# Object Detection: Convolutional Neural Networks and YOLO

Ben Cravens

2023

# What is Object Detection, why YOLO?

- Detect instances of objects in images
- Drawing a bounding box around detected objects
- Report class and confidence score (0.0-1.0)
- Two types: two stage, one stage
- Two stage too slow for realtime detection!
- YOLO first one stage detector, enabled real time detection

# Object Detection: Graphical Explanation

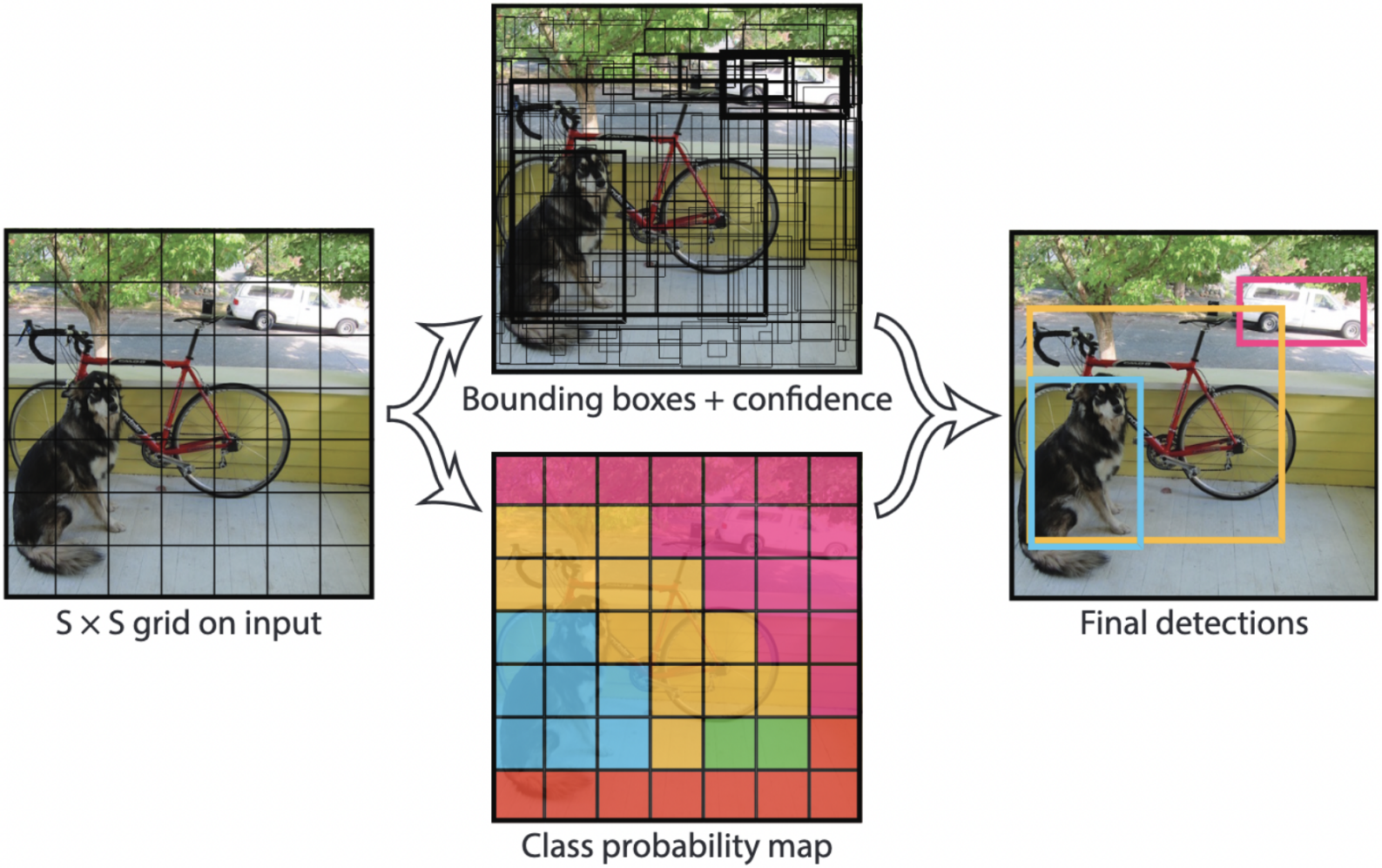



Figure: From YOLOv1 paper

# CNN(1/4) What are they?

- Convolutional, pooling, and fully connected layers
- Alternate convolutional (finds patterns) and pooling (downsamples)
- Finally feeds to fully connected to combine activations to make a classification

$$(f * g)[n] = \Sigma_{m=-\infty}^{\infty} f[m]g[n-m] \tag{1}$$

- Convolution flips the image g horizontally
- Slides the filter $f$ (which is a grid of numbers) across the image $g$, computing the element wise sum of the filter and the patch it overlaps at each point.
- A spatially invariant measure of "similarity" of $f$ and $g$, high activation if there is a feature in $g$ similar to $f$

# CNN(2/4) Convolution

$$(f * g)[n] = \Sigma_{m=-\infty}^{\infty} f[m]g[n-m] \tag{2}$$

- Weights in the filter $f$ are learned to detect patterns in the image.
- A large output (or "activation") corresponds to a high likelihood the feature $f$ represents occurs in $g$.
- At each convolution layer there may be several filters which each learn different patterns.

# CNN (3/4) Pooling Layers

- Between the convolution layers are pooling layers
- Have no learnable weights
- Simply downsample the image
- Saves memory and compute
- Prevents overfitting
- Gives greater generalisation

# CNN (4/4) Intuition for Object Detection

- Able to learn to detect complex higher order shapes.
- Filters at the initial layers learn simplest features
- I.e corners and lines.
- Activations combined from left to right,
- Later activation layers representing higher level abstract features (such as the "outline" of a face made by combining corners and lines).
- Last layers are feed-forward layers which learn to combine the "presence" of different high-level features to determine a classification probability.

# YOLOv1

- Directly predicts bounding boxes and their classes from input images
- Divides the image into a grid.
- Each grid cell is "responsible" for an object if its center is located inside the grid cell
- The object can be larger than the cell.
- YOLO then predicts B bounding boxes for each grid square.
- Each bounding box also has an associated confidence score
- Given this "object detected" confidence score, YOLO is able to calcuate a likelihood of each class of object being centered in a given grid cell

# YOLO visualised

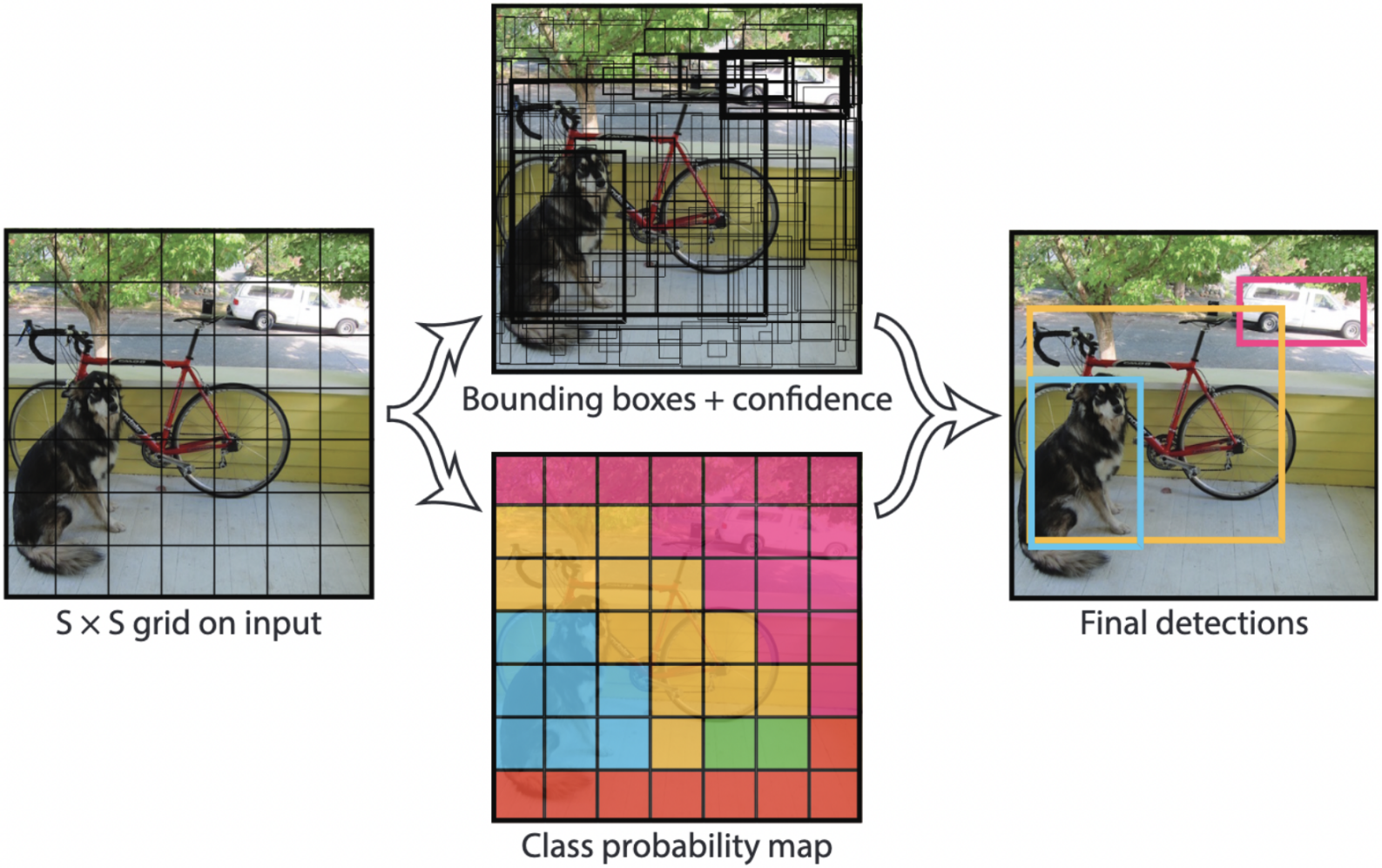
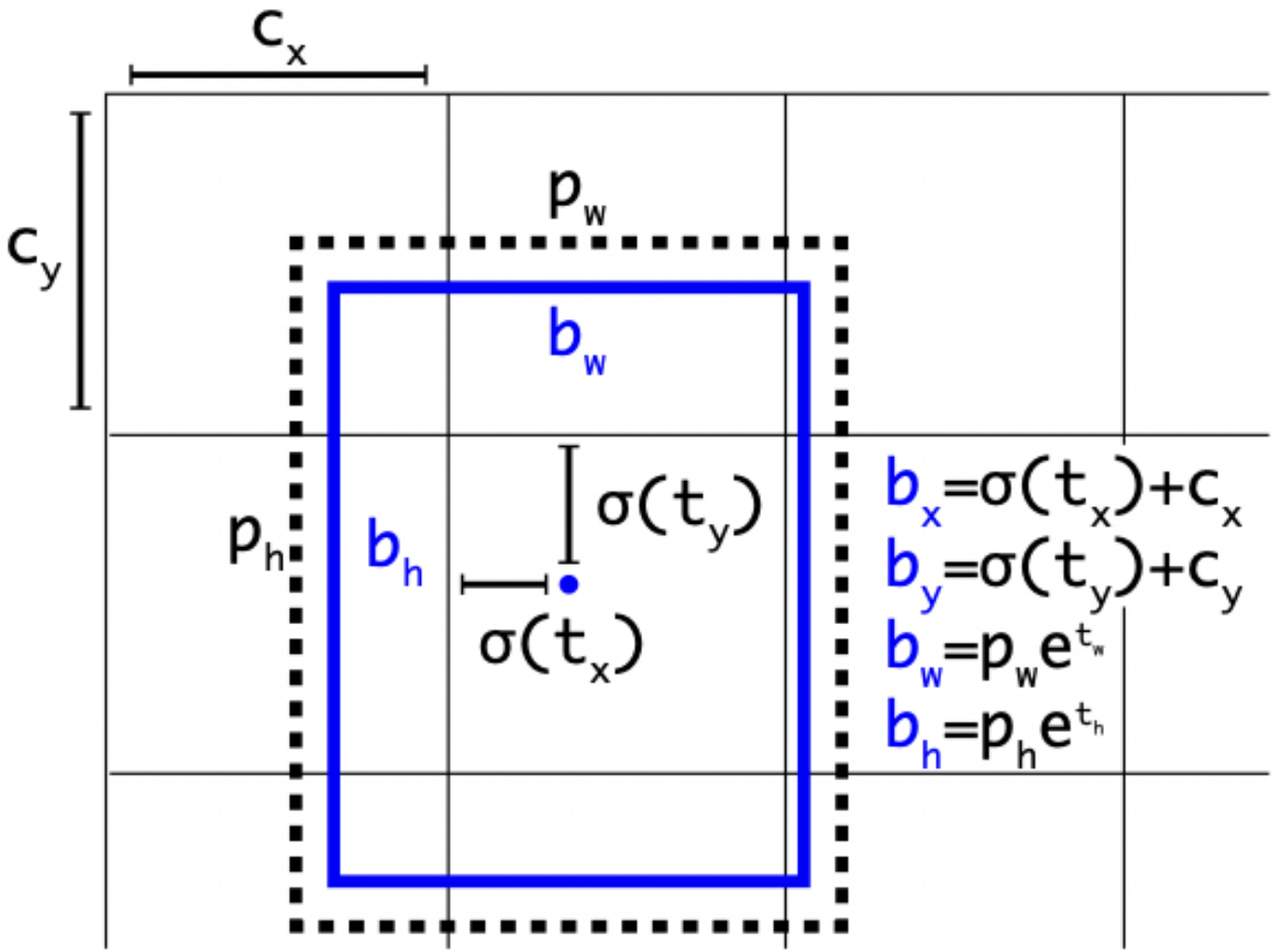



Figure: From YOLOv1 paper

# Architecture

- A deep CNN
- Has 24 convolutional layers followed by 2 fully connected layers.
- The convolutional layers perform feature extraction
- Fully connected layers predict the bounding box based on features

# YOLOv2 upgrade: Anchor boxes



# YOLOv3 upgrade: Darknet-53

- Multi-scale prediction (small,medium,large)
- Multi-class prediction
- Anchor box template k means clustering
- Bigger feature extraction backbone: Darknet-53

# YOLOv* upgrade: Lots of stuff!

- YOLOv5 / v8: ultralytics
- YOLOv7
- Tonnes of development, too complicated to summarise