1. Discussion leading towards YOLO!

2. What is YOLO?
   - Introduction
   - Unified Object Detection
   - The design of cost function

3. Experimental analysis
   - Comparisons

4. Limitations of YOLO

5. Summary
The Infographic!

Anup Deshmukh, Pratheeksha Nair
Seminar
March 28, 2018
Little bit of History

- Object Detection using Hog Features (2005)
- Region-based Convolutional Neural Networks (R-CNN) (Oct 2014)
- Fast (R-CNN) (Sept 2015)
- Faster (R-CNN) (Jan 2016)
Next Subsection

1. Discussion leading towards YOLO!

2. What is YOLO?
   - Introduction
     - Unified Object Detection
     - The design of cost function

3. Experimental analysis
   - Comparisons

4. Limitations of YOLO

5. Summary
Introduction

- Modelling the Regression problem
- Using bounding boxes with associated class prob.
- Single DNN - end to end optimization
- Full image as input (Understanding of context)
- Extremely fast (45fps)
Discussion leading towards YOLO!

What is YOLO?

Introduction
Unified Object Detection
The design of cost function

Experimental analysis
Comparisons

Limitations of YOLO

Summary
Unified Object Detection

- YOLO sees entire image during both training and testing
- Single regression problem (image pixels to bounding boxes and classes)
Unified Object Detection

- YOLO sees entire image during both training and testing
- Single regression problem (image pixels to bounding boxes and classes)
Unified Object Detection - Workflow

- YOLO divides input image into $S \times S$ grid

- If the center of an object falls inside a grid cell, that grid cell is responsible for detecting the object
Unified Object Detection - Workflow

- Each grid cell predicts $B$ bounding boxes and confidence scores for these boxes.

- Confidence of box being accurate and box containing an object:
  
  \[
  \text{Confidence} = \text{Pr}(\text{object}) \times \text{IOU} \text{truth} \times \text{IOU} \text{pred}
  \]

- Each bounding box is represented as $[x, y, w, h, \text{conf}]$. 

---

Anup Deshmukh, Pratheeksha Nair

Seminar

March 28, 2018 10 / 25
Unified Object Detection - Workflow

- Each grid cell predicts B bounding boxes and confidence scores for these boxes.
- Confidence of box being accurate and box containing an object:

\[
\text{Confidence} = \Pr(\text{object}) \times IOU_{true}^{pred}
\]
Unified Object Detection - Workflow

- Each grid cell predicts $B$ bounding boxes and confidence scores for these boxes.
- Confidence of box being accurate and box containing an object:
  \[
  \text{Confidence} = \Pr(\text{object}) \times IOU_{\text{pred}}^{\text{truth}}
  \]
- Each bounding box is represented as $[x, y, w, h, \text{conf}]$. 
Unified Object Detection - Workflow

- Each grid cell calculates $C_i = Pr(Class_i|Object)$ for all classes of objects
- In each grid cell, one box is chosen and $C_i$ is calculated
- This is encoded as a tensor of size $[S \times S \times (B*5 + C)]$
Unified Object Detection - Workflow

- Each grid cell calculates $C_i = Pr(Class_i|Object)$ for all classes of objects
- In each grid cell, one box is chosen and $C_i$ is calculated
- This is encoded as a tensor of size $[S \times S \times (B*5 + C)]$
Unified Object Detection - Testing

- The conditional class probabilities and confidence of predicted bounding boxes are multiplied

$$Pr(Class_i | Object) \times Pr(Object) \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth}$$

- For each predicted box, class specific confidence scores
- Confidence of predicted box fitting the object and object belonging to that class
Quick overview of YOLO
1 Discussion leading towards YOLO!

2 What is YOLO?
   Introduction
   Unified Object Detection
   The design of cost function

3 Experimental analysis
   Comparisons

4 Limitations of YOLO

5 Summary
The design

- The authors have used sum-squared error because it is easy to optimize.
- Sum-squared loss should not equally weight errors with object equally with error when there is no object.
- Sum-squared loss also should not equally weight errors in large boxes and small boxes.
Cost function

Term 1

\[ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \]

\[ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \]

\[ + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{i,j}^{\text{obj}} (C_i - \hat{C}_i)^2 \]
Cost function

Term 1

\[
\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} (C_i - \hat{C}_i)^2
\]

Term 2

\[
+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
+ \sum_{i=0}^{S^2} 1_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\]
Discussion leading towards YOLO!

What is YOLO?
- Introduction
- Unified Object Detection
- The design of cost function

Experimental analysis
- Comparisons

Limitations of YOLO

Summary
DPM - The only other model that runs in real time

Table:

<table>
<thead>
<tr>
<th></th>
<th>DPM</th>
<th>YOLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>processes</td>
<td>processes 30fps</td>
<td>processes 45fps</td>
</tr>
<tr>
<td>mAP</td>
<td>mAP is 26.1%</td>
<td>mAP is 63.4%</td>
</tr>
</tbody>
</table>
**Table:**

<table>
<thead>
<tr>
<th>RCNN</th>
<th>YOLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>processes 5-18fps</td>
<td>processes 45fps</td>
</tr>
<tr>
<td>3x times better at localizing</td>
<td>lags in localizing accuracy</td>
</tr>
<tr>
<td>far more background errors</td>
<td>analyzing image as a whole</td>
</tr>
</tbody>
</table>
The Generalization

- PICASSO and People-Art Dataset
- YOLO generalizes better to new domains better than others
- RCNN drops considerably (Selective search is tuned only for natural images)
- YOLO has good performance (models shape, size and relationship between objects and their locations)
Limitations

• Errors are treated same in small bounding boxes versus large bounding boxes.

• Small objects that appear in groups
YOLO is modelled as a super-fast regression problem which takes an input image and learns the class probabilities and bounding box coordinates.
References

- **The Infographics:** https://medium.com/@nikasa1889/the-modern-history-of-object-recognition-infographic-aea18517c318
Questions/Comments?
Thank You!