More Imaging Tasks

- Classification
- Objects
  - Object Localization
  - Object Detection
- Keypoint detection
  - Human pose detection
  - Facial recognition
- Segmentation
  - Instance/Scene segmentation
  - Medical imaging

Object Localization

- Where is an object in the image?

Object Localization

- Where is an object in the image?
  - Input: image of a cat
  - Label: location of cat

Object Localization

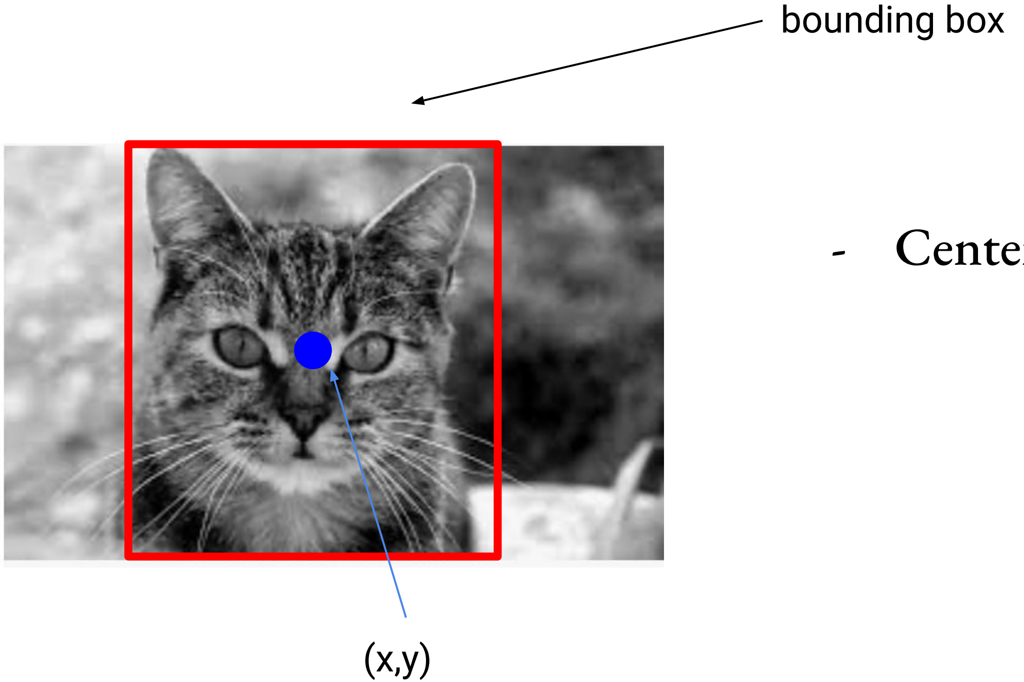
- Where is an object in the image?
  - Input: image of a cat
  - Label: bounding box coordinates

bounding box

- Center/Dimensions: (x, y, w, h)

Object Localization

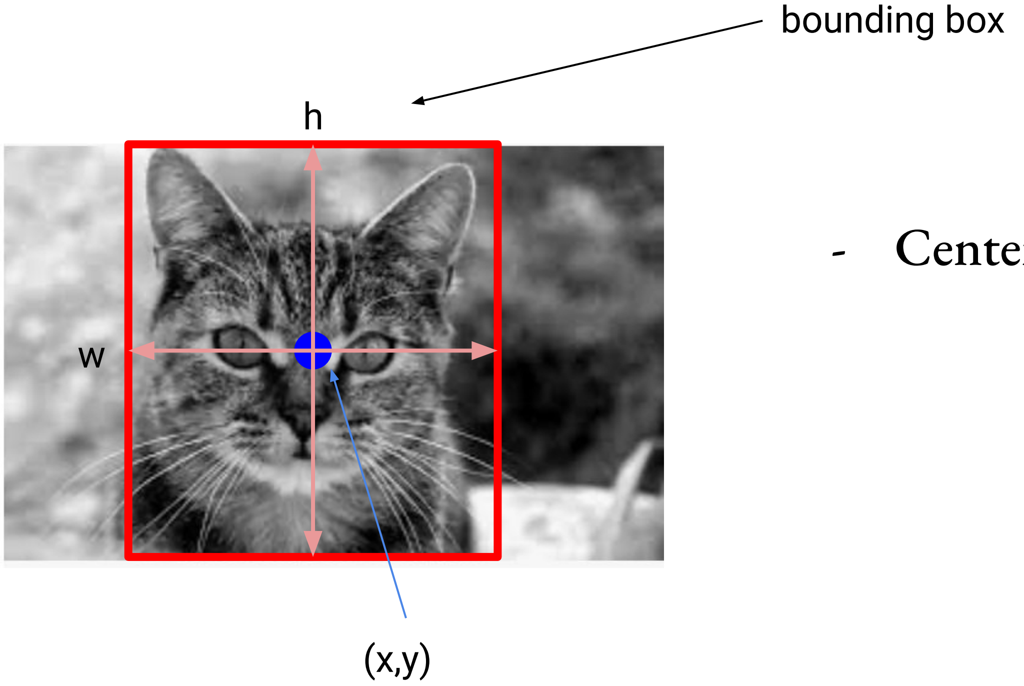
- Where is an object in the image?
  - Input: image of a cat
  - Label: bounding box coordinates



- Center/Dimensions: (x, y, w, h)

Object Localization

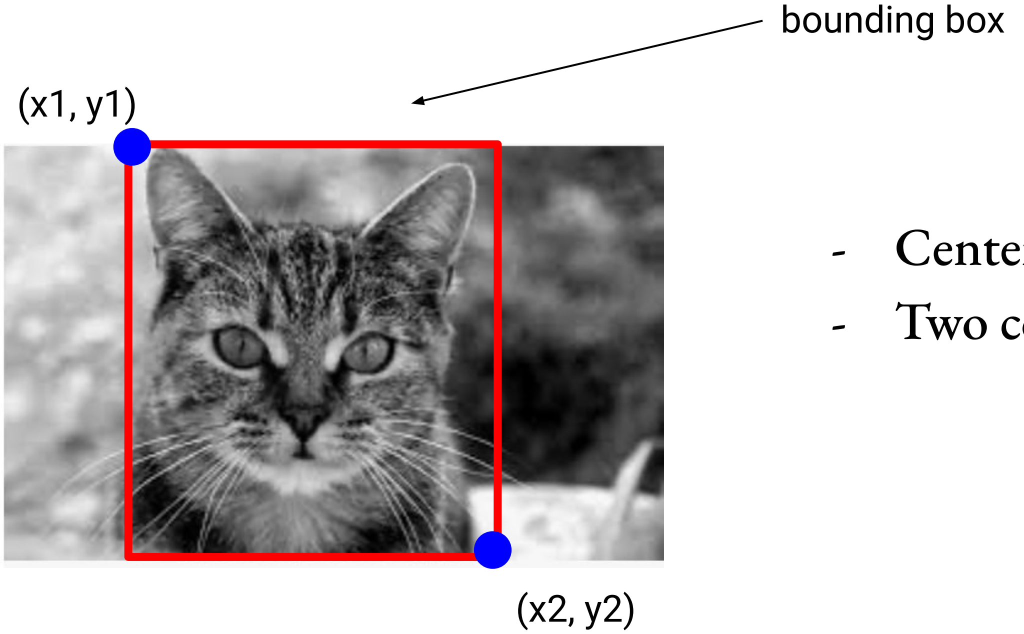
- Where is an object in the image?
  - Input: image of a cat
  - Label: bounding box coordinates



- Center/Dimensions: (x, y, w, h)

Object Localization

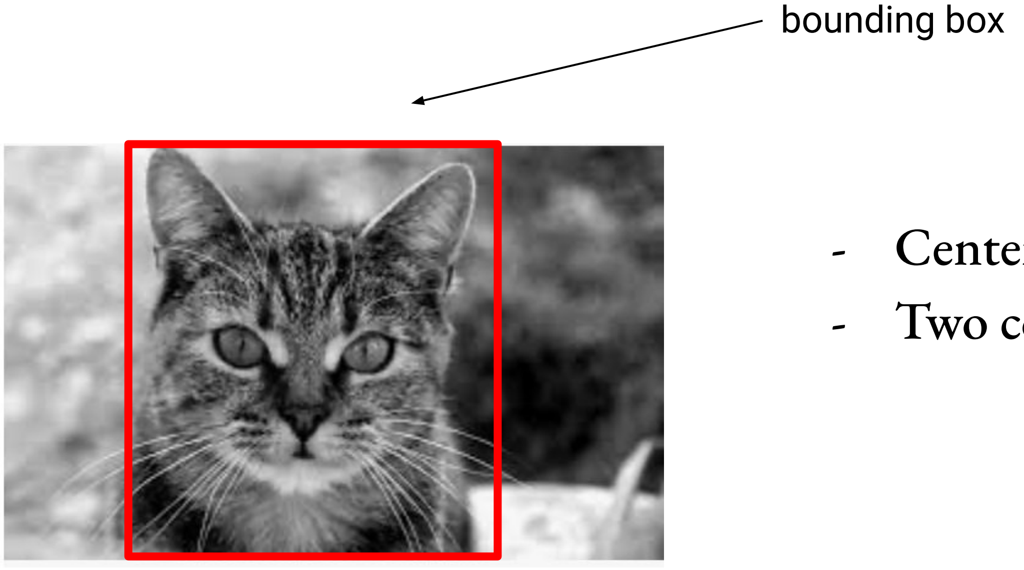
- Where is an object in the image?
    - Input: image of a cat
    - Label: bounding box coordinates



- Center/Dimensions: (x, y, w, h)
- Two corners: (x1, y1, x2, y2)

Object Localization

- Where is an object in the image?
  - Input: image of a cat
  - Label: bounding box coordinates



- Center/Dimensions: (x, y, w, h)
- Two corners: (x1, y1, x2, y2)

Object Localization

- Where is an object in the image?
  - Input: image of a cat
  - Label: bounding box coordinates

bounding box

regression

- Center/Dimensions: (x, y, w, h)
- Two corners: (x1, y1, x2, y2)

Classification + Localization

- Simultaneously answer:
  - Does this photo contain a cat?
  - If so where is the cat?

cat

Classification + Localization

- Simultaneously answer:
  - Does this photo contain a cat?
  - If so where is the cat?

cat

(1, 150, 175, 50, 70) <-> (class, x, y, w, h)

Classification + Localization

- Simultaneously answer:
  - Does this photo contain a cat?
  - If so where is the cat?

cat

(1, 150, 175, 50, 70) <-> (class, x, y, w, h)

Classification: BCELoss(predicted class, true class)

Localization: MSELoss(predicted coords, true coords)

Classification + Localization

- Simultaneously answer:
  - Does this photo contain a cat?
  - If so where is the cat?

cat

(1, 150, 175, 50, 70) <-> (class, x, y, w, h)

Class label

$$Loss = BCELoss + Y * MSELoss$$

Classification + Localization

- Simultaneously answer:
  - Does this photo contain a cat?
  - If so where is the cat?

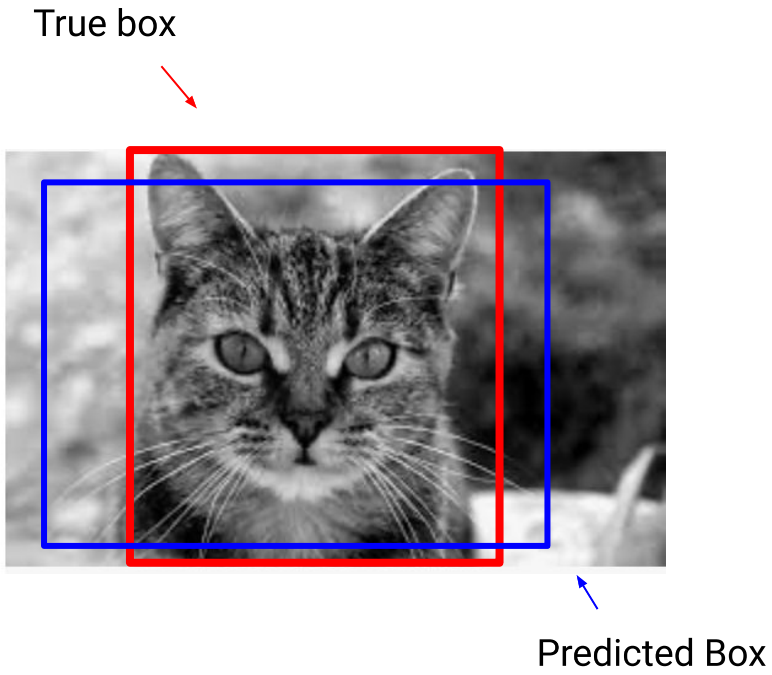
cat

(1, 150, 175, 50, 70) <-> (class, x, y, w, h)

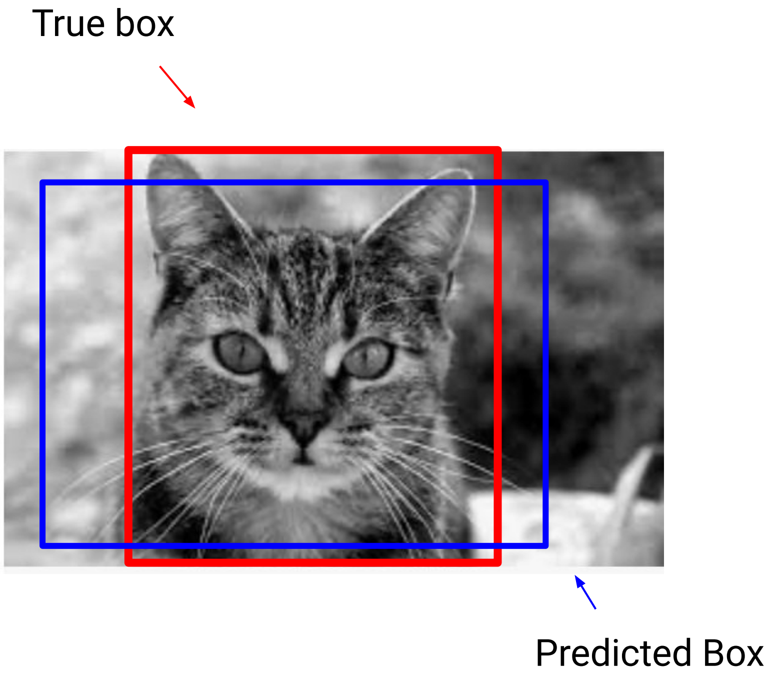
Class label

$$Loss = BCELoss + \gamma(Y * MSELoss)$$

Intersection over Union for Localization



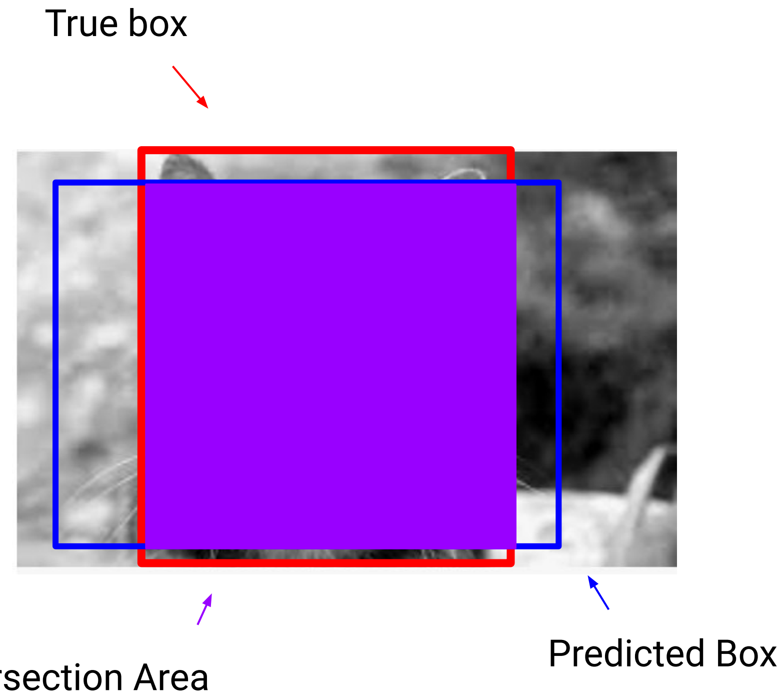
Intersection over Union for Localization



$$\text{IoU: } \frac{Intersection\ Area}{Union\ Area}$$

- Score from 0 to 1
- Higher is better

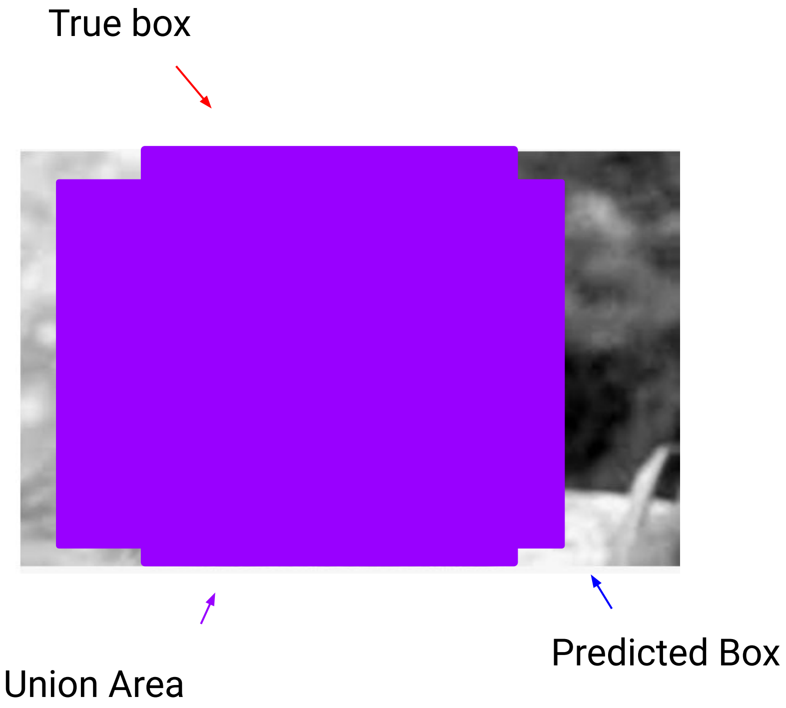
Intersection over Union for Localization



IoU: $$\frac{Intersection\ Area}{Union\ Area}$$

- Score from 0 to 1
- Higher is better

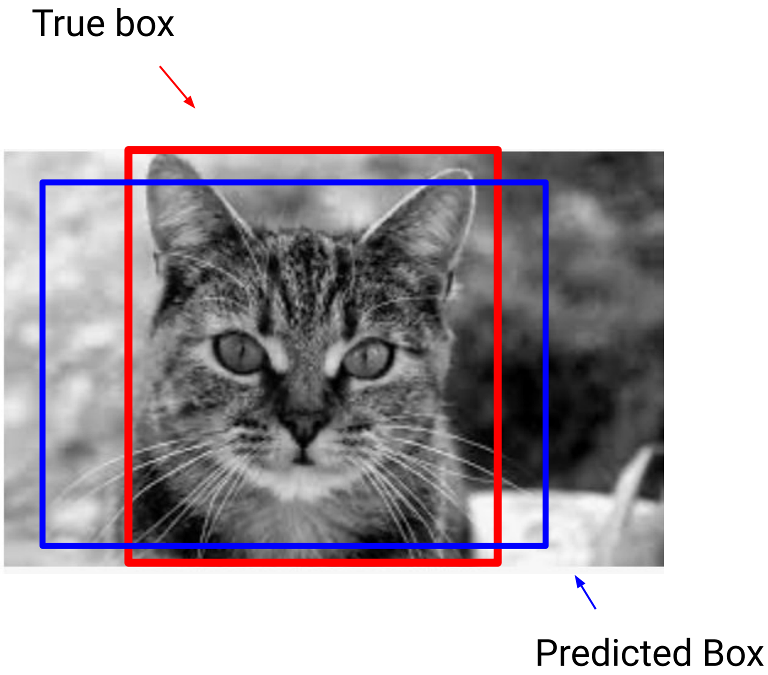
Intersection over Union for Localization



IoU: $\dfrac{Intersection\ Area}{Union\ Area}$

- Score from 0 to 1
- Higher is better

Intersection over Union for Localization



IoU: $\frac{Intersection\ Area}{Union\ Area}$

- Score from 0 to 1
- Higher is better
- Areas can be computed from box coordinates
- Union given by

Box1 Area + Box2 Area - Intersection Area

Multi-Label Classification + Localization

- What about multiple classes in the same photo?

Multi-Label Classification + Localization

- What about multiple classes in the same photo?
- k classes -> k binary classification tasks + localization

Classes
- Pedestrian
- Stop sign
- Car
- Traffic Light

Multi-Label Classification + Localization

- What about multiple classes in the same photo?
- k classes -> k binary classification tasks + localization

Classes
- Pedestrian
- Stop sign
- Car
- Traffic Light

Label

$$(0, 0, 0, 0, 0)$$
$$(1, 200, 300, 10, 10)$$
$$(1, 400, 100, 300, 100)$$
$$(0, 0, 0, 0, 0)$$

Multi-Label Classification + Localization

- What about multiple classes in the same photo?
- k classes -> k binary classification tasks + localization

Classes

- Pedestrian
- Stop sign
- Car
- Traffic Light

Label

classes

$$(0, 0, 0, 0, 0)$$
$$(1, 200, 300, 10, 10)$$
$$(1, 400, 100, 300, 100)$$
$$(0, 0, 0, 0, 0)$$

Multi-Label Classification + Localization

- What about multiple classes in the same photo?
- k classes -> k binary classification tasks + localization

Classes

- Pedestrian
- Stop sign
- Car
- Traffic Light

Label

Bounding box coordinates

classes

$$(0, 0, 0, 0, 0)$$
$$(1, 200, 300, 10, 10)$$
$$(1, 400, 100, 300, 100)$$
$$(0, 0, 0, 0, 0)$$

Object Detection

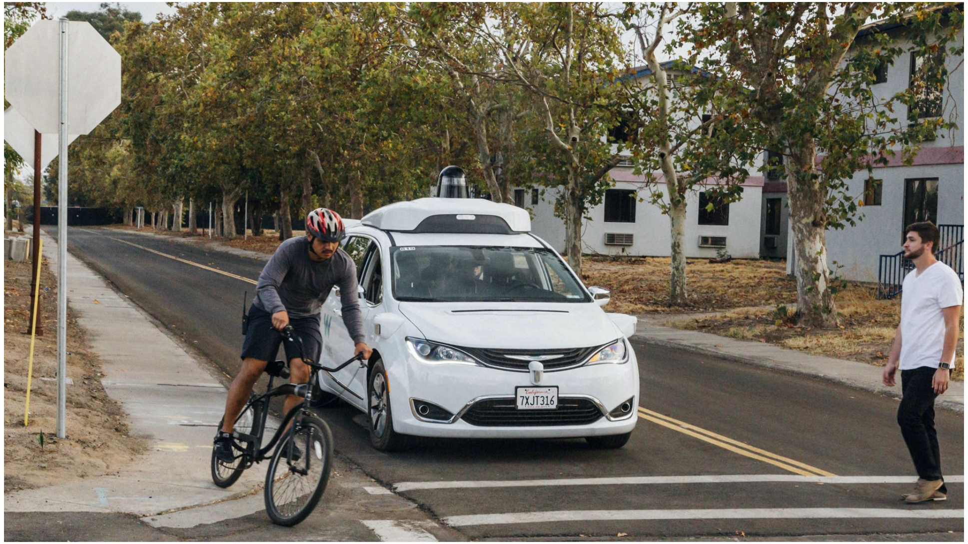
- Classify and locate all objects in an image

Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
  1. Propose region for object classification + localization
  2. Perform object classification + localization on this region

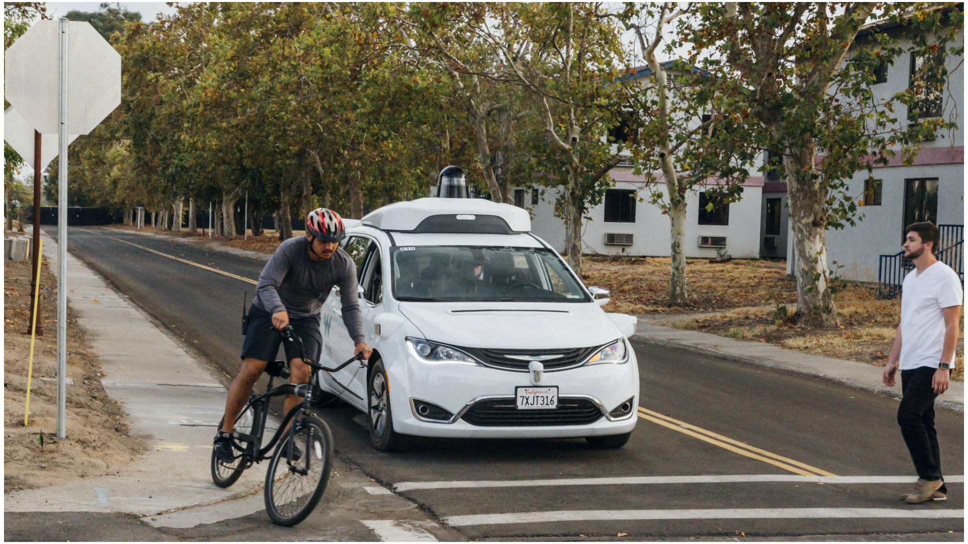
Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
  1. Propose region for object classification + localization
  2. Perform object classification + localization on this region



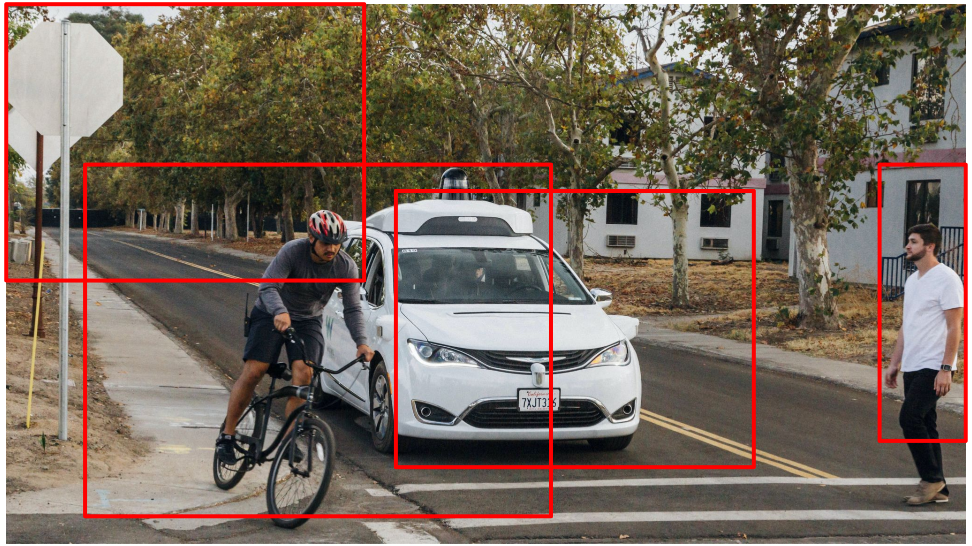
Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
  1. Propose region for object classification + localization
  2. Perform object classification + localization on this region



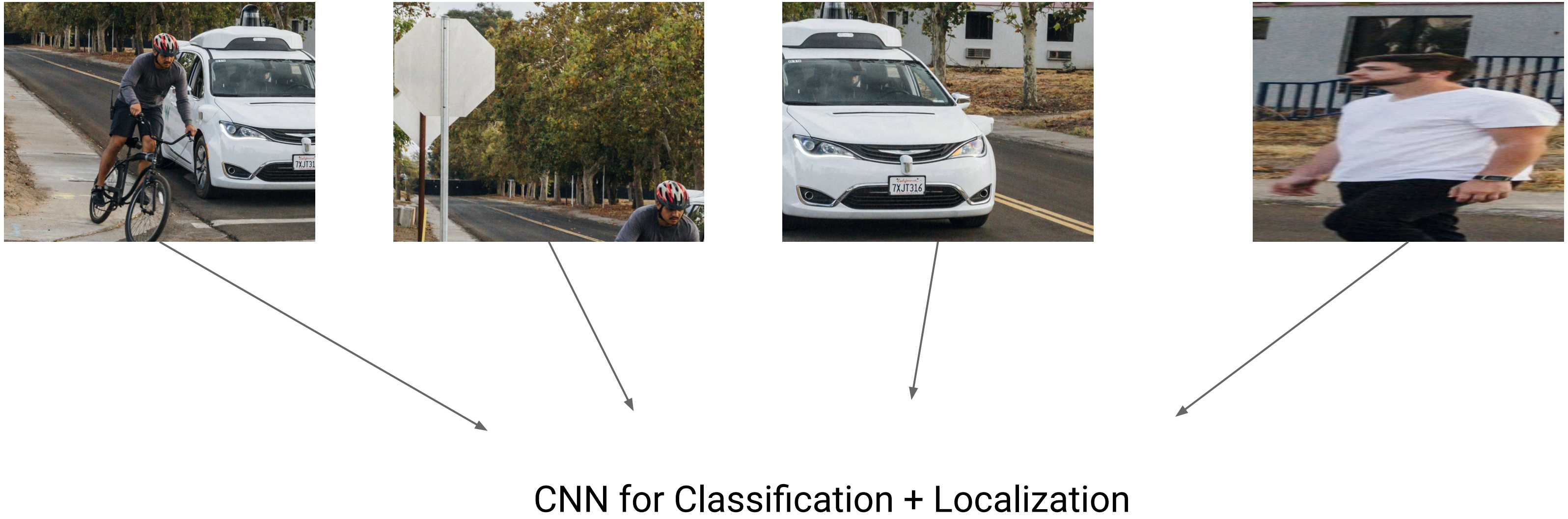
Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
  1. Propose region for object classification + localization
  2. Perform object classification + localization on this region



Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
  1. Propose region for object classification + localization
  2. Perform object classification + localization on this region

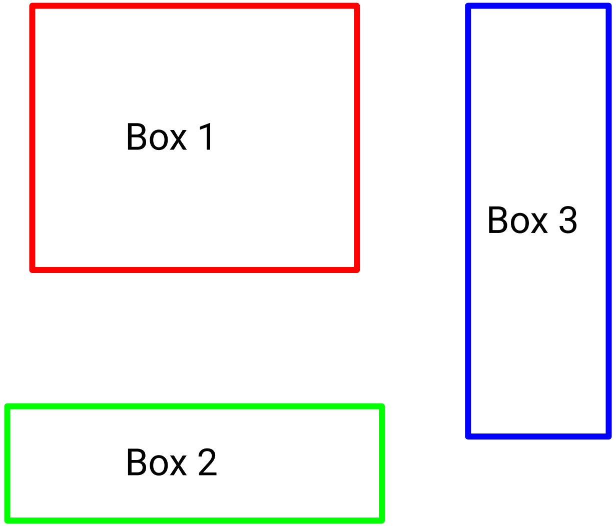


Object Detection

- Classify and locate all objects in an image
- Region Proposals (RCNNs)
- You Only Look Once (YOLO)
  1. Sort each bounding box label into various "anchor boxes"
  2. Divide Image into a grid
  3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)

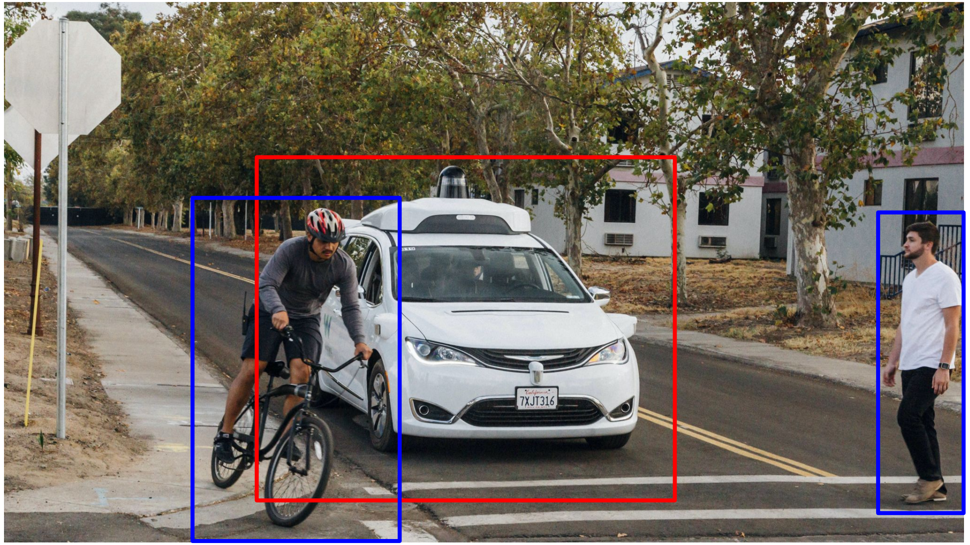
YOLO

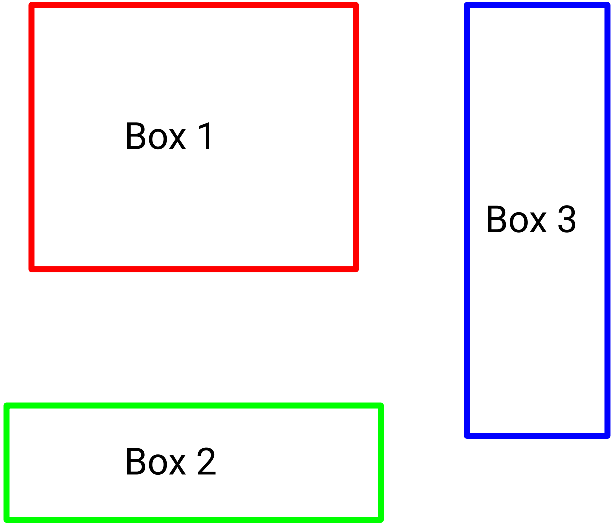
1. Sort each bounding box label into various “anchor boxes”
2. Divide Image into a grid
3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)



YOLO

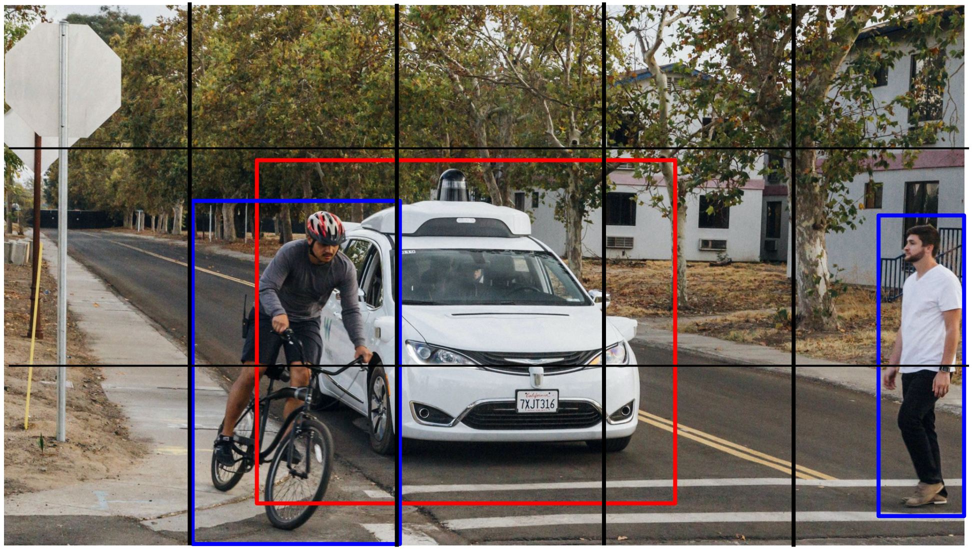
1. Sort each bounding box label into various “anchor boxes”
2. Divide Image into a grid
3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)

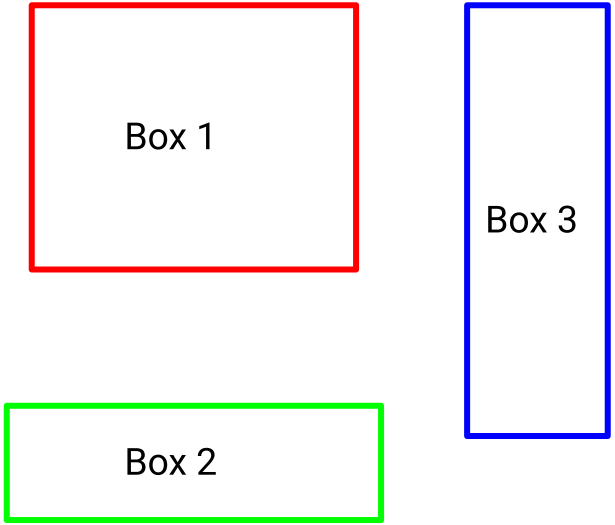




YOLO

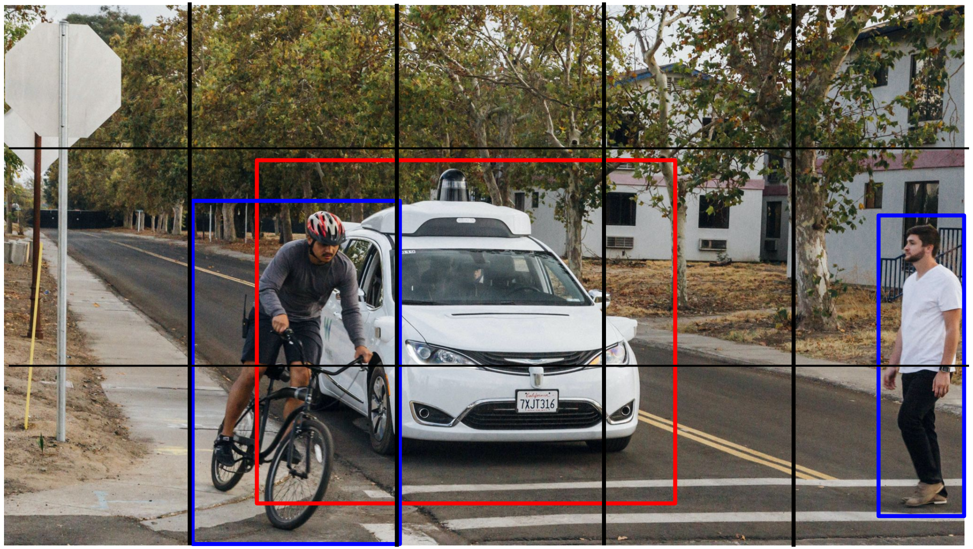
1. Sort each bounding box label into various “anchor boxes”
2. Divide Image into a grid
3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)

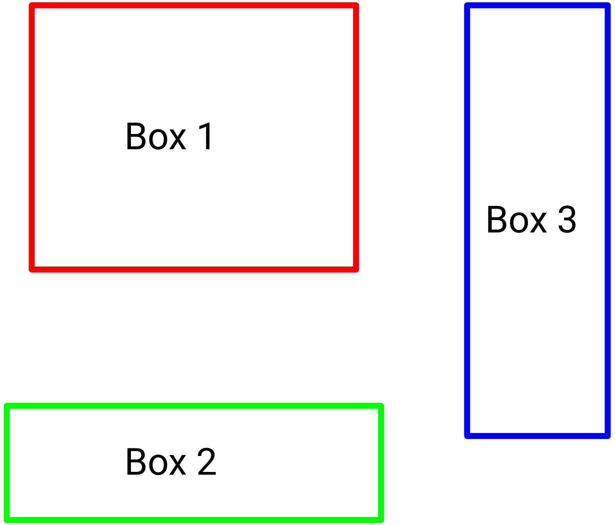




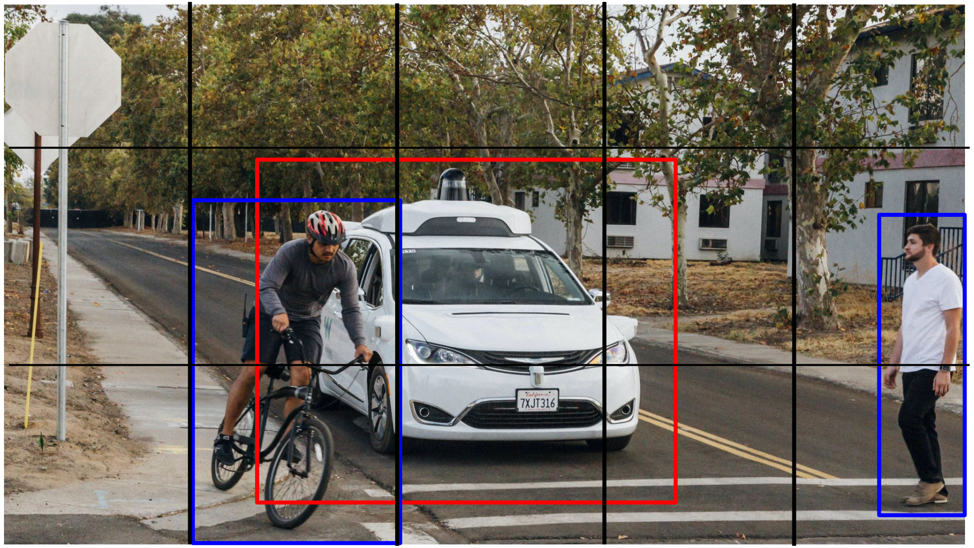
YOLO

1. Sort each bounding box label into various "anchor boxes"
2. Divide Image into a grid
3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)





YOLO



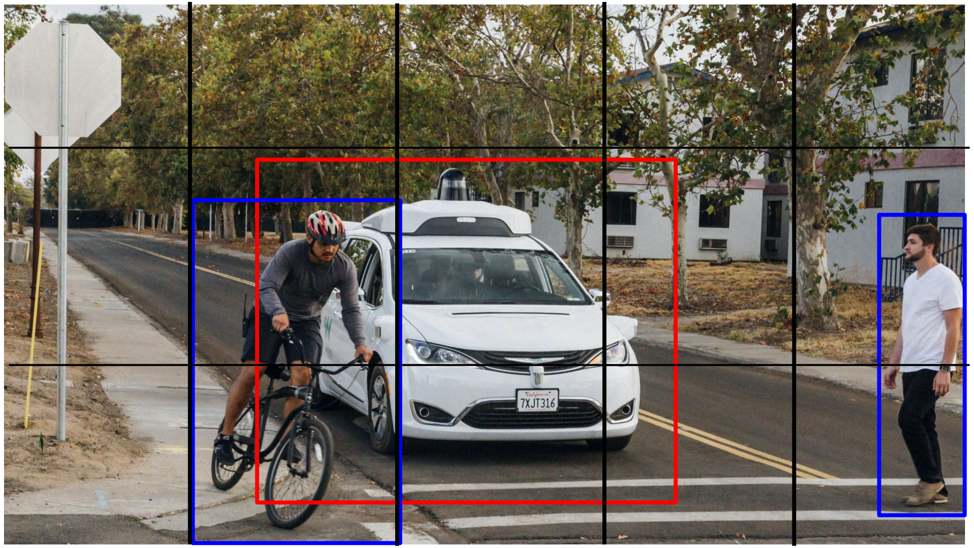
$$\begin{bmatrix} p_c^1 \\ c_1^1 \\ c_2^1 \\ \vdots \\ c_k^1 \\ x^1 \\ y^1 \\ w^1 \\ h^1 \\ p_c^2 \\ c_1^2 \\ c_2^2 \\ \vdots \\ c_k^2 \\ x^2 \\ y^2 \\ w^2 \\ h^2 \\ p_c^3 \\ c_1^3 \\ c_2^3 \\ \vdots \\ c_k^3 \\ x^3 \\ y^3 \\ w^3 \\ h^3 \end{bmatrix}$$

Probability an object appears matching anchor box 1

Class predictions for this box

Bounding box coords for this box

YOLO



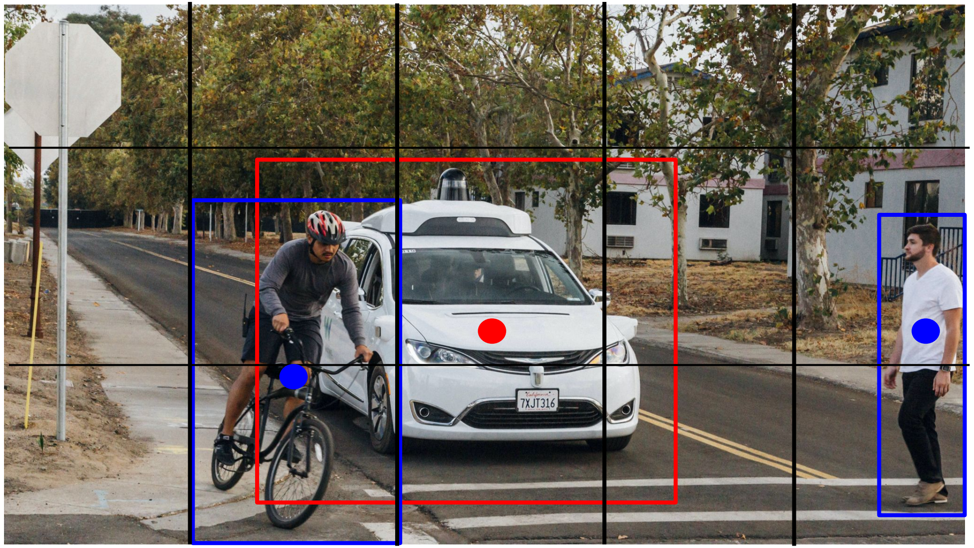
$$\begin{bmatrix} p_c^1 \\ c_1^1 \\ c_2^1 \\ \vdots \\ c_k^1 \\ x^1 \\ y^1 \\ w^1 \\ h^1 \\ p_c^2 \\ c_1^2 \\ c_2^2 \\ \vdots \\ c_k^2 \\ x^2 \\ y^2 \\ w^2 \\ h^2 \\ p_c^3 \\ c_1^3 \\ c_2^3 \\ \vdots \\ c_k^3 \\ x^3 \\ y^3 \\ w^3 \\ h^3 \end{bmatrix}$$

Probability an object appears matching anchor box 2

Class predictions for this box

Bounding box coords for this box

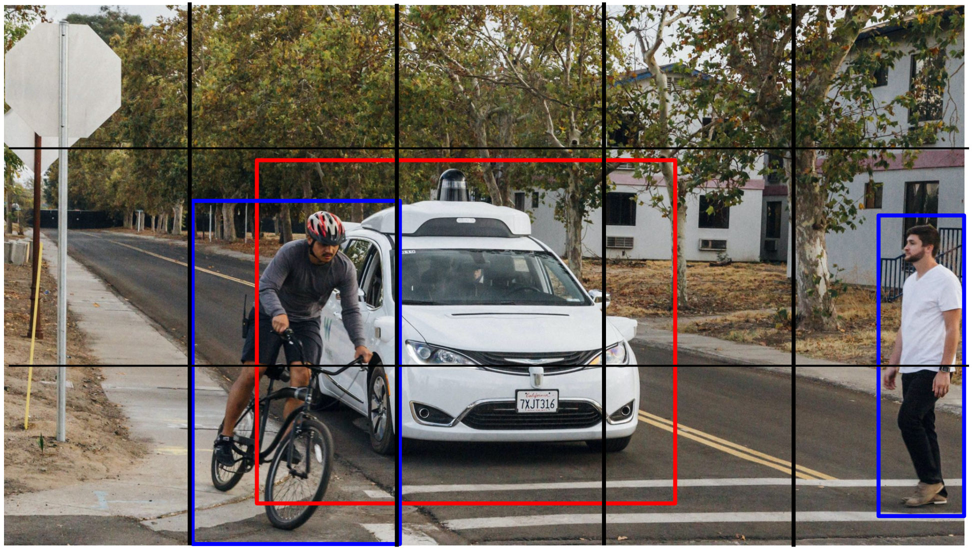
YOLO

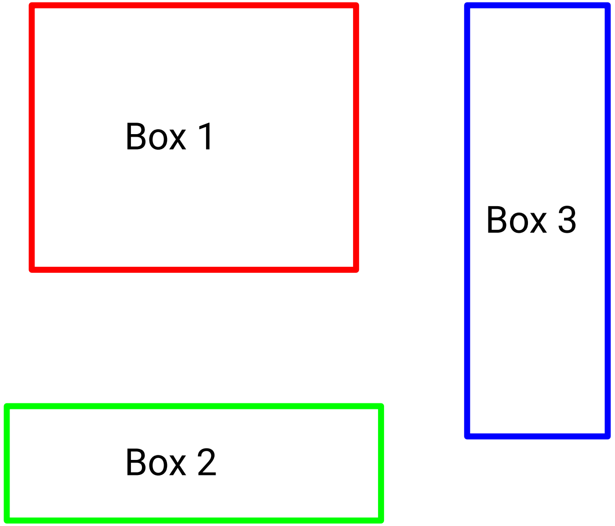


$$\begin{bmatrix}
(p_c^1)^1 & \dots & (p_c^1)^N \\
(c_1^1)^1 & & (c_1^1)^N \\
c_2^1)^1 & & (c_2^1)^N \\
\vdots & & \vdots \\
(c_k^1)^1 & & (c_k^1)^N \\
(x^1)^1 & & (x^1)^N \\
(y^1)^1 & & (y^1)^N \\
(w^1)^1 & & (w^1)^N \\
(h^1)^1 & & (h^1)^N \\
(p_c^2)^1 & & (p_c^2)^N \\
(c_1^2)^1 & & (c_1^2)^N \\
(c_2^2)^1 & & (c_1^2)^N \\
\vdots & & \vdots \\
(c_k^2)^1 & & (c_k^2)^N \\
(x^2)^1 & & (x^2)^N \\
(y^2)^1 & & (y^2)^N \\
(w^2)^1 & & (w^2)^N \\
(h^2)^1 & & (h^2)^N \\
(p_c^3)^1 & & (p_c^3)^N \\
(c_1^3)^1 & & (c_1^3)^N \\
(c_2^3)^1 & & (c_2^3)^N \\
\vdots & & \vdots \\
(c_k^3)^1 & & (c_k^3)^N \\
(x^3)^1 & & (x^3)^N \\
(y^3)^1 & & (y^3)^N \\
(w^3)^1 & & (w^3)^N \\
(h^3)^1 & \dots & (h^3)^N
\end{bmatrix}$$

YOLO

1. Sort each bounding box label into various “anchor boxes”
2. Divide Image into a grid
3. For each grid square predict whether the center of a bounding box appears in the grid square (for each anchor box)

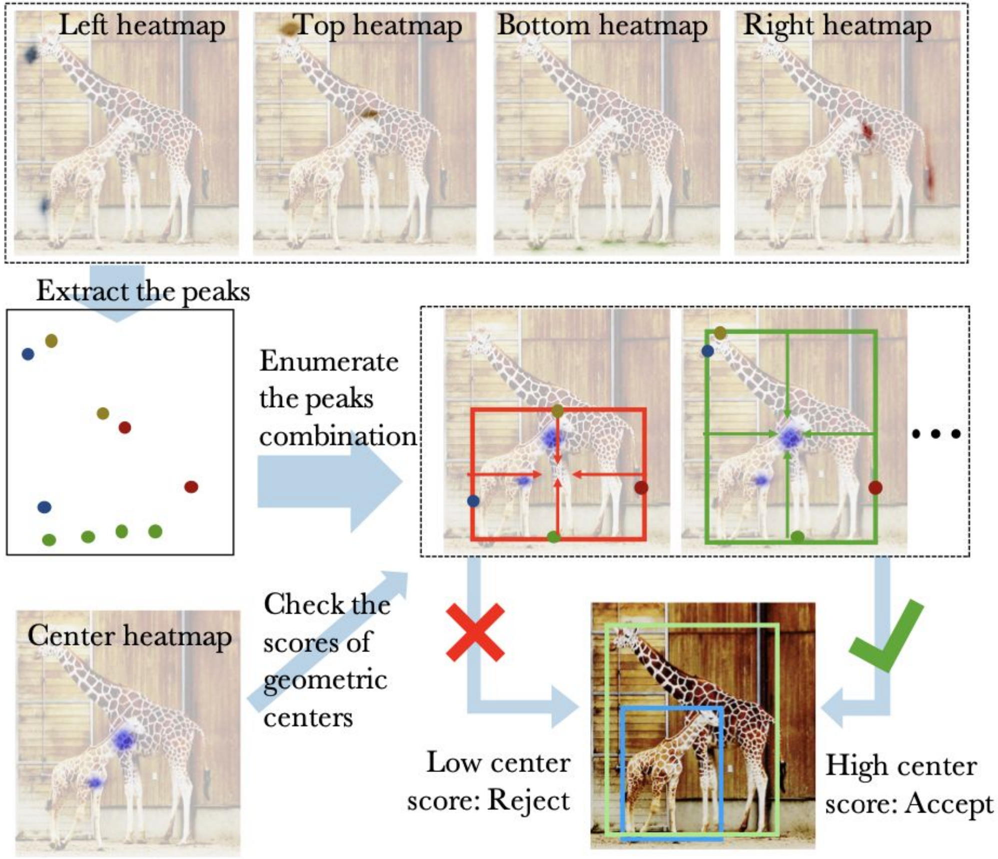





7XJT316
Box 1
Box 2
Box 3

Other Methods

- Some methods output a "heatmap" or series of heatmaps
- Probability each pixel is a keypoint of the bounding box and its class
- Examples
    - CenterNet
    - ExtremeNet



Bottom-up Object Detection by Grouping Extreme and Center Points

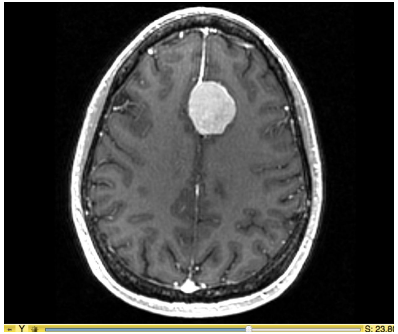
by X. Zhou, J. Zhuo, P. Krähenbühl

Segmentation

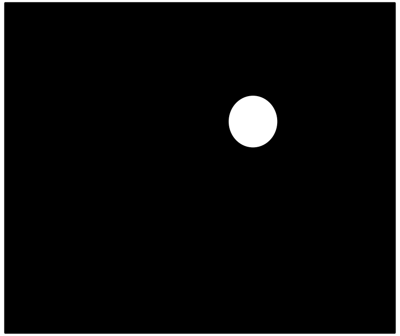
- Instance or Scene Segmentation
- More sophisticated labels
- Output/Labels are masks or series of masks (same size as image)

Segmentation

- Medical Imaging



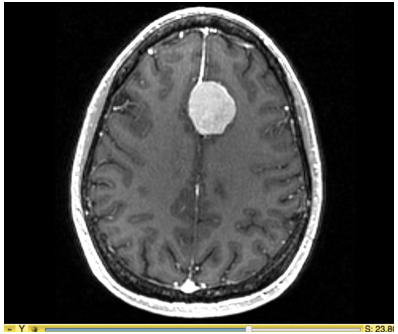
Input



Label

Segmentation

- Medical Imaging

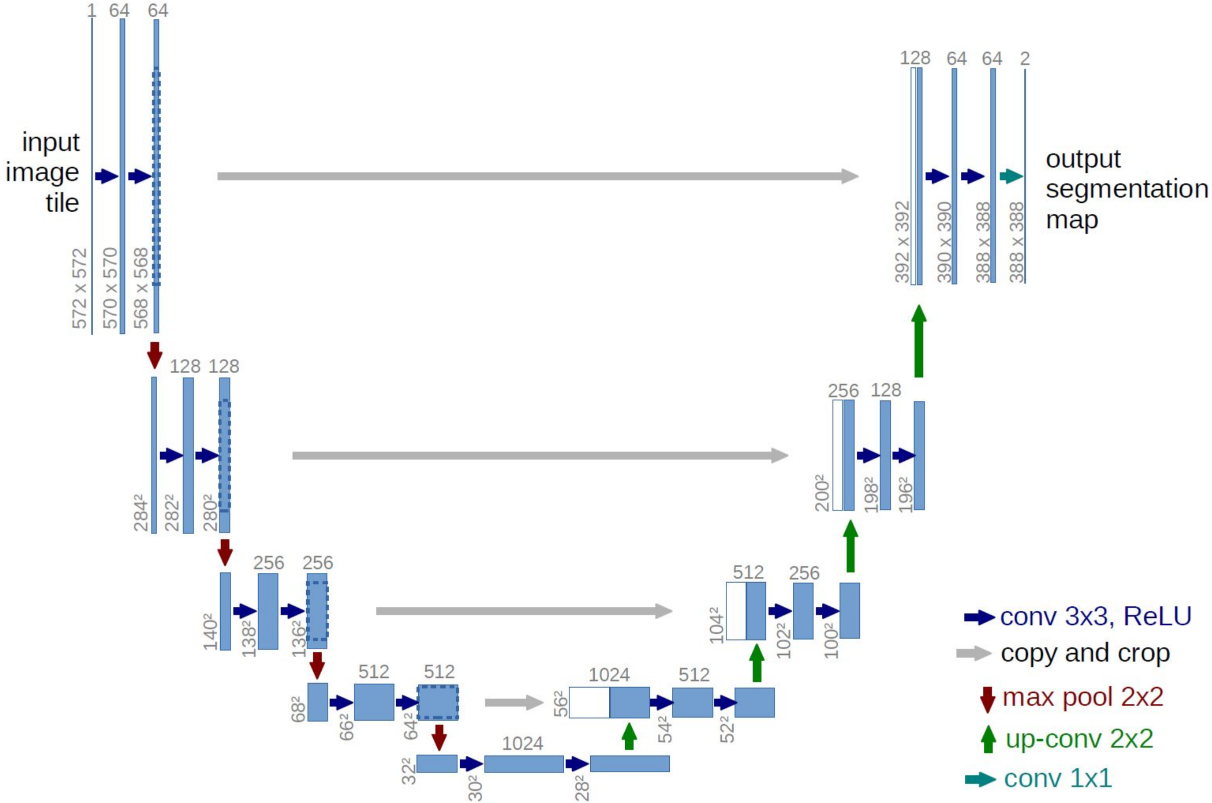


Input

labels

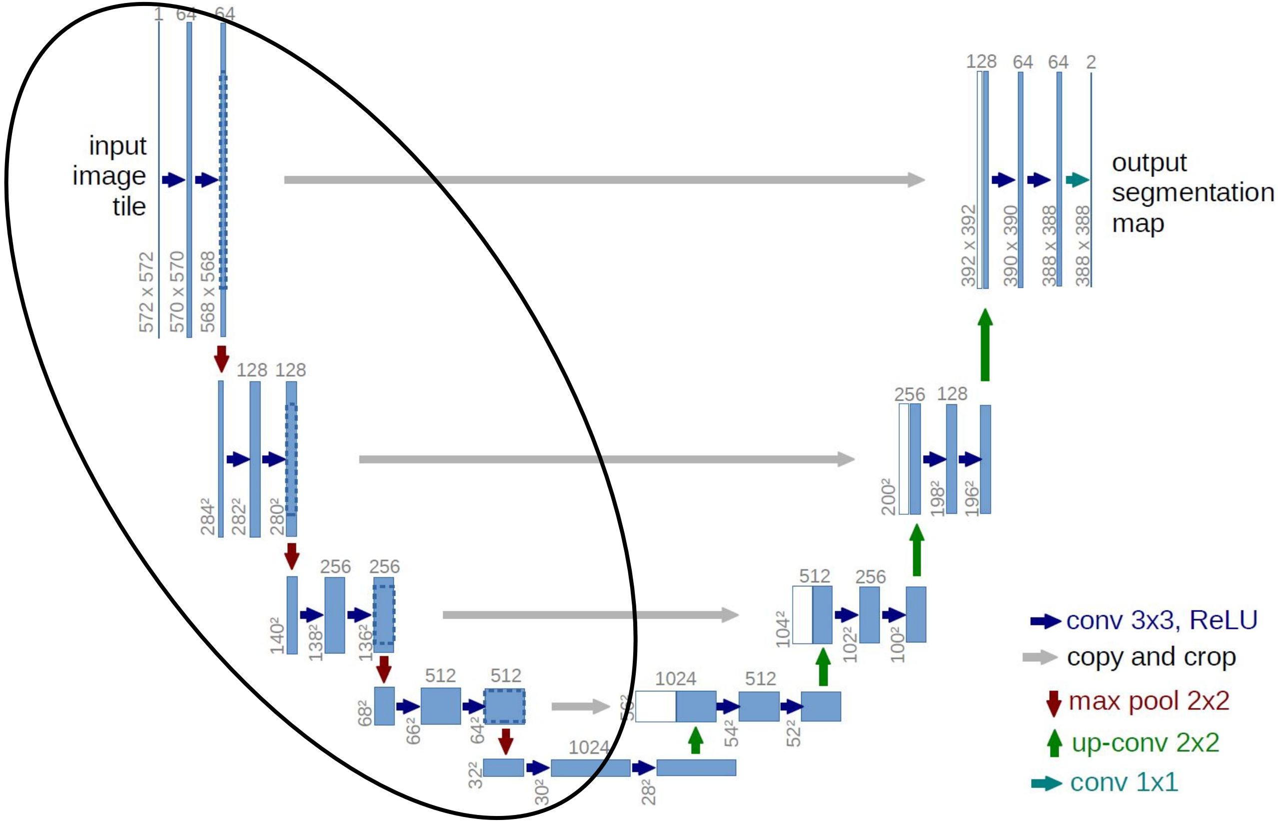
Segmentation

- UNet-like Architecture



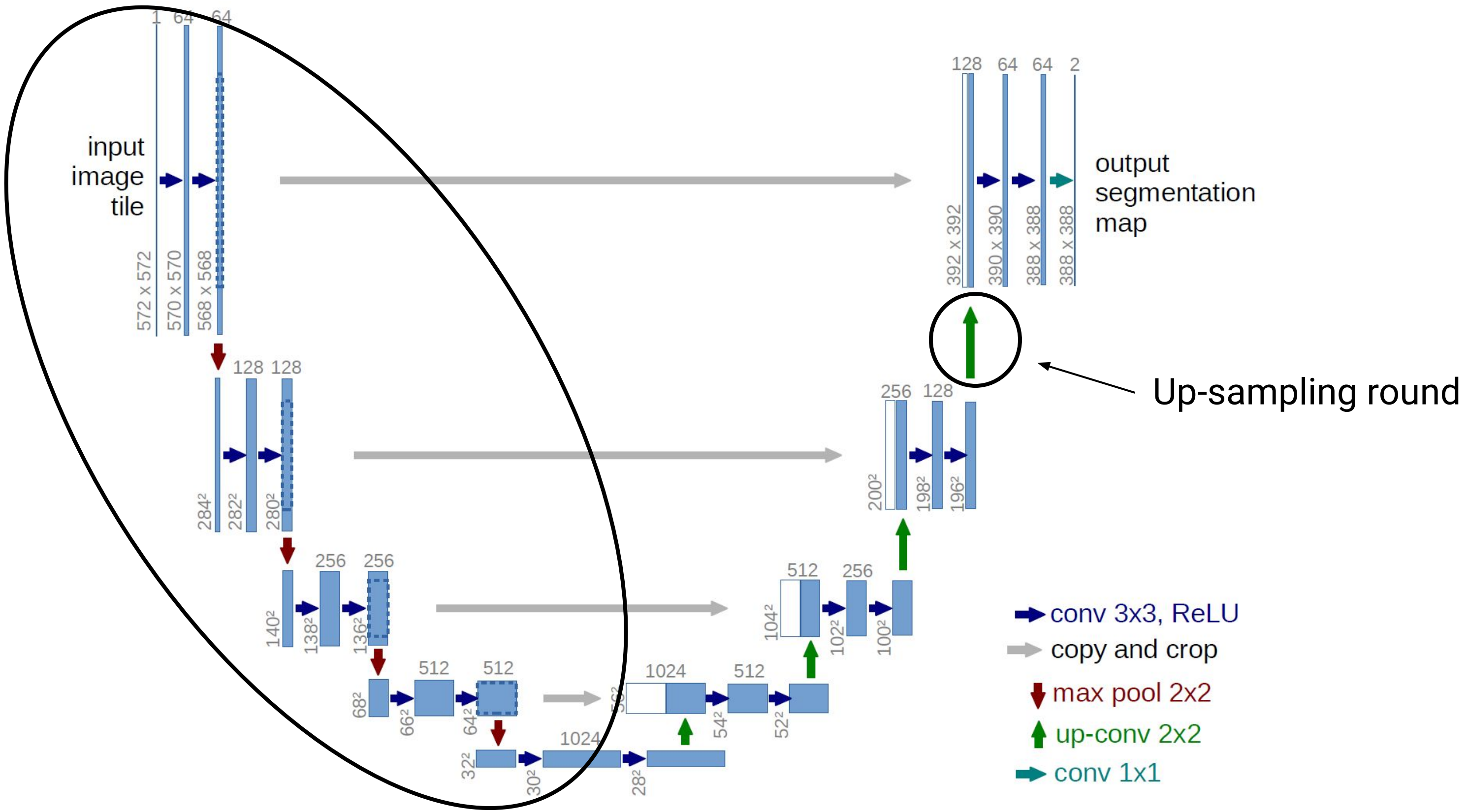
Segmentation

- UNet-like Architecture



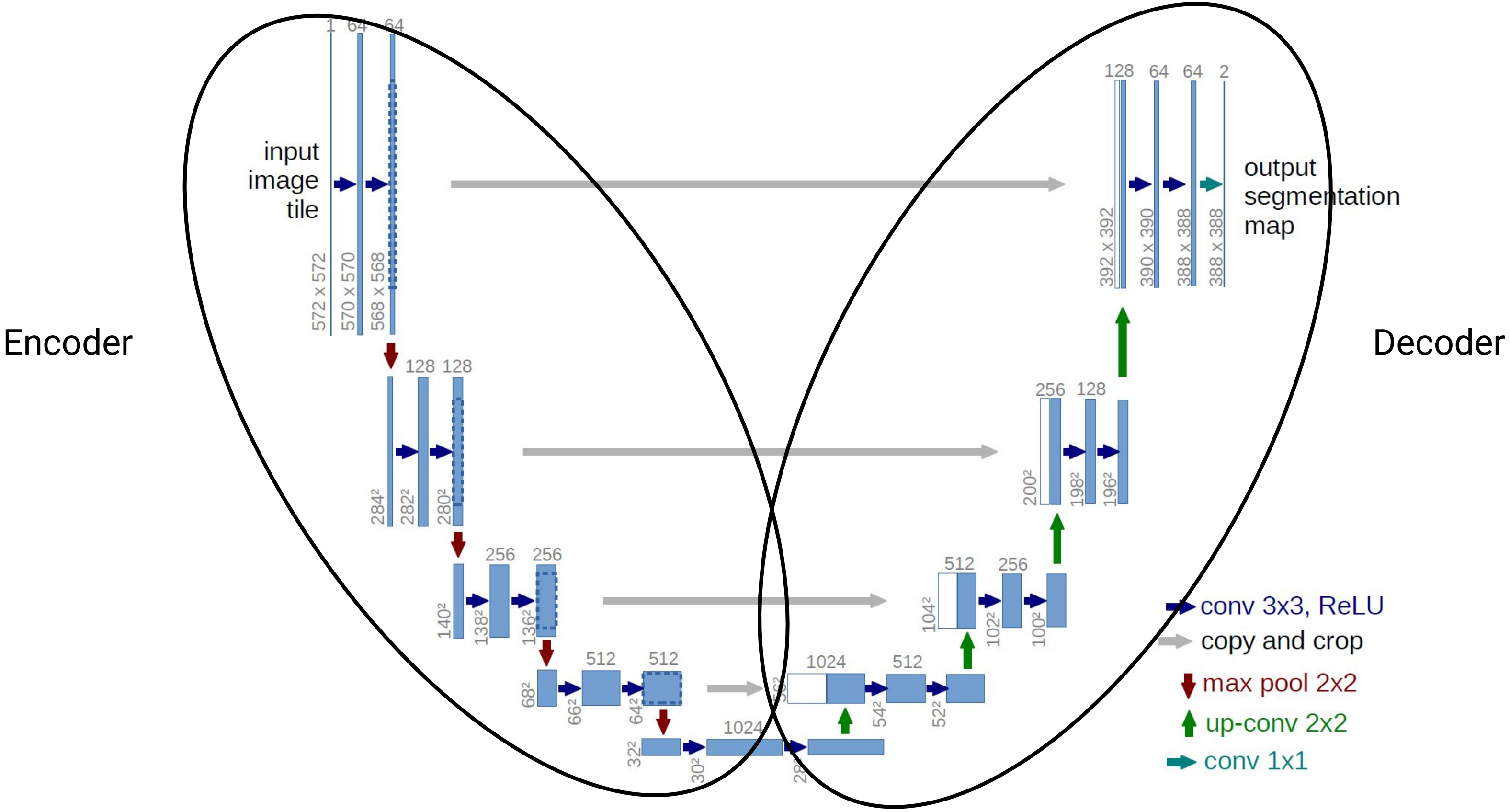
Segmentation

- UNet-like Architecture



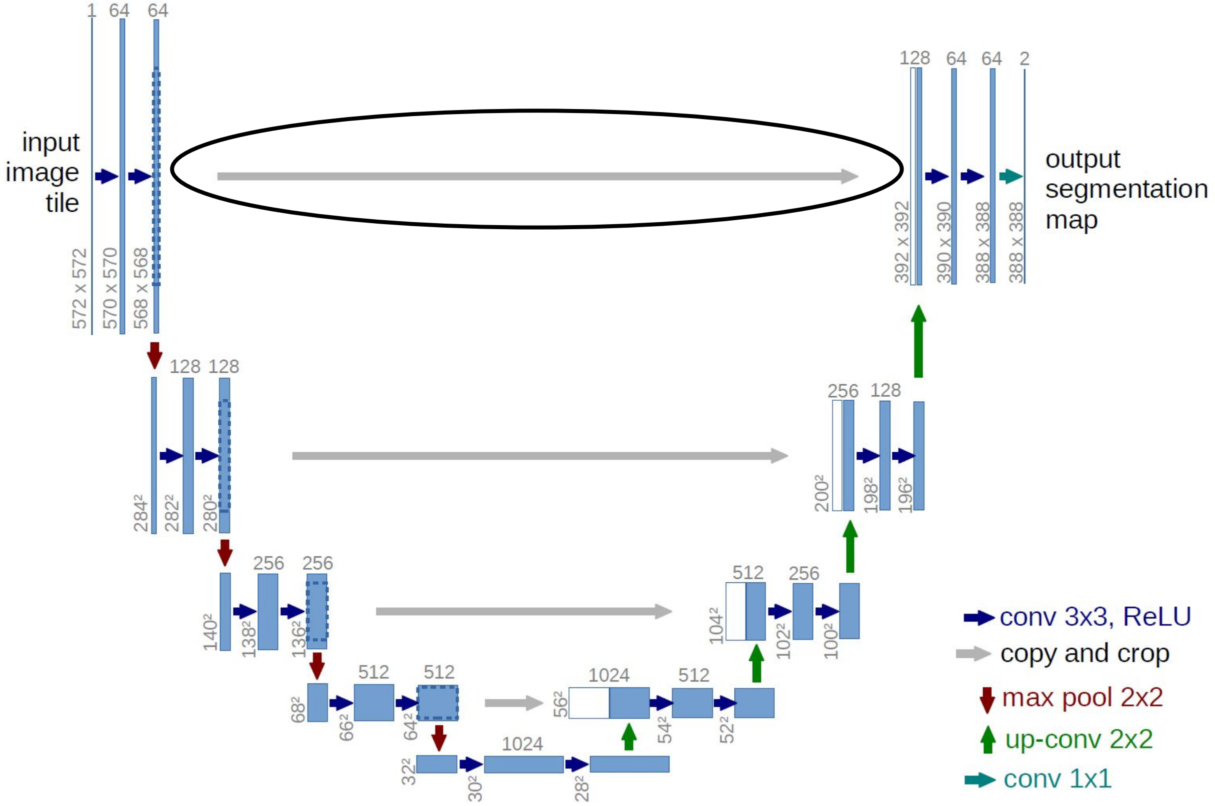
Segmentation

- UNet-like Architecture



Segmentation

- UNet-like Architecture



Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling

$$\begin{bmatrix} \boxed{.5} & .25 \\ 0 & 1 \end{bmatrix}$$

Input

$$\begin{bmatrix} .5 & .5 & .25 & .25 \\ .5 & .5 & .25 & .25 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Output

Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling
- Up-Convolution

$$\begin{bmatrix} .5 & .25 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Input　　　　Filter

Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling
- Up-Convolution

$$\begin{bmatrix} \boxed{.5} & .25 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} \boxed{\begin{matrix} 0 & -.5 \\ .5 & 0 \end{matrix}} & \\ & \end{bmatrix}$$

Input

Filter

Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling
- Up-Convolution

$$\begin{bmatrix} .5 & \boxed{.25} \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & -.5 & 0 & -.25 \\ .5 & 0 & .25 & 0 \\ & & & \\ & & & \end{bmatrix}$$

Input

Filter

Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling
- Up-Convolution

$$\underset{\text{Input}}{\begin{bmatrix} .5 & .25 \\ 0 & 1 \end{bmatrix}} * \underset{\text{Filter}}{\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}} \longrightarrow \begin{bmatrix} 0 & -.5 & 0 & -.25 \\ .5 & 0 & .25 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Up-Sampling

- Increase resolution of the feature map
- Simplest: Un-pooling
- Up-Convolution

$$\begin{bmatrix} .5 & .25 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & -.5 & 0 & -.25 \\ .5 & 0 & .25 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Input

Filter