Machine Learning 410

Lesson 13

Introduction to Object Dectection

Steve Elston

Outline

- Elements of object detection algorithms
- Evolution of object detection algorithms
- Parameterization of bounding boxes
- Evaluation of object detection algorithms
- Multiple prior bounding boxes
- Finding priors for bounding boxes
- Solving the object detection problem
- Working with multiple scales
- Integrating datasets involves complex language problem

Try it yourself! Object detection is widely used commercially
https://cloud.google.com/vision/automl/object-detection/docs/

Elements of Object Detection Algorithms

Object detection algorithms have some common elements

- **Convolutional Neural Network**: CNN creates a feature map which is used to detect and classify objects
- **Candidate bounding boxes**: Multiple candidate bounding boxes are generated for each region
- **Filter bounding boxes**: The probability of an object being in each bounding box (**objectness**), and low probability boxes are filtered
- **Minimal bounding boxes**: The size of the bounding boxes is adjusted to best fit the objects
- **Classification**: The objects in each bounding box is classified

Evolution of Object Detection Algorithms

Object detection algorithms

- Erhan et. al., 2013, Scalable Object Detection using Deep Neural Networks, introduced the R-CNN algorithm the first widely accepted deep learning object detection algorithm. R-CNN demonstrated a significant improvement in object recognition accuracy. However, this algorithm is too slow for real-time video processing.
- Girshick, 2015, Fast R-CNN simplified the required computations but still struggled with real-time video.

Evolution of Object Detection Algorithms

Object detection algorithms

- Ren et. al., 2016, Faster R-CNN algorithm, but computational complexity of the algorithm was still rather high.
- He, et. al. in 2018 Mask R-CNN algorithm exhibits significantly improved object detection accuracy, particularly when there are large numbers of objects, such as flock of birds or a crowd of people. While not efficient enough for real-time video, but accurate for complex scenes

Evolution of Object Detection Algorithms

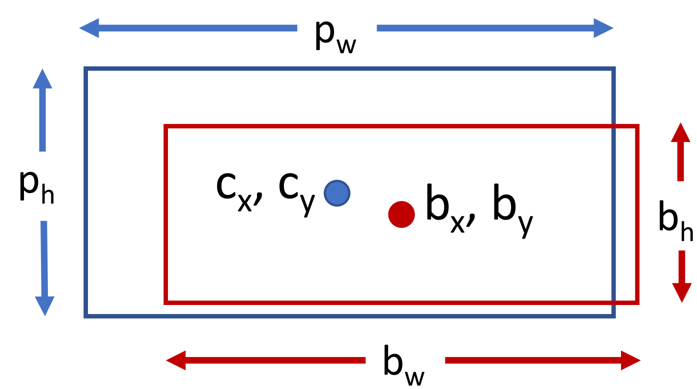
Real-time object detection algorithms

- Lui et. al., 2016, Single shot Multibox Detector performs bounding box fitting, object detection, and classification in one step. This single shot algorithm provides real time performance for video
- Redmon, et. al. 2016, You Only Look Once: Unified, Real-Time Object Detection (YOLO) is an alternative single shot detector. YOLO version 1 suffered from low accuracy
- Redmon, et. al., 2016, YOLO 9000: Better, Faster, Stronger (aka YOLO v2) made several improvements over the original algorithm. Included the combination of efficient CNN, larger, integrated training data set.
- Redmon, et. al., 2016, YOLOv3: An Incremental Improvement, primarily new CNN.

Parameterization of Bounding Boxes

Need a stable parameterization of 4 parameters of bounding box

- Start with a prior for the bounding box
  - c_x, c_y is center of the prior
  - p_w is the width prior
  - p_h is the height prior
- The compute the best fit box
  - b_x, b_y is center of bounding box
  - b_w is the width of the bounding box
  - b_h is the Hight of the bounding box

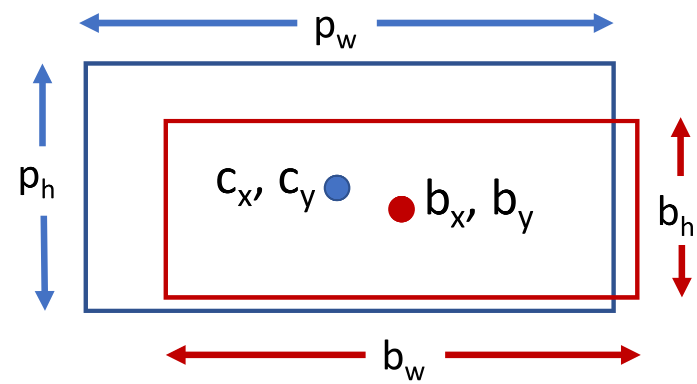


Parameterization of Bounding Boxes

Need a stable parameterization of 4 parameters of bounding box

- A naive approach is to solve a linear system of equations for parameters, t_x, t_y, t_u, t_h:

$$b_x = t_x + c_x$$
$$b_y = t_y + c_y$$
$$b_w = p_w * t_u$$
$$b_h = p_h * t_h$$



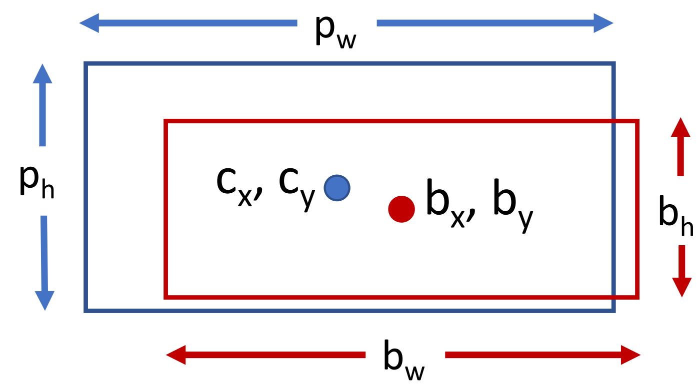
- But parameters of the bounding box are unconstrained!
- Solution can be unstable

Parameterization of Bounding Boxes

Need a stable parameterization of 4 parameters of bounding box

- A better parameterization is:

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$p_0 = Pr(object) * IoU(b,\ object) = \sigma(t_0)$$



- The bounding box is now constrained and the parameterization is stable
- p_0 is the probability the box contains an object

Evaluation of object detection

How can we evaluate a the bounding boxes computed with object detection?

- Compare the computed bounding box with the marked bounding box (lable)
- Use the ratio of the area of the intersection divided by the area of the union
- Intersection over union or IoU metric
- Range:
  - 0.0 – no overlap
  - 1.0 – perfect match

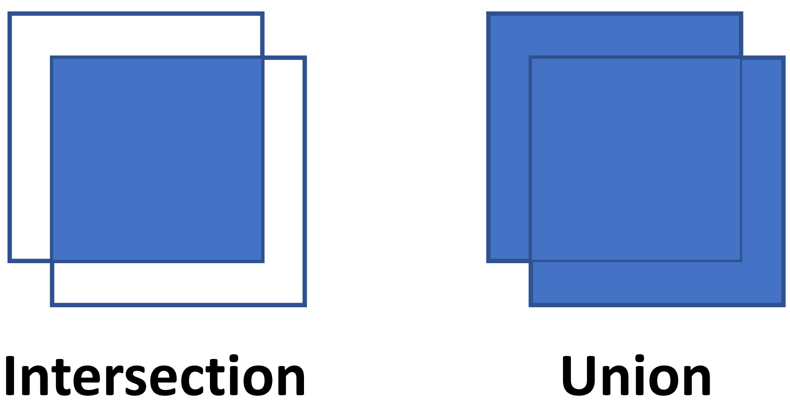
Evaluation of object detection

Need multiple criteria to evaluate object detection

- Is there an object in the box?
  - Can use ML metrics like **accuracy**
- Is the object correctly classified?
  - Typically use mean average precision – mAP
  - Average precision over all objects detected
  - Precision = true positives/(true positives + false postitives)
- Is the bounding box correct?

Evaluation of object detection

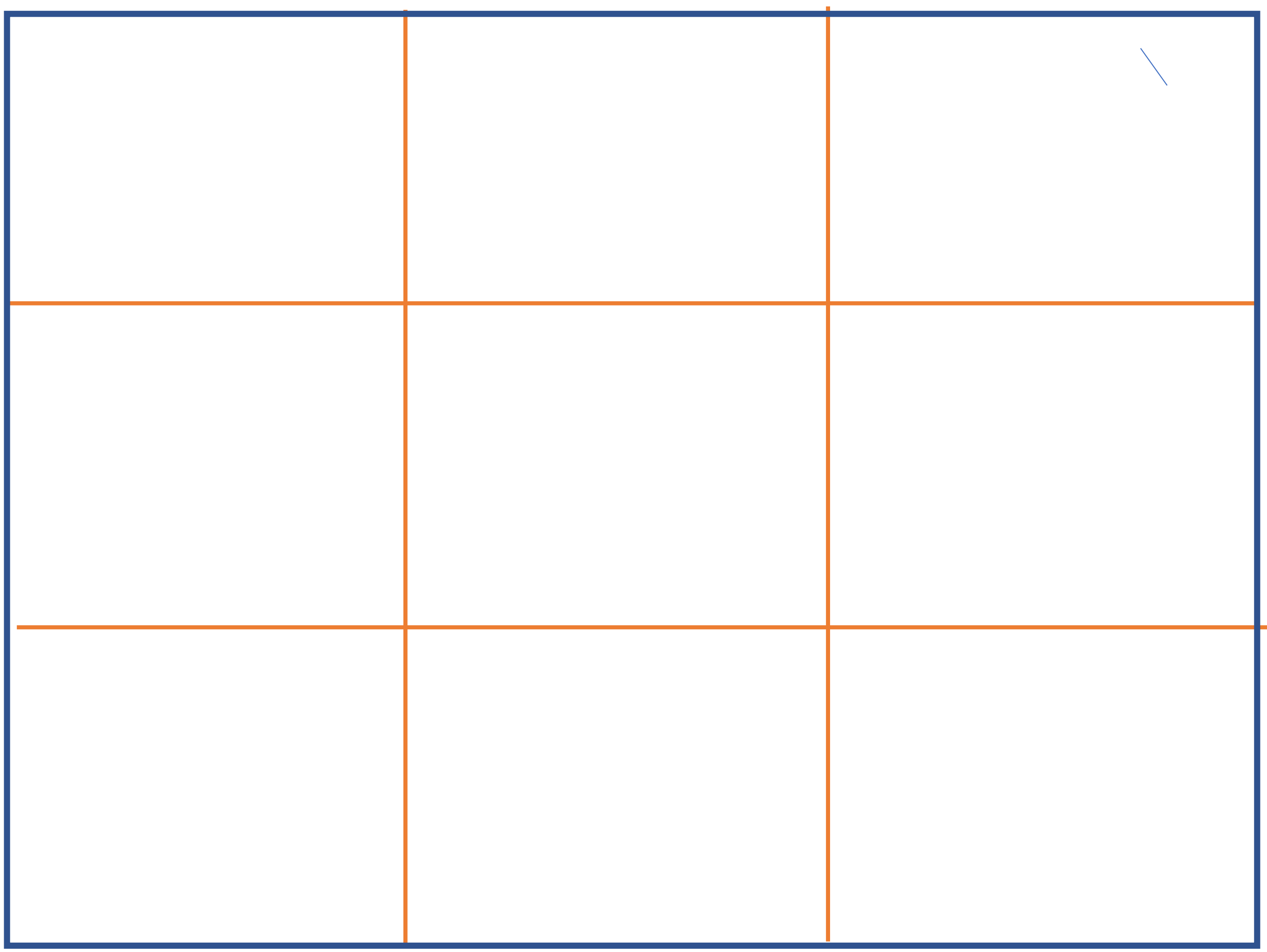
How can we evaluate a the bounding boxes computed with object detection?



$$IoU = \frac{Area\ of\ intersection}{Area\ of\ union}$$

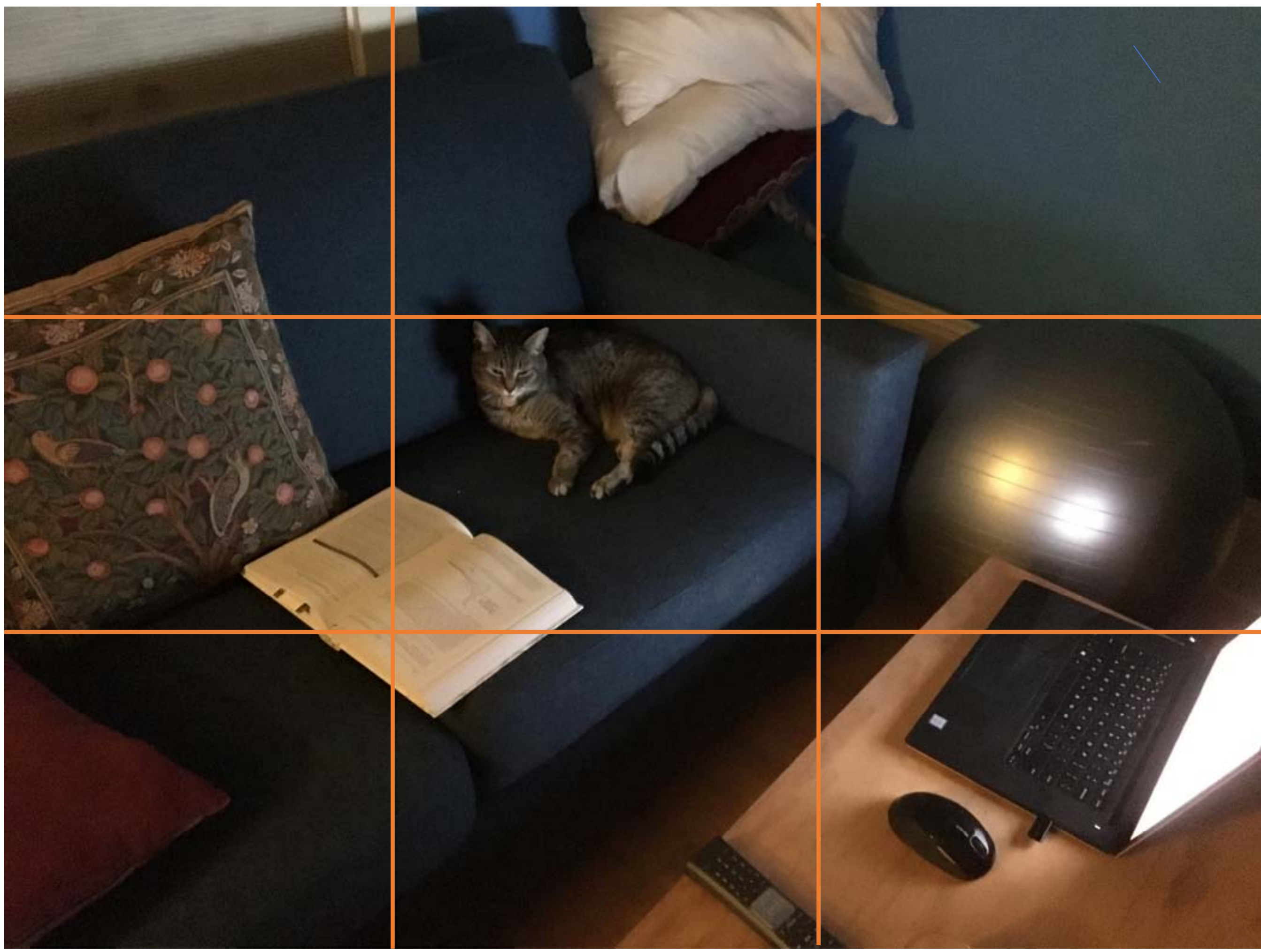
Multiple Prior Bounding Boxes

- Images can contain many objects
- Use a grid to divide the image
- Can fit bounding boxes to with centroids in each of the grid cells
- Use odd grid dimensions so there is a centroid at the center of image



Multiple Prior Bounding Boxes

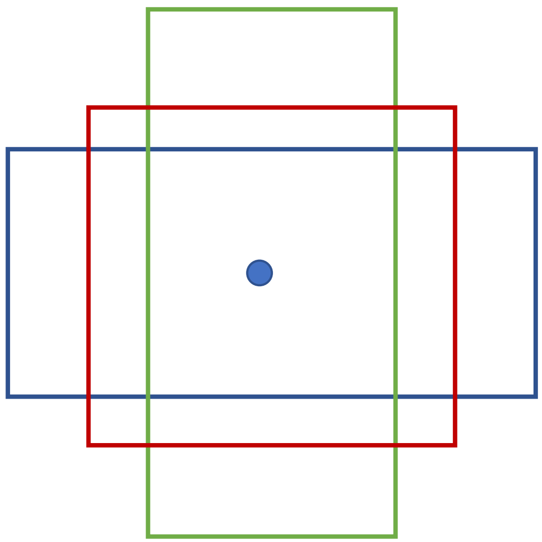
- Images contain many objects
- Impose grid over image
- Locate objects on the grid



Multiple Prior Bounding Boxes

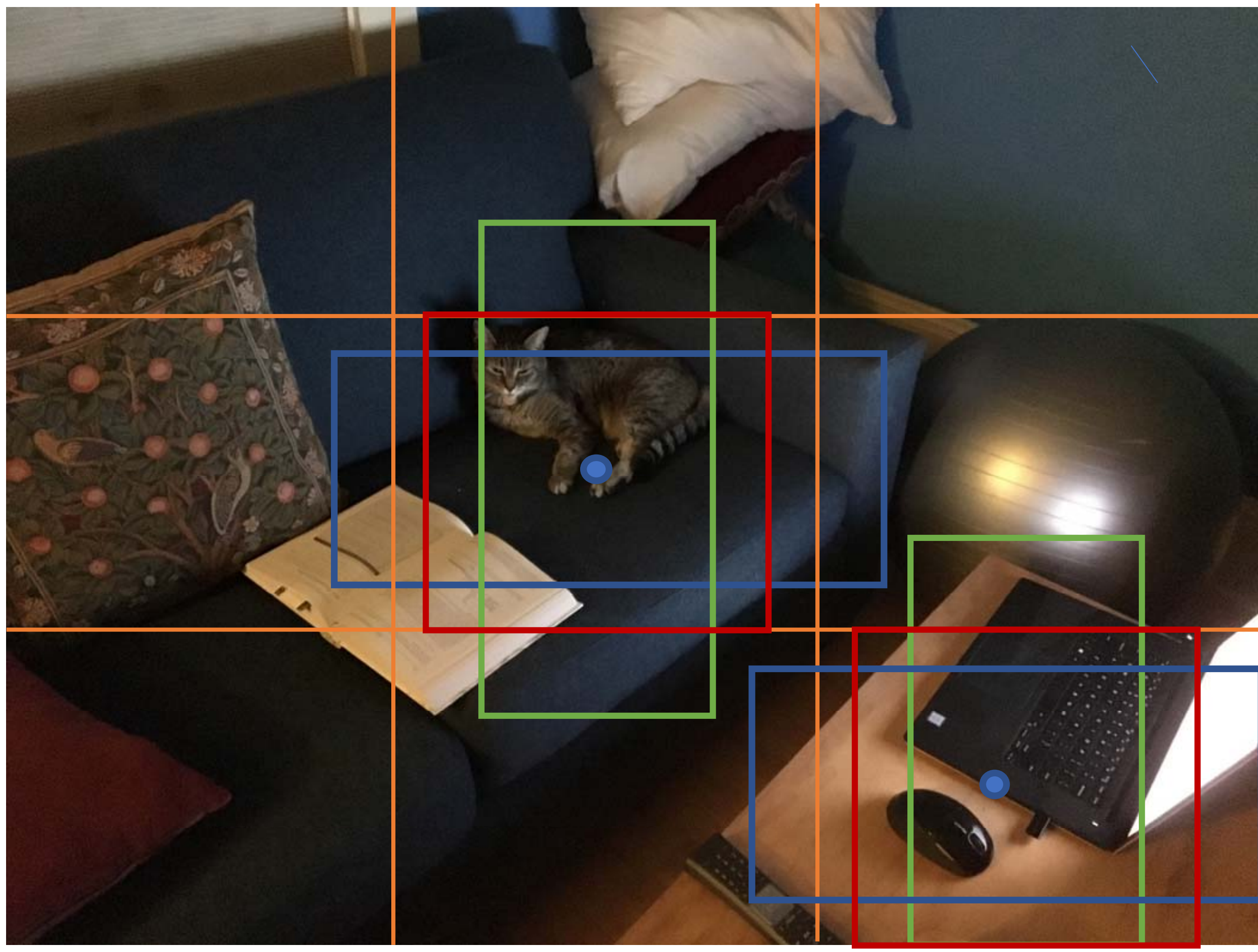
There are many possible bounding box proposals

- Start with a first bounding box proposal, with centroid
- Boxes with different aspect ratios and same centroid



Multiple Prior Bounding Boxes

- Multiple objects
- Multiple prior bounding box candidates



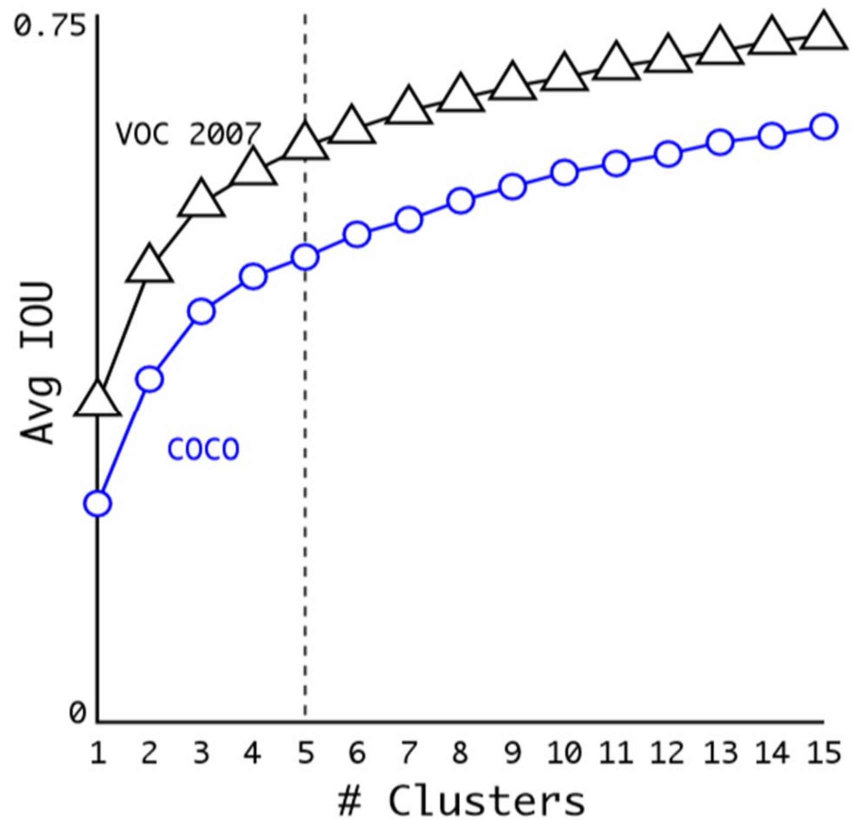
Finding Priors for Bounding Boxes

Good priors are required for solution

- Hand picked priors are inefficient
- Use k-means clustering with distance metric

d(box, centroid) = 1.0 – IOU(box, centroid)

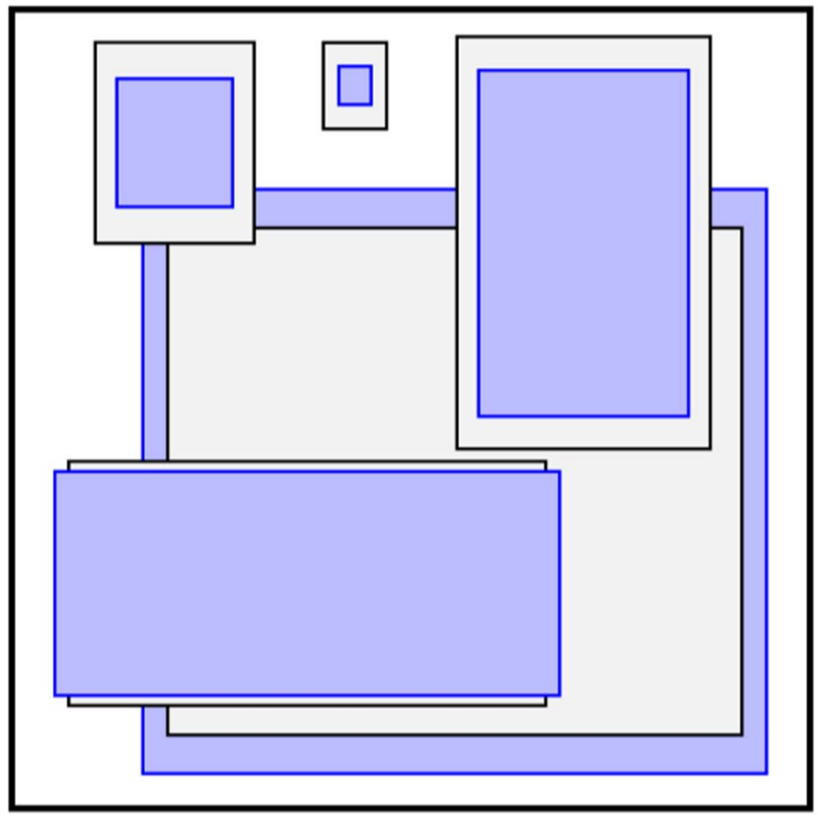
- How to choose *k*?
- Use k = 5
- Conservative value prevent overfitting



Finding Priors for Bounding Boxes

Good priors are required for solution

- Priors for VOC and COCO
- For both data sets tall and narrow priors are favored



Solving Object Detection Problem

Solve as object detection as regression problem

Find bounding box, objectness, and category (c_1,c_2,…,c_n), as label for regression problem

$$\hat{y} = \begin{bmatrix} b_x \\ b_y \\ b_w \\ b_h \\ p_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Solving Object Detection Problem

Solve as object detection as regression problem

- Can formulate the problem with label for multiple bounding boxes.
- Solve as regression problem in **one step**

$$\hat{y} = \begin{bmatrix} b_x \\ b_y \\ b_w \\ b_h \\ p_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ b_x \\ b_y \\ b_w \\ b_h \\ p_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Solving Object Detection Problem

Solve as object detection as regression problem

Find most probable bounding box with **non-max suppression algorithm**:

```
Filter all boxes with p0 below threshold, say 0.5
While( more than one overlapping box ):
   Select the remaining boxes with the highest probability.
   Compute the IoU for overlapping bounding boxes.
   Filter out bounding boxes with IoU below threshold, say 0.6.
```

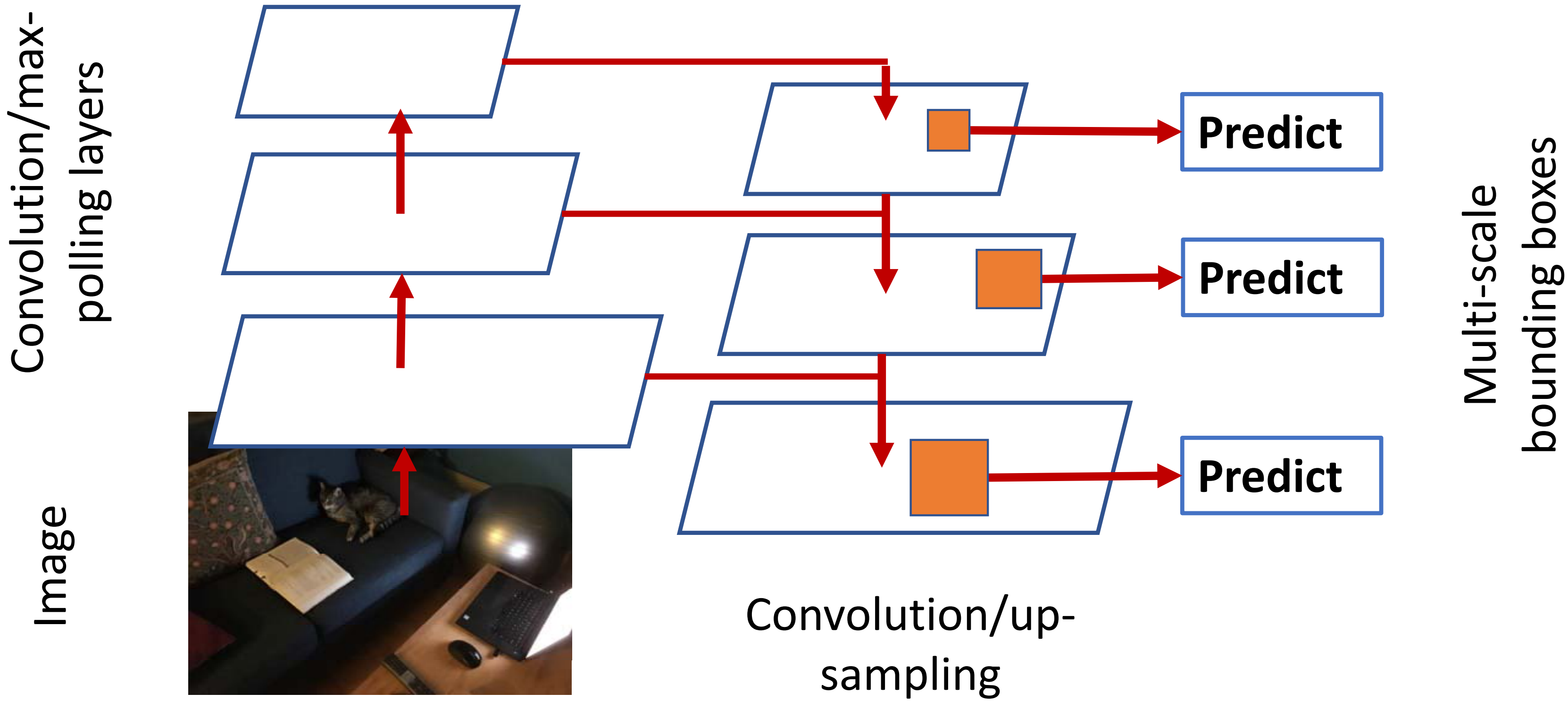
Working with multiple scales

Images contain objects a multiple scales

- Need to detect objects across wide range of scale
- Is trade-off between semantics and detail
  - Large scale has better semantics
  - Fine scale has more detail
- Deep neural network architecture produces multiple scales
  - Convolution with max pooling reduces detail
  - Deeper layers with better symantics

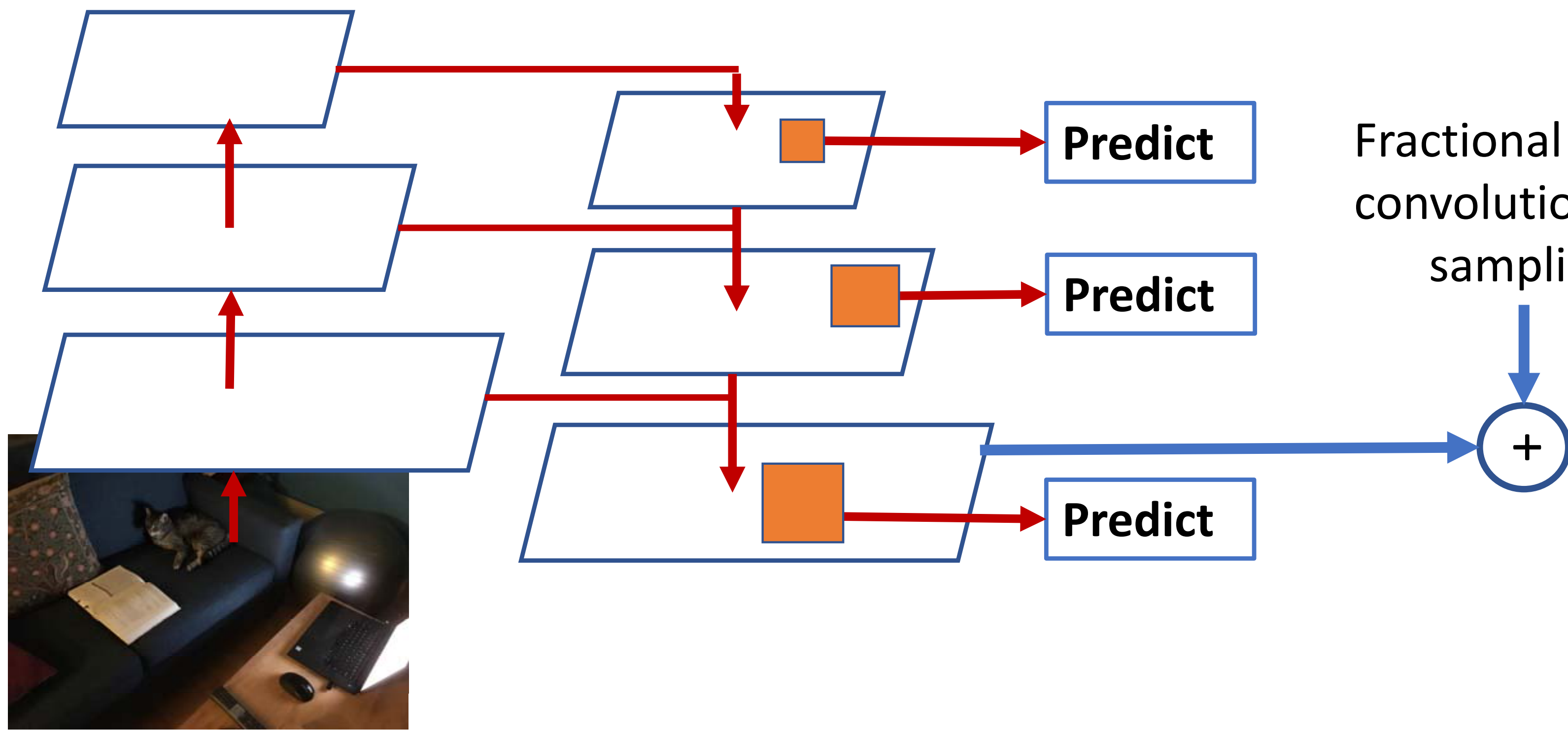
Working with multiple scales

Convolutional neural network with multi-scale feature map (pyramid)



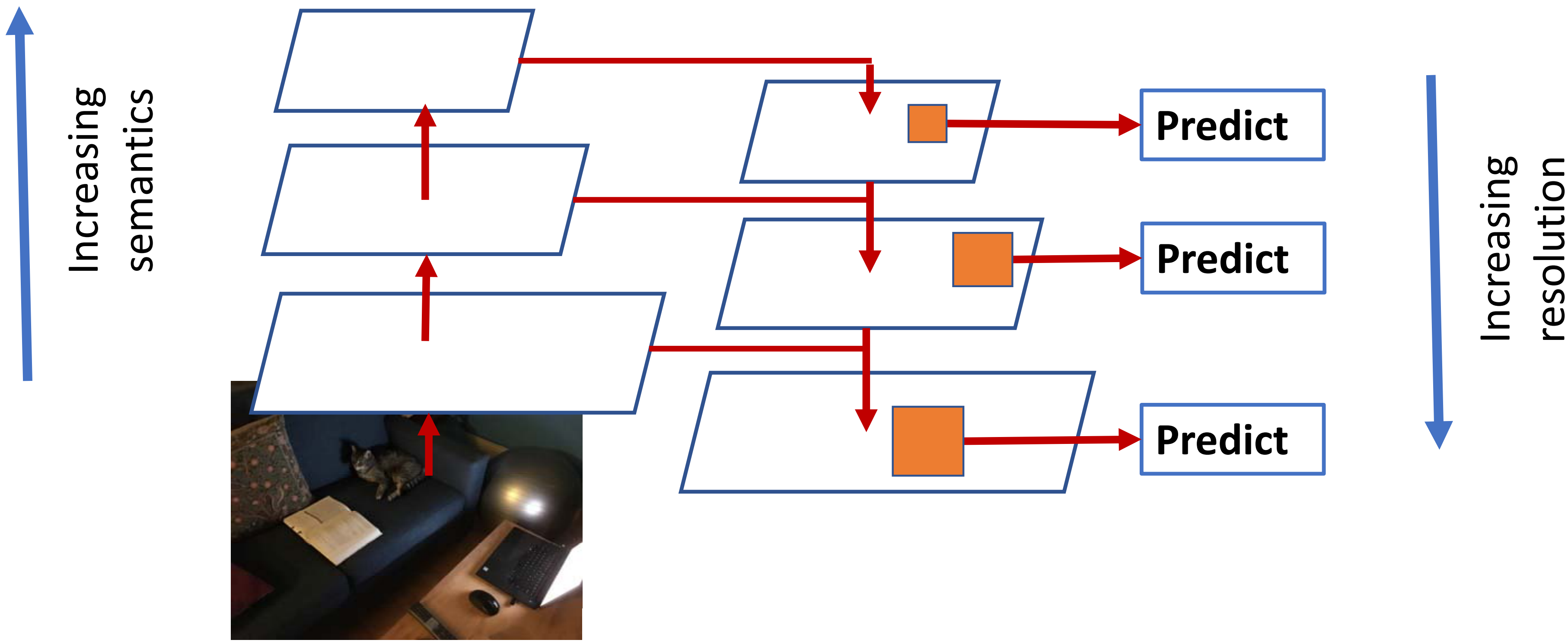
Working with multiple scales

Convolutional neural network with multi-scale feature map (pyramid)



Working with multiple scales

Convolutional neural network with multi-scale feature map (pyramid)



Integrating Datasets

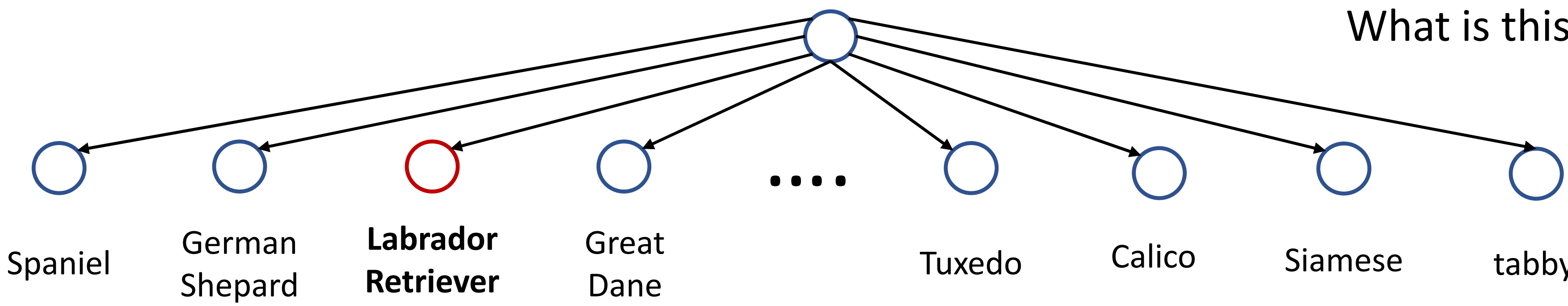
Need to integrate multiple datasets

- Difference in number of cases between training datasets
  - ImageNet is extensive, but only for classification, no bounding boxes
  - Classification datasets with marked bounding box are more limited
- Must integrate these datasets for training
  - ImageNet uses compound words, e.g. Labrador retriever
  - Marked bounding box data uses simple words: e.g. retriever or dog
- Semantics of the classification categories are rather different!
  - Must resolve mismatch to integrate datasets

Semantics of Language is Complex

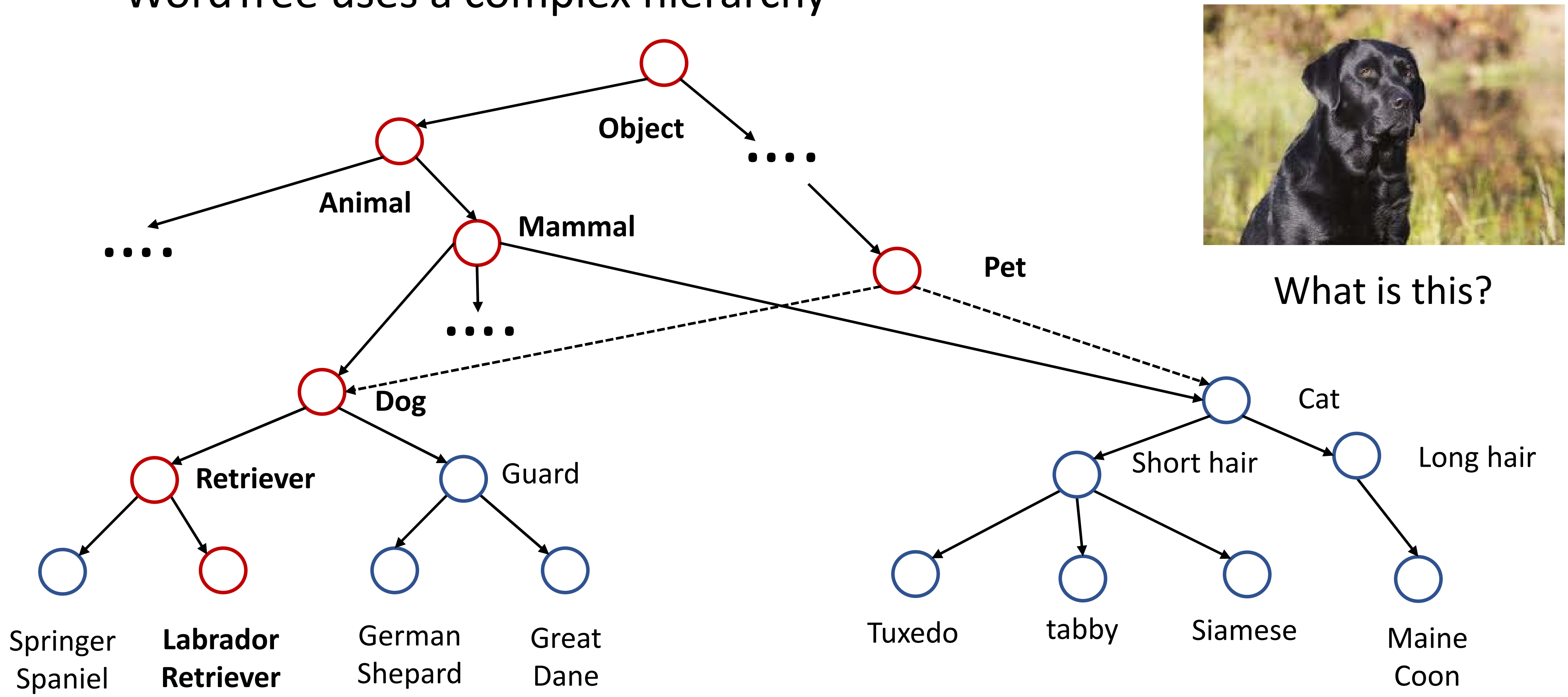
ImageNet uses a flat hierarchy for classification

What is this?



Semantics of Language is Complex

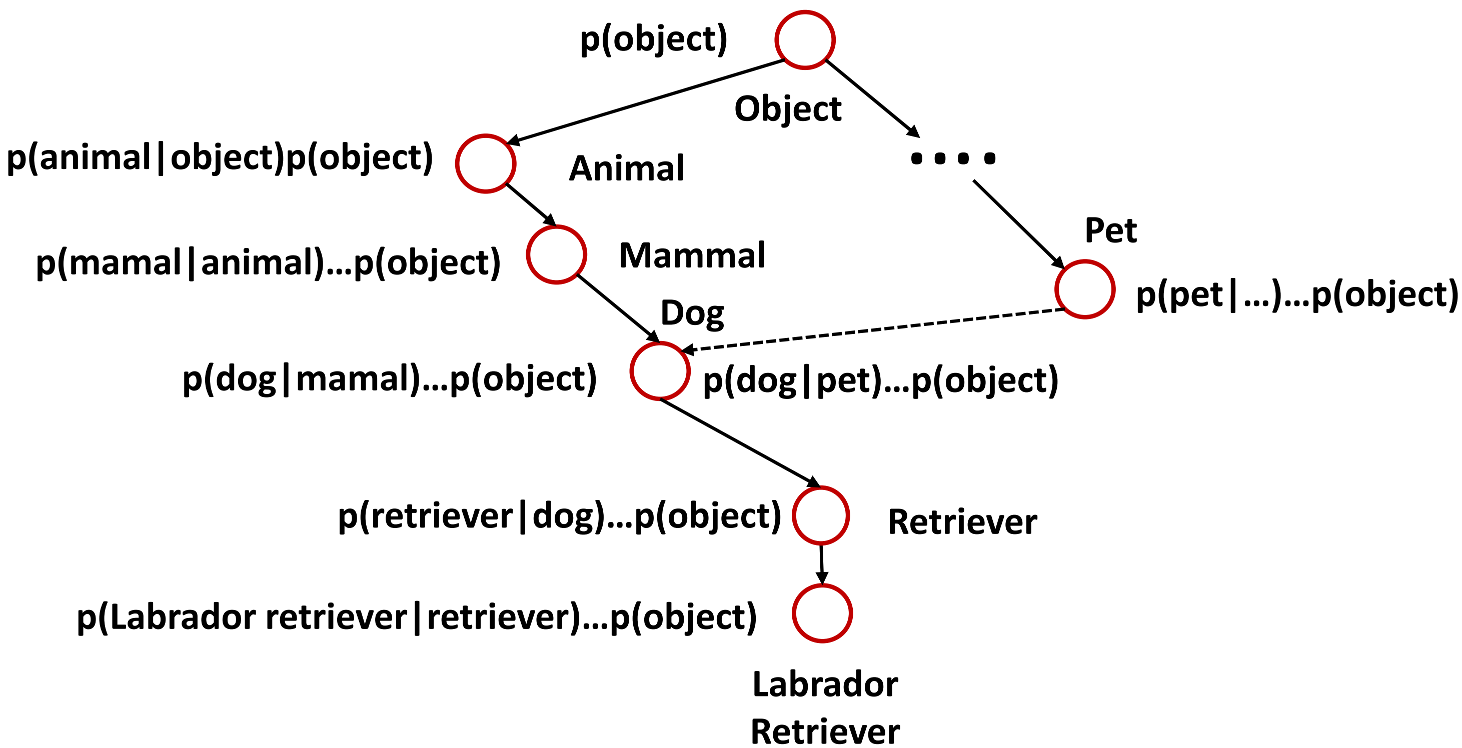
- Human language classifies the same object into multiple categories
- WordTree uses a complex hierarchy



What is this?

Semantics of Language is Complex

- Human language classifies the same object into multiple categories
- What are the conditional probabilities?
- Computation depends on semantics!



What is this?

Integrating Datasets

Need to integrate multiple datasets

- Integration of the datasets requires integration of classification terms
- Integrate terms by shortest path on WordTree
- Use common term to integrate bounding box and extensive classification categories