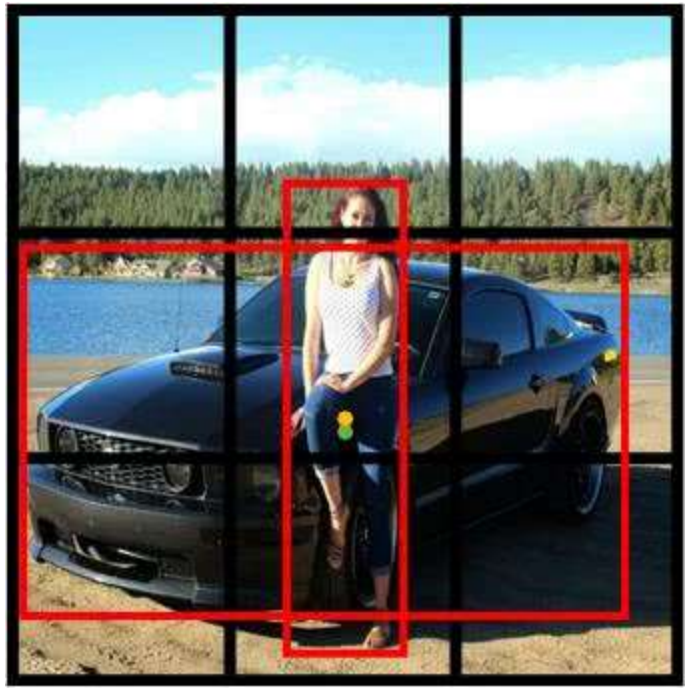
13. 4. Anchor Boxes

Object detection algorithms usually …

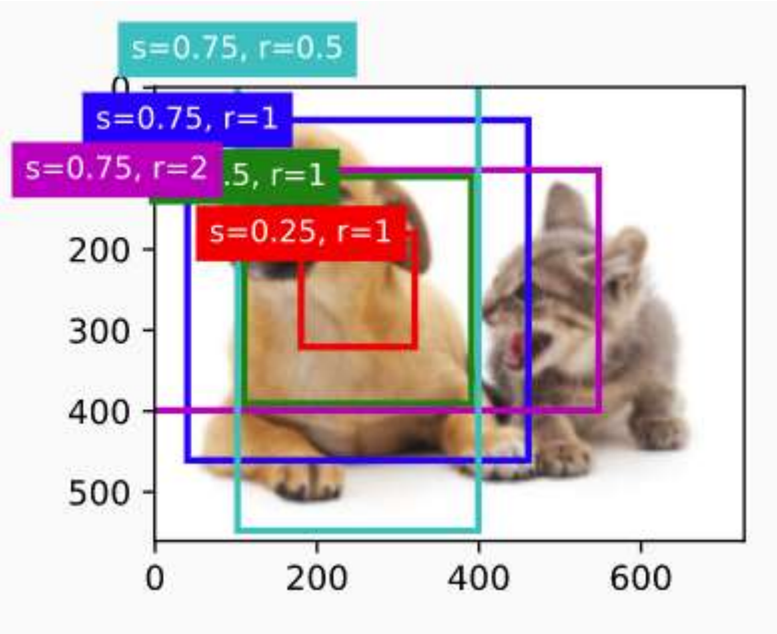
- sample **many regions** in the input image
- determine whether these regions **contain object of interest**
- adjust **the edges of the regions** to predict bounding box accurately



13. 4. Anchor Boxes

13. 4. 1. Generating Multiple Anchor Boxes

- Input image has a height of **h** and width of **w**
- Generate anchor boxes with different shapes centered on each pixel
- Size parameter $s \in (0,1]$
- Aspect ratio $r > 0$ (ratio between width and height)
- $s_1, \dots, s_n$ and $r_1, \dots, r_m$ => whnm anchor boxes
- $(s_1, r_1), (s_1, r_2), \dots, (s_1, r_m), (s_2, r_1), (s_3, r_1), \dots, (s_n, r_1)$ => $\mathrm{wh}(\mathrm{n} + \mathrm{m} - 1)$



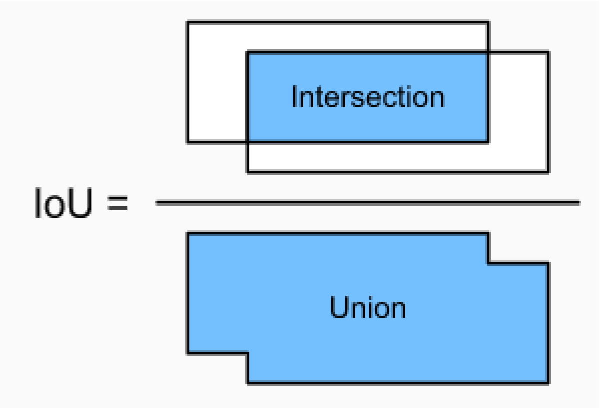
13. 4. Anchor Boxes

13. 4. 2. Intersection over Union (IoU)

- How to measure the similarity between anchor box and the ground-truth bounding box?
- Jaccard index

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

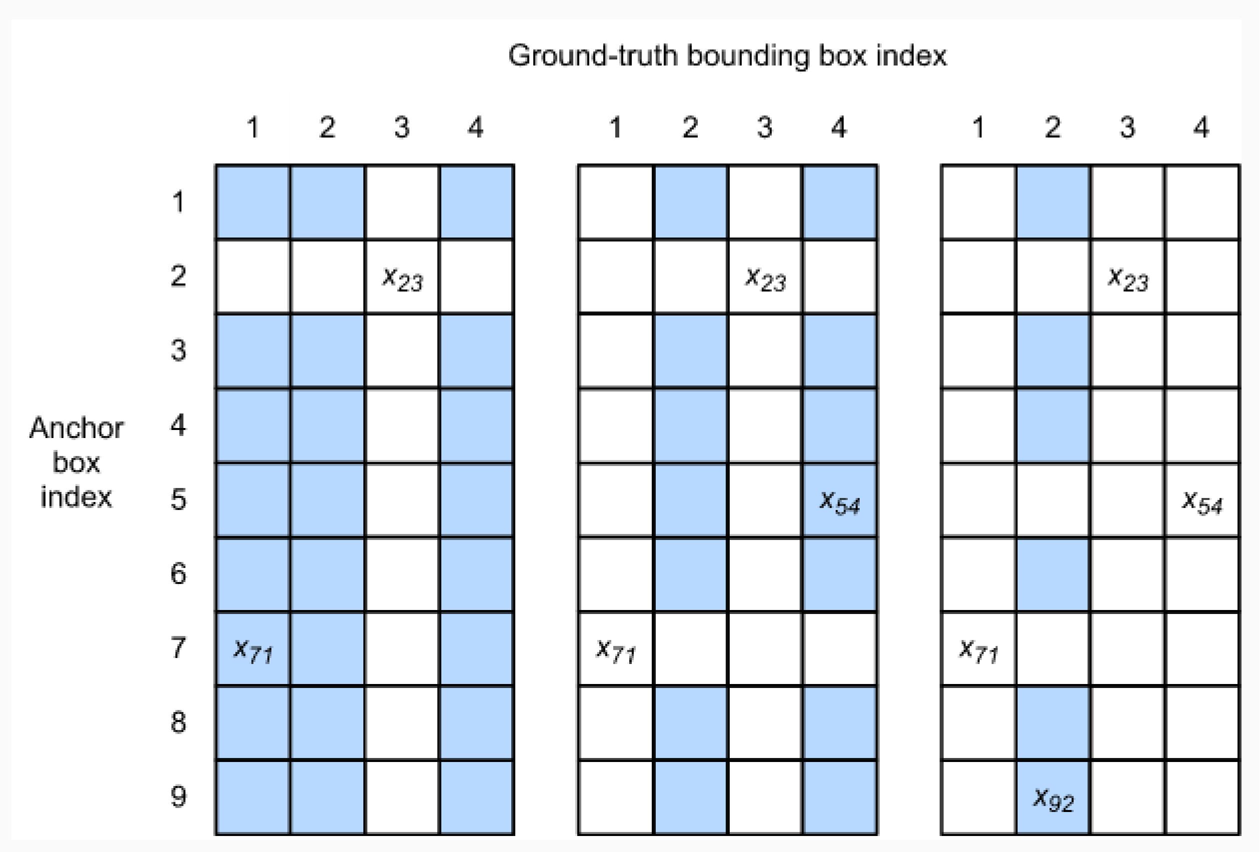
- 0 indicates there are no overlapping pixels, while 1 indicates that two bounding boxes are equal



13. 4. Anchor Boxes

13. 4. 3. Labeling Training Set Anchor Boxes

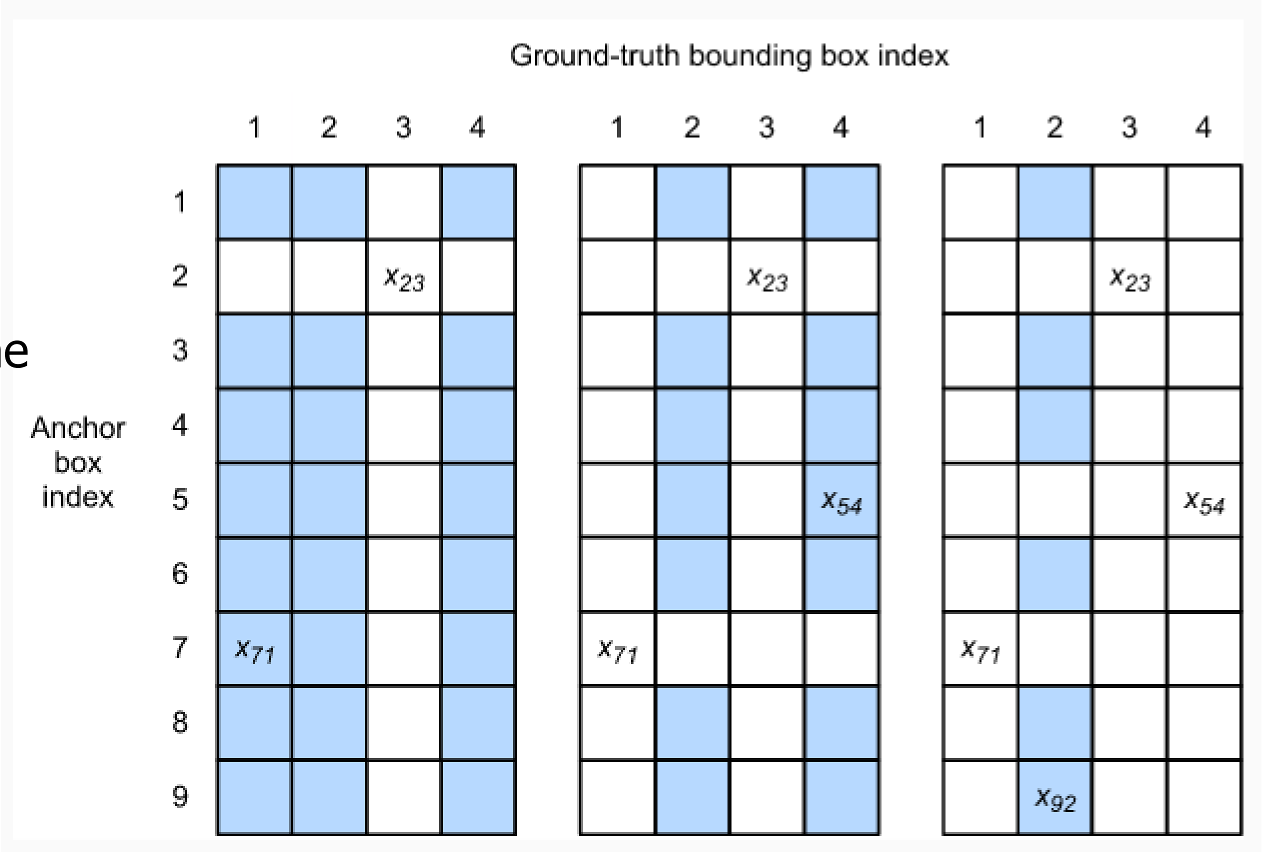
- To train the object detection model
  - The category of target contained in anchor box (category)
  - The offset of the ground-truth bounding box relative to anchor box (offset)
- In object detection
  - Generate multiple anchor boxes
  - Predict the categories and offsets for each anchor box
  - Adjust anchor box position according to the predicted offset
  - Filter out the prediction bounding box



13. 4. Anchor Boxes

13. 4. 3. Labeling Training Set Anchor Boxes

- $A_1, A_2, \ldots, A_{n_a}$, anchor boxes in the images
- $B_1, B_2, \ldots, B_{n_b}$, ground-truth bounding boxes
- $X \in \mathbb{R}^{n_a \times n_b}$, each element means IoU
- Find the largest element in the largest element in the matrix X
- Discard all elements in the previously used row and column in the matrix
- Find the largest remaining element in the matrix X until all elements in the n_b columns in the matrix X are discarded
- Traverse only $n_a - n_b$ anchor boxes and find bounding box having largest IoU greater than threshold

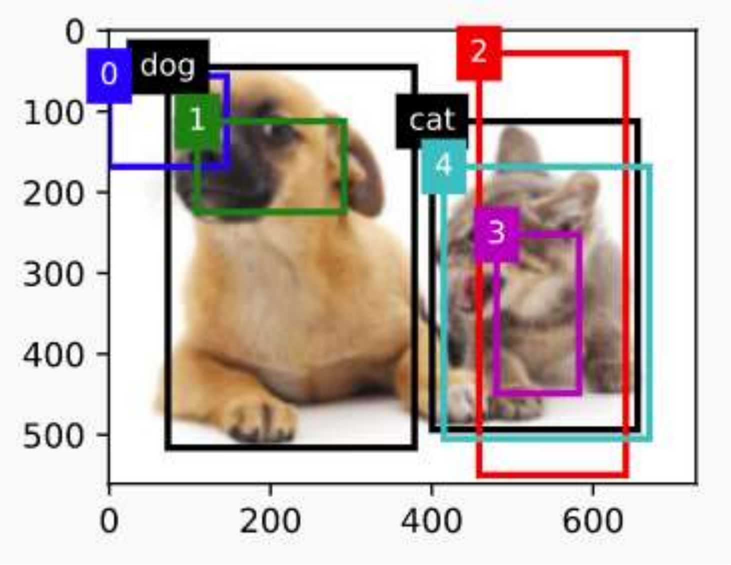


13. 4. Anchor Boxes

13. 4. 3. Labeling Training Set Anchor Boxes

$$\left(\frac{\frac{x_b - x_a}{w_a} - \mu_x}{\sigma_x}, \frac{\frac{y_b - y_a}{h_a} - \mu_y}{\sigma_y}, \frac{\log \frac{w_b}{w_a} - \mu_w}{\sigma_w}, \frac{\log \frac{h_b}{h_a} - \mu_h}{\sigma_h} \right)$$

- Because the positions and sizes of various boxes may vary
- Usually require some special transformations the offset distribution more uniform and easier to fit



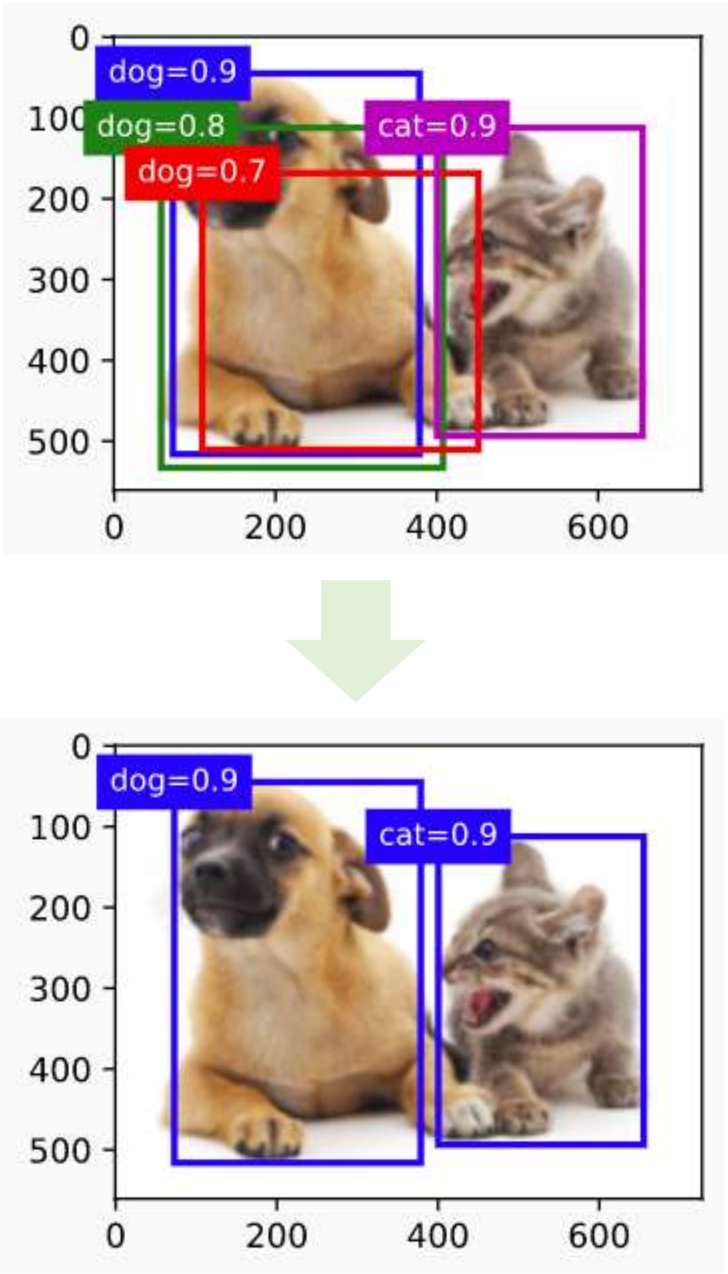

```
labels[2]
```

```
tensor([[0, 1, 2, 0, 2]])
```

13. 4. Anchor Boxes

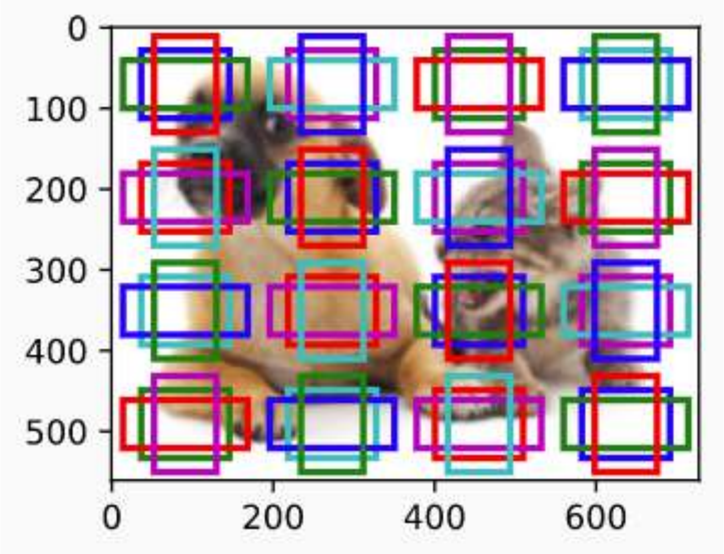
13. 4. 4. Bounding Boxes for Prediction

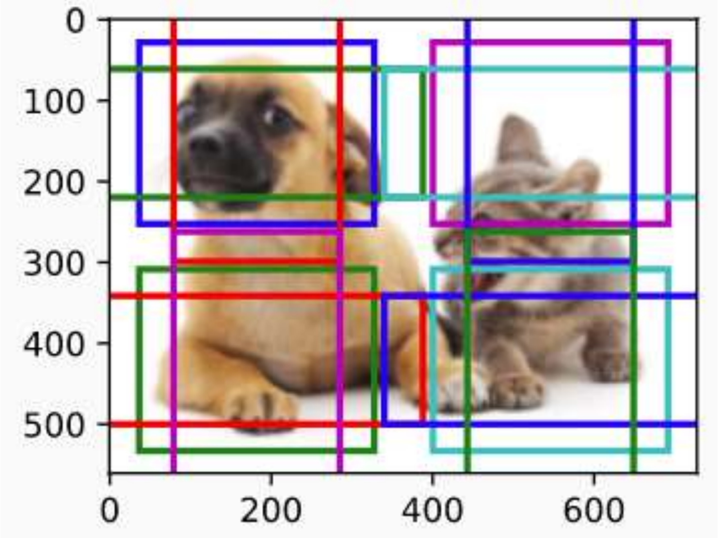
- During model prediction phase, first **generate multiple anchor boxes** for the image and the predict **categories** and **offsets** one by one
- This can make many anchor boxes for the same target
- To simplify the results, we need to remove similar prediction bounding boxes by using non-maximum suppression (NMS)
- For prediction bounding box, the model calculates the prediction probability for each category
- Assume the largest predicted probability is p with category of B
- On the same image, sort the prediction bounding boxes with predicted category and remove the bounding box having IoU greater than threshold

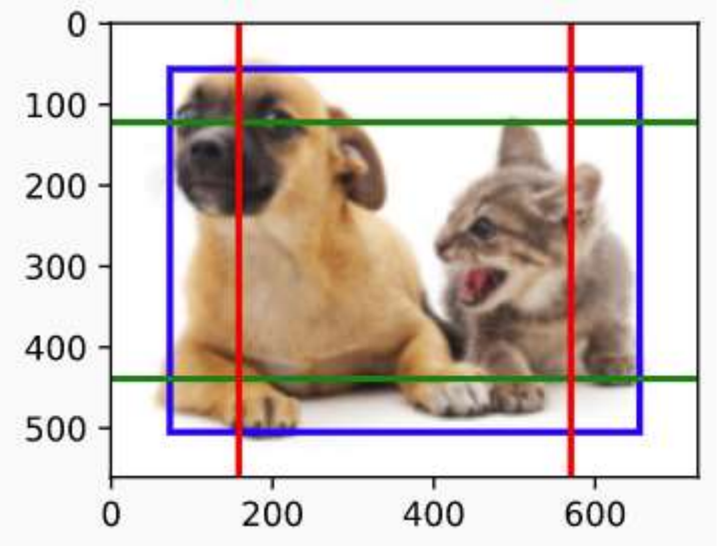


13. 5. Multiscale Object Detection

- There could be two many anchor boxes if we compute on each pixel of the image
  - If we have 561 x 728 image with 5 anchor boxes => over two million anchor boxes
- Apply uniform sampling on a small portion of pixels from the image
- When using smaller anchor boxes to detect small objects, we can sample more regions
- When using larger anchor boxes to detect larger objects, we can sample fewer regions







13. 6. The Object Detection Dataset

- There are no small datasets like MNIST or Fashion-MNIST, in the object detection field
- In order to quickly test models, we are going to assemble a small dataset
- First, generate 1000 banana images of different angles and sizes
- Then, collect a series of background images and place a banana image at a random position
- It can be simple artificial dataset