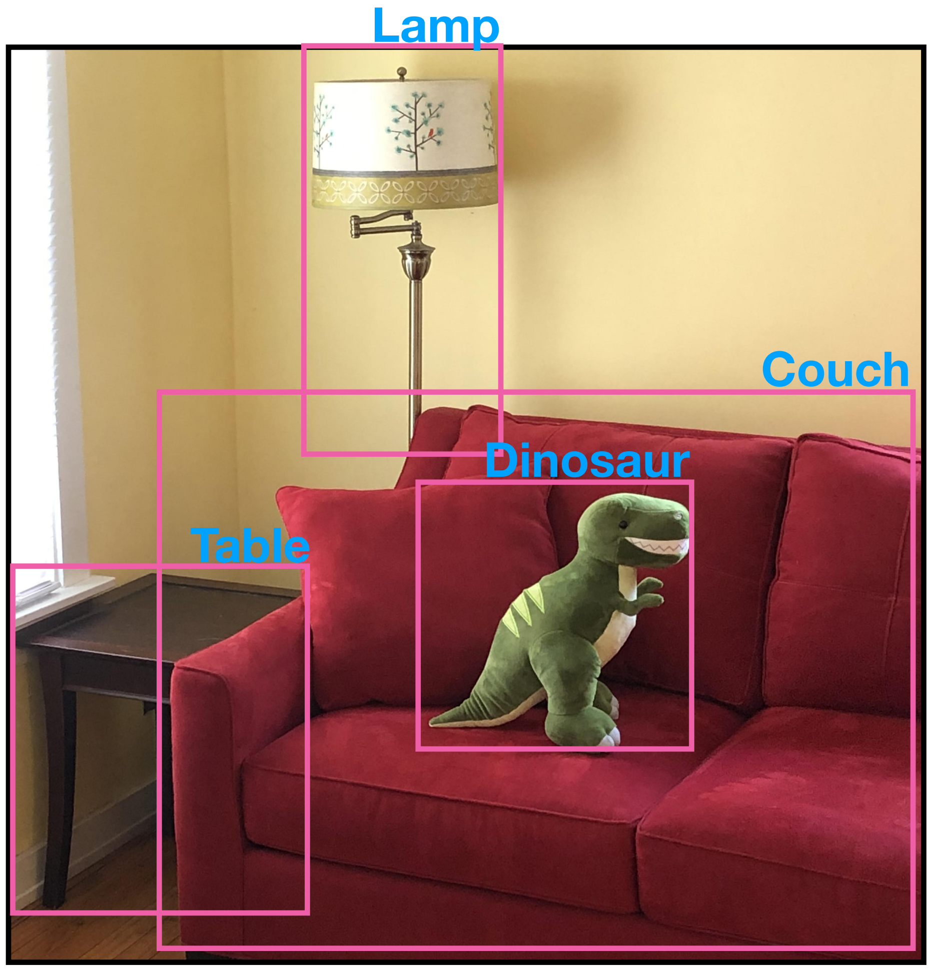
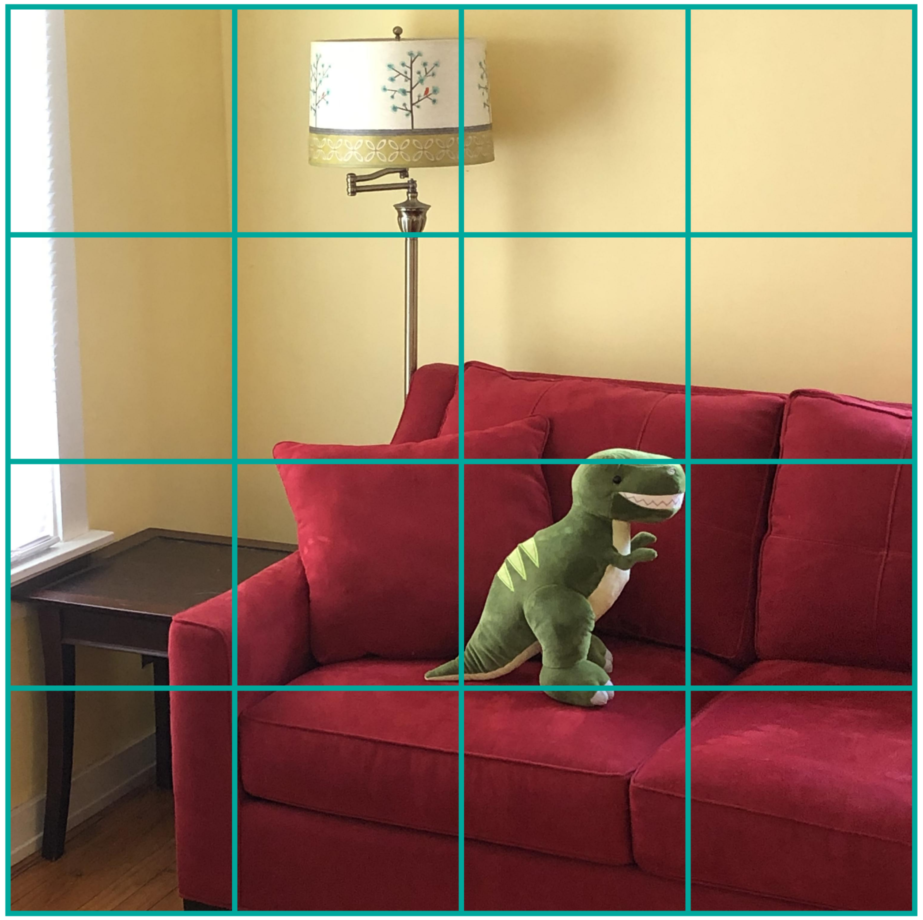
Object Detection (YOLO Algorithm)



Goal: In an image with multiple objects, identify:

- where the objects are
- what type of object each is

Object Detection (YOLO Algorithm)



Goal: In an image with multiple objects, identify:

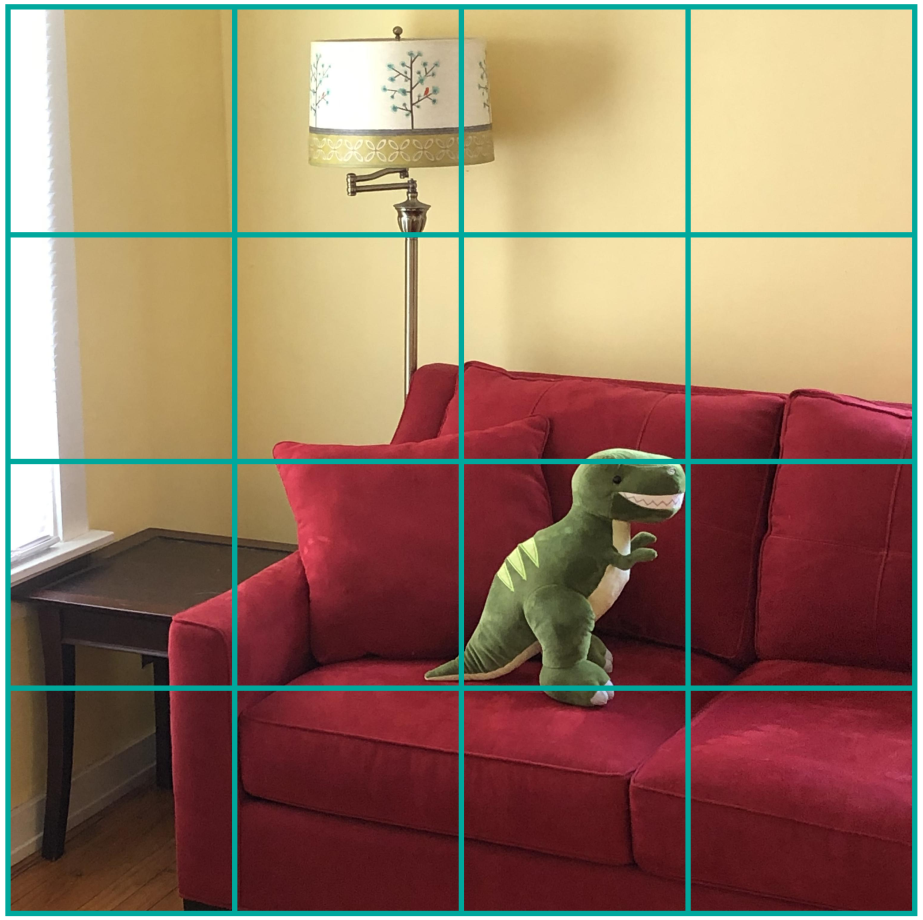
- where the objects are
- what type of object each is

Overview of Strategy:

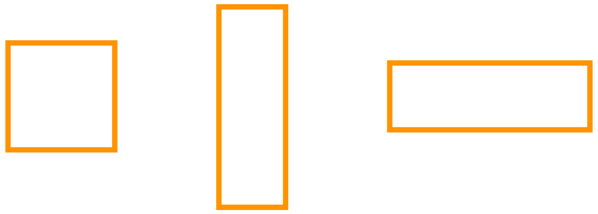
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image

Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



Goal: In an image with multiple objects, identify:

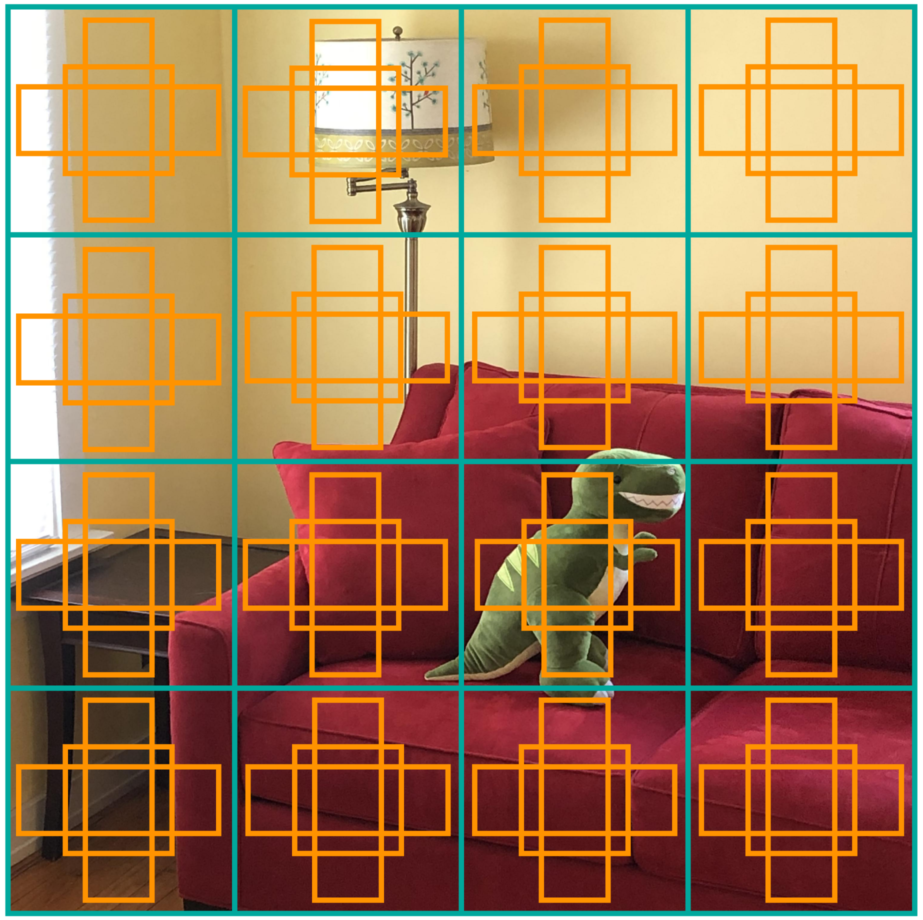
- where the objects are
- what type of object each is

Overview of Strategy:

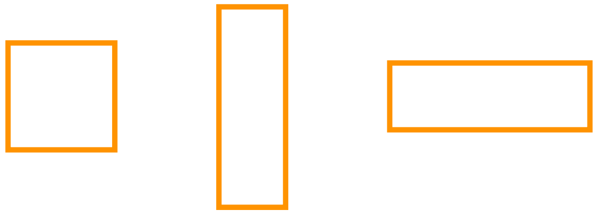
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations

Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



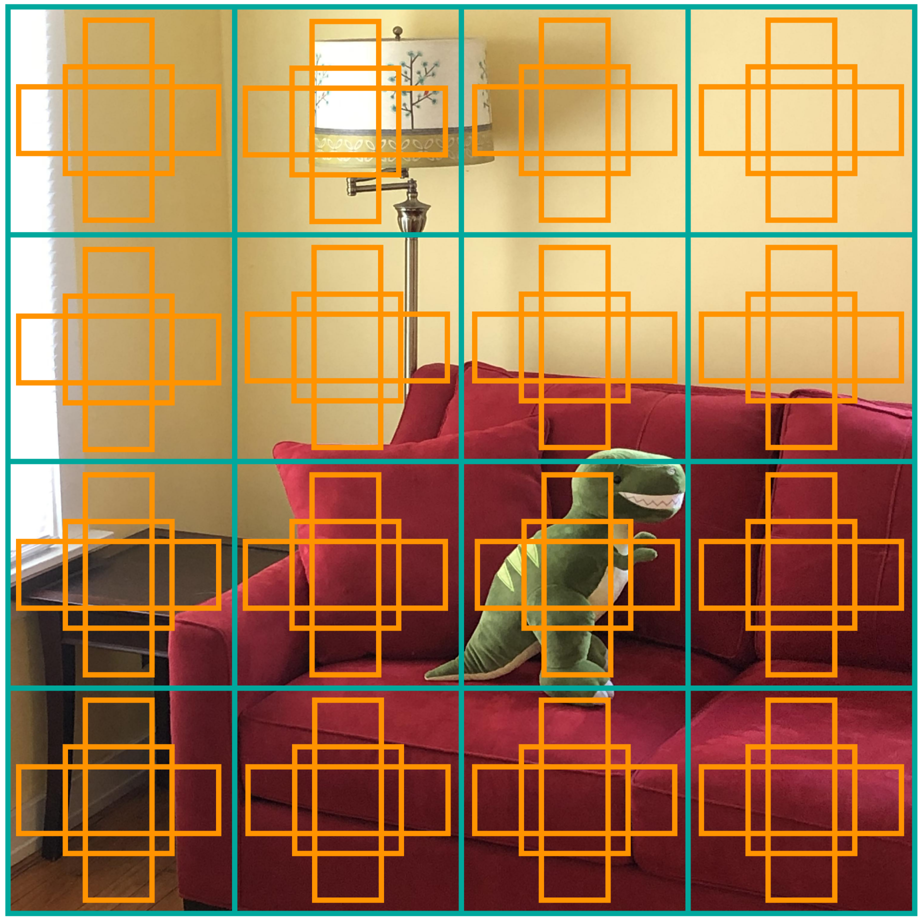
Goal: In an image with multiple objects, identify:

- where the objects are
- what type of object each is

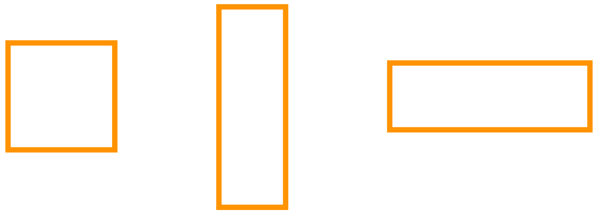
Overview of Strategy:
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations

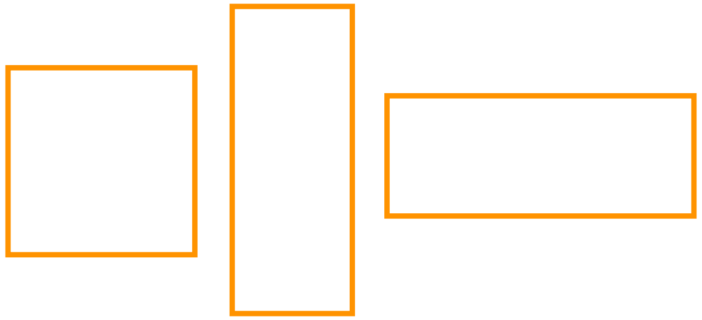
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

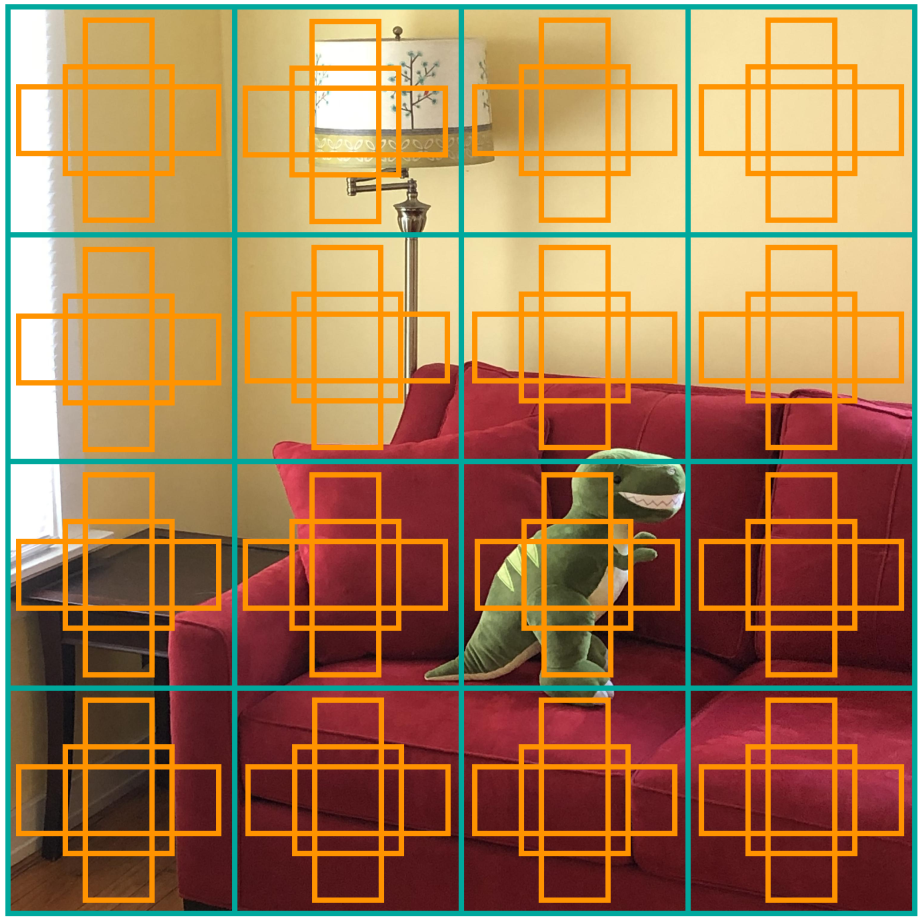
- where the objects are
- what type of object each is

Overview of Strategy:

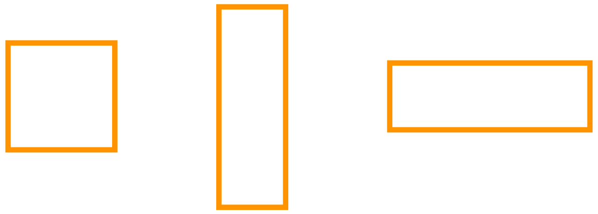
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations

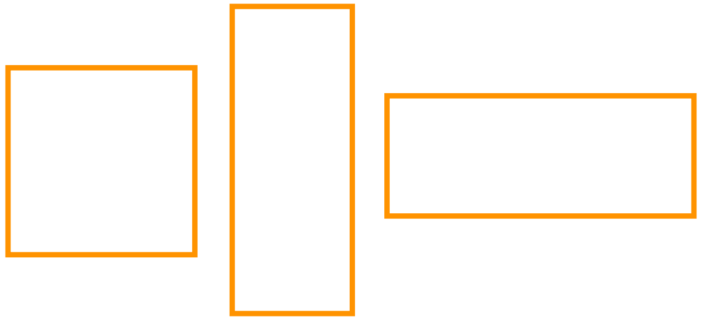
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

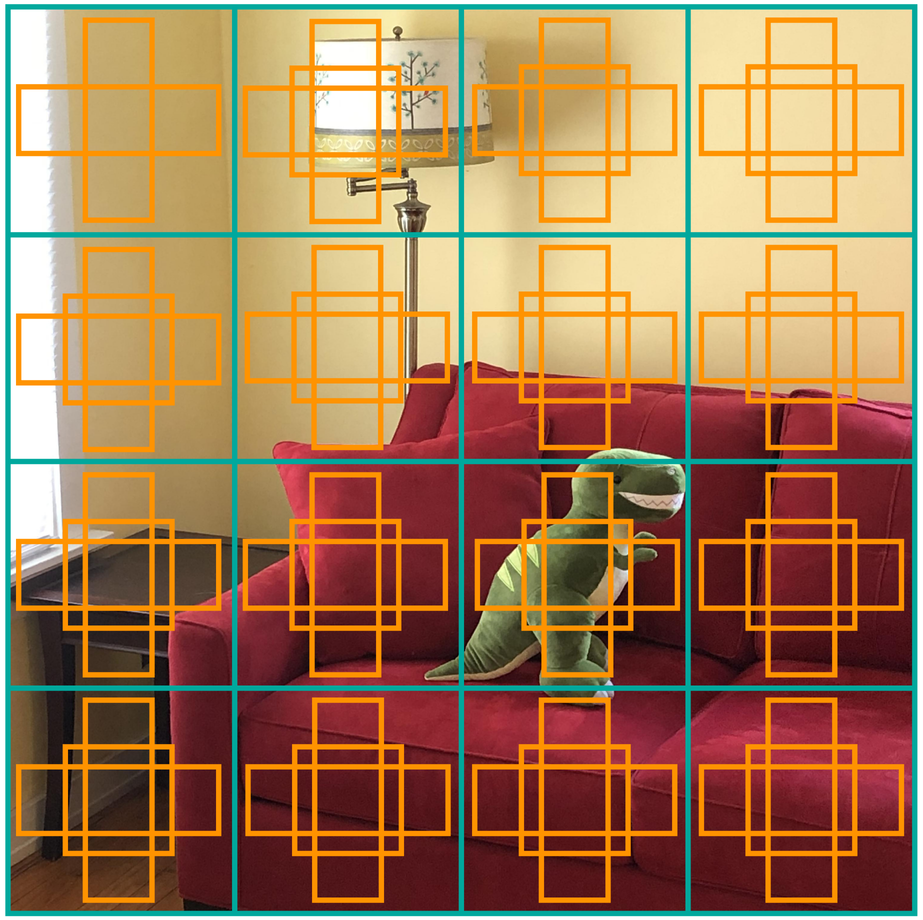
- where the objects are
- what type of object each is

Overview of Strategy:

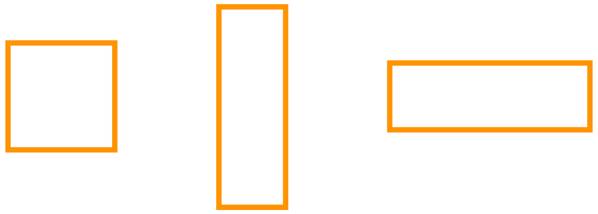
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

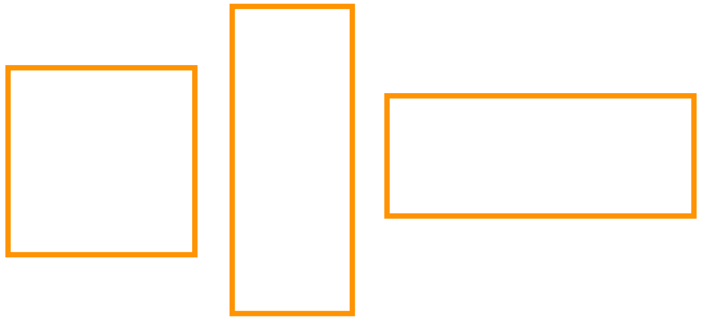
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

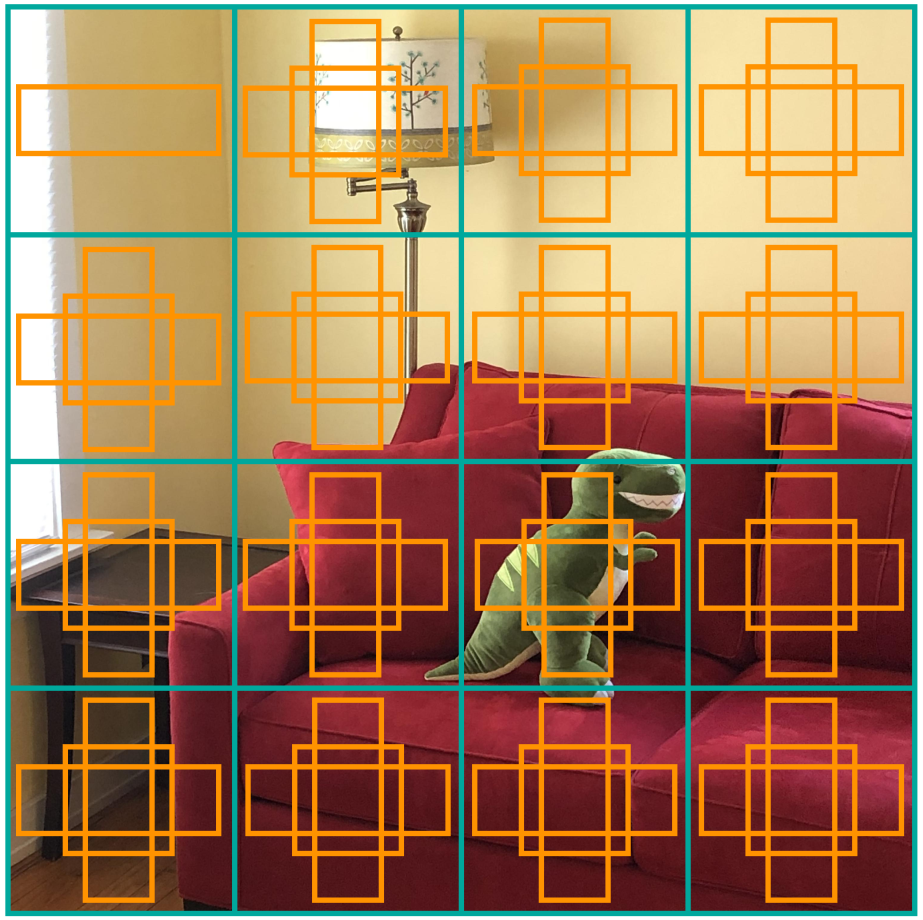
- where the objects are
- what type of object each is

Overview of Strategy:

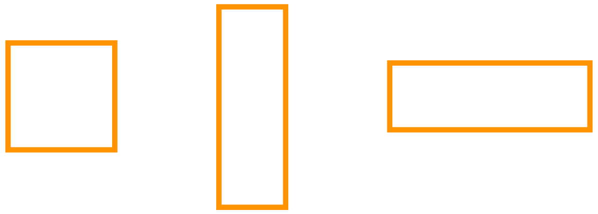
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

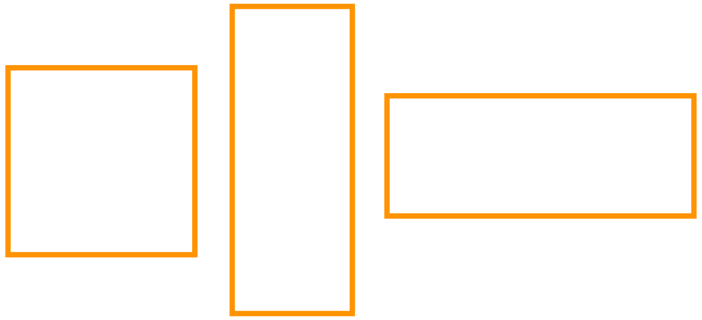
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

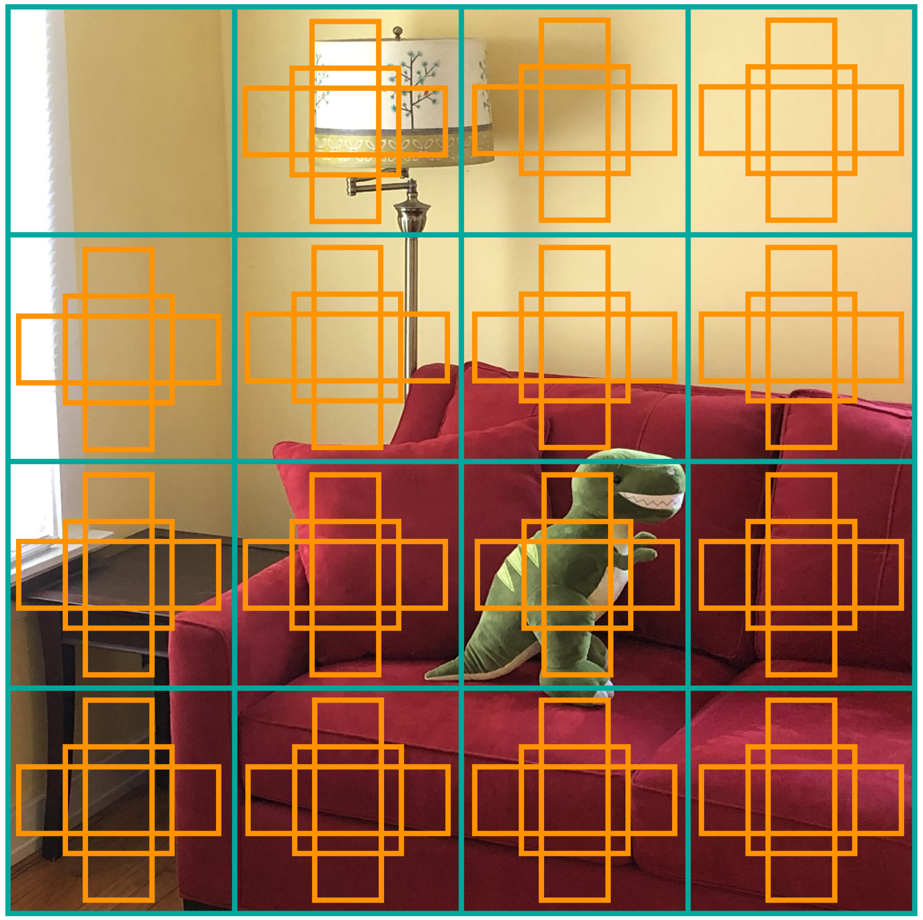
- where the objects are
- what type of object each is

Overview of Strategy:

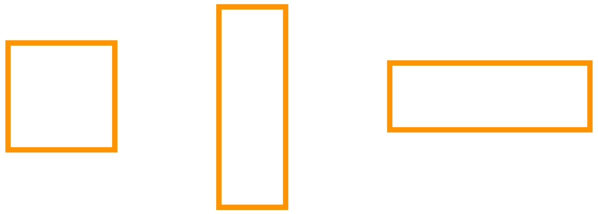
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

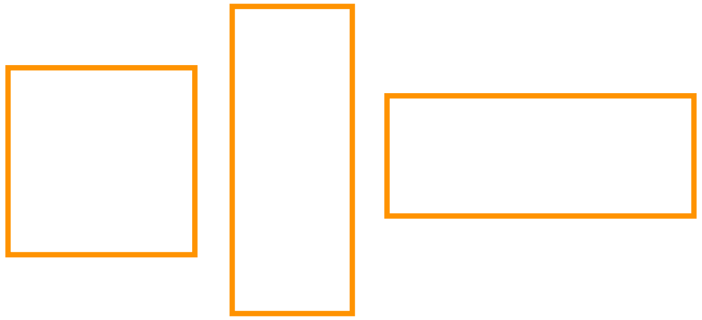
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

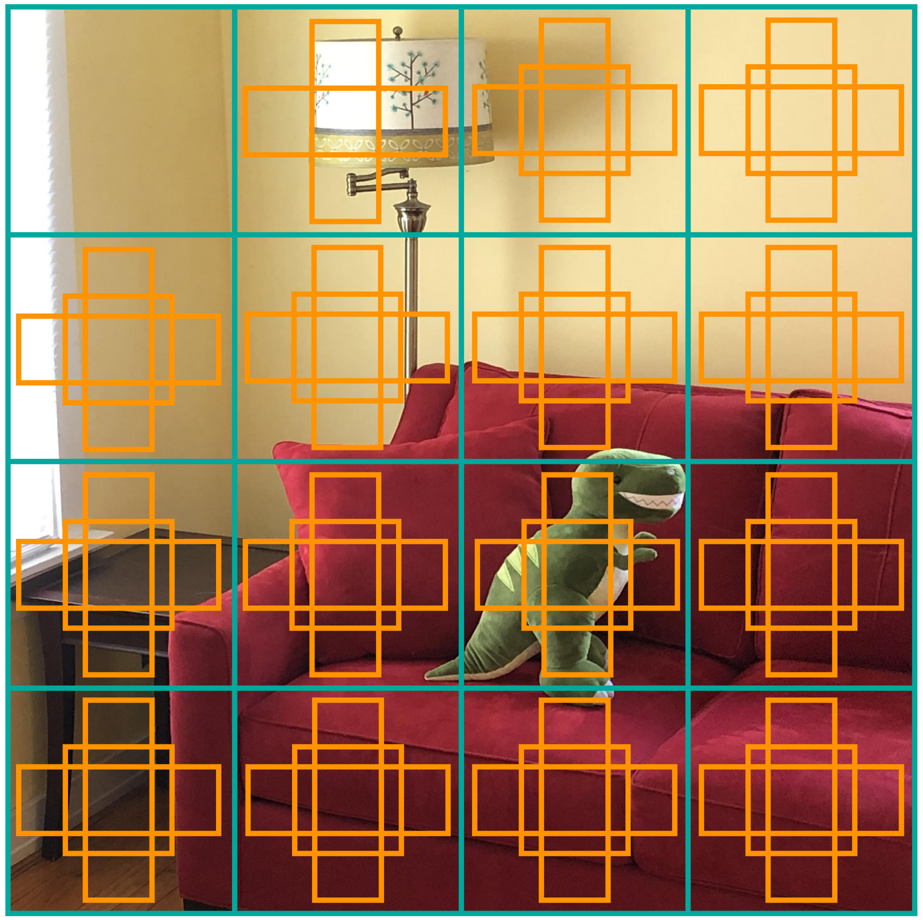
- where the objects are
- what type of object each is

Overview of Strategy:

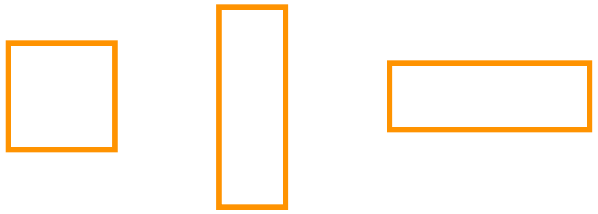
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
    - Hope each cell contains a small number of objects
    - Narrow down location within image
- For each cell, associate some anchor boxes
    - Same anchor boxes used in each cell
    - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
    1. Is there an object in the image that both:
        - Has its center within this cell? (not necessarily the full object)
        - Has size and orientation roughly similar to this anchor box?

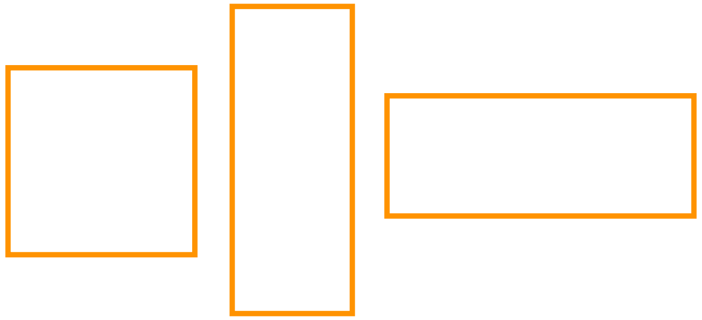
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

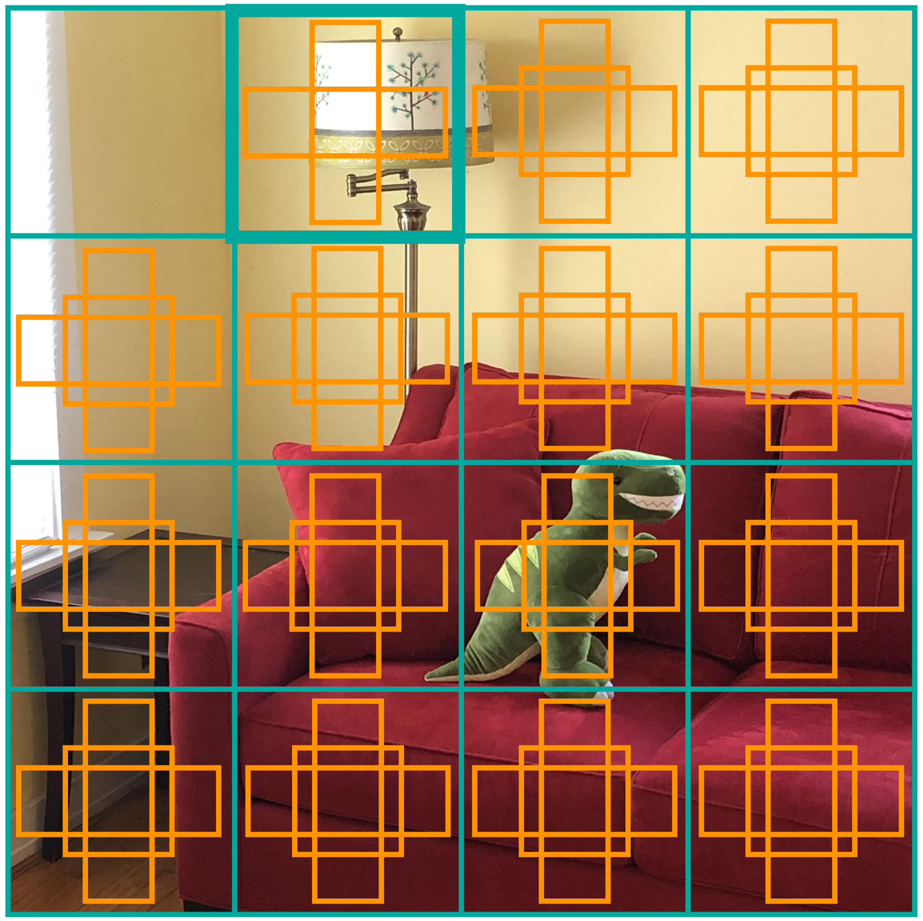
- where the objects are
- what type of object each is

Overview of Strategy:

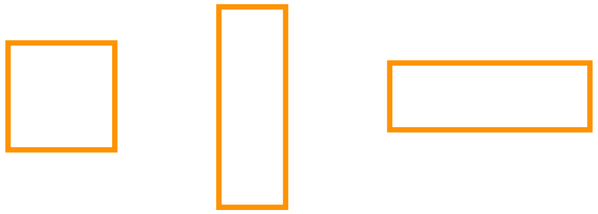
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

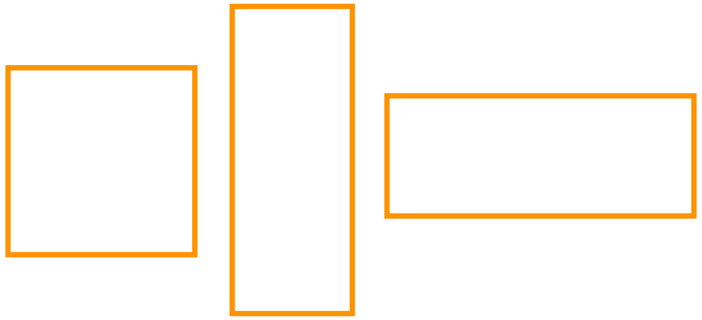
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

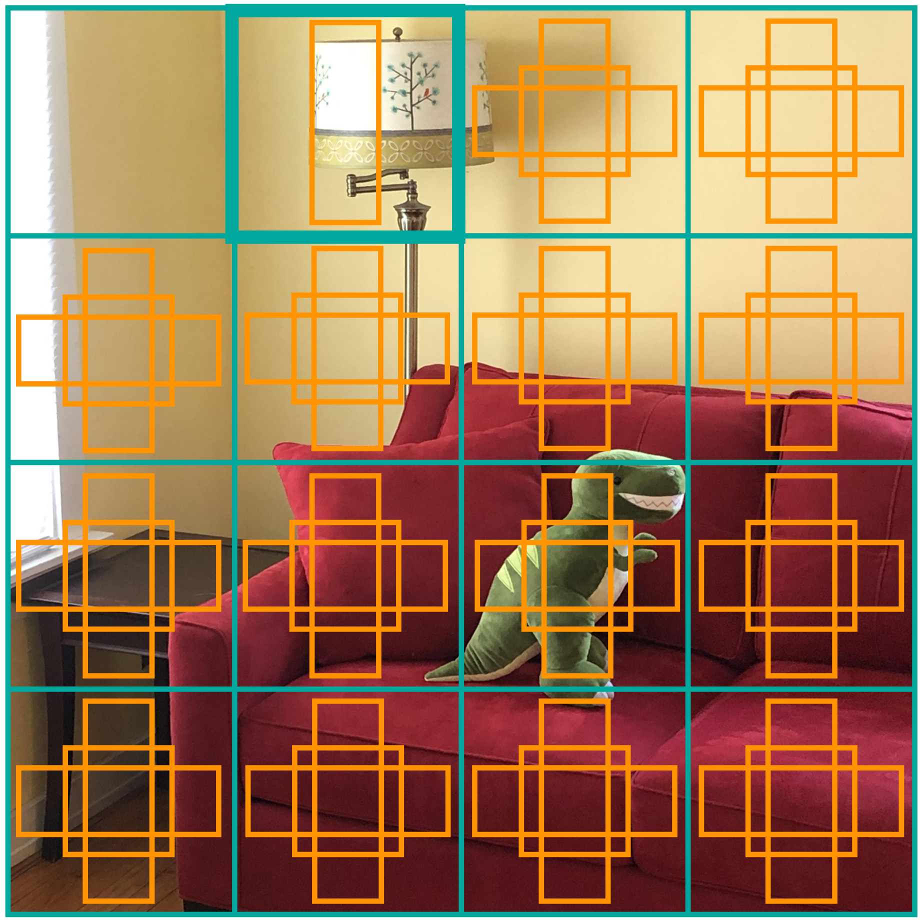
- where the objects are
- what type of object each is

Overview of Strategy:

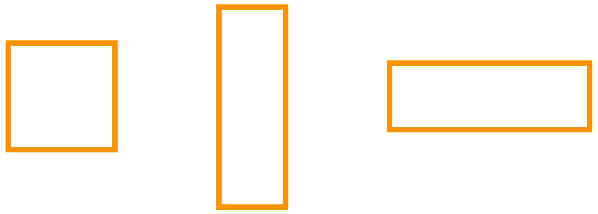
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

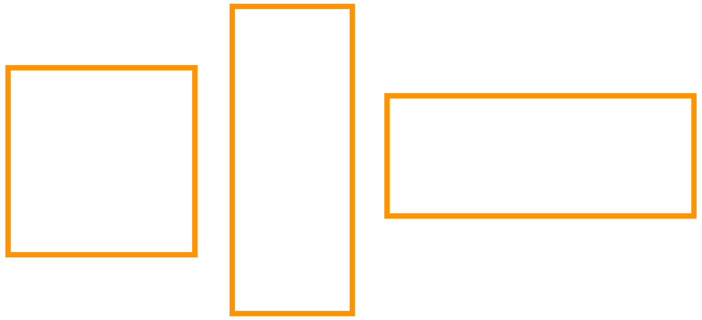
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

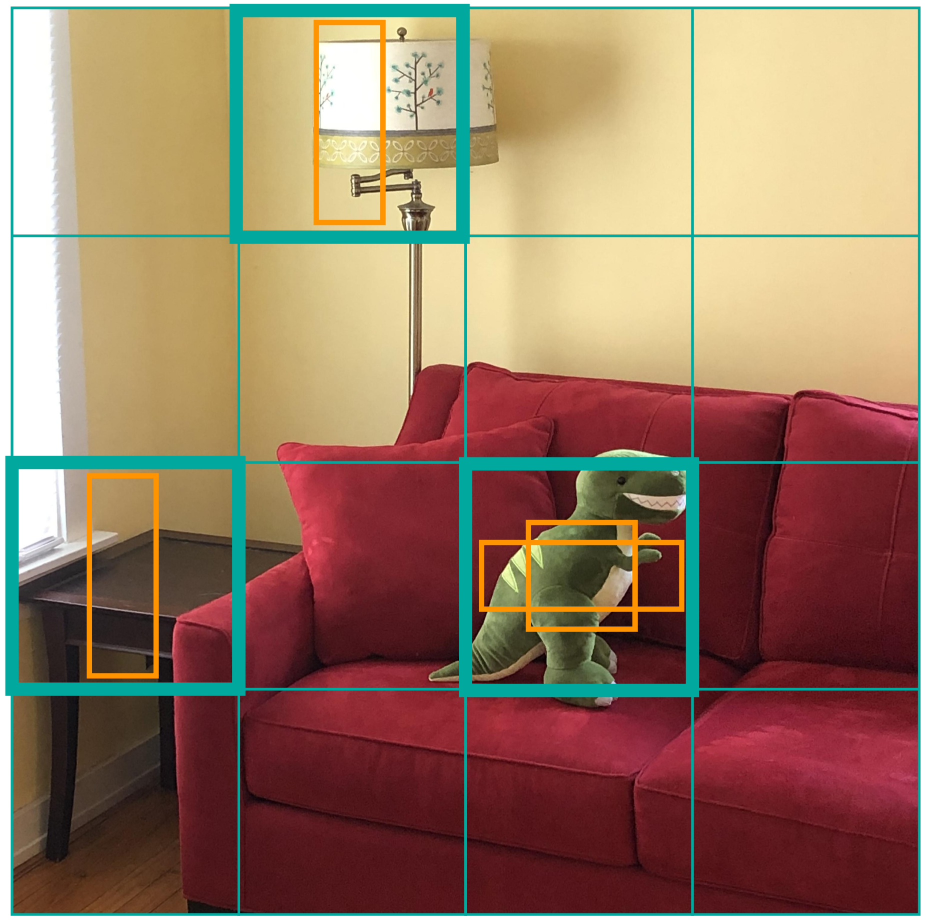
- where the objects are
- what type of object each is

Overview of Strategy:

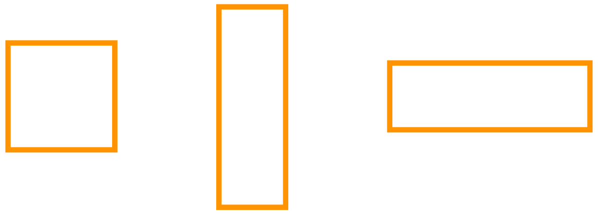
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

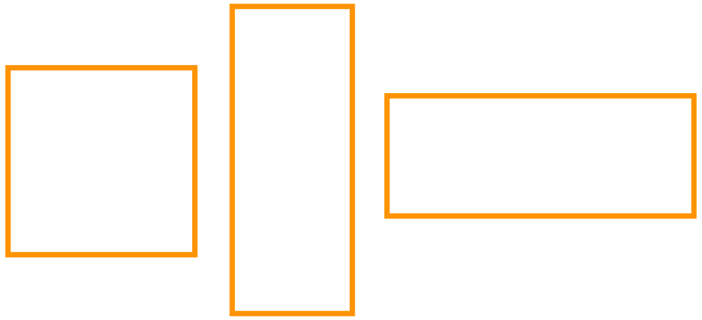
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

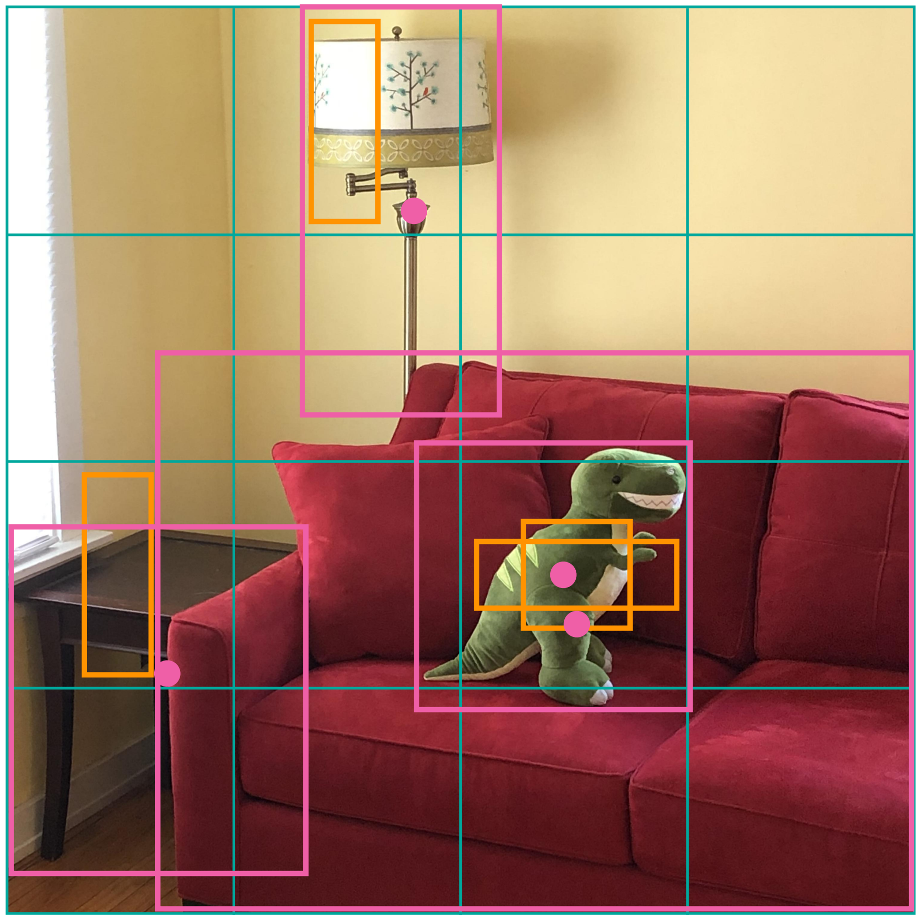
- where the objects are
- what type of object each is

Overview of Strategy:

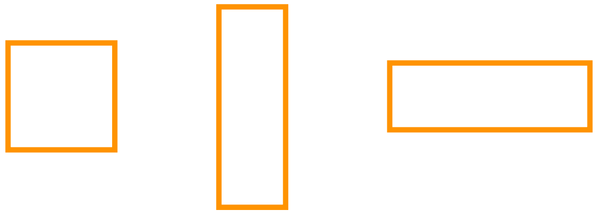
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?

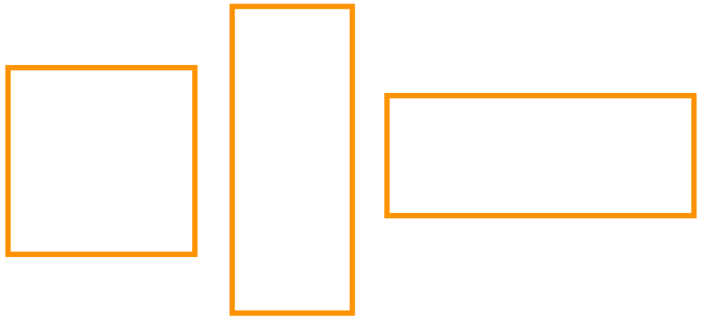
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

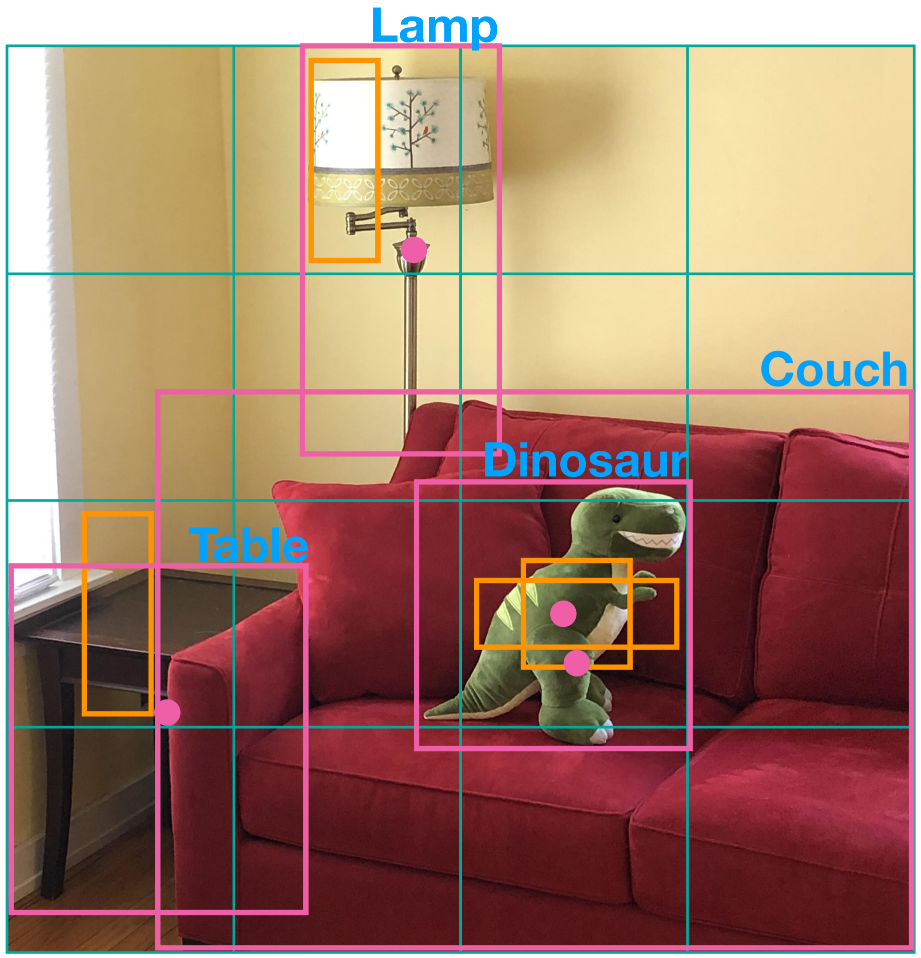
- where the objects are
- what type of object each is

Overview of Strategy:

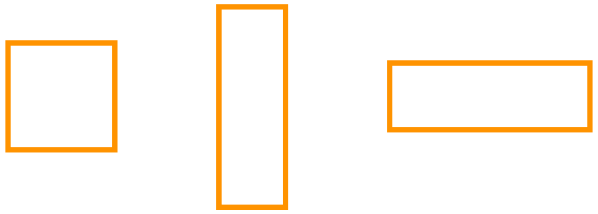
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?
  2. If there is an object, what is the object's location?
     - Where is its center, (b_x, b_y)?
     - What are its width and height, (b_w, b_h)?

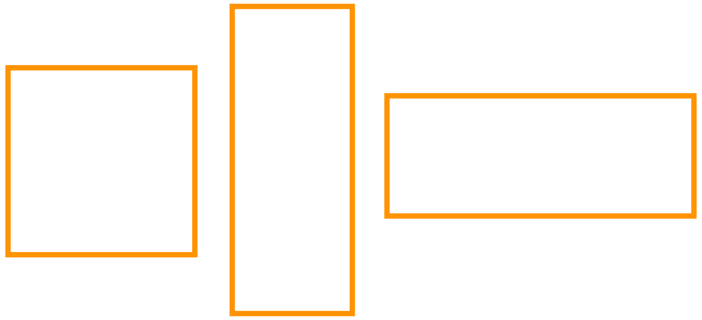
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



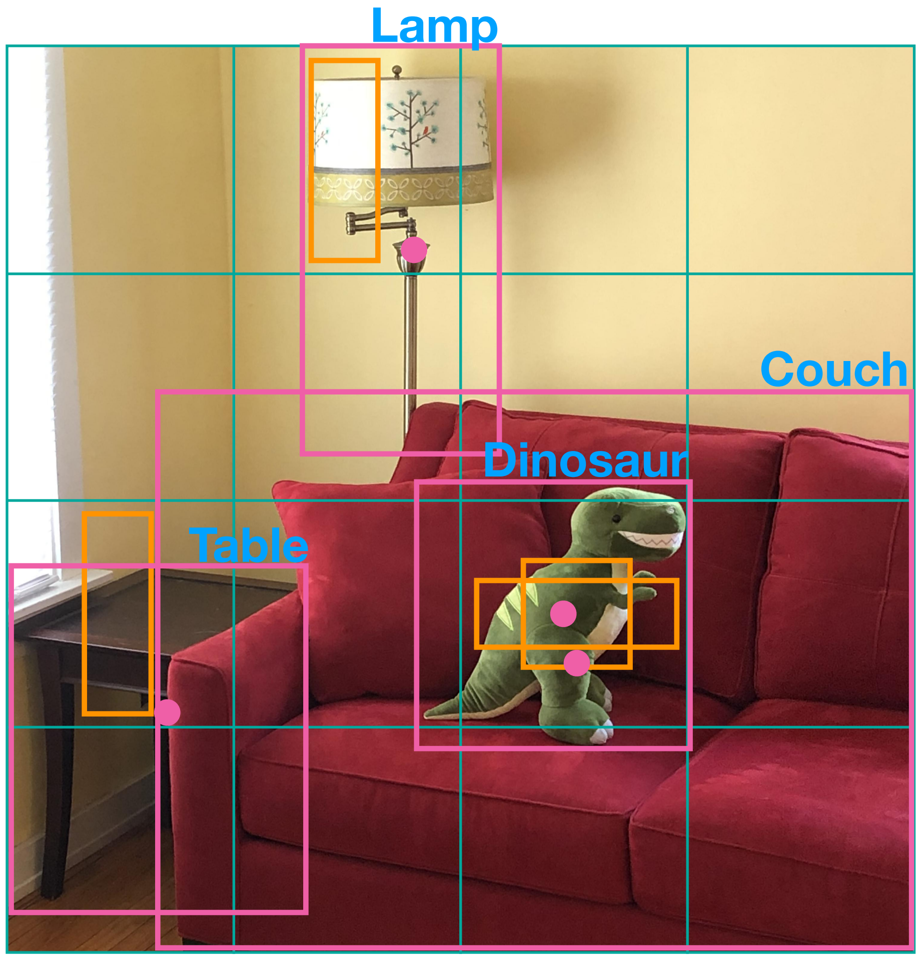
Goal: In an image with multiple objects, identify:

- where the objects are
- what type of object each is

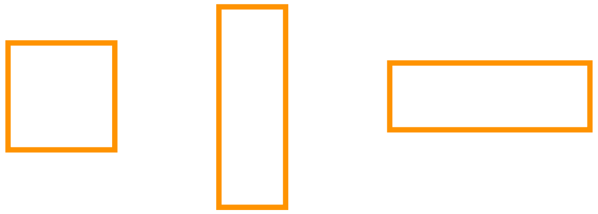
Overview of Strategy:
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?
  2. If there is an object, what is the object's location?
     - Where is its center, (b_x, b_y)?
     - What are its width and height, (b_w, b_h)?
  3. If there is an object, what class of object is it?
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…

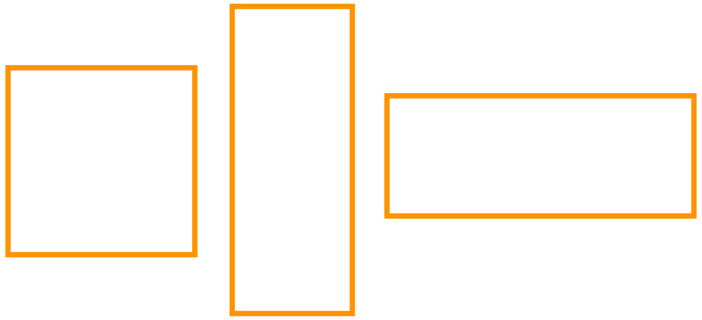
Object Detection (YOLO Algorithm)



Anchor boxes: one copy of each is associated with every grid cell



(in real life, also include different sizes)



Goal: In an image with multiple objects, identify:

- where the objects are
- what type of object each is

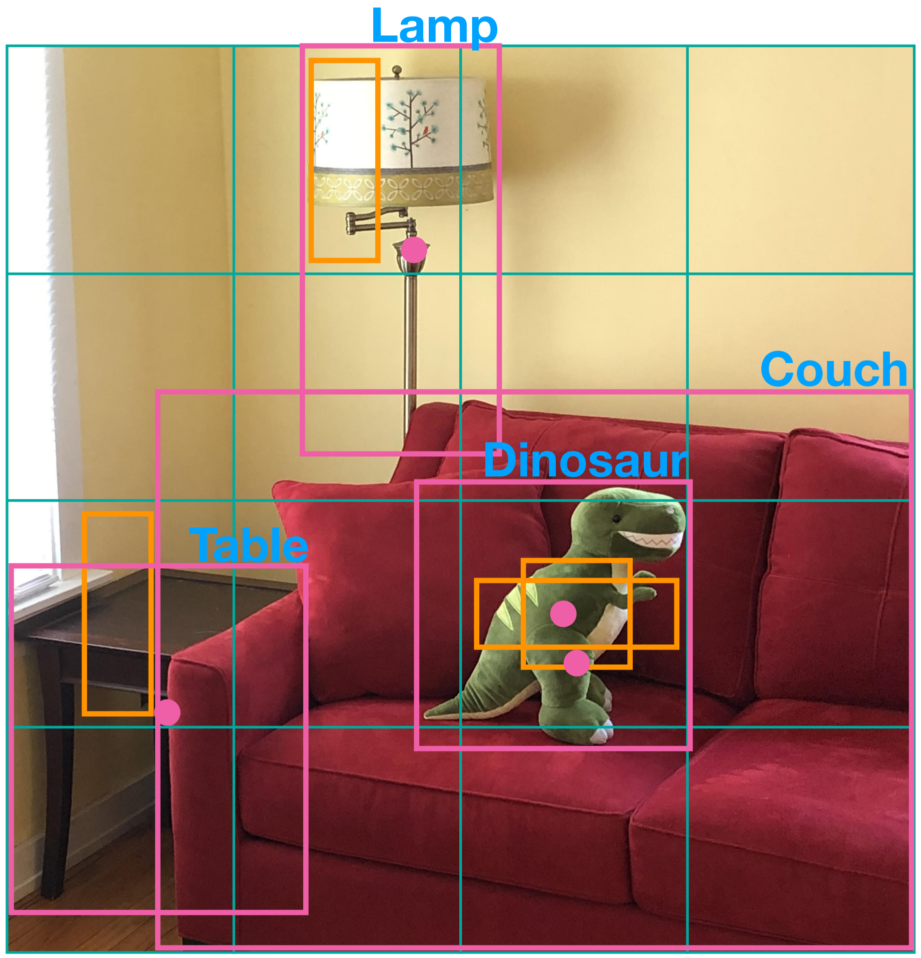
Overview of Strategy:
A neural network does the following steps:

- Divide image into a grid of cells. We'll try to identify objects within each cell
  - Hope each cell contains a small number of objects
  - Narrow down location within image
- For each cell, associate some anchor boxes
  - Same anchor boxes used in each cell
  - Anchor boxes have a range of sizes and orientations
- For each combination of cell and anchor box, ask 3 questions:
  1. Is there an object in the image that both:
     - Has its center within this cell? (not necessarily the full object)
     - Has size and orientation roughly similar to this anchor box?
  2. If there is an object, what is the object's location?
     - Where is its center, (b_x, b_y)?
     - What are its width and height, (b_w, b_h)?
  3. If there is an object, what class of object is it?
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…

For validation/test set predictions, post-processing is done outside of the neural network after predictions are made (illustrated later):

- Keep only boxes with high probability of being an object.
- Keep only boxes that don't overlap too much with each other

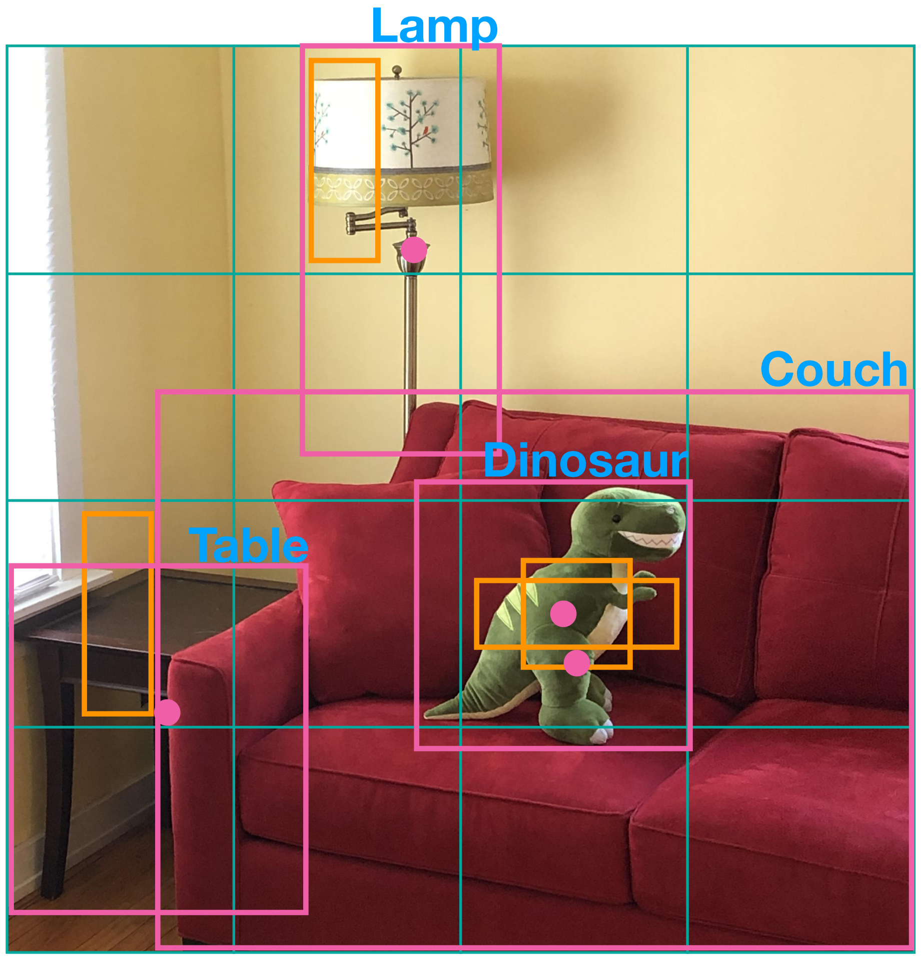
Set Up for Output, Activation, and Loss



Consider one combination of cell and anchor box.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 if contains object, 0 if not
   - Binary classification
   - Sigmoid activation function in output unit
   - Binary cross-entropy loss

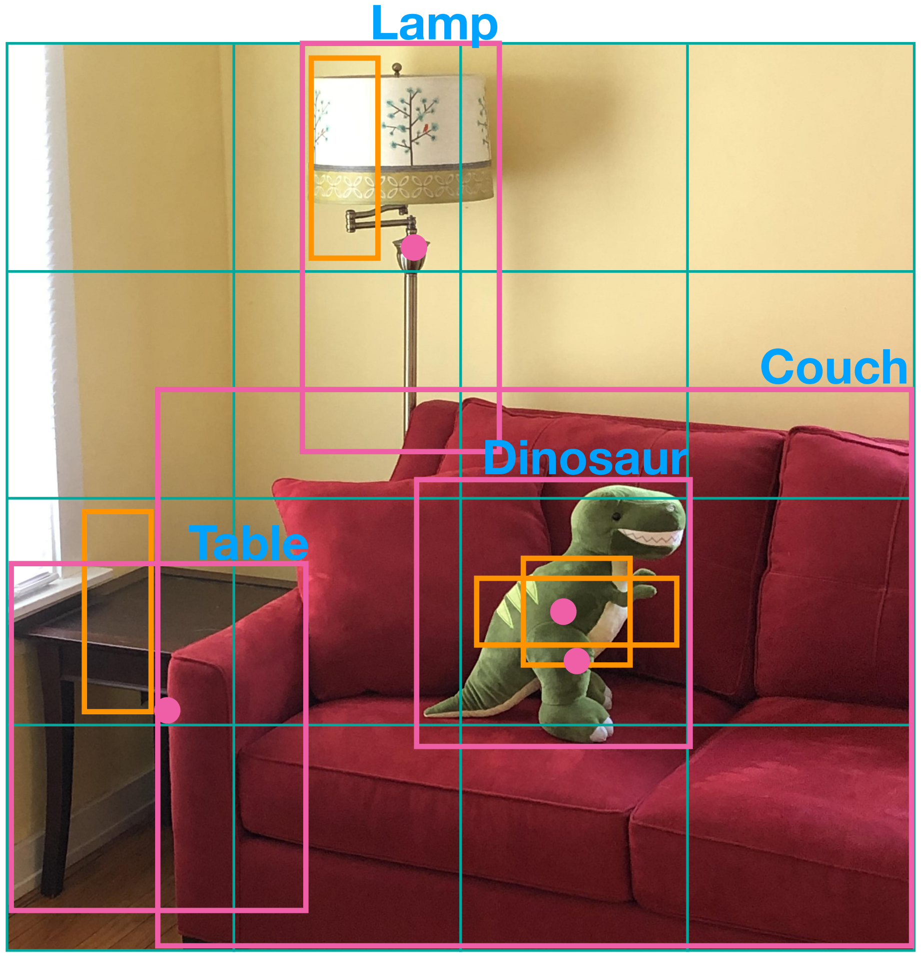
Set Up for Output, Activation, and Loss



Consider one combination of cell and anchor box.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 if contains object, 0 if not
   - Binary classification
   - Sigmoid activation function in output unit
   - Binary cross-entropy loss
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h); center coordinates, width, and height of box
   - Regression
   - Linear activation in output unit
   - Mean squared error loss

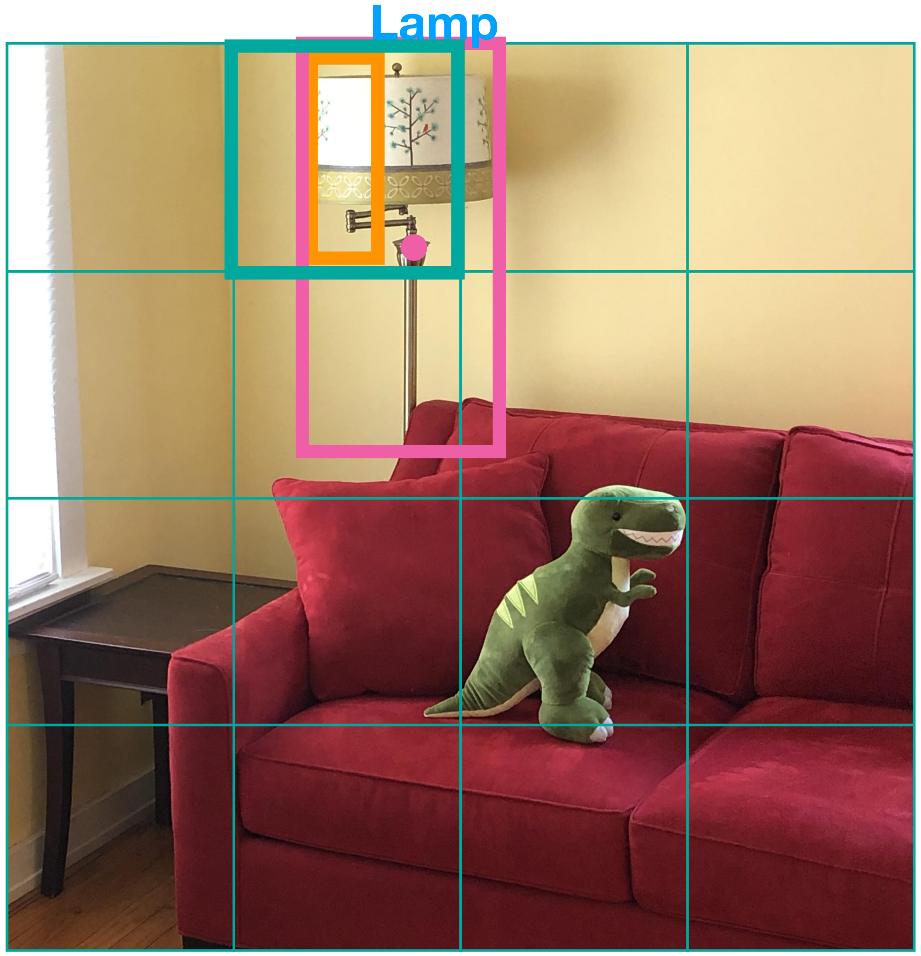
Set Up for Output, Activation, and Loss



Consider one combination of cell and anchor box.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 if contains object, 0 if not
   - Binary classification
   - Sigmoid activation function in output unit
   - Binary cross-entropy loss
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h); center coordinates, width, and height of box
   - Regression
   - Linear activation in output unit
   - Mean squared error loss
3. If there is an object, what class of object is it?
   - Y = one-hot encoding of object type; K possible categories
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…
   - Multi-class classification
   - Softmax activation in output unit
   - Categorical cross-entropy loss

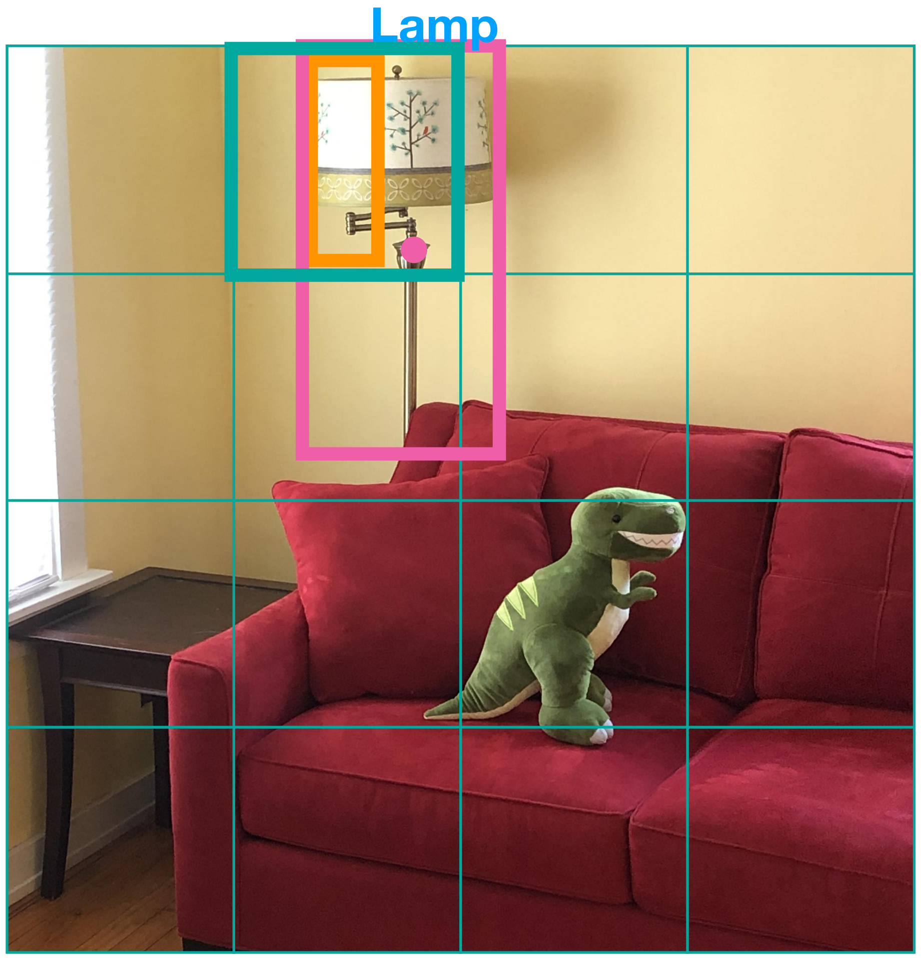
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?

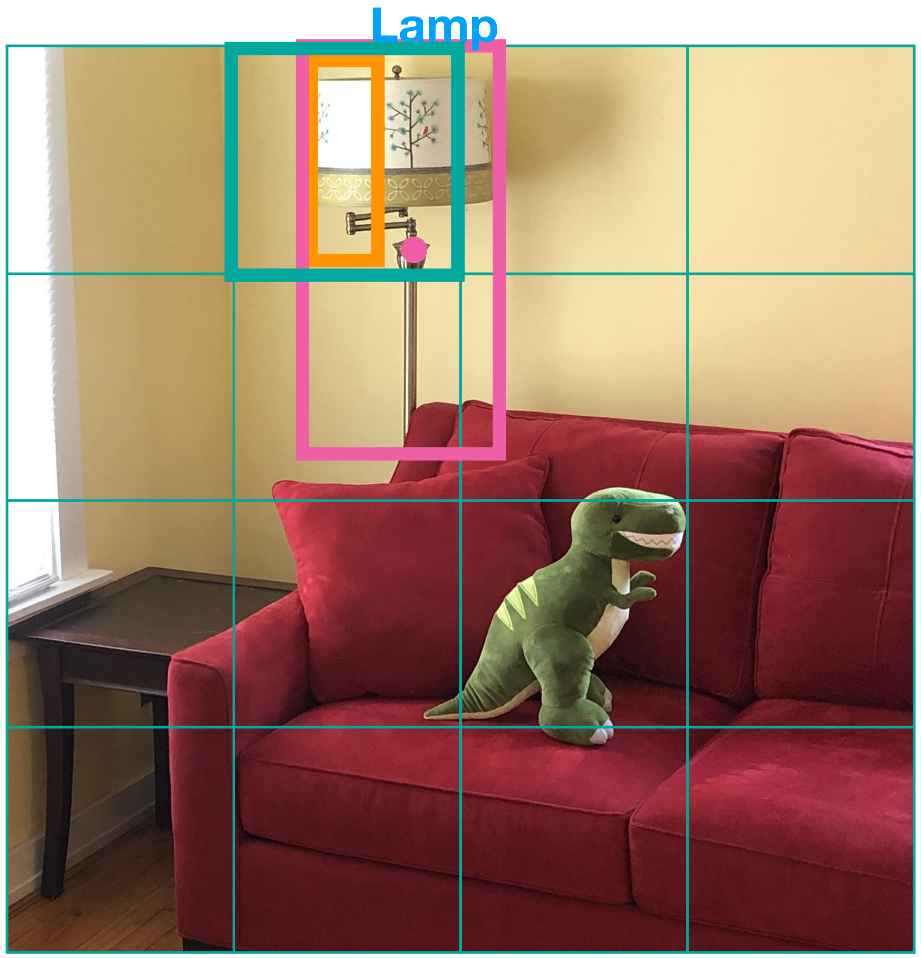
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)

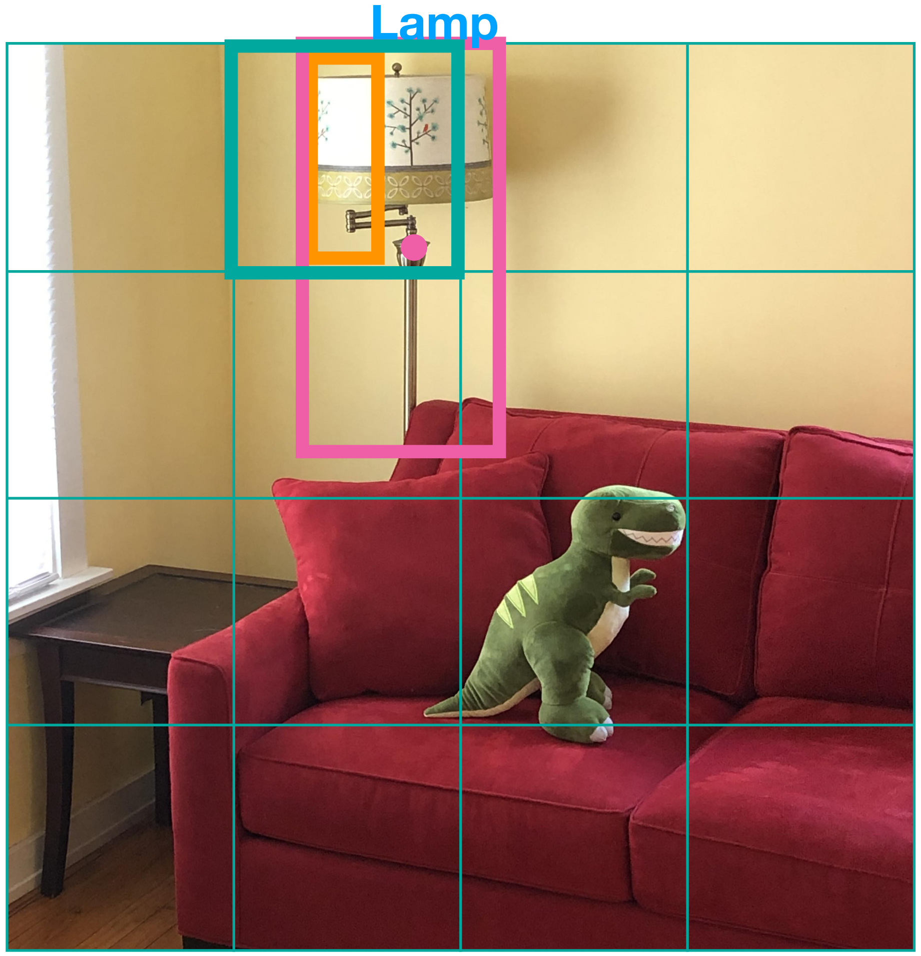
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?

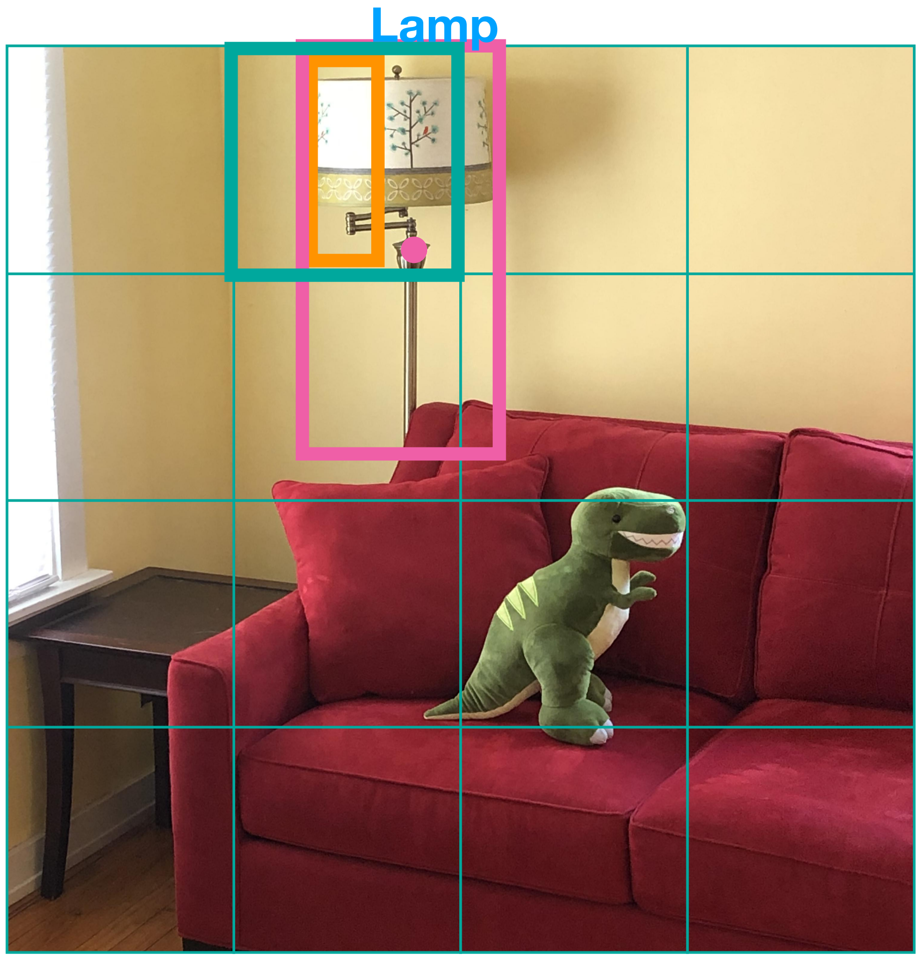
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h) = (0.8, 0.9, 0.8, 1.7)
   - Center coordinates are specified in coordinates relative to this cell
     - Top left corner of cell is (0, 0), lower right corner of cell is (1, 1)
     - Maybe center (pink point) is at (0.8, 0.9)
   - Box width and height are multiples of cell width and height
     - Maybe width and height are (0.8, 1.7) times that of a cell

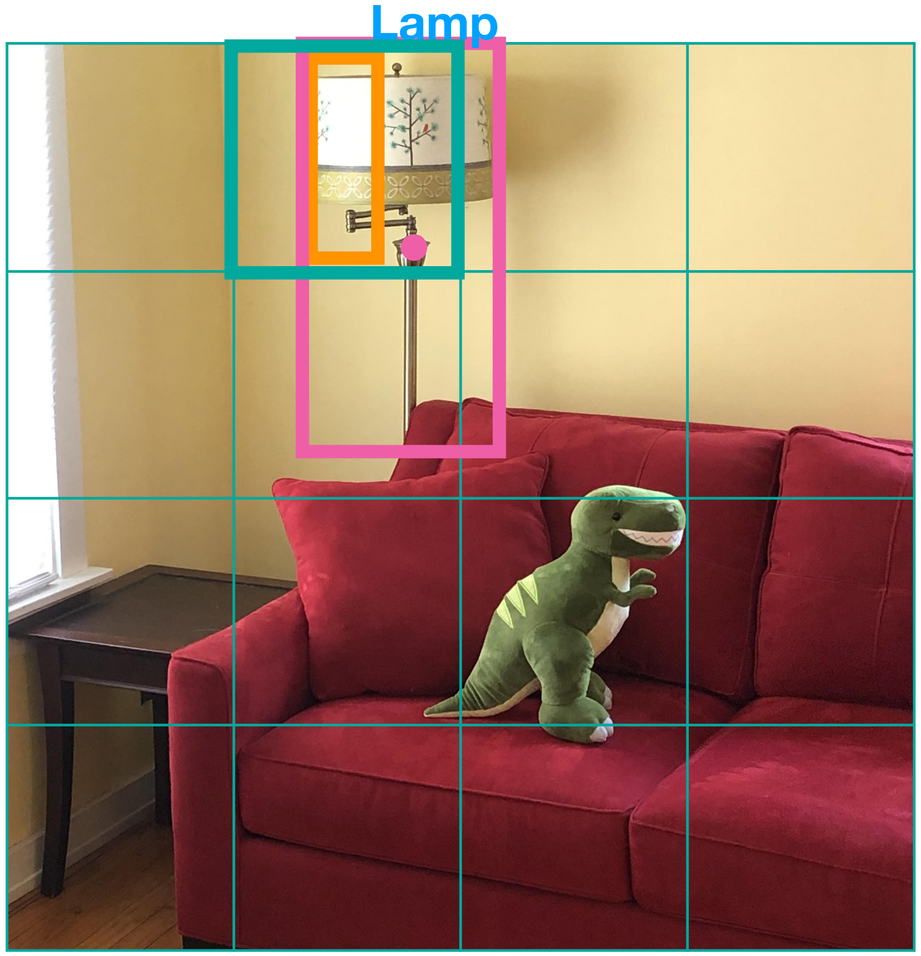
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h) = (0.8, 0.9, 0.8, 1.7)
   - Center coordinates are specified in coordinates relative to this cell
     - Top left corner of cell is (0, 0), lower right corner of cell is (1, 1)
     - Maybe center (pink point) is at (0.8, 0.9)
   - Box width and height are multiples of cell width and height
     - Maybe width and height are (0.8, 1.7) times that of a cell
3. If there is an object, what class of object is it?

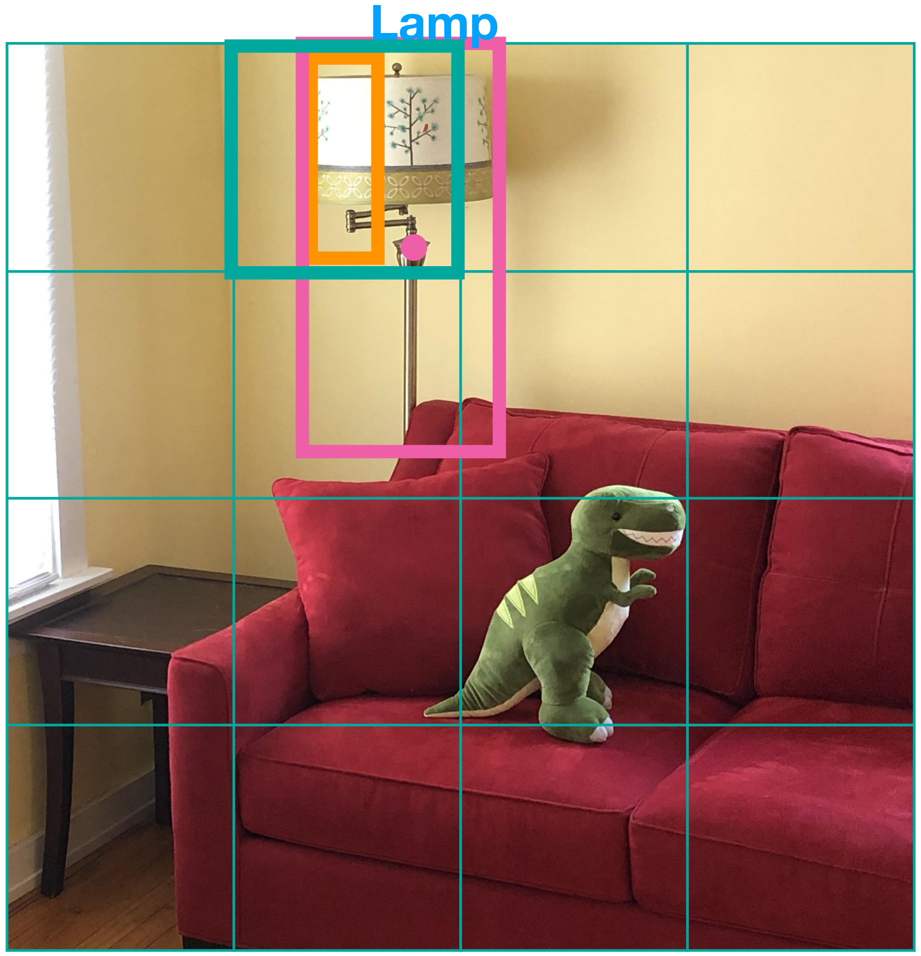
Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h) = (0.8, 0.9, 0.8, 1.7)
   - Center coordinates are specified in coordinates relative to this cell
     - Top left corner of cell is (0, 0), lower right corner of cell is (1, 1)
     - Maybe center (pink point) is at (0.8, 0.9)
   - Box width and height are multiples of cell width and height
     - Maybe width and height are (0.8, 1.7) times that of a cell
3. If there is an object, what class of object is it?
   - y = one-hot encoding of object type; K possible categories
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…
   - y = $\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h) = (0.8, 0.9, 0.8, 1.7)
   - Center coordinates are specified in coordinates relative to this cell
     - Top left corner of cell is (0, 0), lower right corner of cell is (1, 1)
     - Maybe center (pink point) is at (0.8, 0.9)
   - Box width and height are multiples of cell width and height
     - Maybe width and height are (0.8, 1.7) times that of a cell
3. If there is an object, what class of object is it?
   - y = one-hot encoding of object type; K possible categories
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…
   - y = $\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}$

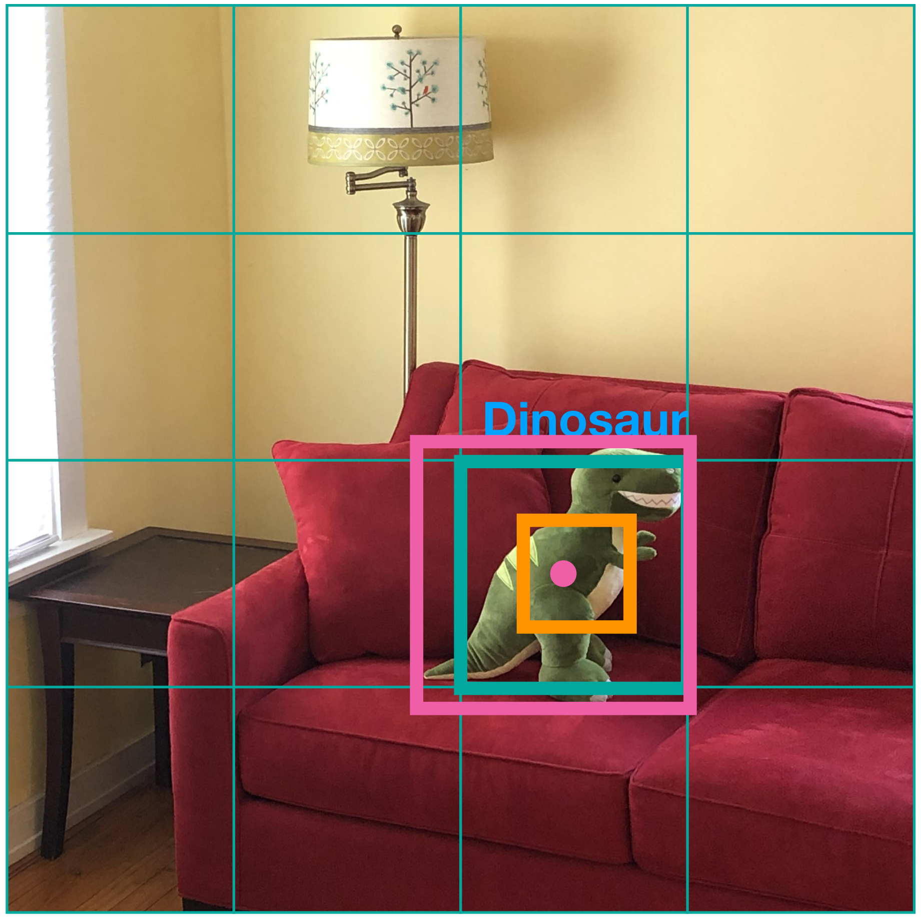
Putting all these pieces together, y = $\begin{bmatrix}1\\0.8\\0.9\\0.8\\1.7\\0\\1\\0\\\vdots\\0\end{bmatrix}$

cell/anchor contains object

center and width/height of object box

class of object

Examples of Output y



Consider the combination of cell and anchor box for the lamp.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 1 (yes, contains object)
2. If there is an object, what is the object's location?
   - y = (b_x, b_y, b_w, b_h) = (0.4, 0.5, 1.2, 1.1)
   - Center coordinates are specified in coordinates relative to this cell
     - Top left corner of cell is (0, 0), lower right corner of cell is (1, 1)
     - Maybe center (pink point) is at (0.4, 0.5)
   - Box width and height are multiples of cell width and height
     - Maybe width and height are (1.2, 1.1) times that of a cell
3. If there is an object, what class of object is it?
   - y = one-hot encoding of object type; K possible categories
     - Dinosaur, lamp, table, couch, car, motorcycle, person,…
   - y = $\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

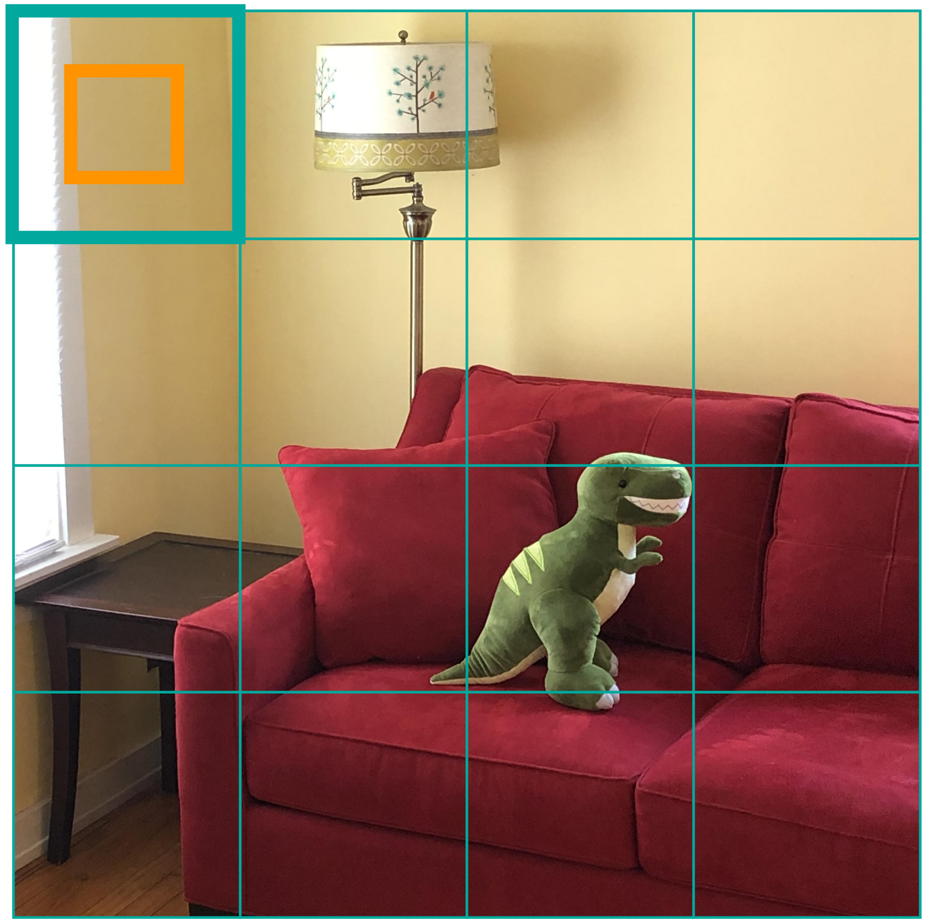
Putting all these pieces together, y = $\begin{bmatrix} 1 \\ 0.4 \\ 0.5 \\ 1.2 \\ 1.1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

cell/anchor contains object

center and width/height of object box

class of object

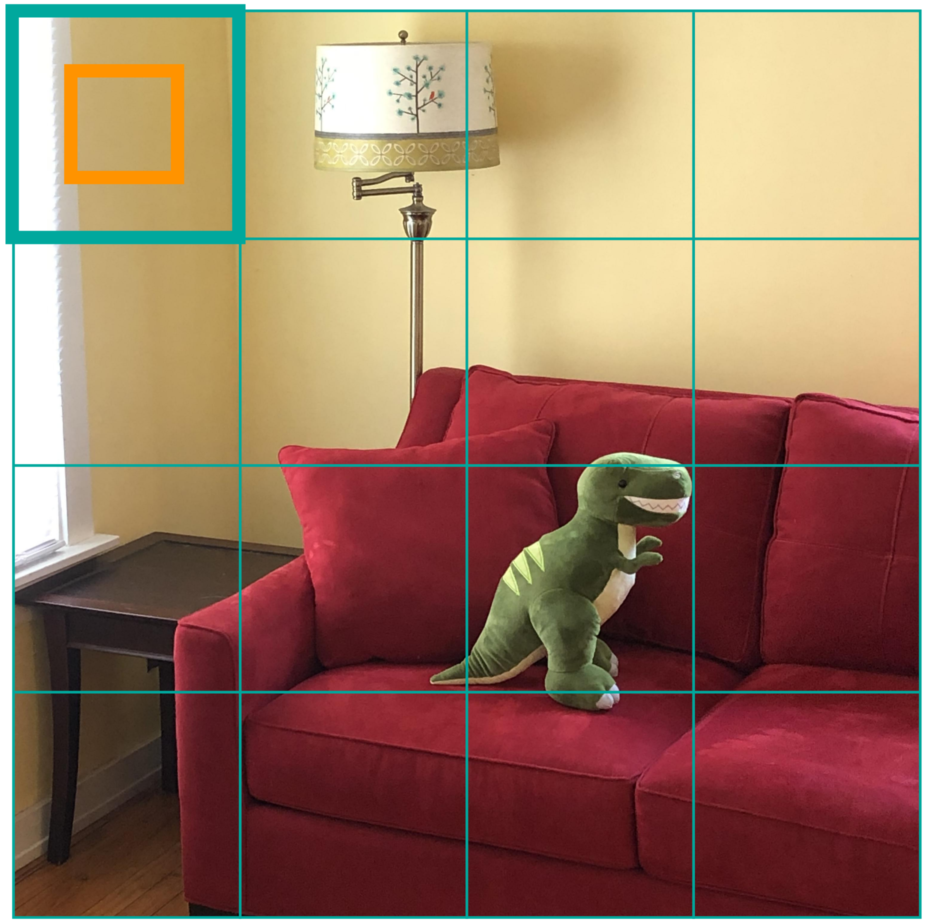
Examples of Output y



Consider a combination of cell and anchor box for the top left cell.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?

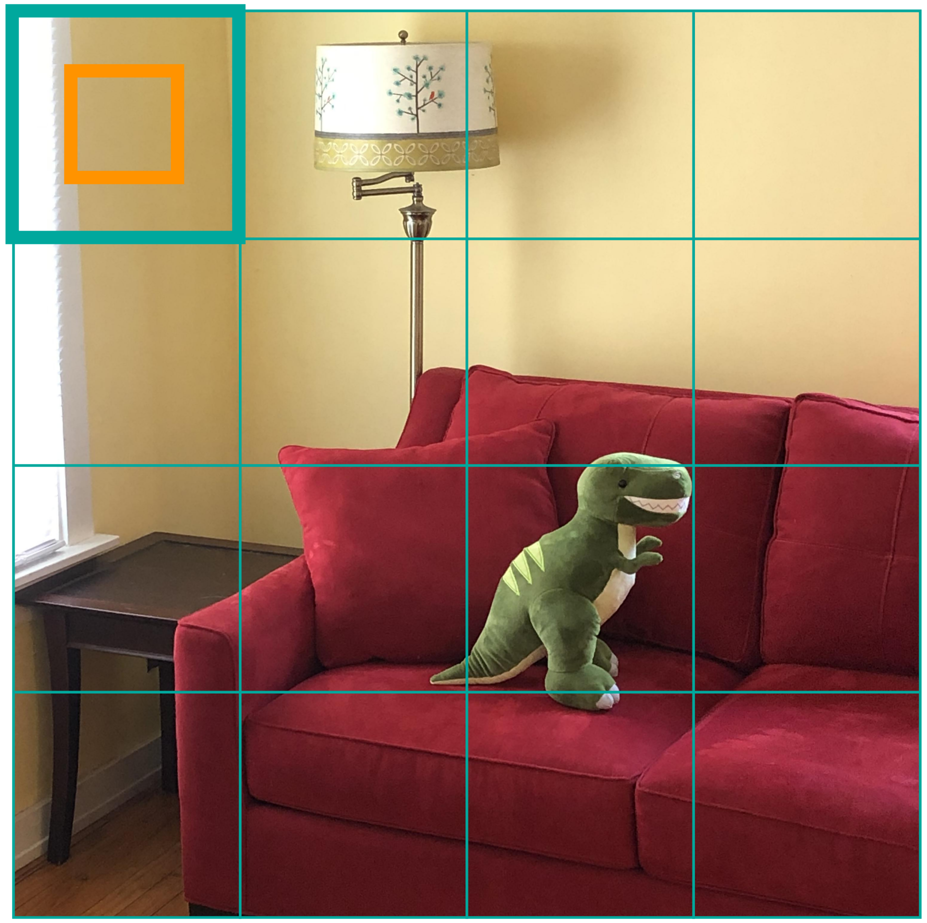
Examples of Output y



Consider a combination of cell and anchor box for the top left cell.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 0 (no, does not contain object)
2. If there is an object, what is the object's location?

Examples of Output y

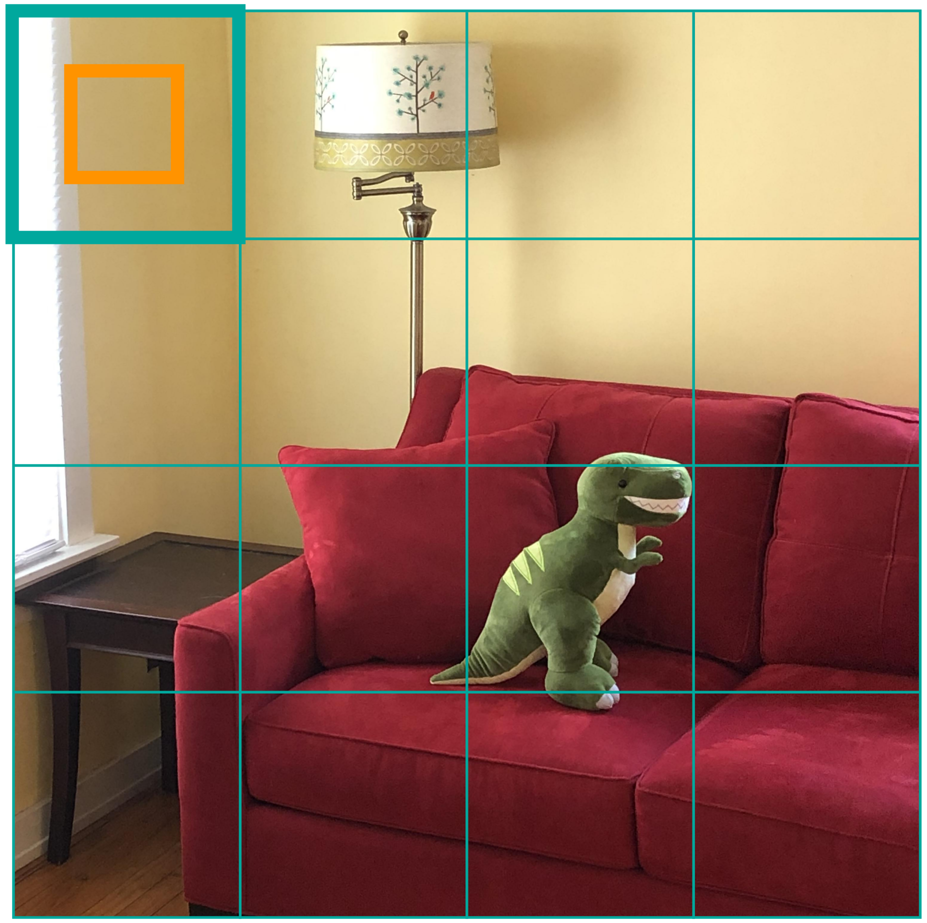


Consider a combination of cell and anchor box for the top left cell.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 0 (no, does not contain object)
2. If there is an object, what is the object's location?
   - Trick question!! No object.
   - y = (b_x, b_y, b_w, b_h) = (__, __, __, __)
   - Doesn't matter how you fill in the blank, this part of y will not contribute to the loss function
3. If there is an object, what class of object is it?
   - Doesn't matter what you use, this part of y will not contribute to the loss function

- y = $\begin{bmatrix} \_ \\ \_ \\ \_ \\ \vdots \\ \_ \end{bmatrix}$

Examples of Output y



Consider a combination of cell and anchor box for the top left cell.
Our neural network must output predictions for our three questions:

1. Does this combination of grid cell and anchor box contain an object?
   - y = 0 (no, does not contain object)
2. If there is an object, what is the object's location?
   - Trick question!! No object.
   - y = (b_x, b_y, b_w, b_h) = (__, __, __, __)
   - Doesn't matter how you fill in the blank, this part of y will not contribute to the loss function
3. If there is an object, what class of object is it?
   - Doesn't matter what you use, this part of y will not contribute to the loss function

   - y = $\begin{bmatrix} \_ \\ \_ \\ \_ \\ \vdots \\ \_ \end{bmatrix}$

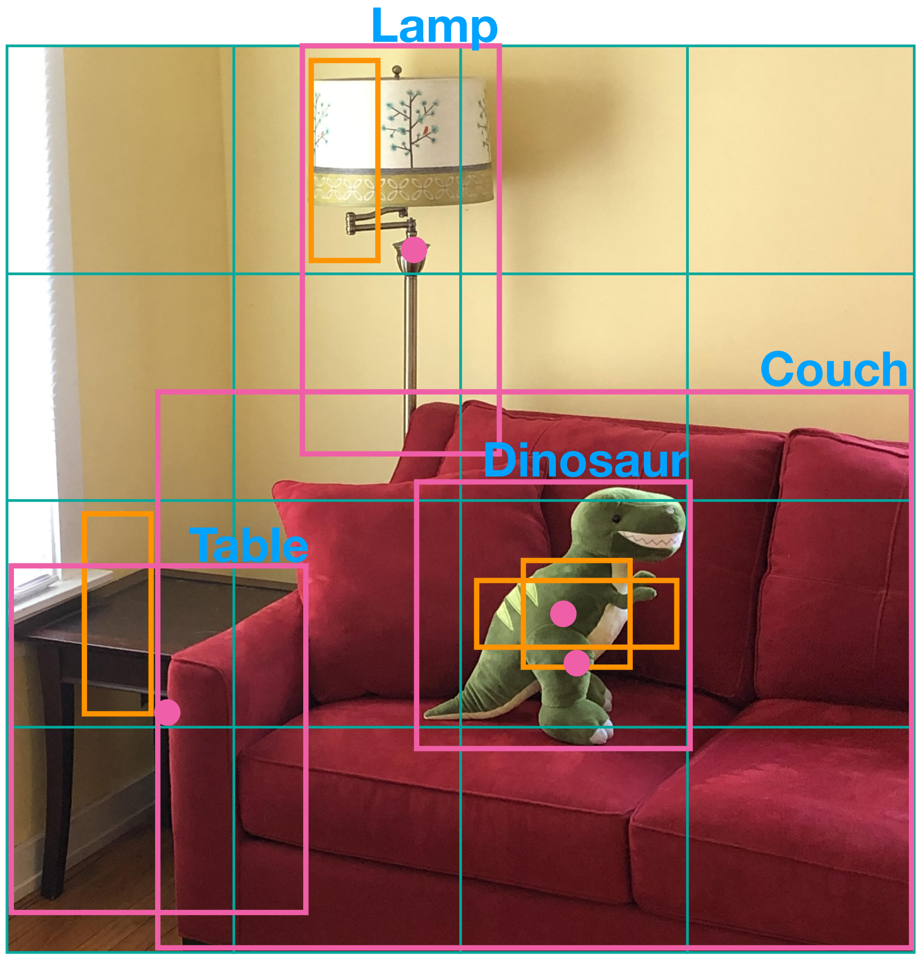
Putting all these pieces together, y = $\begin{bmatrix} 0 \\ \_ \\ \_ \\ \_ \\ \_ \\ \_ \\ \_ \\ \_ \\ \vdots \\ \_ \end{bmatrix}$

cell/anchor doesn't contain object

center and width/height of object box

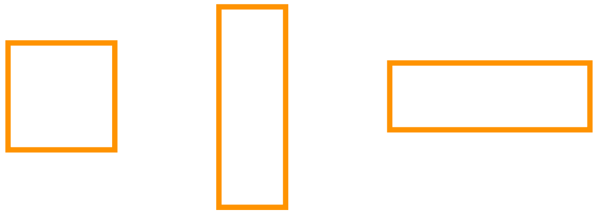
class of object

Examples of Output y

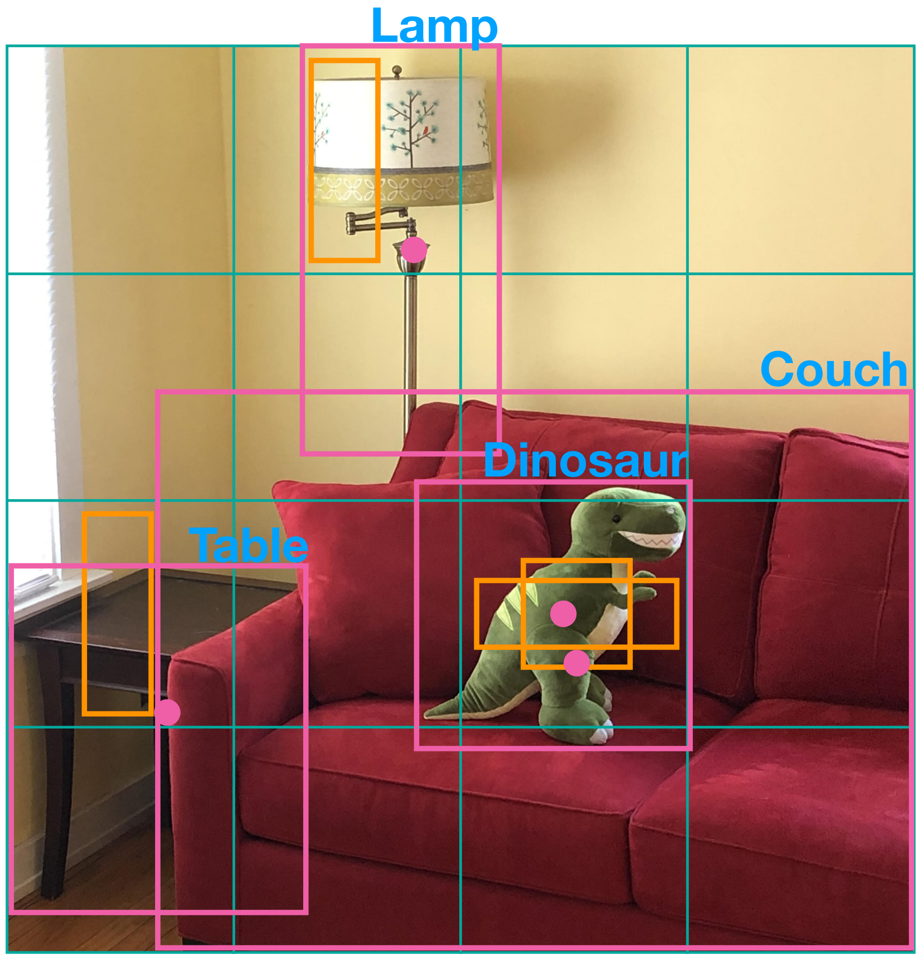


Our network must output predictions for all cells and anchor boxes!
Combined, for this image y is an array of shape **(4, 4, 3 * (1 + 4 + K))**

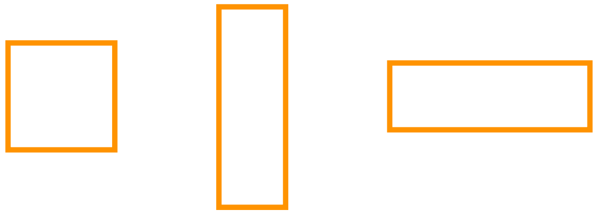
Anchor boxes: one copy of each is associated with every grid cell



Examples of Output y



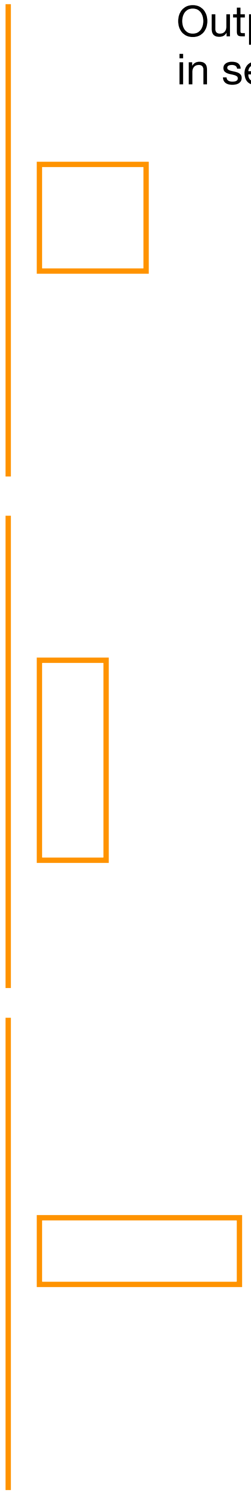
Anchor boxes: one copy of each is associated with every grid cell



Our network must output predictions for all cells and anchor boxes!
Combined, for this image y is an array of shape **(4, 4, 3 * (1 + 4 + K))**

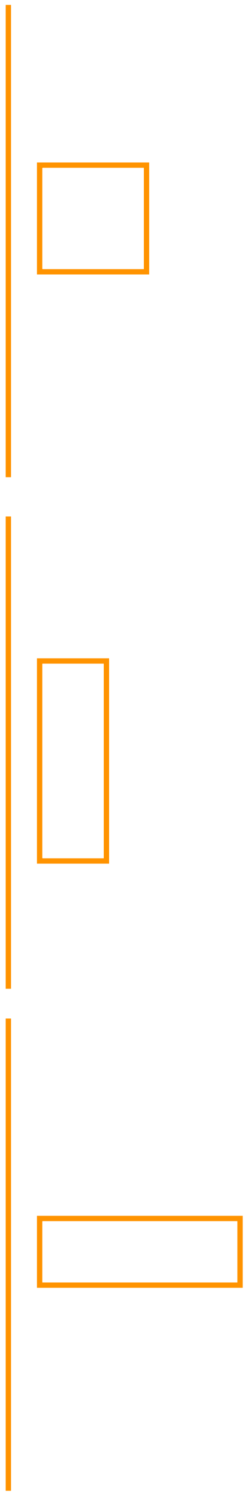
Outputs for all 3 anchor boxes in top left grid cell:

$$y[0, 0, :] = \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ \vdots \\ - \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ \vdots \\ - \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ \vdots \\ - \end{bmatrix}$$



Outputs for all 3 anchor boxes in second grid cell of top row:

$$y[0, 1, :] = \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ \vdots \\ - \\ 1 \\ 0.8 \\ 0.9 \\ 0.8 \\ 1.7 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ \vdots \\ - \end{bmatrix}$$



Loss Function

Contributions from a single cell and anchor box:

Case 1: Cell/anchor box contains an object

$$J^{cell,\,anchor}(b,w) = J_{object}^{cell,\,anchor}(b,w) + J_{location}^{cell,\,anchor}(b,w) + J_{class}^{cell,\,anchor}(b,w)$$

Binary cross-entropy for object existence

Mean squared error for object location

Categorical cross-entropy for object class

Loss Function

Contributions from a single cell and anchor box:

Case 1: Cell/anchor box contains an object

$$J^{cell,\, anchor}(b,w) = J^{cell,\, anchor}_{object}(b,w) + J^{cell,\, anchor}_{location}(b,w) + J^{cell,\, anchor}_{class}(b,w)$$

Binary cross-entropy for object existence

Mean squared error for object location

Categorical cross-entropy for object class

Case 2: Cell/anchor box does not contain an object

$$J^{cell,\, anchor}(b,w) = J^{cell,\, anchor}_{object}(b,w)$$

Binary cross-entropy for object existence

Loss Function

Contributions from a single cell and anchor box:

Case 1: Cell/anchor box contains an object

$$J^{cell,\,anchor}(b,w) = J^{cell,\,anchor}_{object}(b,w) + J^{cell,\,anchor}_{location}(b,w) + J^{cell,\,anchor}_{class}(b,w)$$

Binary cross-entropy for object existence

Mean squared error for object location

Categorical cross-entropy for object class

Case 2: Cell/anchor box does not contain an object

$$J^{cell,\,anchor}(b,w) = J^{cell,\,anchor}_{object}(b,w)$$

Binary cross-entropy for object existence

Contributions from all cells and anchor boxes in image (i):

$$J^{(i)}(b,w) = \sum_{cell} \sum_{anchor} J^{(i)\,cell,\,anchor}(b,w)$$

Loss Function

Contributions from a single cell and anchor box:

Case 1: Cell/anchor box contains an object

$$J^{cell,\,anchor}(b,w) = J_{object}^{cell,\,anchor}(b,w) + J_{location}^{cell,\,anchor}(b,w) + J_{class}^{cell,\,anchor}(b,w)$$

Binary cross-entropy for object existence

Mean squared error for object location

Categorical cross-entropy for object class

Case 2: Cell/anchor box does not contain an object

$$J^{cell,\,anchor}(b,w) = J_{object}^{cell,\,anchor}(b,w)$$

Binary cross-entropy for object existence

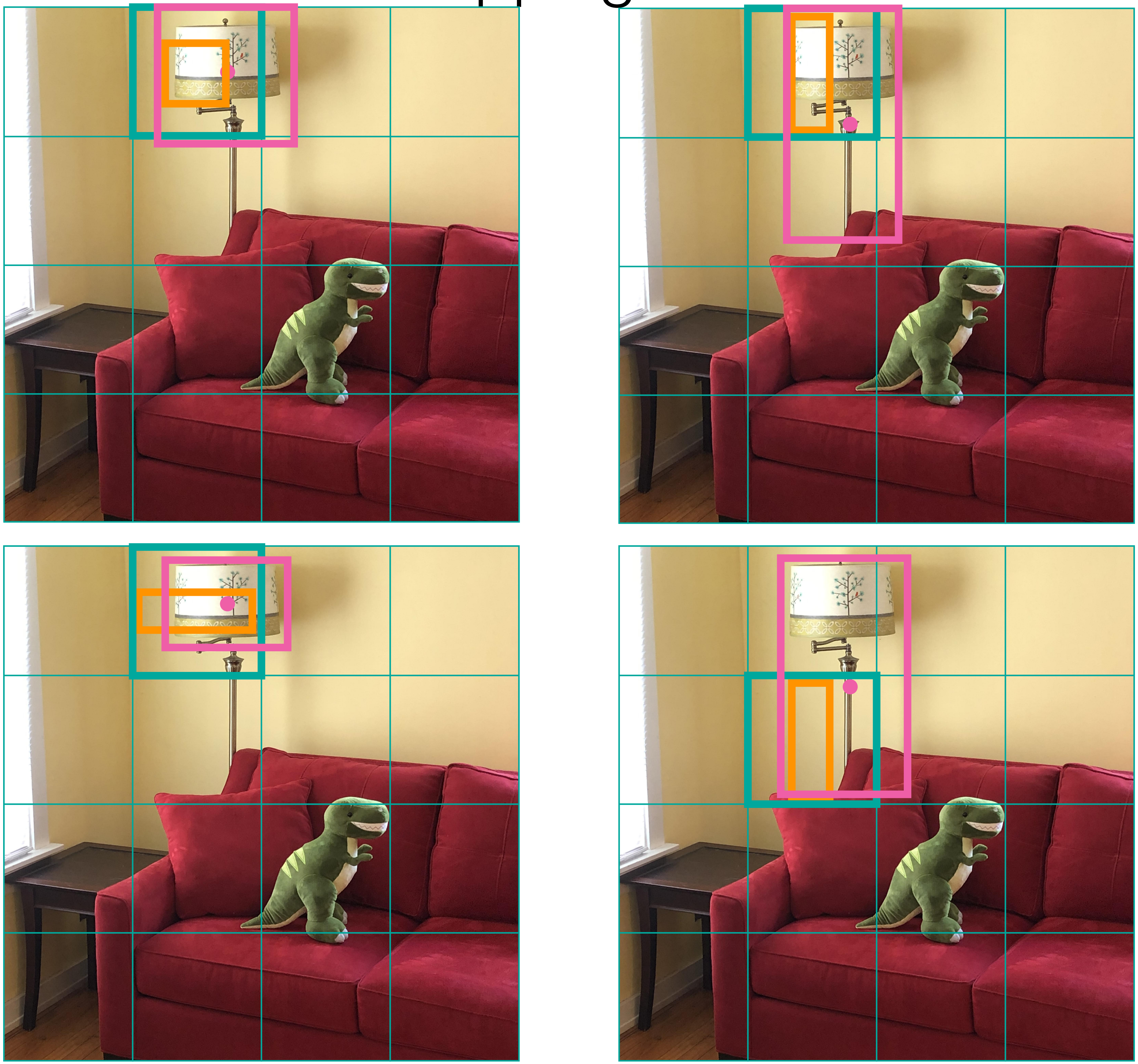
Contributions from all cells and anchor boxes in image (i):

$$J^{(i)}(b,w) = \sum_{cell} \sum_{anchor} J^{(i)\,cell,\,anchor}(b,w)$$

Contributions from all images i = 1, …, m:

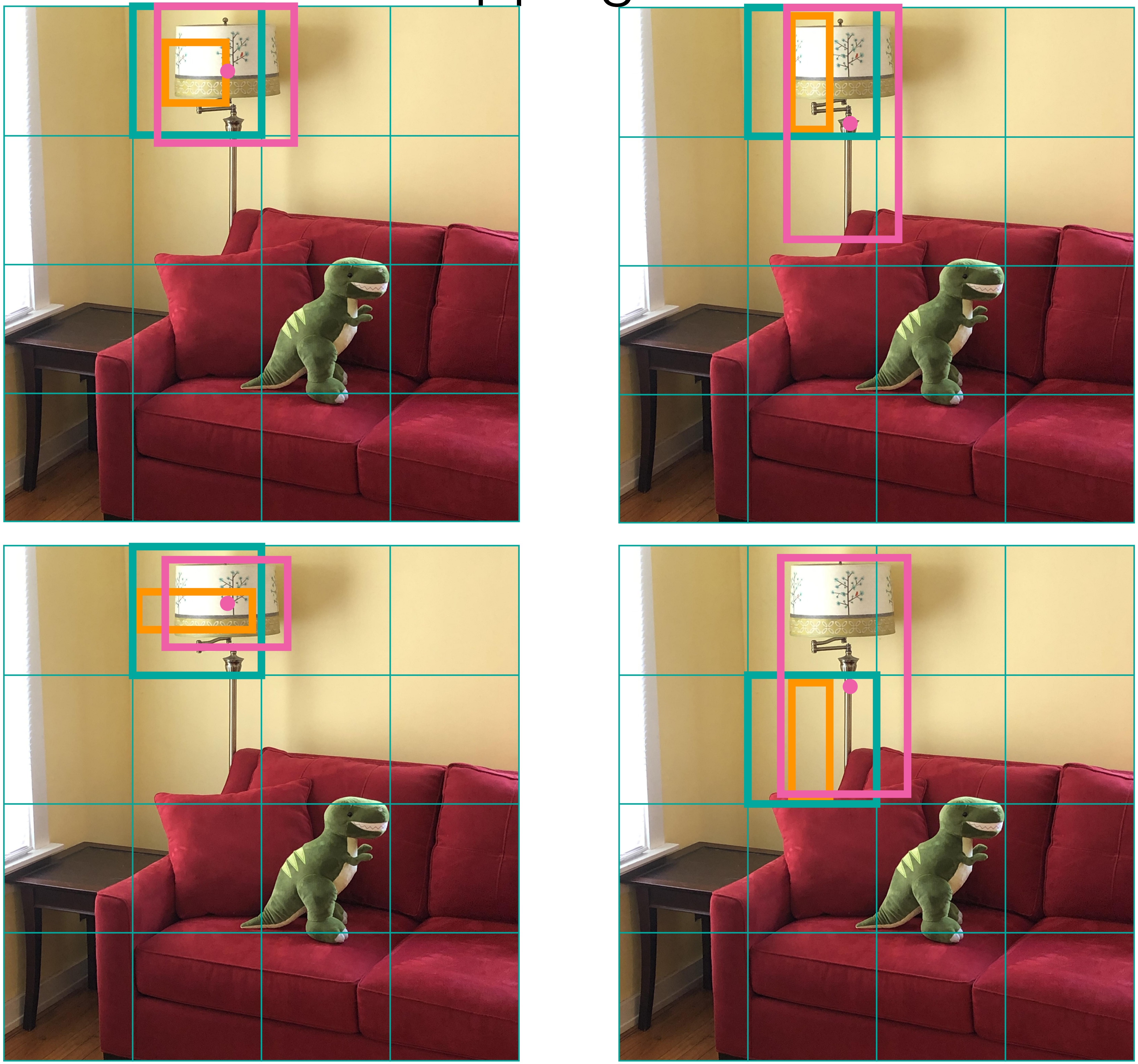
$$J(b,w) = \sum_{i=1}^{m} J^{(i)}(b,w)$$

Overlapping Test Set Predictions



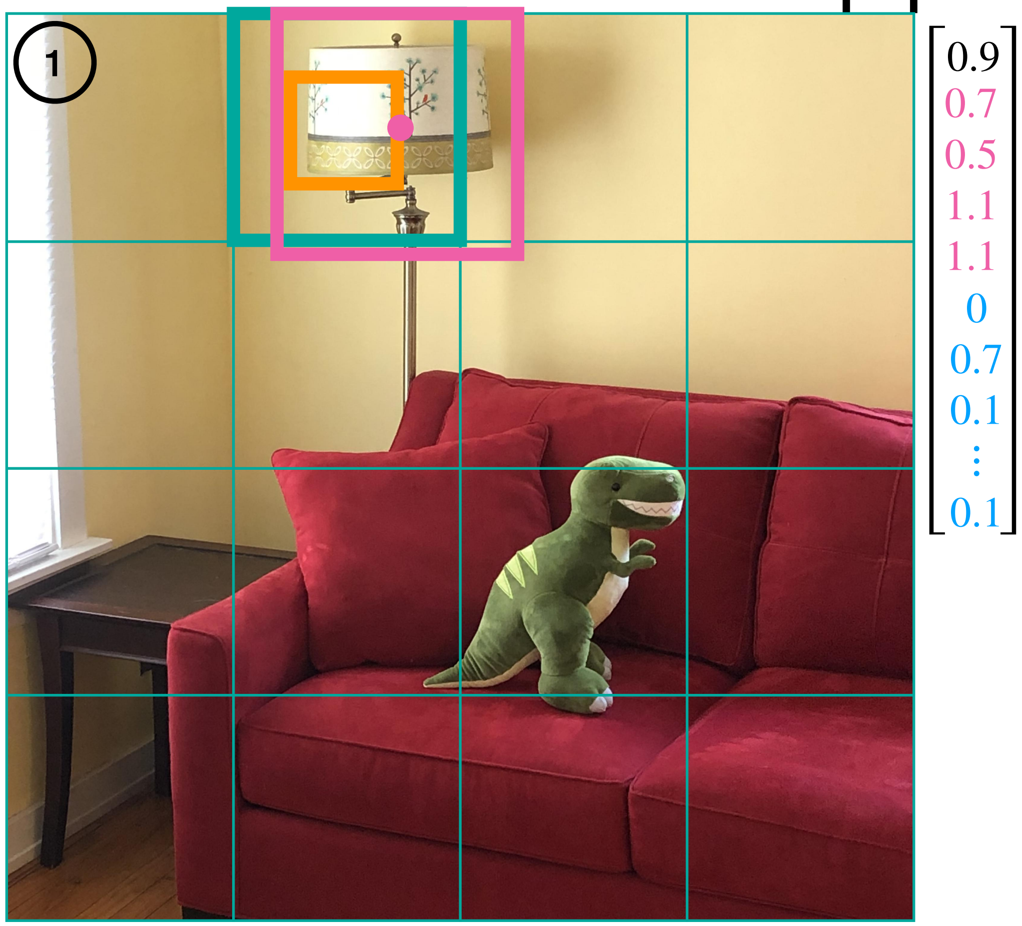
- When we make test set predictions, the same object may be identified for multiple cells and anchor boxes.

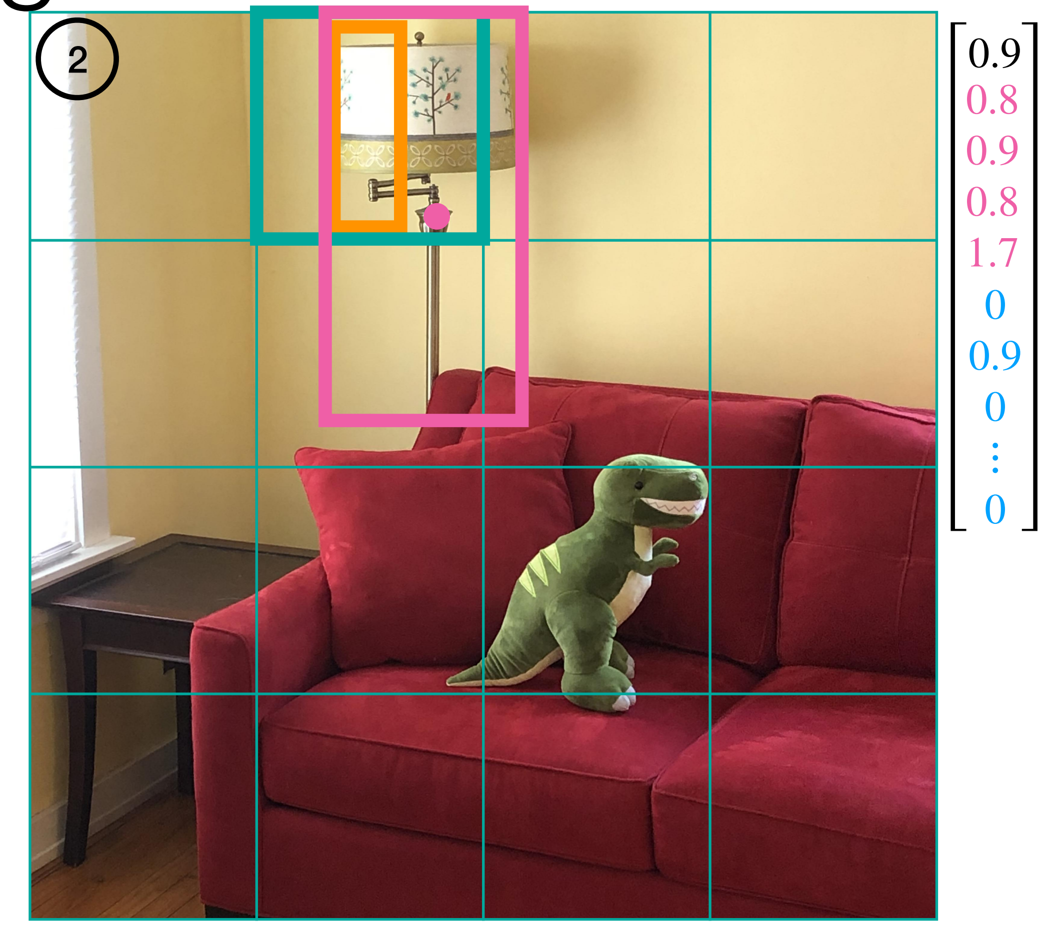
Overlapping Test Set Predictions

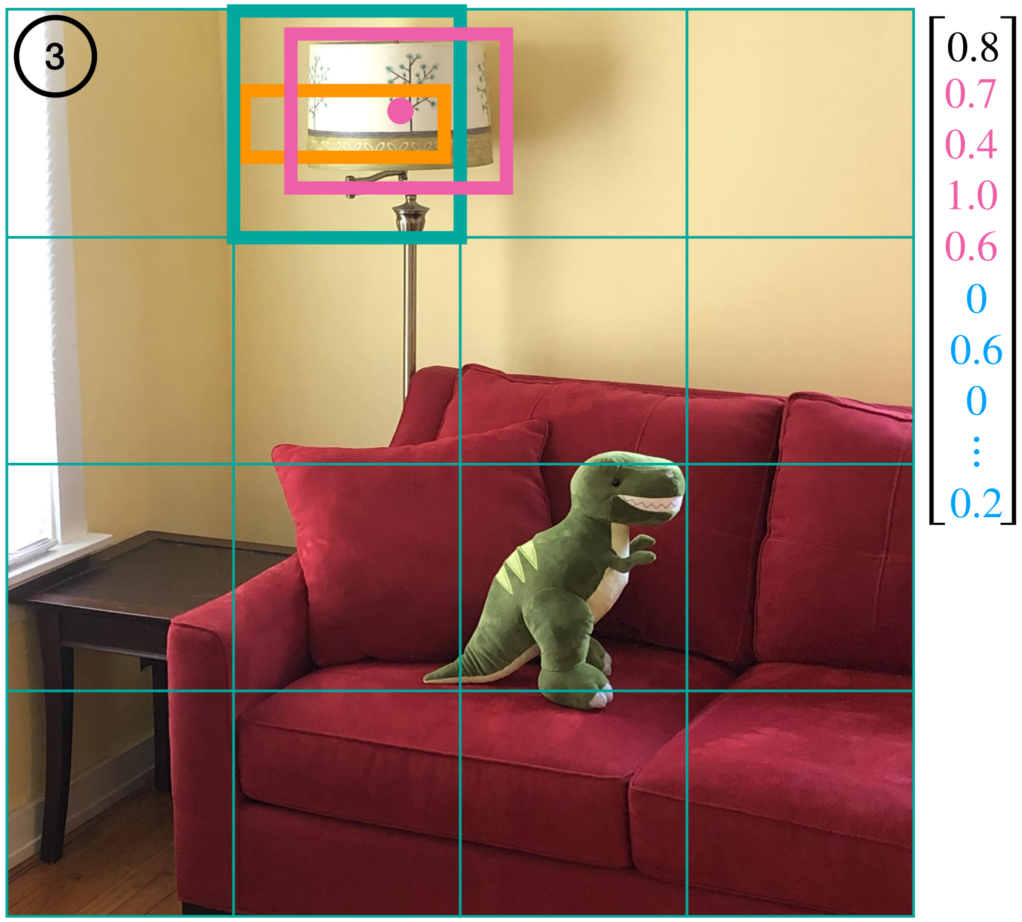


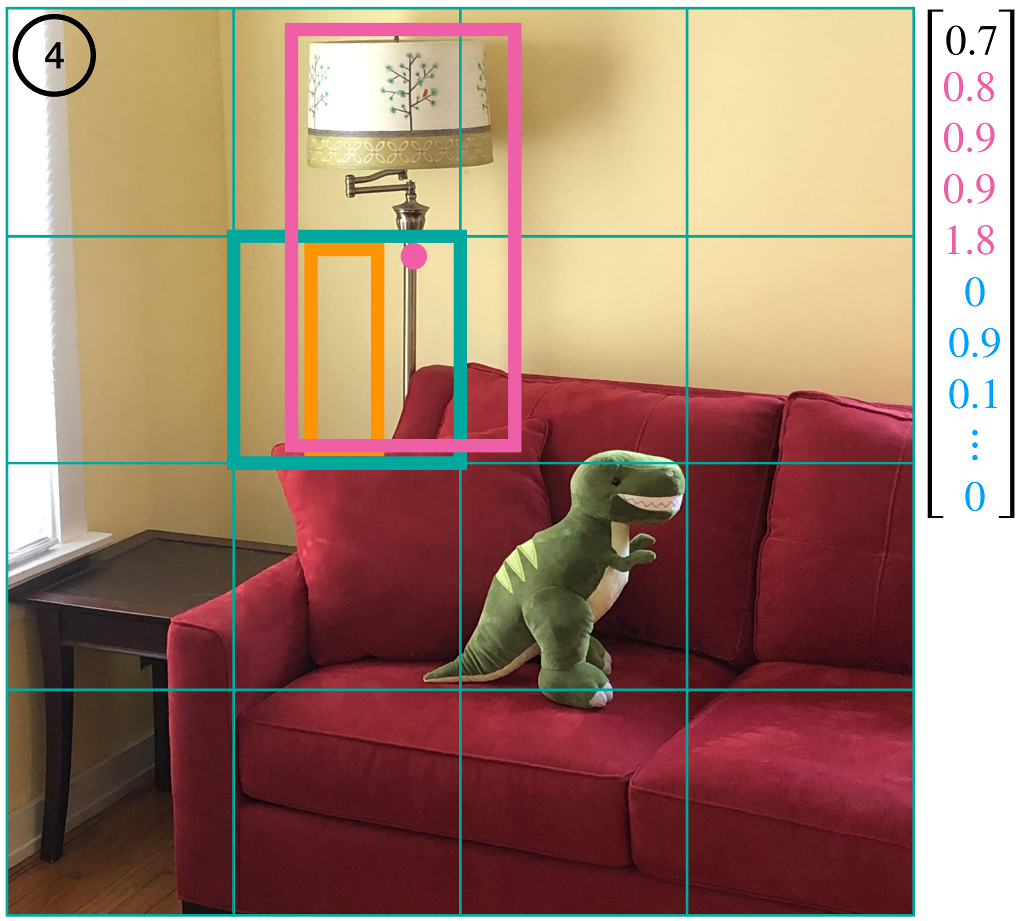
- When we make test set predictions, the same object may be identified for multiple cells and anchor boxes.
- To resolve:
  - Keep boxes where the probability that there is an object of a given class is high
  - Eliminate boxes that overlap a lot

Overlapping Test Set Predictions









- Probability that there is an object that is a lamp:

$$P(\textbf{object that is a lamp}) = P(\textbf{object}) \times P(\textbf{lamp} \mid \textbf{object})$$

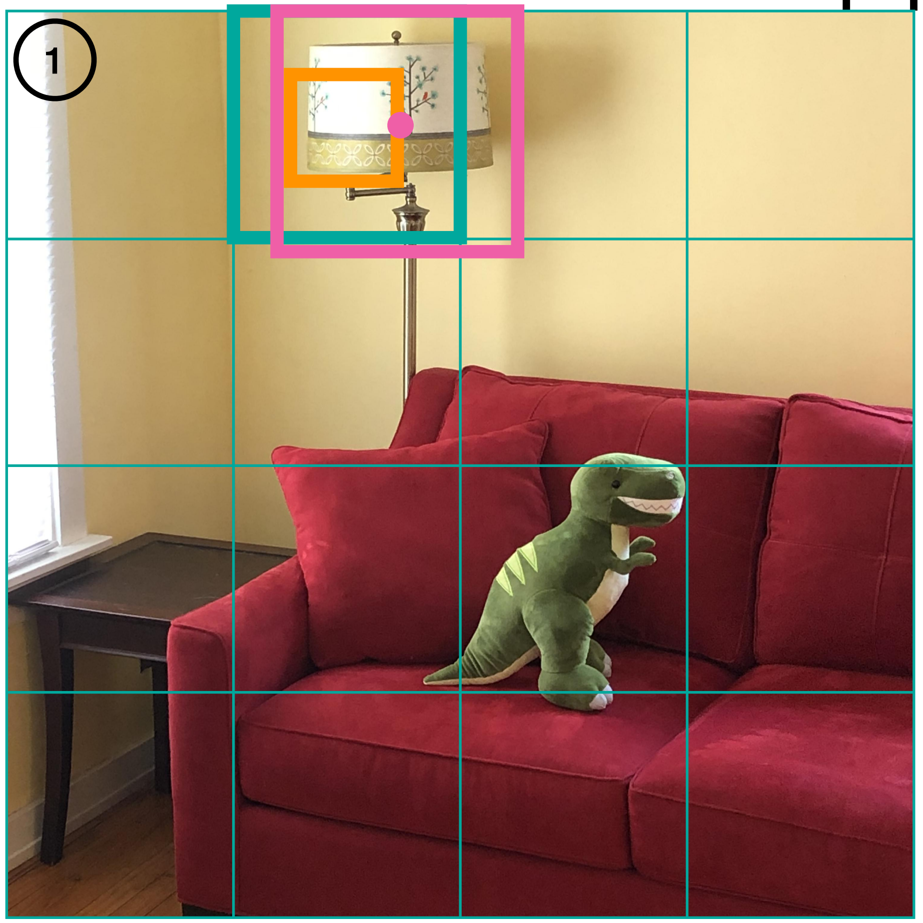
1 $0.9 \times 0.7 = 0.63$

2 $0.9 \times 0.9 = 0.81$

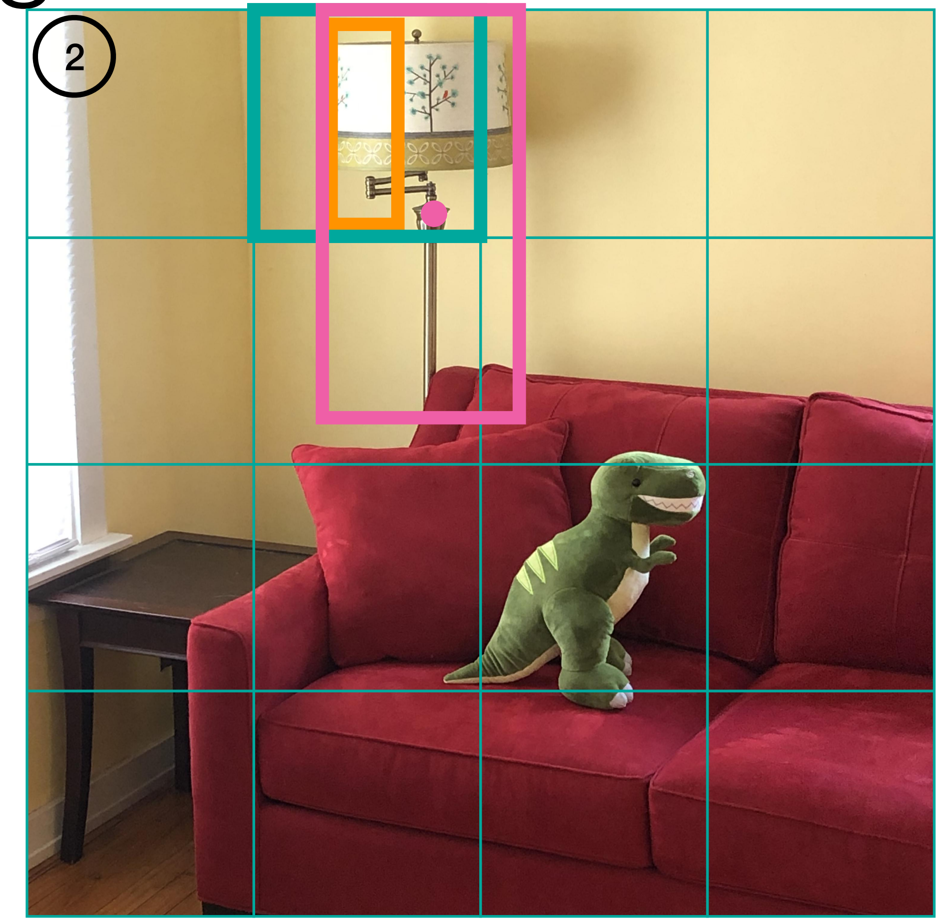
3 $0.8 \times 0.6 = 0.48$

4 $0.7 \times 0.9 = 0.63$

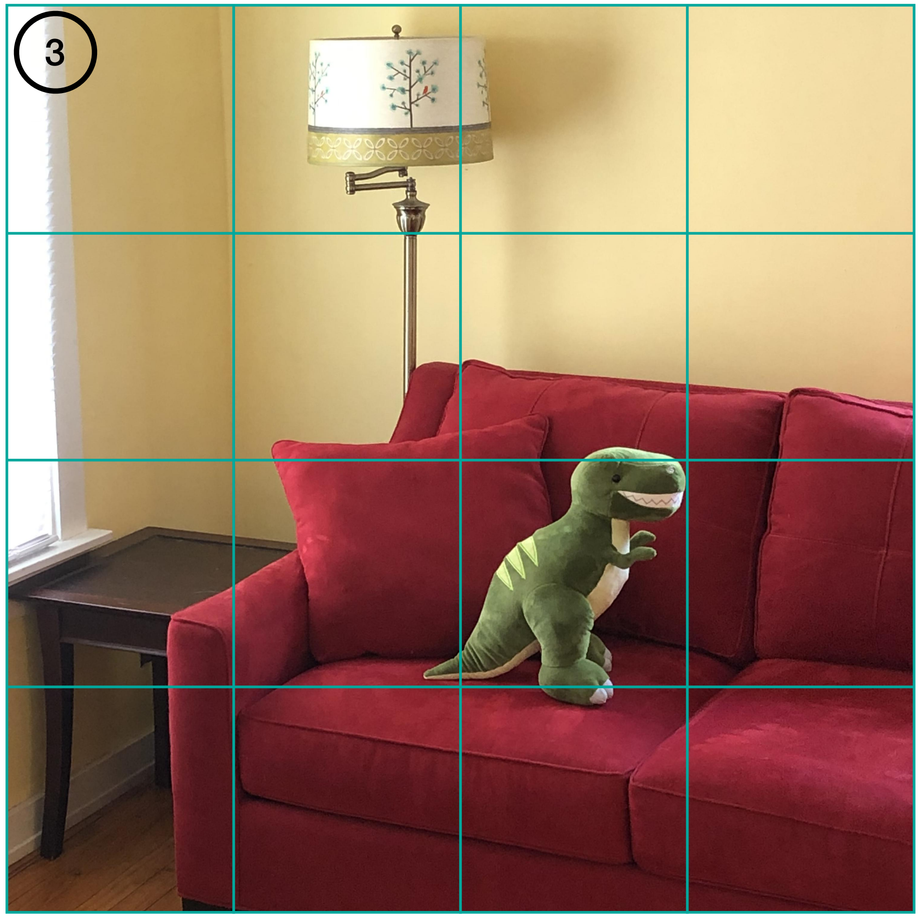
Overlapping Test Set Predictions



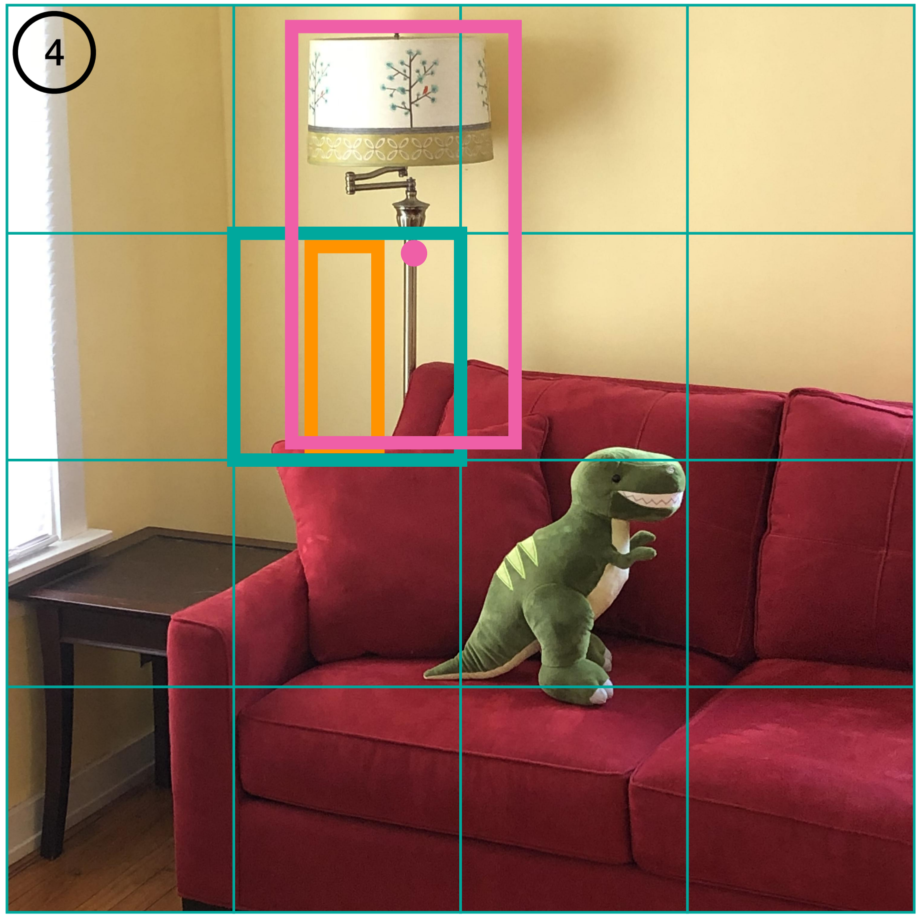
$$\begin{bmatrix} 0.9 \\ 0.7 \\ 0.5 \\ 1.1 \\ 1.1 \\ 0 \\ 0.7 \\ 0.1 \\ \vdots \\ 0.1 \end{bmatrix}$$



$$\begin{bmatrix} 0.9 \\ 0.8 \\ 0.9 \\ 0.8 \\ 1.7 \\ 0 \\ 0.9 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 0.8 \\ 0.7 \\ 0.4 \\ 1.0 \\ 0.6 \\ 0 \\ 0.6 \\ 0 \\ \vdots \\ 0.2 \end{bmatrix}$$



$$\begin{bmatrix} 0.7 \\ 0.8 \\ 0.9 \\ 0.9 \\ 1.8 \\ 0 \\ 0.9 \\ 0.1 \\ \vdots \\ 0 \end{bmatrix}$$

- Probability that there is an object that is a lamp:

$P($**object that is a lamp**$)$
$= P($**object**$) \times P($**lamp | object**$)$

① $0.9 \times 0.7 = 0.63$

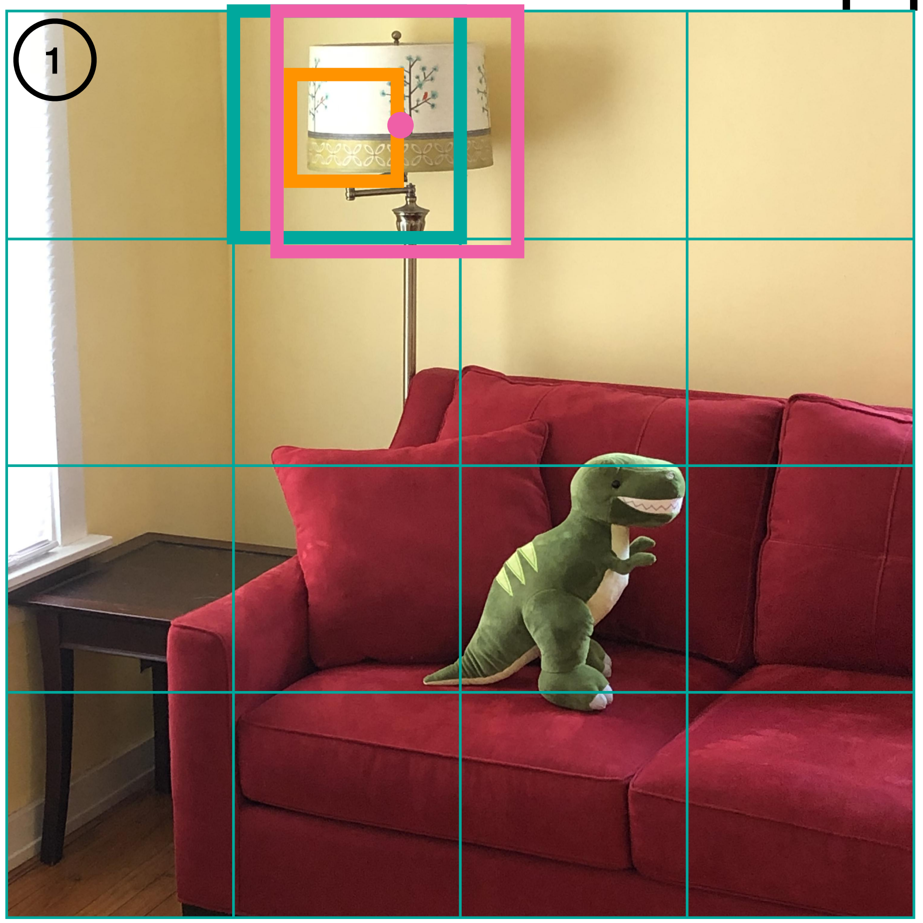
② $0.9 \times 0.9 = 0.81$

~~③ $0.8 \times 0.6 = 0.48$~~

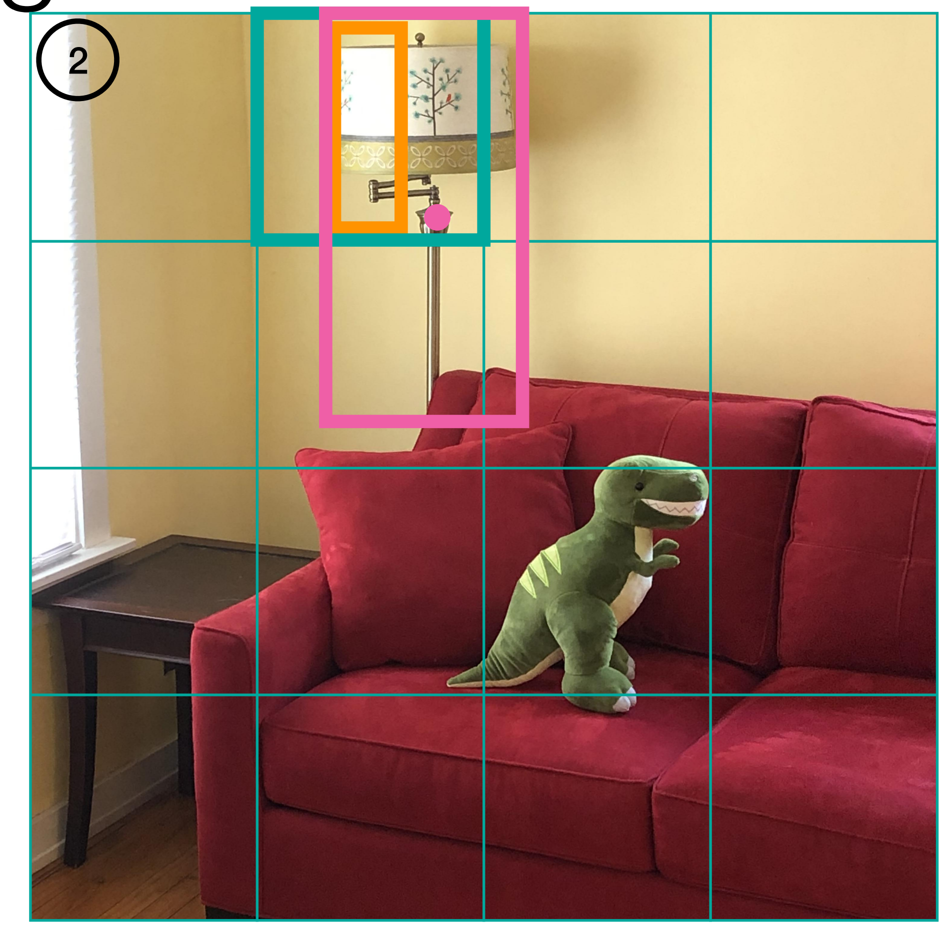
④ $0.7 \times 0.9 = 0.63$

- Since box ③ assigns probability less than 0.5 that there is a lamp, we will not consider it further

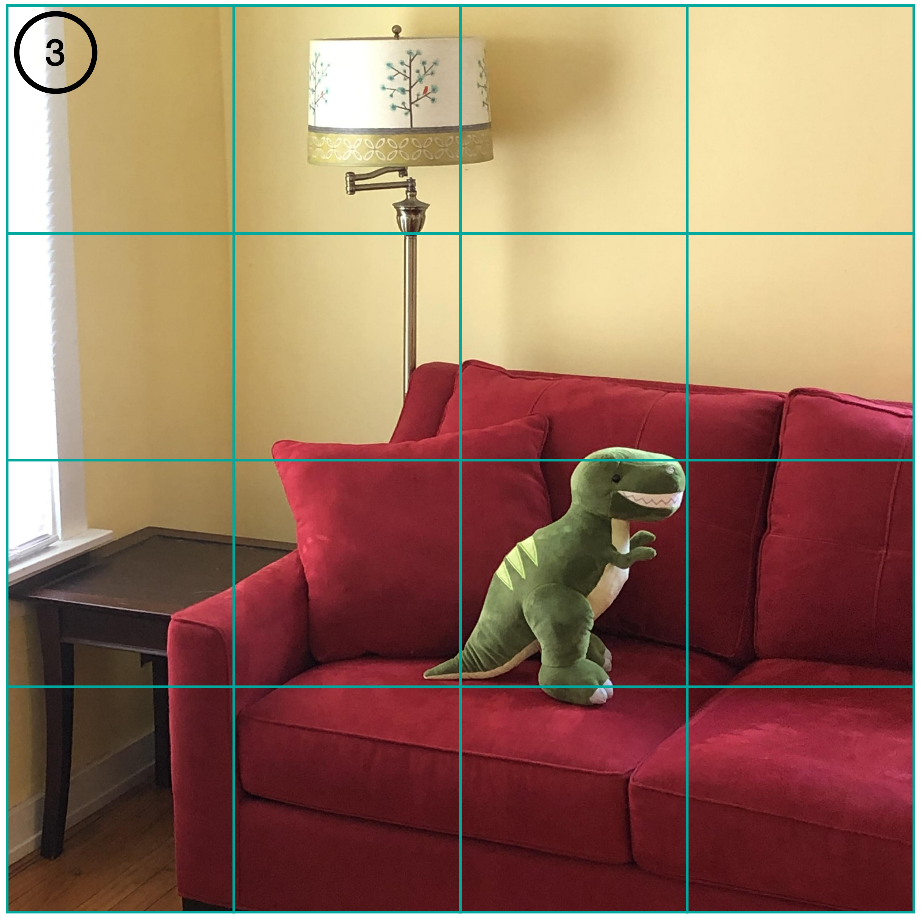
Overlapping Test Set Predictions



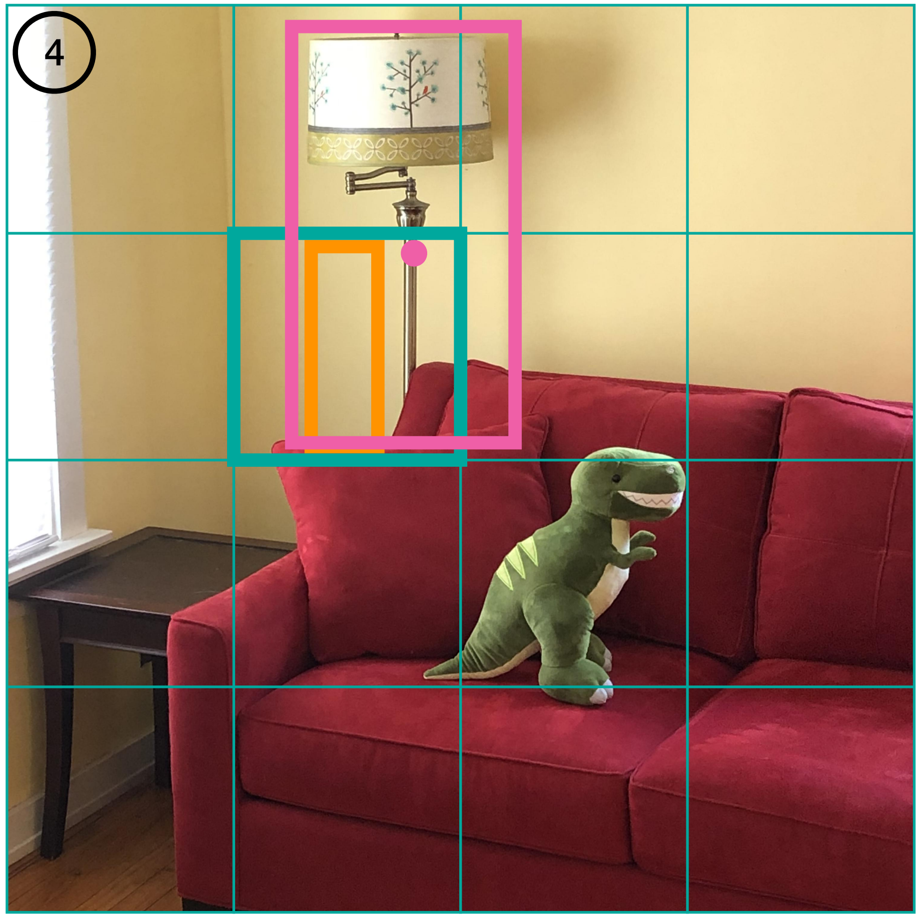
$$\begin{bmatrix} 0.9 \\ 0.7 \\ 0.5 \\ 1.1 \\ 1.1 \\ 0 \\ 0.7 \\ 0.1 \\ \vdots \\ 0.1 \end{bmatrix}$$



$$\begin{bmatrix} 0.9 \\ 0.8 \\ 0.9 \\ 0.8 \\ 1.7 \\ 0 \\ 0.9 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 0.8 \\ 0.7 \\ 0.4 \\ 1.0 \\ 0.6 \\ 0 \\ 0.6 \\ 0 \\ \vdots \\ 0.2 \end{bmatrix}$$



$$\begin{bmatrix} 0.7 \\ 0.8 \\ 0.9 \\ 0.9 \\ 1.8 \\ 0 \\ 0.9 \\ 0.1 \\ \vdots \\ 0 \end{bmatrix}$$

- Probability that there is an object that is a lamp:

$$P(\textbf{object that is a lamp}) = P(\textbf{object}) \times P(\textbf{lamp} \mid \textbf{object})$$

① $0.9 \times 0.7 = 0.63$

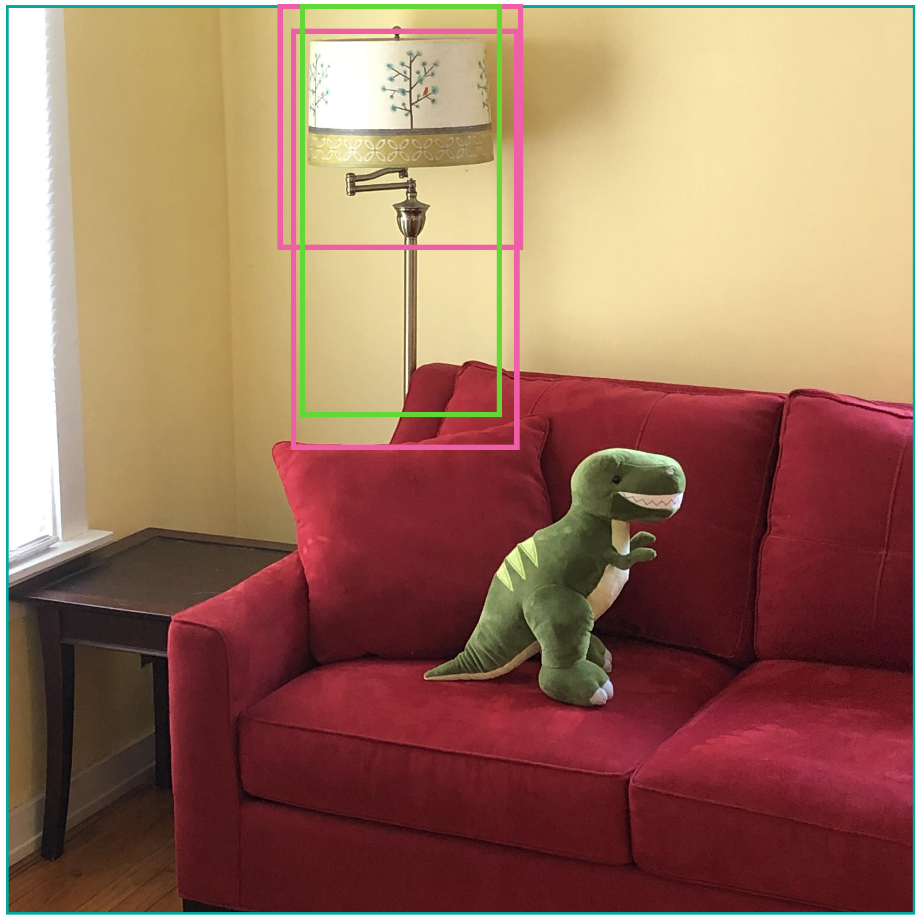
② $0.9 \times 0.9 = 0.81$

~~③ $0.8 \times 0.6 = 0.48$~~

④ $0.7 \times 0.9 = 0.63$

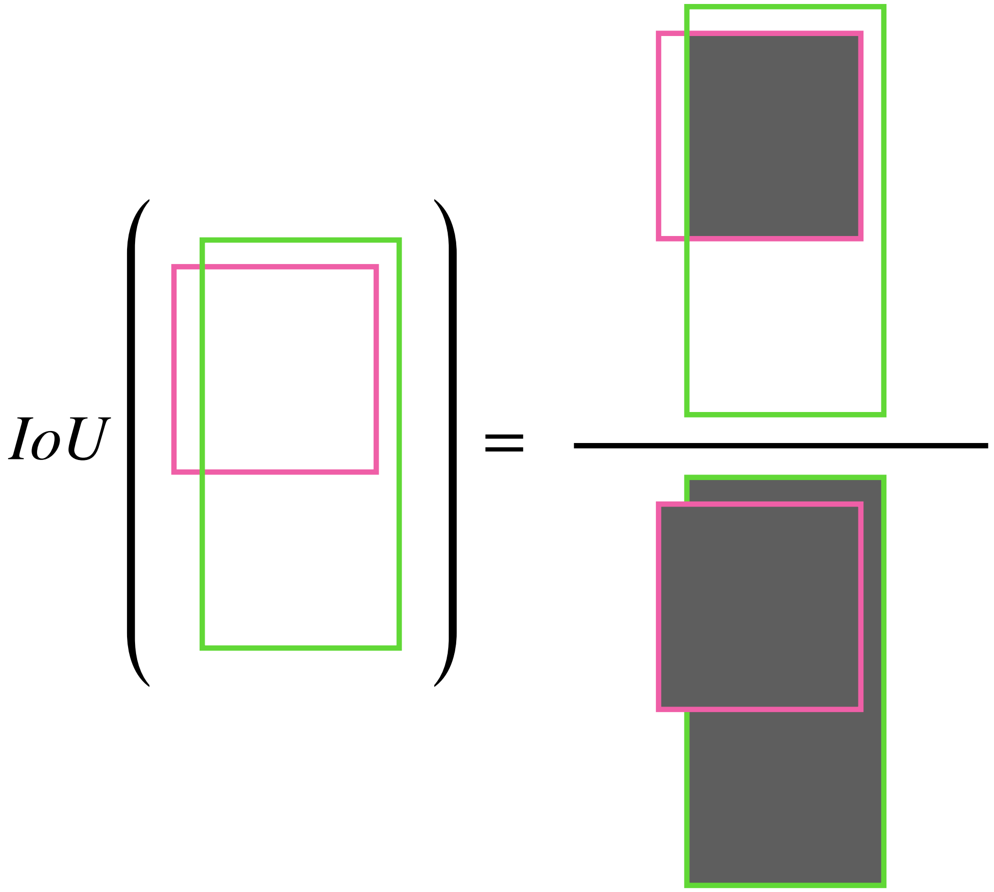
- Since box ③ assigns probability less than 0.5 that there is a lamp, we will not consider it further
- Since box ② assigns highest probability that there is a lamp, we'll definitely keep it.

Overlapping Test Set Predictions



For each of the remaining boxes (① and ④):

1. Calculate the **Intersection over Union (IoU)** with box ②



2. If IoU ≥ 0.5, eliminate the box

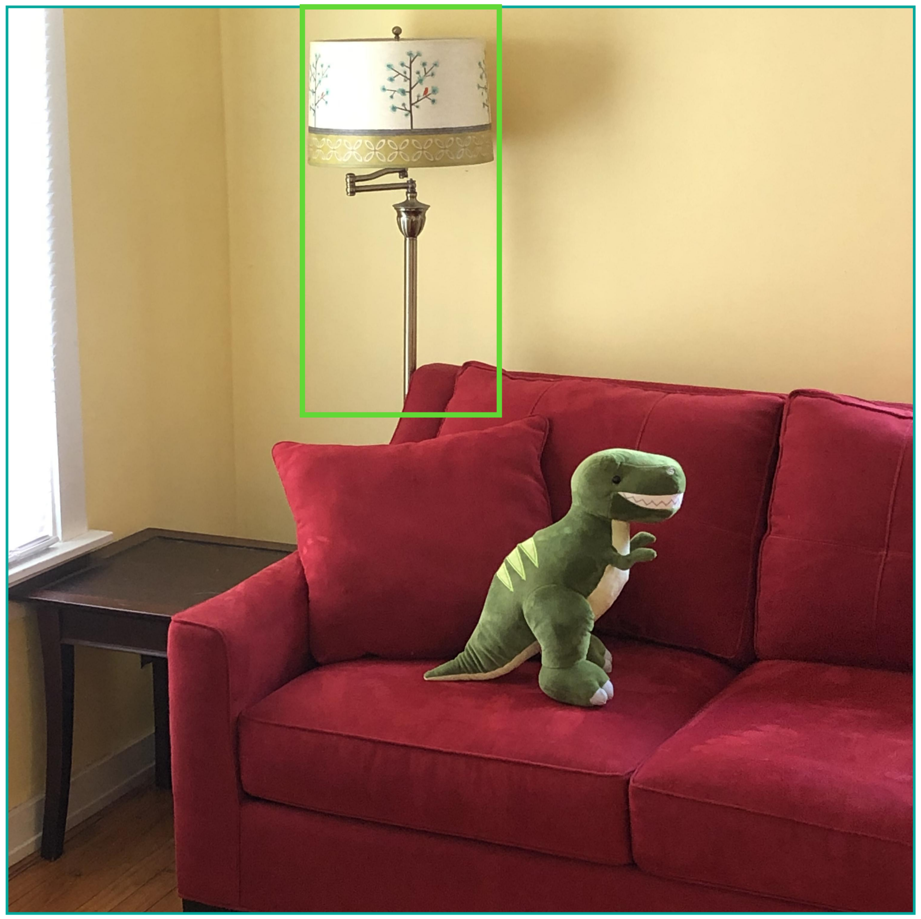
① $0.9 \times 0.7 = 0.63$

② $0.9 \times 0.9 = 0.81$

~~③ $0.8 \times 0.6 = 0.48$~~

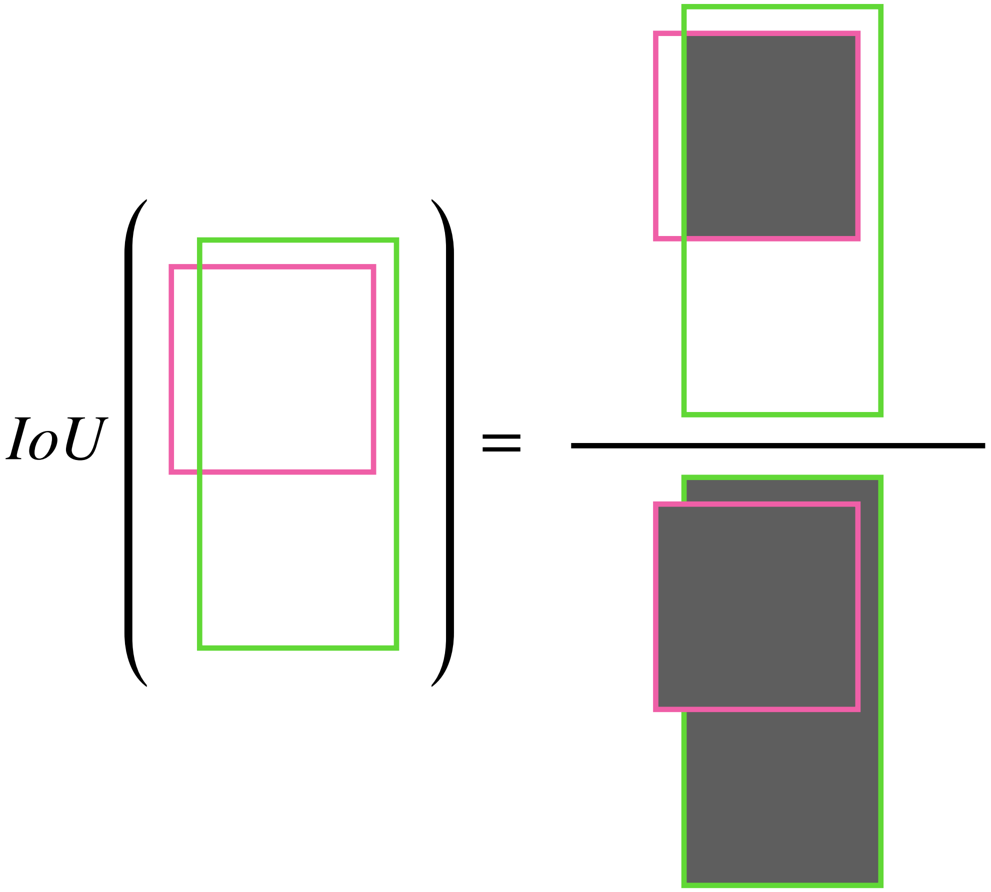
④ $0.7 \times 0.9 = 0.63$

Overlapping Test Set Predictions



For each of the remaining boxes (① and ④):

1. Calculate the **Intersection over Union (IoU)** with box ②



2. If IoU ≥ 0.5, eliminate the box

~~① $0.9 \times 0.7 = 0.63$~~

② $0.9 \times 0.9 = 0.81$

~~③ $0.8 \times 0.6 = 0.48$~~

~~④ $0.7 \times 0.9 = 0.63$~~