15-494/694: Cognitive Robotics

Dave Touretzky

Lecture 15:

Object recognition with YOLO
(You Only Look Once)

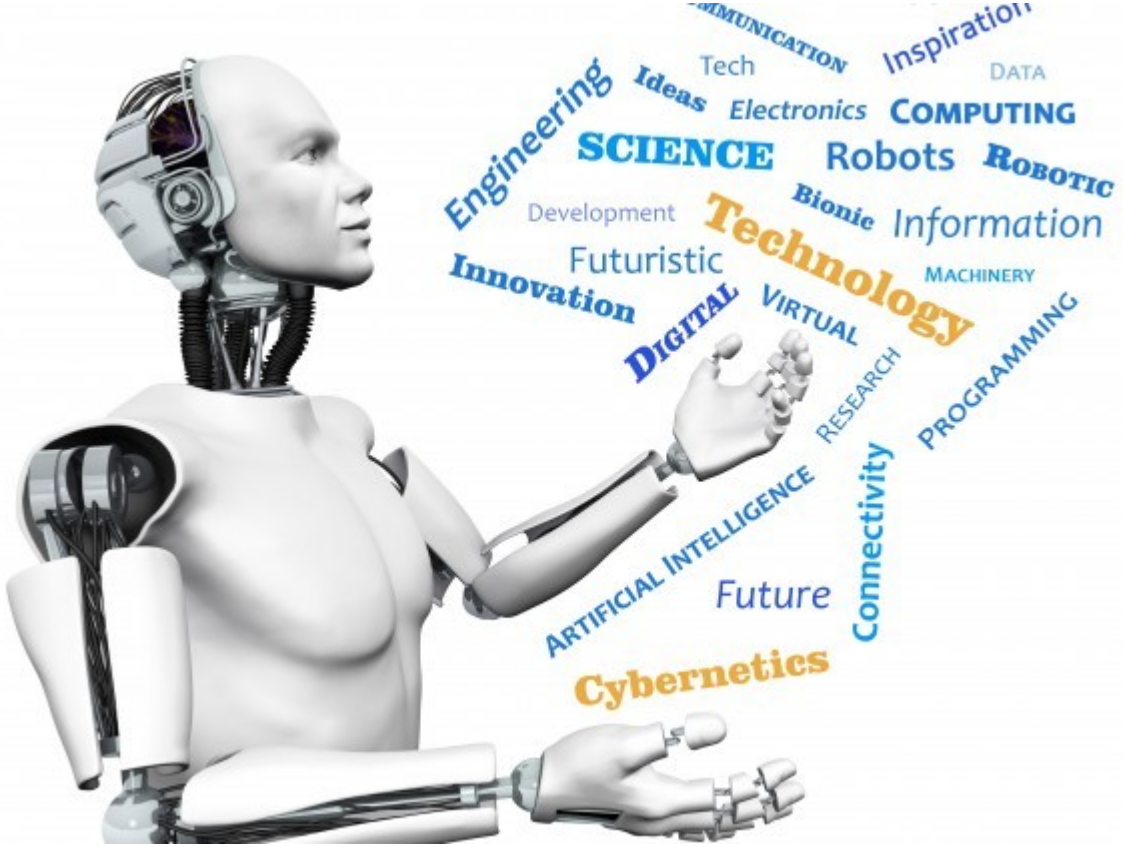
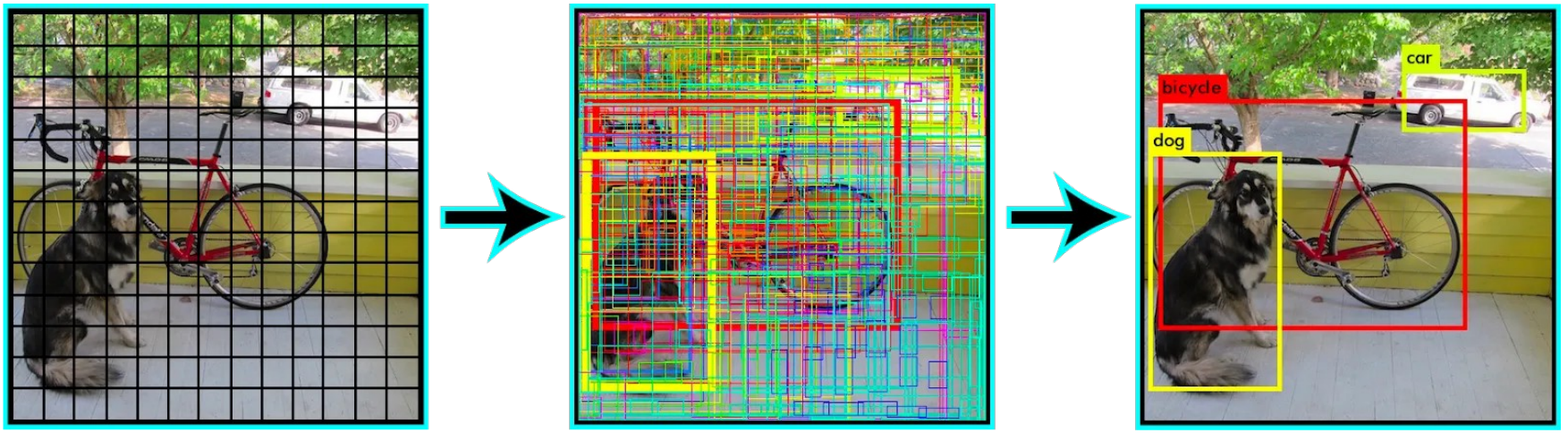


Image from http://www.futuristgerd.com/2015/09/10

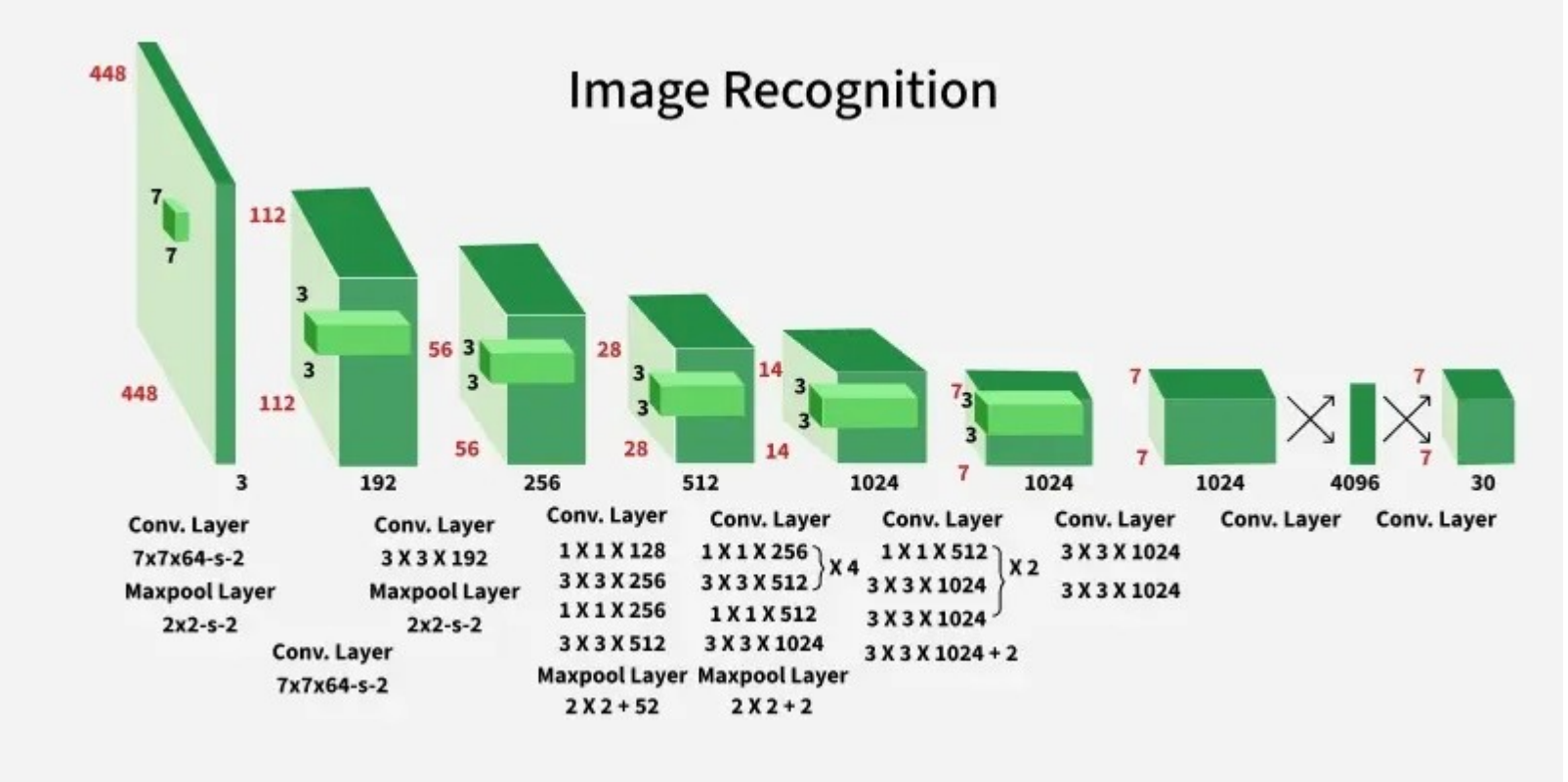
One Pass Object Recognition

- YOLO uses a grid of object detectors.
- Candidate objects are winnowed down using non-max suppression.

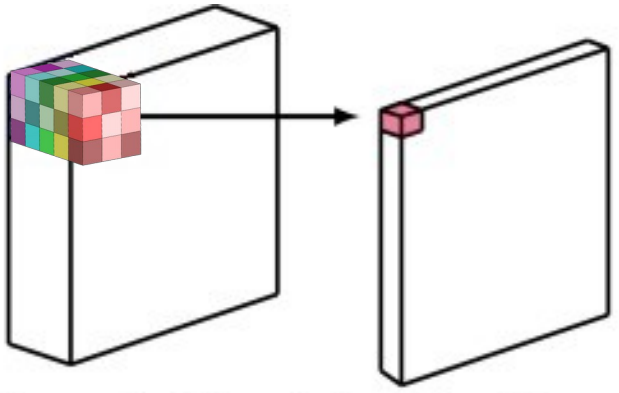


Deep Neural Network

- Mix of depth-wise and point-wise convolutions.



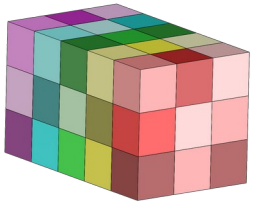
Separable Convolutions

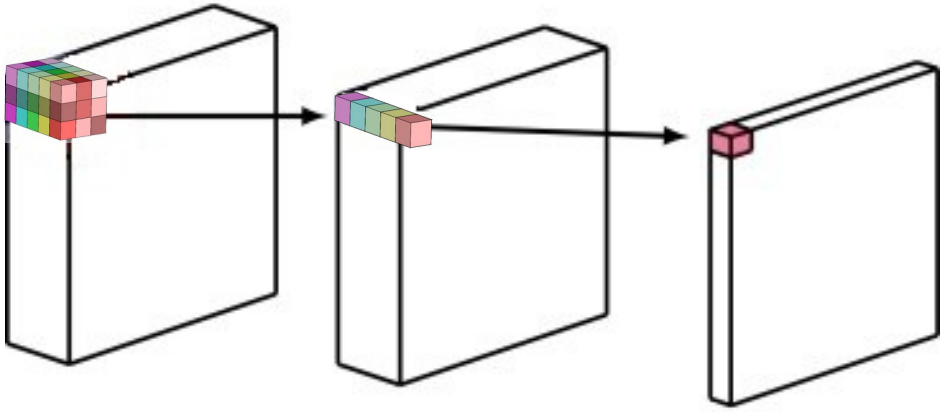


(a) Conventional Convolutional Neural Network

3x3x6 kernel covering 6 channels

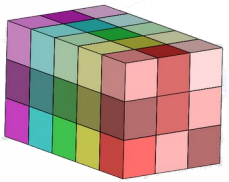
3x3x6 = 54 weights



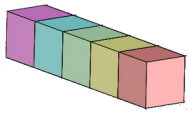


(b) Depthwise Separable Convolutional Neural Network

Depthwise convolution: one 3x3 kernel applied to all 6 channels



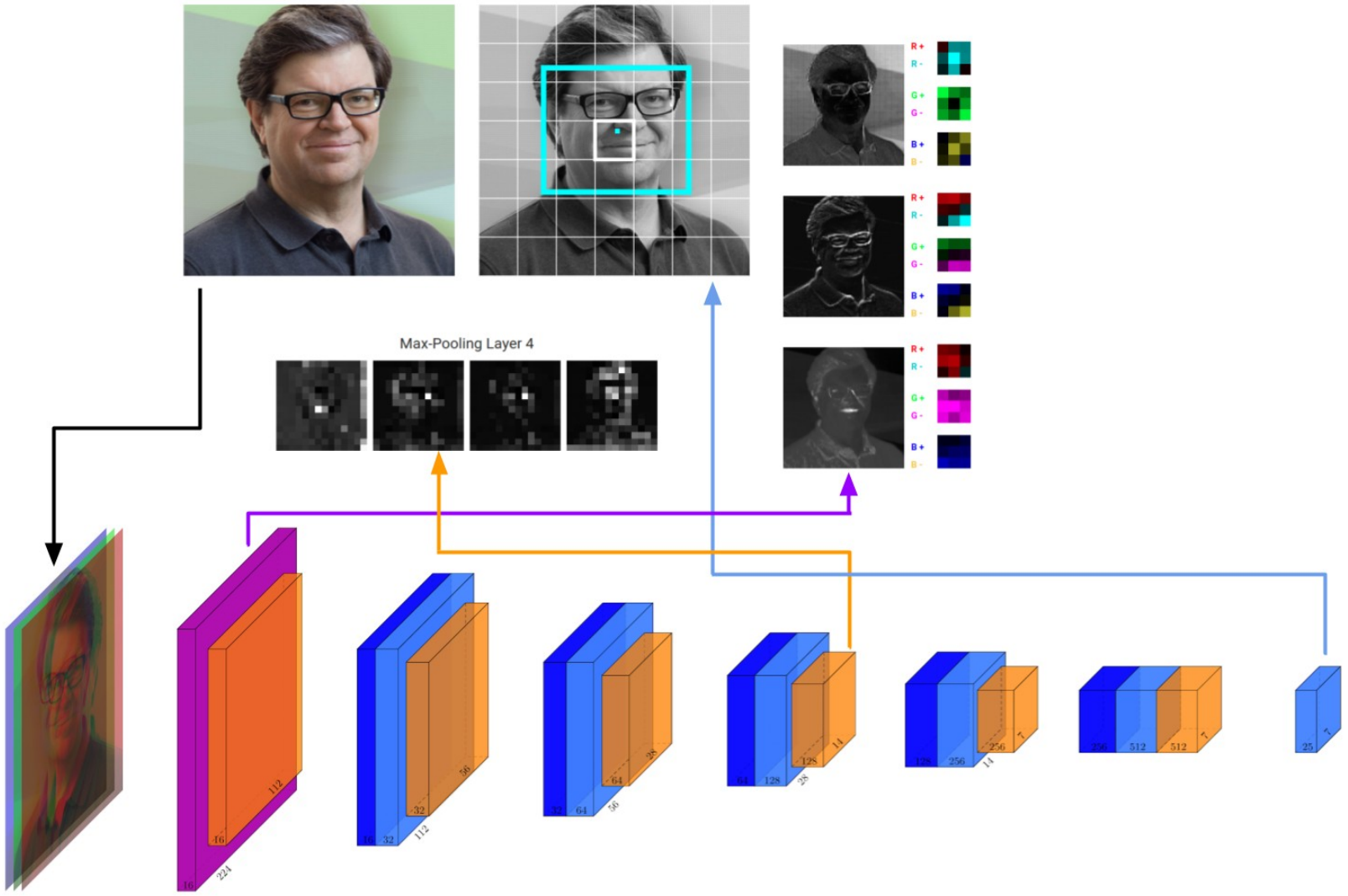
Pointwise convolution: linear weighted combination of one pixel's values across all 6 channels



Combined network:
3x3 + 6 = 15 weights

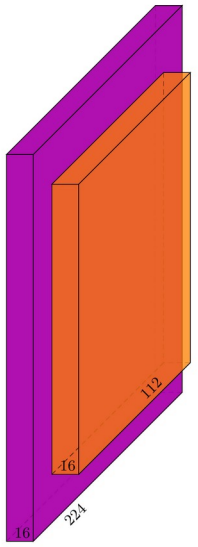
Image modified from
https://www.sciencedirect.com/science/article/pii/S0168169918318696

TinyYOLOV2 Face Recognition



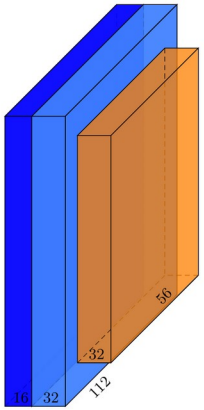
TinyYOLOV2 Architecture

Input Image
R, G, B Channels
224×224×3



conv1: [3,3,3]/1 × 16
224×224×16

max1: [2,2,1]/2 × 16
112×112×16



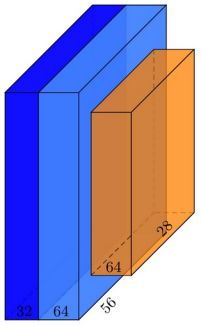
conv2: [3,3,1]/1 × 16
112×112×16

conv3: [1,1,16]/1 × 32
112×112×32

max2: [2,2,1]/2 × 32
56×56×32

Conv & Pool Key: kernelType [kw,hw,inchan]/stride × outchan
Layer Dimension Key: width × height × channels

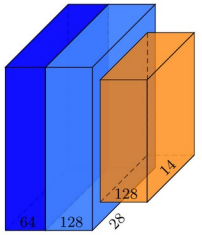
Purple: Full Convolution
Dark Blue: Depthwise Convolution
Light Blue: Pointwise Convolution
Orange: Max Pooling



conv4: [3,3,1]/1 × 32
56×56×32

conv5: [1,1,32]/1 × 64
56×56×64

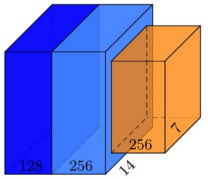
max3: [2,2,1]/2 × 64
28×28×64



conv6: [3,3,1]/1 × 64
28×28×64

conv7: [1,1,64]/1 × 128
28×28×128

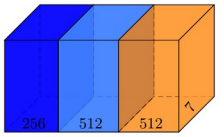
max4: [2,2,1]/2 × 128
14×14×128



conv8: [3,3,1]/1 × 128
14×14×128

conv9: [1,1,128]/1 × 256
14×14×256

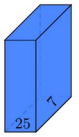
max5: [2,2,1]/2 × 256
7×7×256



conv10: [3,3,1]/1 × 256
7×7×256

conv11: [1,1,256]/1 × 512
7×7×512

max6: [2,2,1]/1 × 512
7×7×512



conv12: [1,1,512]/1 × 25
7×7×25

Layer depths are drawn with logarithmic scaling. Max pooling layer widths and heights are shrunk by $\frac{1}{4}$ instead of $\frac{1}{2}$.

Evolution of YOLO

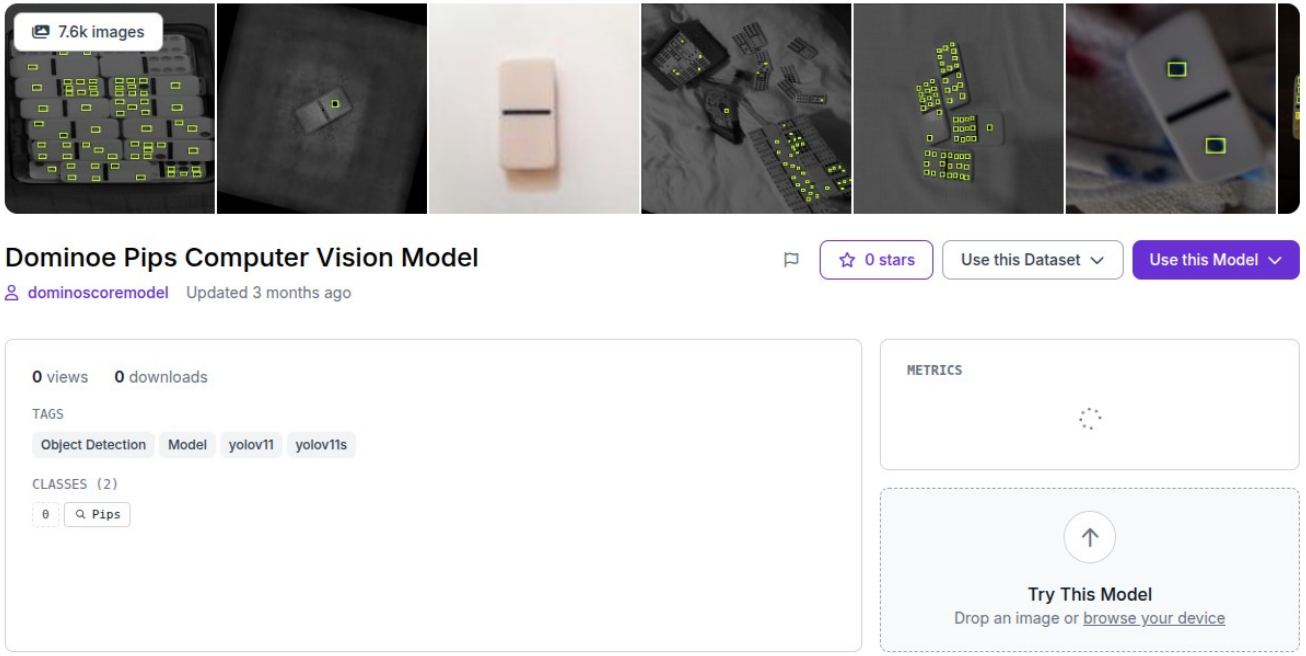
- YOLOv1: first real-time detector
- YOLOv2: anchor boxes
- YOLOv3: multi-scale detection
- YOLOv4: training tricks revolution
- YOLOv5: PyTorch adoption (Ultralytics)
- YOLOv6-7: hardware optimization
- YOLOv8-11: anchor-free modern YOLO
- YOLO26: latest version from Ultralytics

Ultralytics

- Company that has developed their own versions of YOLO
- Code is available: “pip install ultralytics”
- Free to play with; license required for commercial use

RoboFlow

- Provides tools for training computer vision systems, including YOLO systems.
- Publicly available datasets, including dominoes.



Later Developments

- Latest YOLO versions incorporate “attention” idea from Transformer neural networks.
- YOLO World: instead of providing examples of each class to look for, you can use a text prompt, e.g., “cat”.