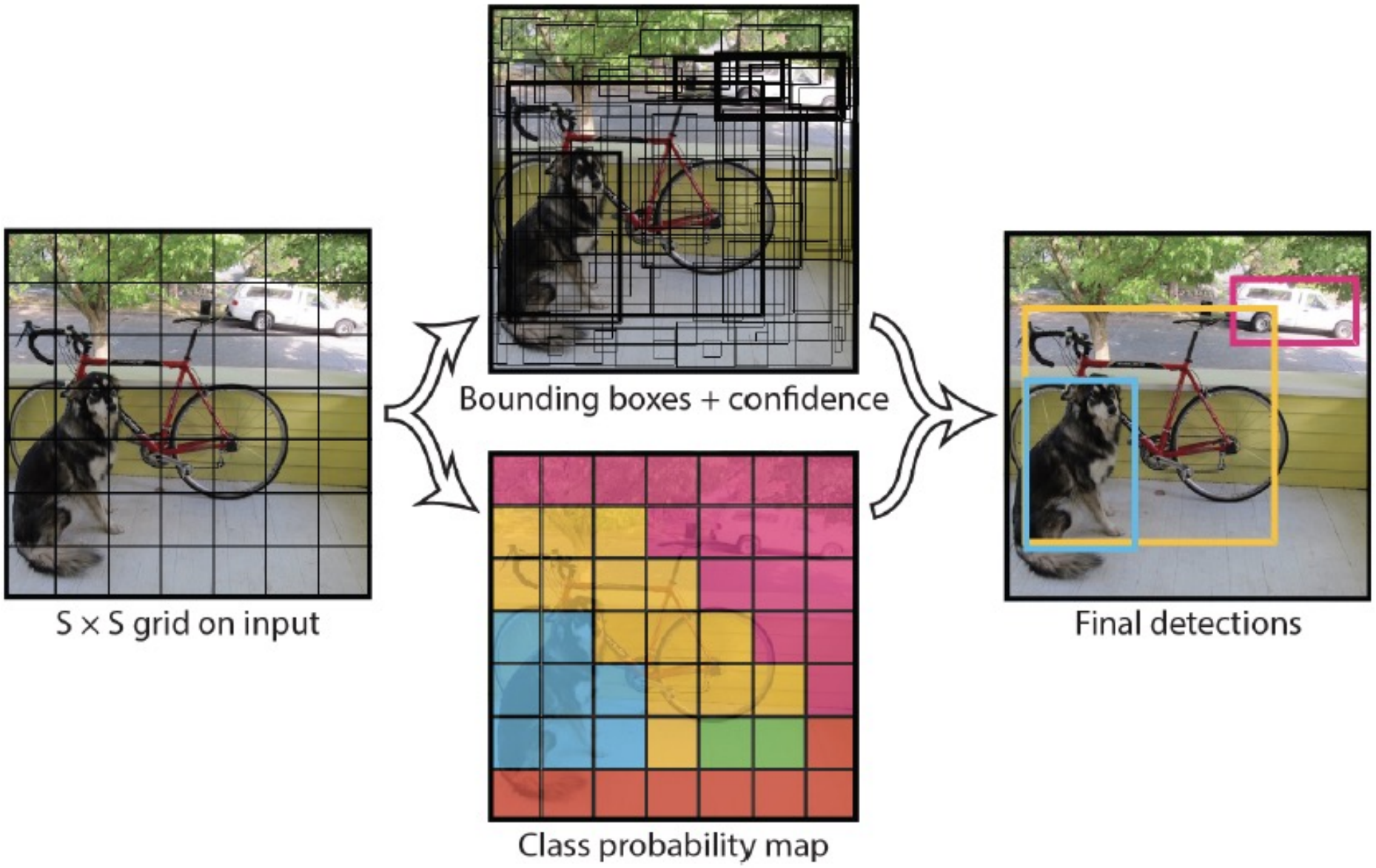
Object detection: Where and What together

D.A. Forsyth

Yolo

- Predict box objectness score, label scores indep.

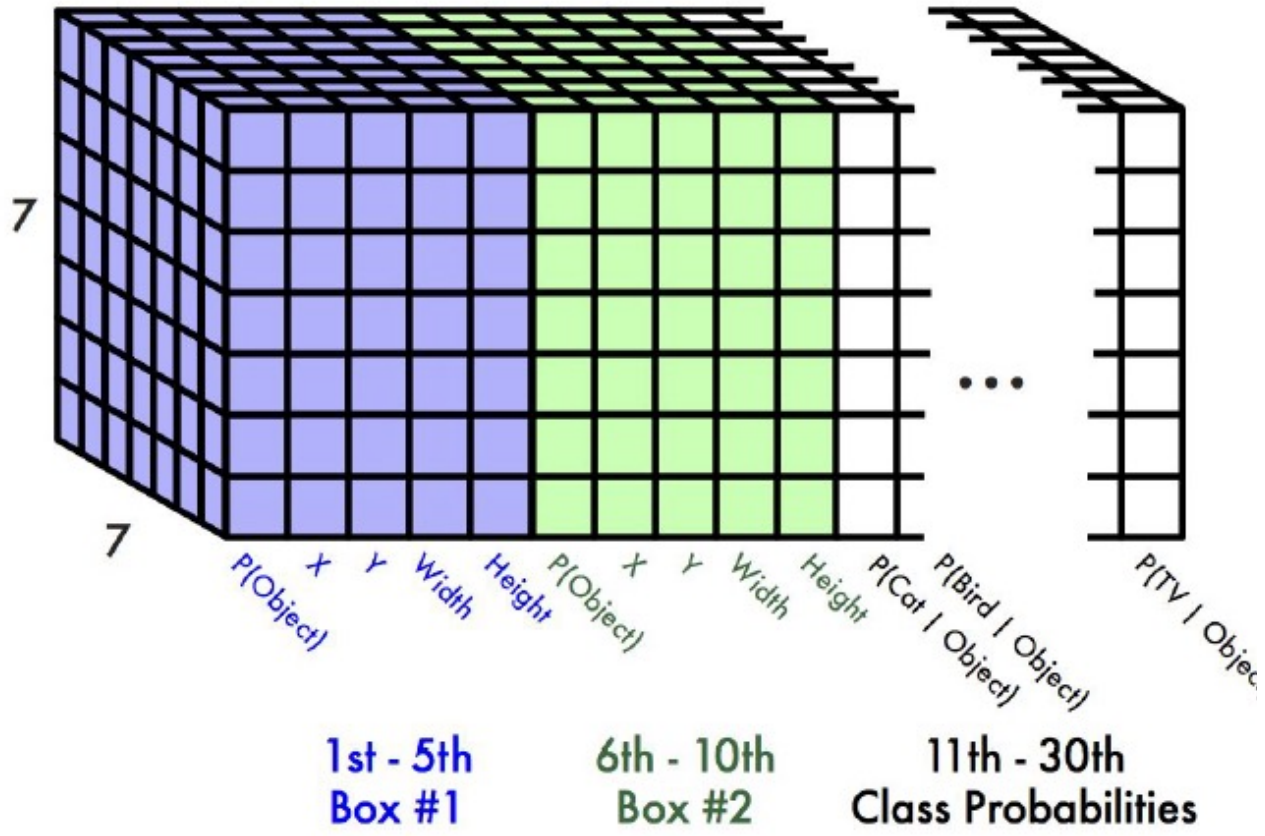
- Process:
  - split image into grid
  - each cell predicts box location, confidence
  - each cell predicts a class probability
  - multiply box and class probabilities
  - cleanup
    - threshold, NMS


Bounding boxes + confidence
S × S grid on input
Class probability map
Final detections

This parameterization fixes the output size

Each cell predicts:

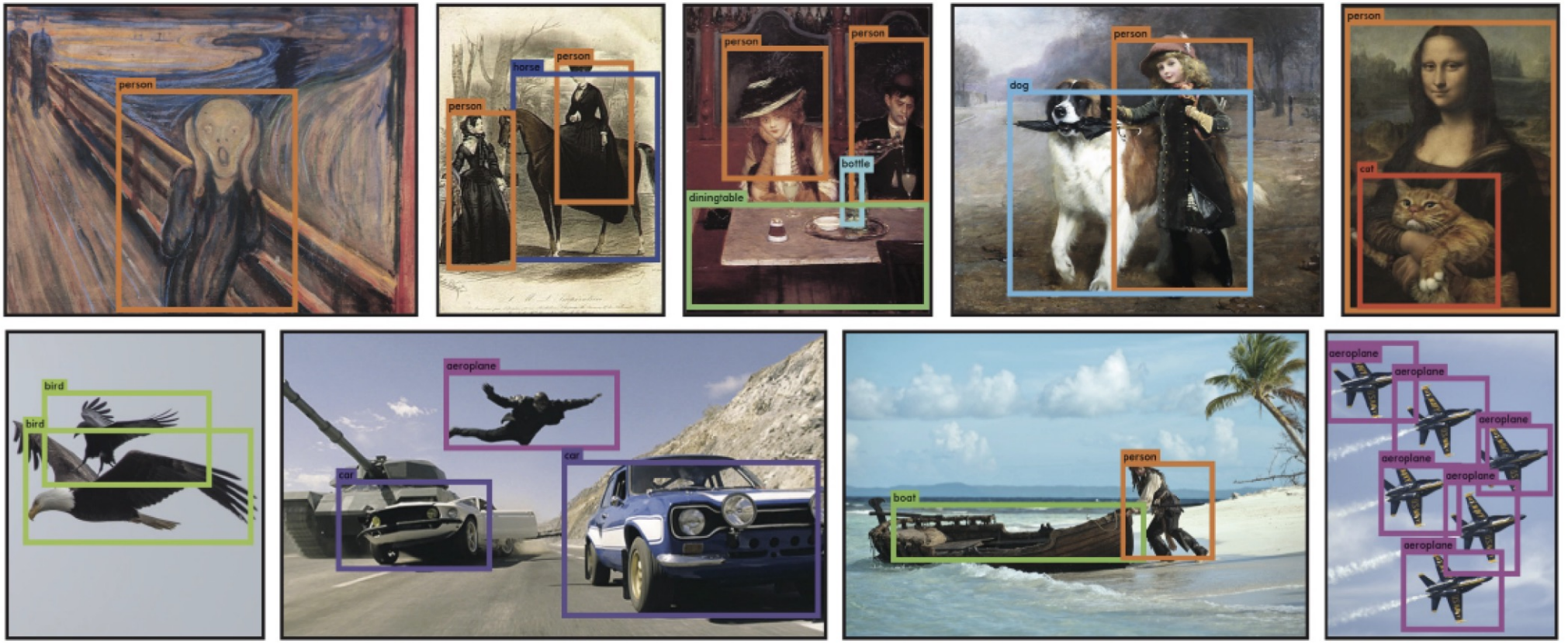
- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities



For Pascal VOC:

- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

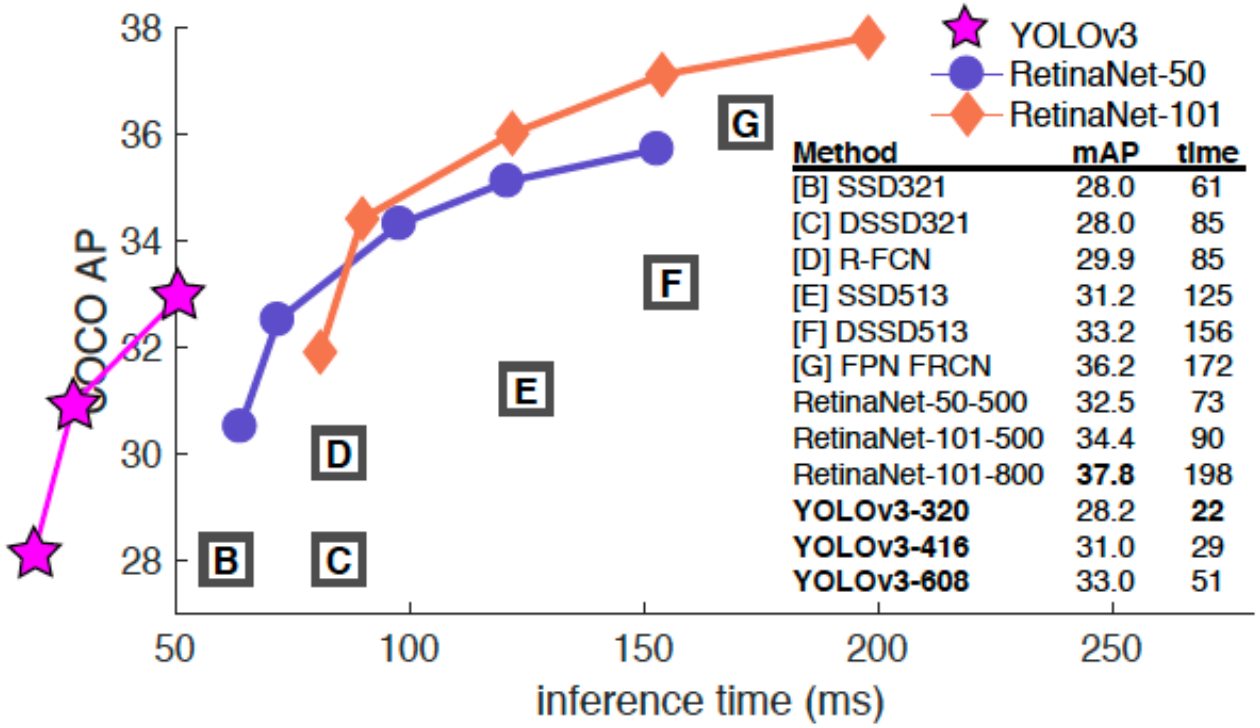
7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**



- Original caption: "*YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane*"

Credit: Figure 5 of You Only Look Once: Unified, Real-Time Object Detection
Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

Speed/accuracy payoffs



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

Figure from: "*YOLOv3: An Incremental Improvement*"

Yolo variants are maintained

- At V.11 now
- Still very widely used
- Training is relatively straightforward