These slides are distributed under the Creative Commons License. DeepLearning.AI makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite DeepLearning.AI as the source of the slides.

For the rest of the details of the license, see https://creativecommons.org/licenses/by-sa/2.0/legalcode
Object Detection

Object localization
What are localization and detection?

Image classification

Classification with localization

Detection

"Car"

1 object

"Car"

"Car"

multiple objects
Classification with localization

1 - pedestrian
2 - car
3 - motorcycle
4 - background

\( b_x, b_y, b_h, b_w \)

\( (0,0) \)
Defining the target label $y$

1. pedestrian
2. car
3. motorcycle
4. background

Need to output $b_x, b_y, b_h, b_w, \text{class label (1-4)}$

$$L(y, y') = \begin{cases} 
(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 \\
\vdots + (\hat{y}_n - y_n)^2 & \text{if } y_1 = 1 \\
(\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 
\end{cases}$$

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

is there an object?

Andrew Ng
Object Detection

Landmark detection
Landmark detection

\[ b_x, b_y, b_h, b_w \]

\[ l_{x1}, l_{y1}, l_{x2}, l_{y2}, \ldots, l_{x_{64}}, l_{y_{64}} \]
Car detection example

Training set:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
</tr>
</tbody>
</table>

$\rightarrow$ ConvNet $\rightarrow y$
Sliding windows detection

\[ \text{ConvNet} \rightarrow O \]

\[ \text{ConvNet} \]

\[ \text{Computational cost} \]
Object Detection

Convolutional implementation of sliding windows
Turning FC layer into convolutional layers

14 × 14 × 3 → 5 × 5 → 10 × 10 × 16 → MAX POOL 2 × 2 → 5 × 5 × 16 → FC → 400 → FC → 400 → softmax (4)

14 × 14 × 3 → 5 × 5 → 10 × 10 × 16 → MAX POOL 2 × 2 → 5 × 5 × 16 → FC 5 × 5 → 400 → FC 1 × 1 → 400 → 1 × 1 × 4 → 1 × 1 × 4
Convolution implementation of sliding windows

[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]
Convolution implementation of sliding windows

28×28 → 5×5
16×16 → MAX POOL 2×2 → 12×12 → 5×5 → 12×12 → 1×1 → 1×1 → 8×8×4

Andrew Ng
Object Detection

Bounding box predictions
Output accurate bounding boxes
YOLO algorithm

Labels for training
For each grid cell:

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Specify the bounding boxes

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Object Detection

Intersection over union
Evaluating object localization

More generally, IoU is a measure of the overlap between two bounding boxes.
Object Detection

Non-max suppression
Non-max suppression example
Non-max suppression example
Non-max suppression example
Non-max suppression algorithm

Each output prediction is:

Discard all boxes with $p_c \leq 0.6$

While there are any remaining boxes:

- Pick the box with the largest $p_c$
  Output that as a prediction.
- Discard any remaining box with $\text{IoU} \geq 0.5$ with the box output in the previous step
Overlapping objects:

Anchor box 1:

Anchor box 2:

\[
y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_n \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}
\]

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Anchor box algorithm

Previously:
Each object in training image is assigned to grid cell that contains that object’s midpoint.

With two anchor boxes:
Each object in training image is assigned to grid cell that contains object’s midpoint and anchor box for the grid cell with highest IoU.

Output $y$:

3 x 2 x 8

(grid cell, anchor box)

Output $y$:

3 x 3 x 16
3 x 3 x 2 x 8
Anchor box example

Anchor box 1: Anchor box 2:

\[
y = \begin{bmatrix}
  p_c \\
  b_x \\
  b_y \\
  b_w \\
  b_h \\
  c_1 \\
  c_2 \\
  c_3
\end{bmatrix}
\]
Object Detection

Putting it together: YOLO algorithm
Training

1 - pedestrian
2 - car
3 - motorcycle

\[ y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \]

\[ y \text{ is } 3 \times 3 \times 2 \times 8 \]

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Making predictions

\[ y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \]

\[ \Rightarrow \begin{bmatrix} \cdots \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \]
Outputting the non-max supressed outputs

- For each grid call, get 2 predicted bounding boxes.
- Get rid of low probability predictions.
- For each class (pedestrian, car, motorcycle) use non-max suppression to generate final predictions.
Object Detection

Region proposals (Optional)
Region proposal: R-CNN

[Giśhik et. al, 2013, Rich feature hierarchies for accurate object detection and semantic segmentation] Andrew Ng
Faster algorithms

- **R-CNN:** Propose regions. Classify proposed regions one at a time. Output label + bounding box.

  - Girshik et. al, 2013. Rich feature hierarchies for accurate object detection and semantic segmentation

- **Fast R-CNN:** Propose regions. Use convolution implementation of sliding windows to classify all the proposed regions.

  - Girshik, 2015. Fast R-CNN

- **Faster R-CNN:** Use convolutional network to propose regions.

  - Ren et. al, 2016. Faster R-CNN: Towards real-time object detection with region proposal networks
Convolutional Neural Networks

Semantic segmentation with U-Net
Object Detection vs. Semantic Segmentation

Input image  
Object Detection  
Semantic Segmentation
Motivation for U-Net

[Chest X-Ray]

[Brain MRI]

[Dong et al., 2017, Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks]

[Novikov et al., 2017, Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs]
Per-pixel class labels

1. Car
0. Not Car
Per-pixel class labels

1. Car
2. Building
3. Road

Segmentation Map
Deep Learning for Semantic Segmentation
Transpose Convolution

Normal Convolution

Transpose Convolution

Andrew Ng
Transpose Convolution

filter $f \times f = 3 \times 3$

padding $p = 1$

stride $s = 2$

Andrew Ng
Deep Learning for Semantic Segmentation
[Ronneberger et al., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation]

Andrew Ng
U-Net

[Andrew Ng, 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation]