise

LOSS FUNCTION

Balancing the Loss:

- λcoord=5: Increases the weight of localization errors (to ensure the boxes are positioned well).
- λnoobj=0.5: Decreases the weight of confidence errors for cells that don't contain objects, ensuring background grid cells don't dominate the loss.

Sum of Squared Errors:

- Each grid cell also predicts class probabilities for the object present.
- Only one set of class probabilities is predicted per grid cell, even if multiple boxes are predicted.

Handling Multiple Boxes:

- Each grid cell predicts multiple bounding boxes, but during training, only one bounding box is assigned as "responsible" for each object based on the highest IoU with the ground truth.
- This specialization improves overall recall by having each box predictor specialize in different object shapes and sizes.

LOSS FUNCTION

Localization loss

GT bbox x-coordinate in the ith cell

GT bbox y-coordinate in the ith cell

Predicted bbox x-coordinate in the ith cell

Predicted bbox y-coordinate in the ith cell

Set to 5 to increase the loss of bounding box predictions

Sum-squared error

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

For each grid cell

For each grid box

'1' if object appears in the ith cell and the jth box detect it, '0' otherwise

GT bbox height in the ith cell

$$+\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Square root to reduce the range of the values

GT bbox width in the ith cell

Predicted bbox width in the ith cell

Predicted bbox height in the ith cell

LOSS FUNCTION

Confidence loss

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(C_i-\hat{C}_i\right)^2\right]$$

GT confidence score

Predicted confidence score

Confidence error when an object is detected in the ith cell

Set to 0.5 to decrease the loss for empty boxes

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left[\left(C_i-\hat{C}_i\right)^2\right]$$

'1' if there is no object in the ith cell, '0' otherwise

Confidence error when an object not detected in the ith cell

LOSS FUNCTION

Clasification loss

Predicted conditional probability of an object of class c appearing in the ith cell

$$+\sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c\in classes} \left[(p_i(c) - \widehat{p}_i(c))^2\right]$$

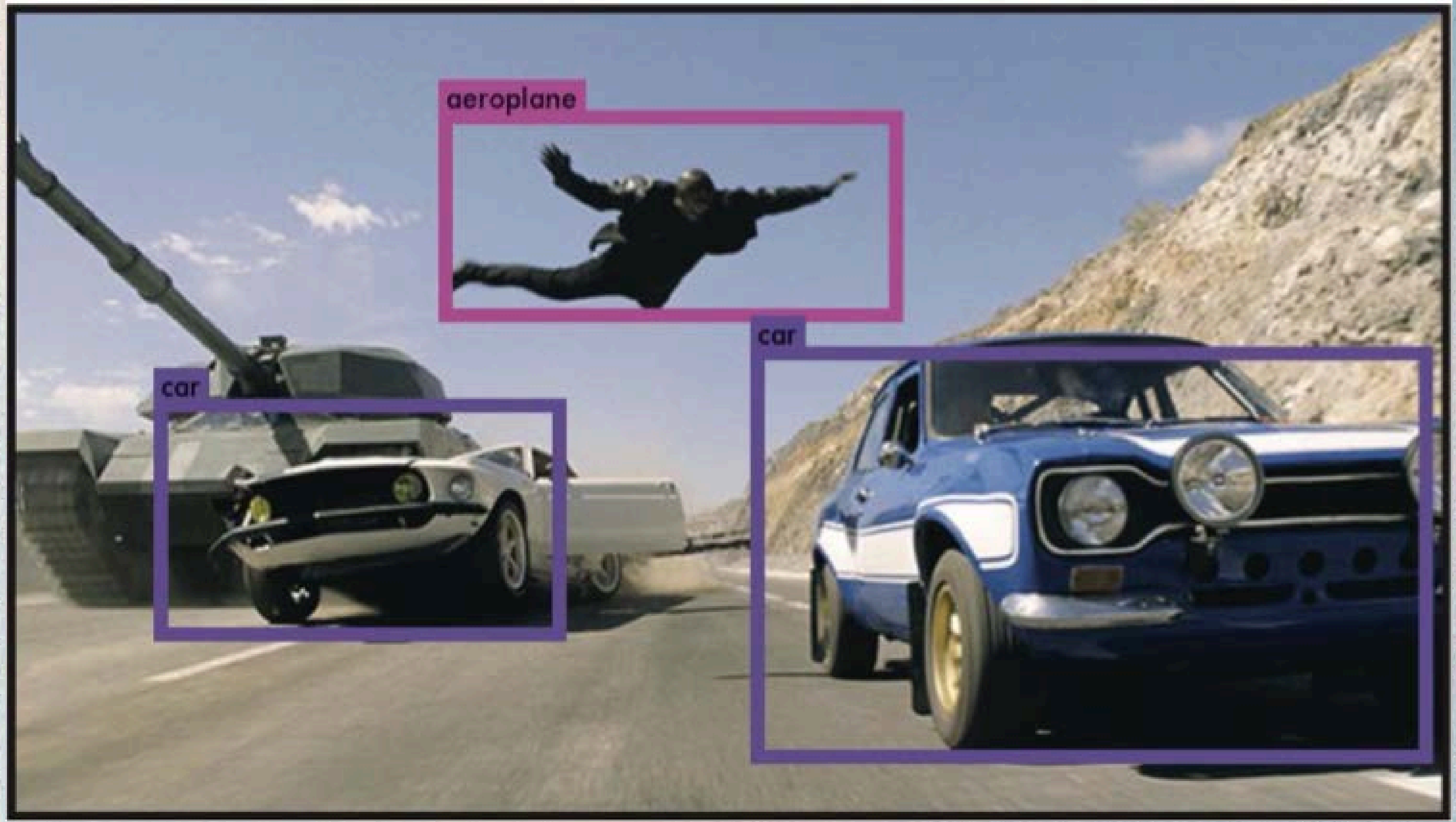
For each grid cell

For each class

GT conditional probability of class c appearing in the ith cell

LIMITATIONS

- YOLO imposes strong spatial constraints on bounding boxes predictions, and struggles with small objects that appear in groups, like flocks of birds.
- As bounding boxes are predicted from learnt data, it struggle to generalize to different aspect ratios.
- Incorrect localization of bounding boxes is the main source of errors of YOLO

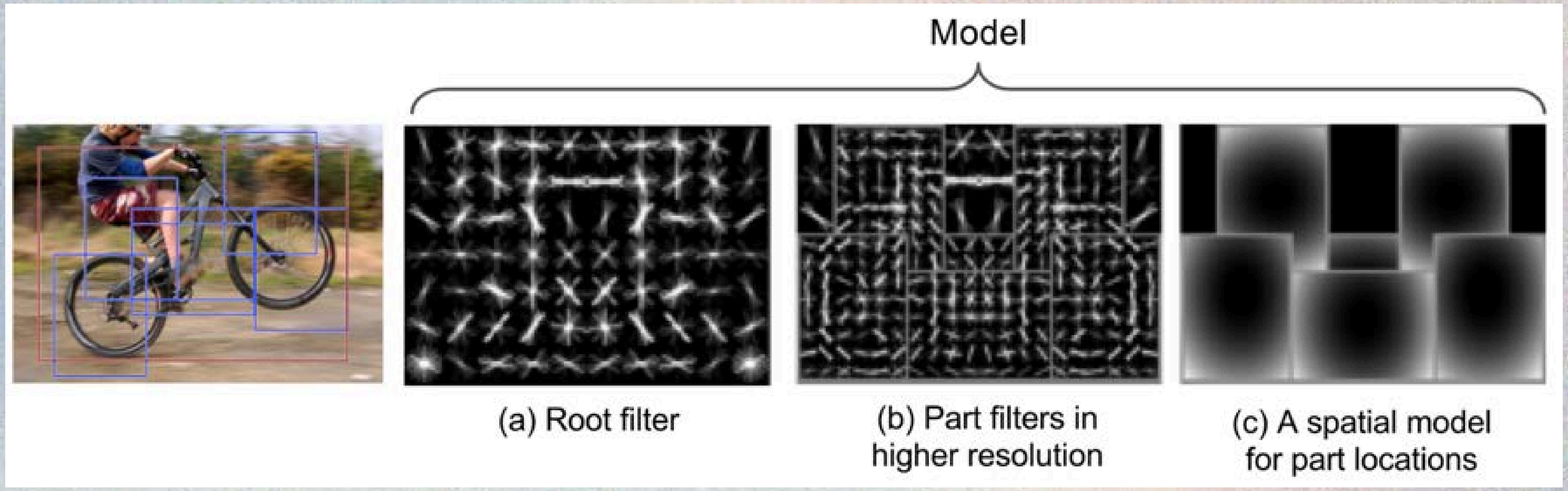


INFERENCE

- During inference on the Pascal VOC dataset, the network predicts 98 bounding boxes per image, along with class probabilities for each.
- The grid design promotes spatial diversity in bounding box predictions.
- Large objects or those near cell borders may be localized by multiple cells, with non-maximal suppression applied to refine the results.

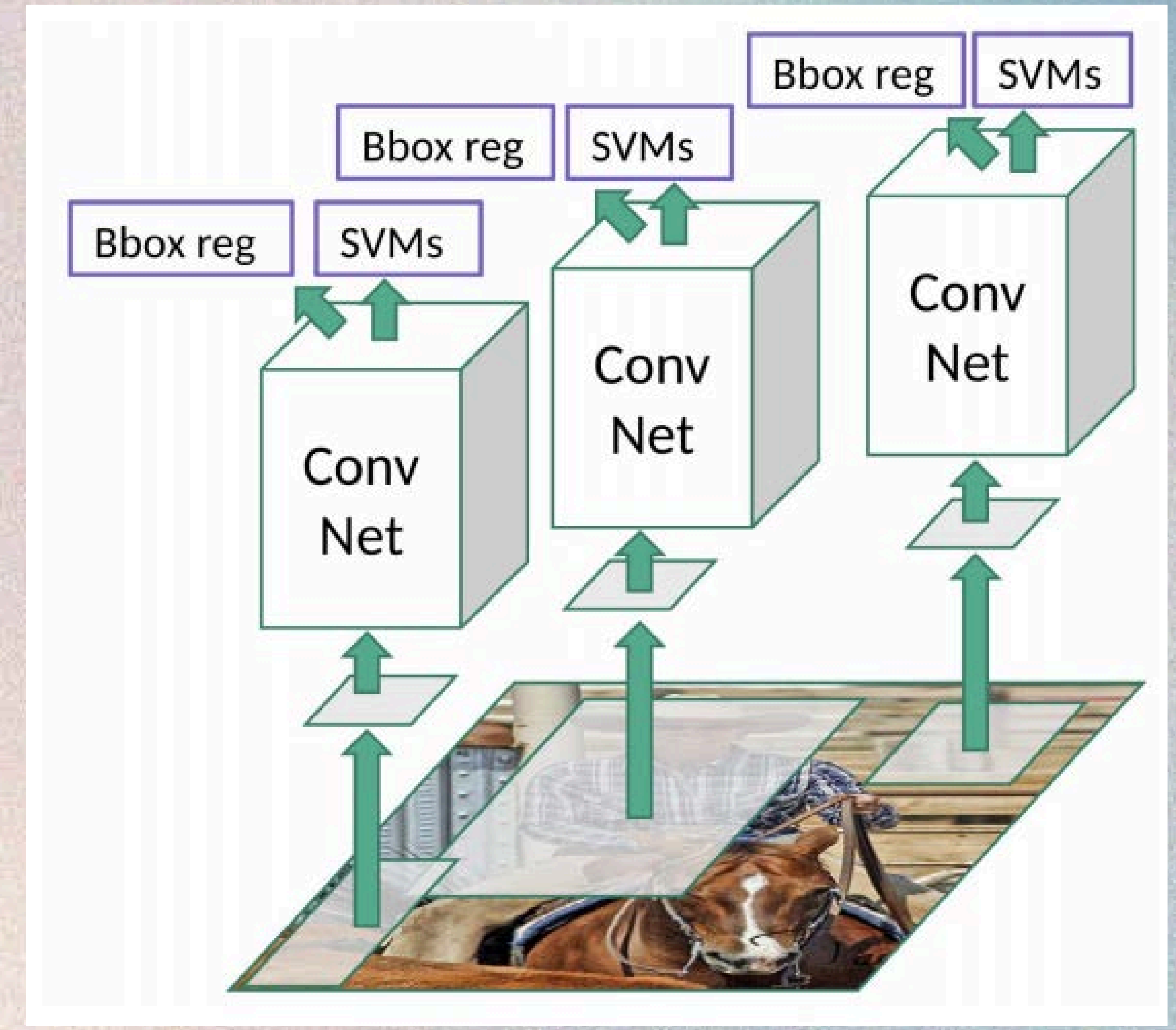
COMPARISON TO OTHER DETECTION SYSTEMS

- **Deformable parts model (DPM) (2010)**
  - sliding window approach --> **disjoint** pipeline
  - it “sees” the object in parts: good when there’s occlusion



COMPARISON TO OTHER DETECTION SYSTEMS

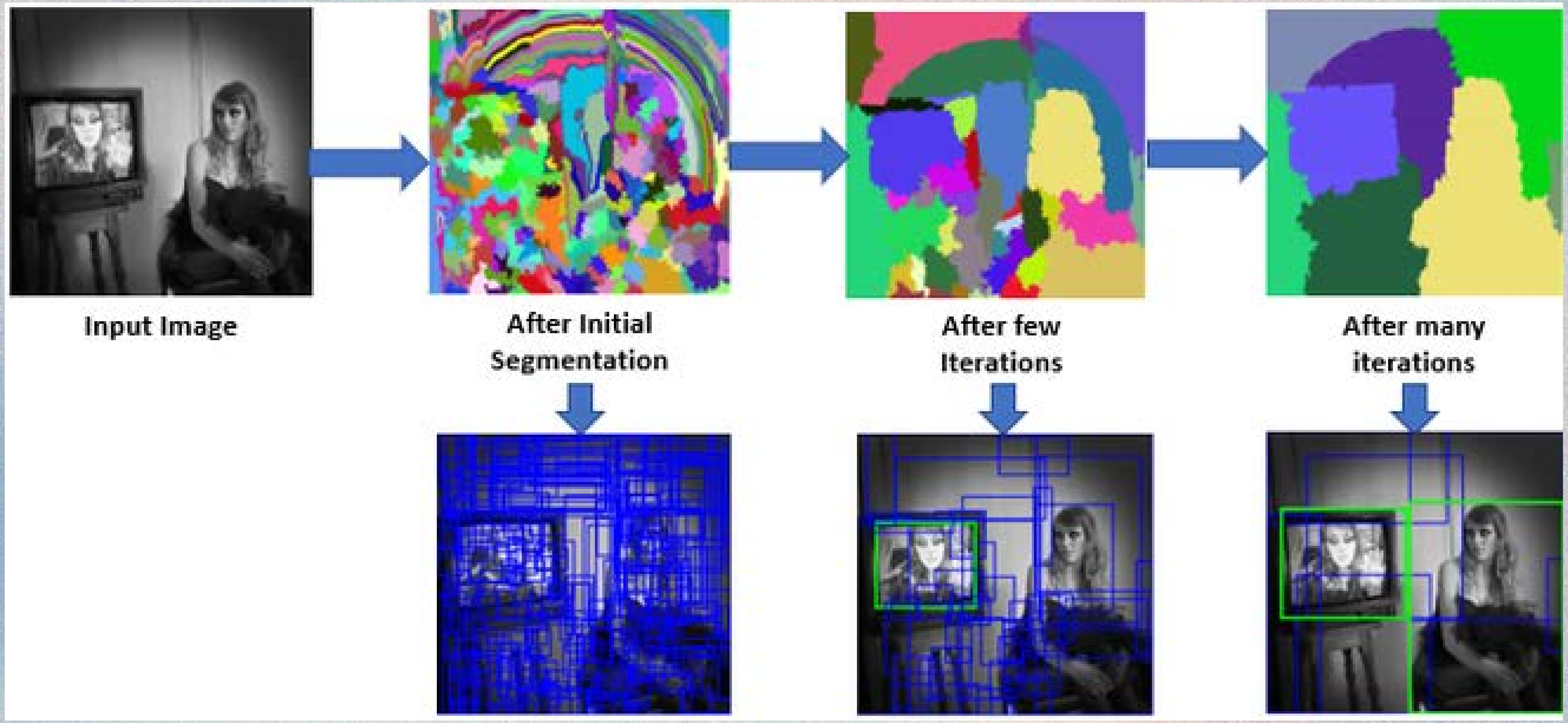
- **R-CNN (2013)**
  - Selective Search for region proposals (candidate regions where objects might be located)
  - CNN to extract features
  - SVM to score the boxes
  - linear model to adjust the bounding boxes
  - non-max suppression to eliminate duplicate detections
  - complex pipeline; each part is tuned indipendently --> very slow (more than 40 seconds per image)



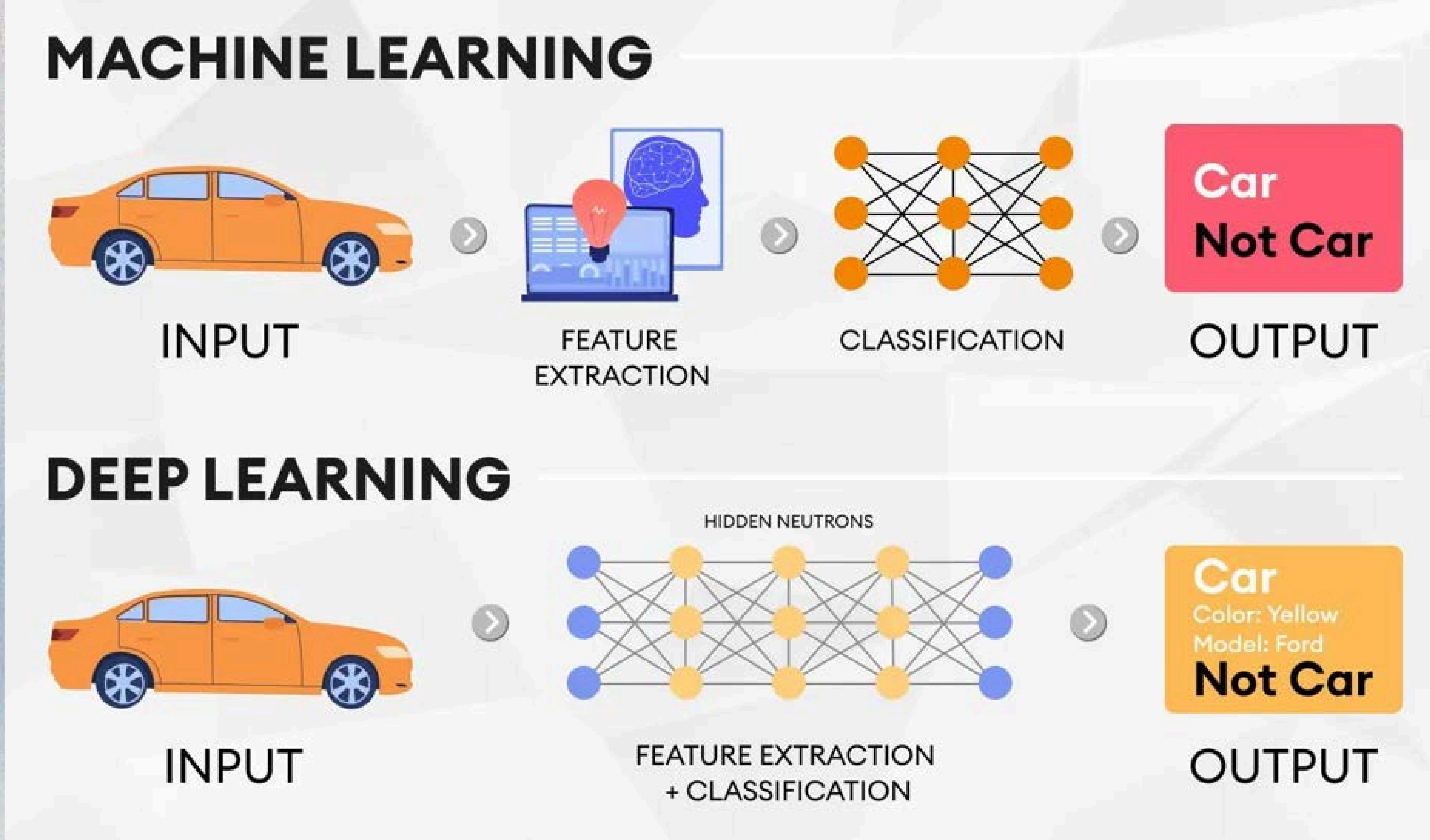
COMPARISON TO OTHER DETECTION SYSTEMS

- **Fast & Faster R-CNN (2015)**
  - speeding up R-CNN
  - neural networks instead of Selective Search
  - no real-time performance
- **Deep MultiBox (2014)**
  - CNN to predict regions of interest, instead of Selective Search
  - not a complete detection system
- **Overfeat (2013)**
  - pioneer in DL-based object detection (~AlexNet)
  - optimises for localisation, cannot reason about global context

SELECTIVE SEARCH



SELECTIVE SEARCH VS CNN

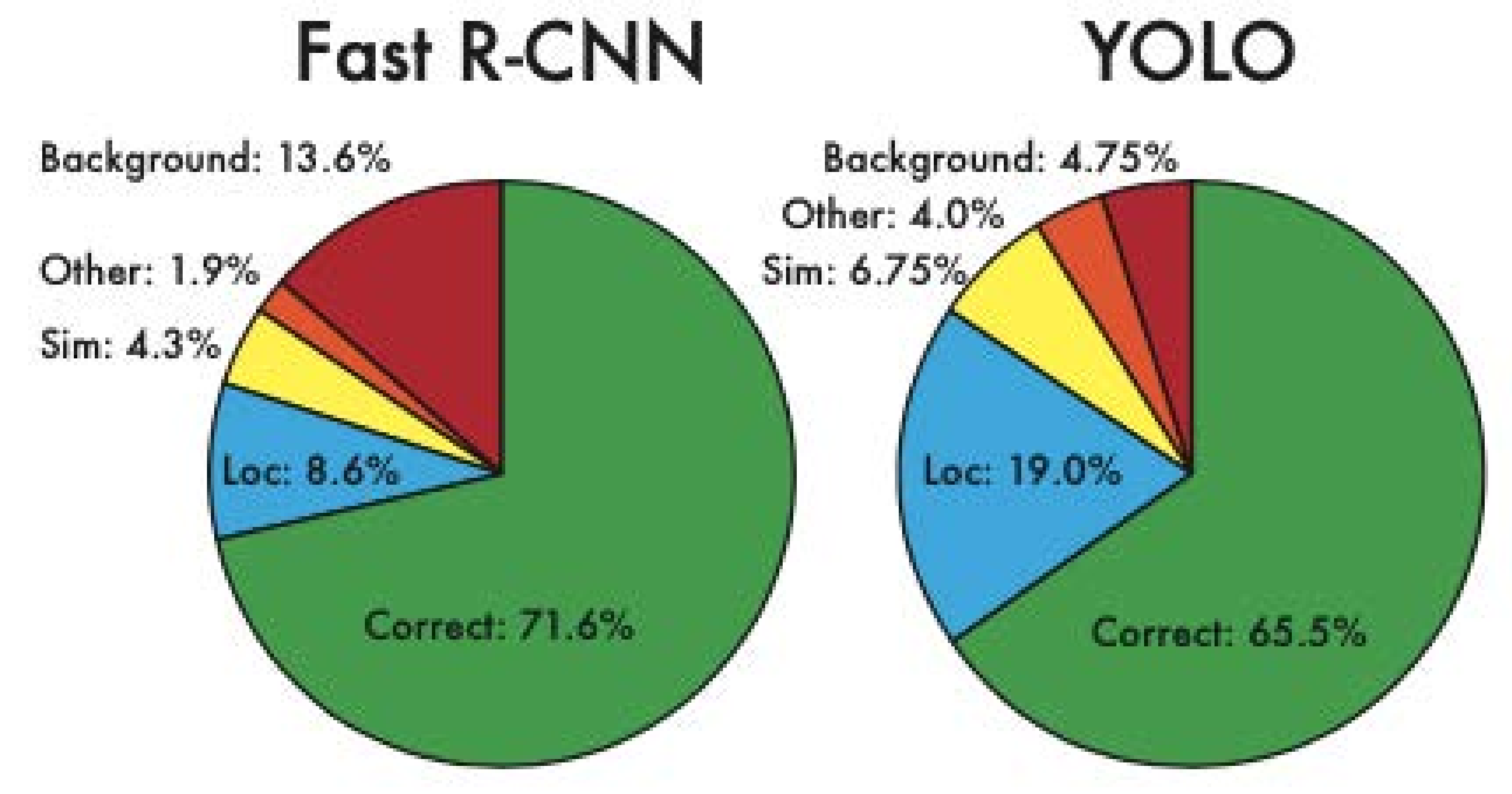


BENEFITS OF A UNIFIED MODEL

- **YOLO is extremely fast**
  - base version: 45 fps
  - fast version: >150 fps
  - can process streaming video real-time (<25 ms latency)
- **YOLO reasons globally**
  - sees entire image during training and test
  - less background errors
- **YOLO is highly generalisable**
  - less likely to break down when applied to new domains

however: **less accuracy**

ENSEMBLE WITH FAST R-CNN

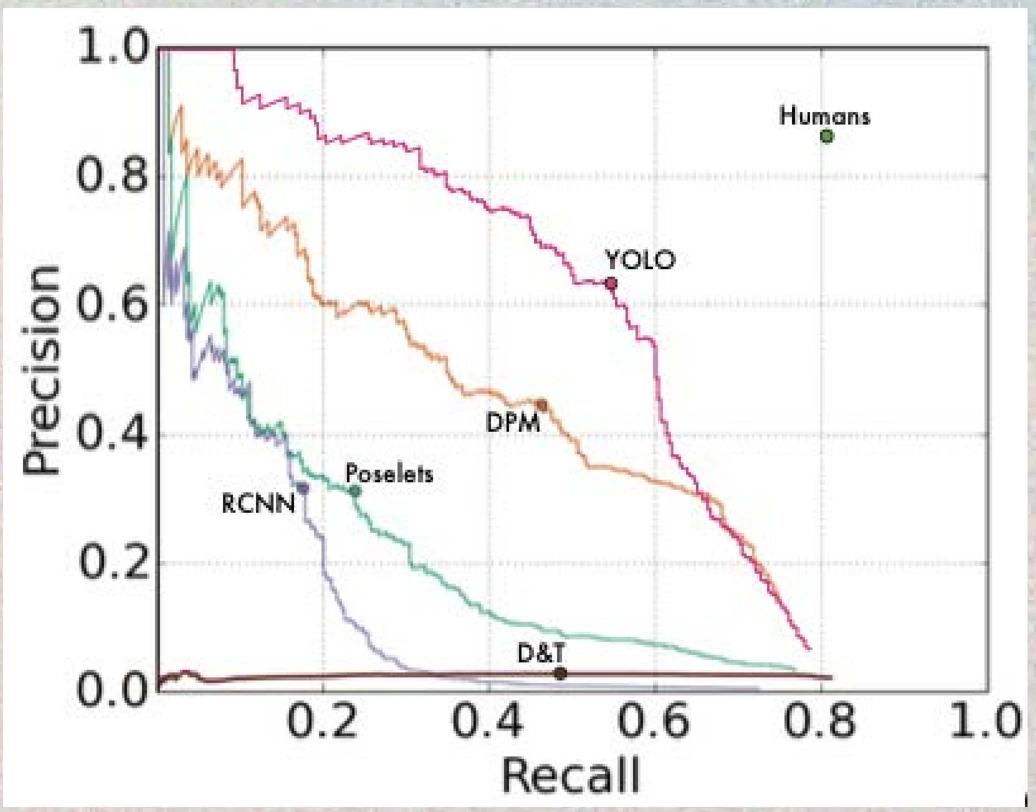


- Fast R-CNN: fewer localisation errors, more background errors
- combination: YOLO to eliminate background detections from Fast R-CNN
  - YOLO checks bounding boxes predicted by R-CNN: boost for similar box
- Fast R-CNN + YOLO: Fast R-CNN mAP increases by 3.2% to 75%
- given YOLO's speed, no significant additional computational time

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

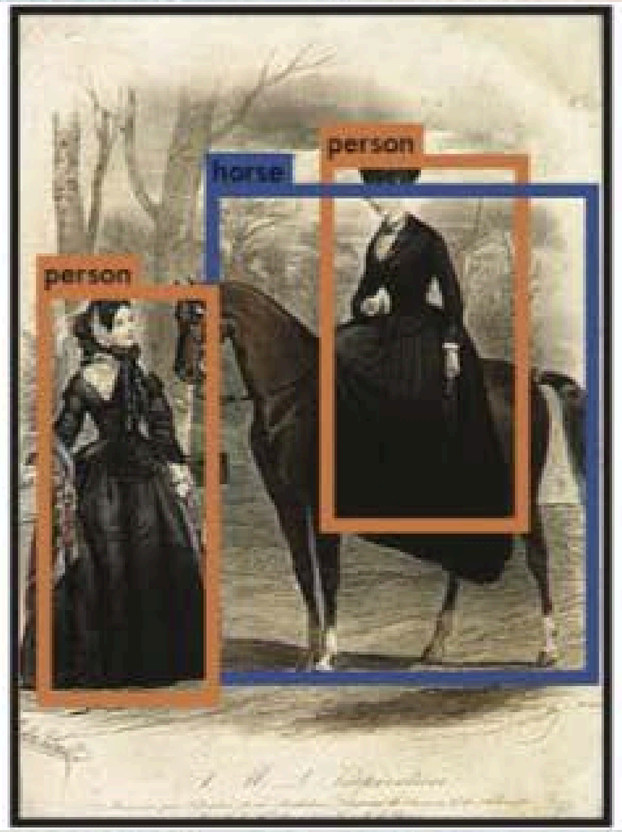
PASCAL VOC LEADERBOARD (NOVEMBER 2015)

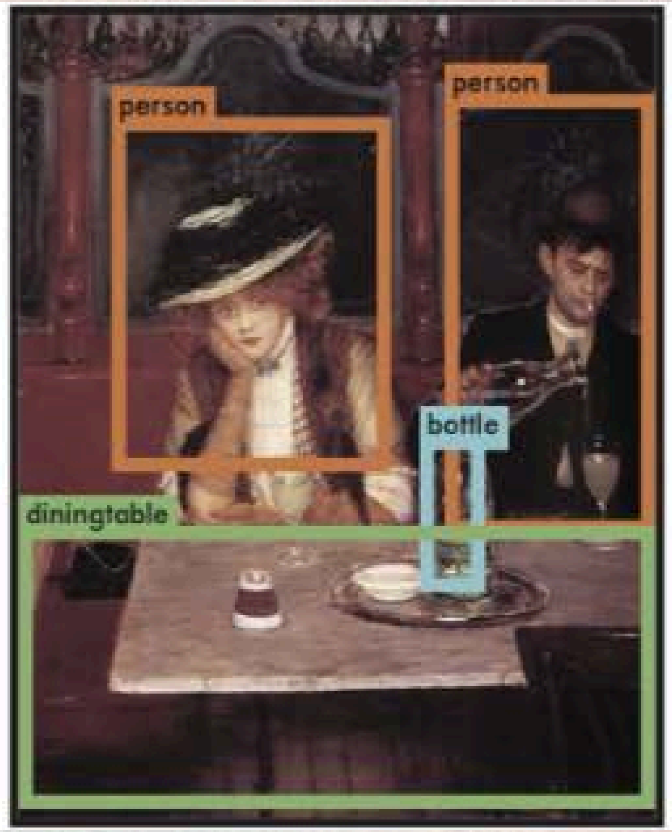
GENERALIZABILITY TO ARTWORK DATASETS

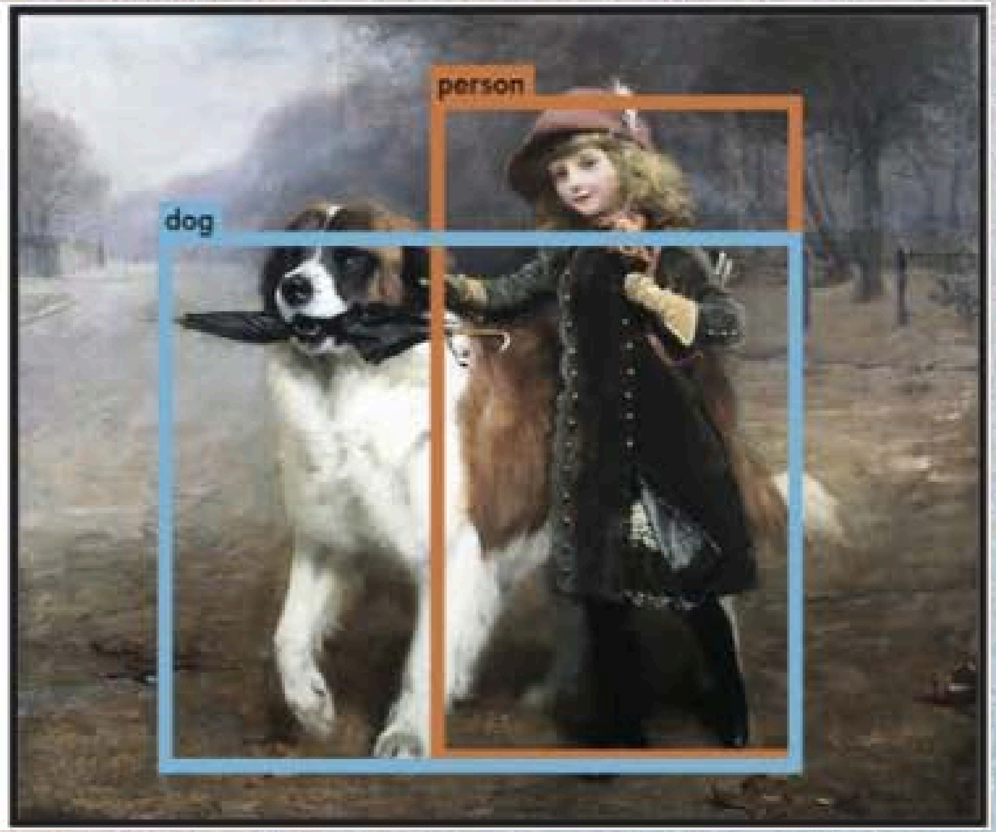


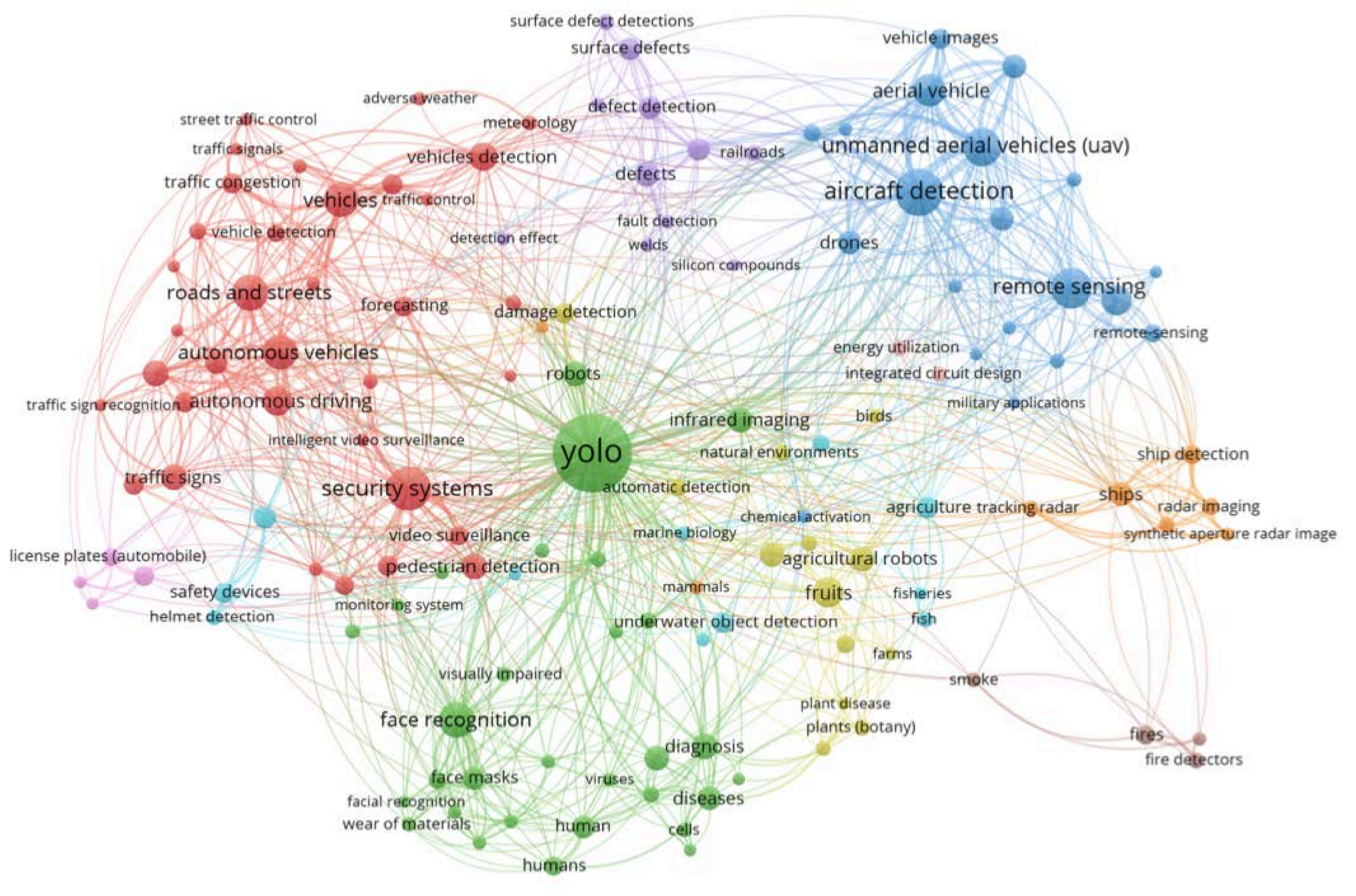
YOLO's AP degrades the least on the artwork datasets, as they're similar in terms of size and shape of objects.

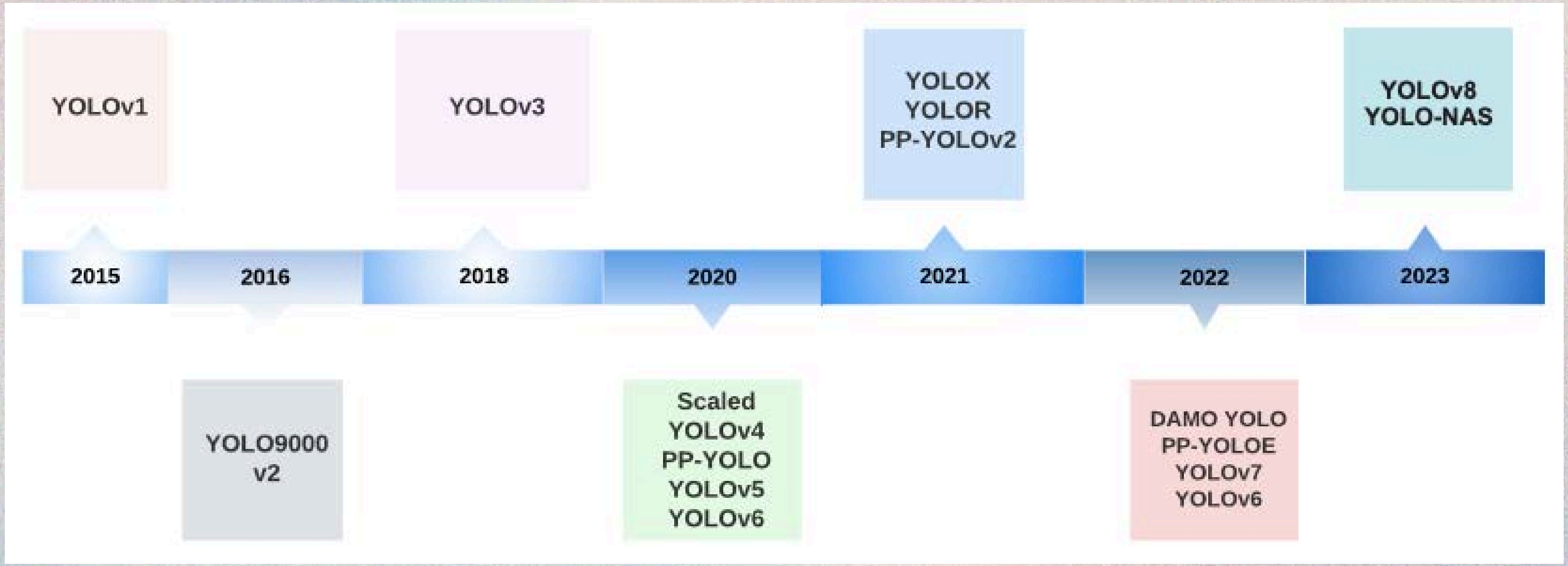
Picasso dataset








surface defect detections
surface defects
vehicle images
aerial vehicle
adverse weather
street traffic control
defect detection
meteorology
traffic signals
vehicles detection
railroads
unmanned aerial vehicles (uav)
traffic congestion
defects
aircraft detection
vehicles
traffic control
fault detection
vehicle detection
detection effect
welds
drones
silicon compounds
roads and streets
remote sensing
forecasting
damage detection
remote-sensing
energy utilization
autonomous vehicles
robots
integrated circuit design
traffic sign recognition
autonomous driving
military applications
birds
infrared imaging
intelligent video surveillance
yolo
natural environments
ship detection
traffic signs
automatic detection
security systems
ships
radar imaging
chemical activation
agriculture tracking radar
video surveillance
marine biology
synthetic aperture radar image
license plates (automobile)
agricultural robots
pedestrian detection
safety devices
mammals
fruits
fisheries
monitoring system
helmet detection
underwater object detection
fish
farms
visually impaired
smoke
plant disease
face recognition
plants (botany)
fires
diagnosis
fire detectors
face masks
viruses
diseases
facial recognition
wear of materials
human
cells
humans


YOLOv1
YOLOv3
YOLOX
YOLOR
PP-YOLOv2
YOLOv8
YOLO-NAS
2015
2016
2018
2020
2021
2022
2023
YOLO9000
v2
Scaled
YOLOv4
PP-YOLO
YOLOv5
YOLOv6
DAMO YOLO
PP-YOLOE
YOLOv7
YOLOv6

final words

THANK YOU