Beyond classification:
Object detection, Segmentation, Human pose estimation

Gül Varol

IMAGINE team, École des Ponts ParisTech

gul.varol@enpc.fr

http://imagine.enpc.fr/~varolg/

@RecVis, 07.11.2023

With many slides from: Weidi Xie, I. Laptev, R. Girshick, K. He, N. Murray, A. Karpathy, L. Fei Fei, J. Johnson, A. Torralba, K. He, S. Yeung, J. Sivic, M. Aubry
Neural Networks

Last week: Neural networks for visual recognition
(G. Varol)

This week: Beyond classification: Object detection, Segmentation, Human pose estimation
(G. Varol)

Next week: Large-scale image and video search
(Josef. Sivic)
Recap: Neural networks for image classification

AlexNet (2012)

VGG-16 (2015)

ResNet (2016)

ViT (2021)
Agenda

• 0. Intro to structured outputs
• 1. Object detection (localization)
• 2. Segmentation
• 3. Human pose estimation

Image credits: Naila Murray
Agenda

• 0. Intro to structured outputs
• 1. Object detection (localization)
• 2. Segmentation
• 3. Human pose estimation

Image credits: Naila Murray
What we would like to do...

• Visual scene understanding
• **What** is in the image and **where**

- Dog 1: Terrier
- Motorbike: Suzuki GSX 750
- Ground: Gravel
- Plant
- Dog 2: Sitting on Motorbike
- Person: John Smith, holding Dog 2
- Wall
- Motorbike: Suzuki GSX 750

• Object categories, identities, properties, activities, relations, …
(Some) Fundamental Tasks in Computer Vision

- **Image Classification**
  - Does the image contain an aeroplane? *(last lecture)*

- **Object Class Detection/Localization**
  - Where are the aeroplanes (if any)?

- **Object Class Segmentation**
  - Which pixels are part of an aeroplane (if any)?
(Some) Fundamental Tasks in Computer Vision

- Representing objects in the image:
  - Class labels
  - Bounding box
  - Semantic pixel-wise labels
  - Instance pixel-wise labels

Lin et al. “Microsoft COCO: Common Objects in Context”
Object Detection

What?

Where?
Object Detection with **Bounding Boxes**

“What?”

“What?”

“Object detection”
Object Detection with **Segmentation Masks**

What?

Where?

“Instance segmentation”
Classification vs. Detection

✓ Dog

Dog

Dog
Problem formulation

\{ \, \text{airplane, bird, motorbike, person, sofa} \, \}
Things vs. Stuff

**Thing** (n): An object with a specific size and shape.

**Stuff** (n): Material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape.

Ted Adelson, Forsyth et al. 1996.

Slide: Geremy Heitz
Panoptic segmentation

Kirillov, He, Girshick, Rother, Dollar CVPR 2019

- **stuff** – amorphous regions of similar texture or material such as grass, sky, road

- **things** – countable objects such as people, animals, tools
Challenges: Scale
Challenges: Occlusion and truncation
Challenges: Background Clutter
Challenges: Intra-class variation
Challenges: How to evaluate object detection?

Images may contain many objects and classes

Localization results may not be precise

Image source

- Ground truth
- Detector output
## COCO Object Detection Average Precision (%)

<table>
<thead>
<tr>
<th></th>
<th>Early 2015</th>
<th>Past (best circa 2012)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM (Pre DL)</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Fast R-CNN (AlexNet)</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Movement to Deep Learning methods: *3x improvement in AP*
COCO Object Detection Average Precision (%)

Progress within DL methods: > 3x!

Past (best circa 2012)  | Early 2015  | Late 2018
---|---|---
DPM (Pre DL)  | Fast R-CNN (AlexNet)  | Faster R-CNN (ResNet-50)
15  | 19  | 29
5  | 19  | 36

Mask R-CNN (X-152-FPN)  | 39
49

~4 years
Steady Progress on Boxes and Masks

- **R-CNN** [Girshick et al. 2014]
- **SPP-net** [He et al. 2014]
- **Fast R-CNN** [Girshick. 2015]
- **Faster R-CNN** [Ren et al. 2015]
- **R-FCN** [Dai et al. 2016]
- **Feature Pyramid Networks + Faster R-CNN** [Lin et al. 2017]
- **Mask R-CNN** [He et al. 2017]
- **Training with Large Minibatches (MegDet)** [Peng, Xiao, Li, et al. 2017]
- **Cascade R-CNN** [Cai & Vasconcelos 2018]
Beyond Boxes and Masks: Human Keypoints

COCO Keypoint Detection Task
[COCO team @ cocodataset.org 2016 - present]
Beyond Boxes and Masks: Human Surfaces

DensePose: Dense Human Pose Estimation In The Wild
[Güler, Neverova, Kokkinos CVPR 2018]
Beyond Boxes and Masks: 3D Shape

Mesh R-CNN
[Gkioxari, Malik, Johnson ICCV 2019]
Agenda

• 0. Intro to structured outputs

• 1. Object detection (localization)

• 2. Segmentation

• 3. Human pose estimation
# Object detection datasets (benchmarks)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Categories</th>
<th>Images</th>
<th>Bounding Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL-VOC</td>
<td>20</td>
<td>11K</td>
<td>27K</td>
</tr>
<tr>
<td>COCO</td>
<td>80 (91 stuff)</td>
<td>328K</td>
<td>2500K</td>
</tr>
<tr>
<td>LVIS</td>
<td>1200</td>
<td>164K</td>
<td>2.2M</td>
</tr>
</tbody>
</table>

PASCAL-VOC [2005-2008]  
COCO [2014-2015]
Evaluating a detector

Test image (previously unseen)
First detection ...

'person' detector predictions
Second detection ...

0.9

0.6

☐ ‘person’ detector predictions
Third detection ...

☐ ‘person’ detector predictions
Compare to ground truth

- ‘person’ detector predictions
- ground truth ‘person’ boxes
Compare to ground truth

- **True positive**
- **False positive**
- **False negative**

‘person’ detector predictions

ground truth ‘person’ boxes
Sort by confidence

0.9 ✓
0.8 X
0.6 ✓
0.5 ✓
0.2 X
0.1 X

true positive (high overlap) i.e., IOU > threshold
false positive (no overlap, low overlap, or duplicate)
false negative (missing detection)

IOU (intersection over union):
intersection
union
Sort by confidence

precision@t = \frac{\text{true positives}@t}{\text{true positives}@t + \text{false positives}@t}

recall@t = \frac{\text{true positives}@t}{\text{ground truth objects}}
Sort by confidence

\[ \text{precision}@t = \frac{\#\text{true positives}@t}{\#\text{true positives}@t + \#\text{false positives}@t} \]

\[ \text{recall}@t = \frac{\#\text{true positives}@t}{\#\text{ground truth objects}} \]

\[ t = 0.5 \]

\[ \sqrt{\checkmark} + \checkmark = 75\% \]

\[ \checkmark = 75\% \]
Sort by confidence

\[
\begin{align*}
\text{precision} @ t &= \frac{\text{true positives} @ t}{\text{true positives} @ t + \text{false positives} @ t} \\
\text{recall} @ t &= \frac{\text{true positives} @ t}{\text{ground truth objects}}
\end{align*}
\]

\[
t = 0.9 \\
\text{precision} @ t = \frac{\checkmark}{\checkmark + \times} = 100\% \\
\text{recall} @ t = \frac{\checkmark}{\times} = 25\%
\]
Average Precision for a (class, IOU threshold) pair

0.9  0.8  0.6  0.5  0.2  0.1
✓   X   ✓   ✓   X   X

Average Precision (AP)
0% is worst
100% is best
mean AP over classes (mAP)

AP(class, threshold): area under PR curve
Average Precision for a class

\[ AP(\text{class}) = \frac{1}{\#\text{thresholds}} \sum_{\text{iou} \in \text{threshold}} AP(\text{class}, \text{iou}) \]

Figure credits: Dollár and Zitnick
Overall Average Precision (%)

\[ AP = \frac{1}{\#\text{classes}} \sum_{\text{class} \in \text{classes}} AP(\text{class}) \]

“AP” is really an average, average, average precision.

- classes
- iou thresholds
- precision @ different recall levels

**Average Precision (AP):**

- AP
- AP\(_{\text{IoU=.50}}\)
- AP\(_{\text{IoU=.75}}\)

- % AP at IoU=.50:.05:.95 (primary challenge metric)
- % AP at IoU=.50 (PASCAL VOC metric)
- % AP at IoU=.75 (strict metric)

**AP Across Scales:**

- AP\(_{\text{small}}\)
- AP\(_{\text{medium}}\)
- AP\(_{\text{large}}\)

- % AP for small objects: area < 32\(^2\)
- % AP for medium objects: 32\(^2\) < area < 96\(^2\)
- % AP for large objects: area > 96\(^2\)
Object detection: naive attempt

Find the chair in this image

This is a chair

Slide credit: A. Torralba
Object detection: naive attempt

Find the chair in this image

Output of normalized correlation

This is a chair

Slide credit: A. Torralba
Object detection: naive attempt

Find the chair in this image

Pretty much garbage
Simple template matching is not going to make it

Slide credit: A. Torralba
Detection by Classification

- Basic component: binary classifier
Detection by Classification

• Detect objects in clutter by **search**

  • **Sliding window**: exhaustive search over position and scale
Detection by Classification

• Detect objects in clutter by **search**

• **Sliding window**: exhaustive search over position and scale
Detection by Classification

- Detect objects in clutter by **search**

- **Sliding window**: exhaustive search over position and scale (can use same size window over a spatial pyramid of images)
Test: Non-maximum suppression (NMS)

- Scanning-window detectors typically result in multiple responses for the same object

To remove multiple responses, a simple greedy procedure called “Non-maximum suppression” is applied:

NMS:
1. Sort all detections by detector confidence
2. Choose most confident detection $d_i$; remove all $d_j$ s.t. $\text{overlap}(d_i, d_j) > T$
3. Repeat Step 2. until convergence
Test: Non-maximum suppression (NMS)

- Scanning-window detectors typically result in multiple responses for the same object

- To remove multiple responses, a simple greedy procedure called “Non-maximum suppression” is applied:

  NMS:
  1. Sort all detections by detector confidence
  2. Choose most confident detection $d_i$; remove all $d_j$ s.t. $\text{overlap}(d_i, d_j) > T$
  3. Repeat Step 2. until convergence
Detection by Classification

• Detect objects in clutter by **search**

Problem: too many windows to run a classifier

• **Sliding window**: exhaustive search over position and scale (can use same size window over a spatial pyramid of images)
Object proposals

Generate and evaluate a few hundred region proposals.

- Proposal mechanism can:
  - take advantage of low-level perceptual organization cues,
  - be category-specific or category-independent, handcrafted or trained.

- Classifier can be slower but more powerful.

Slide credit: Lana Lazebnik
Region proposals: Selective search

1. Merge two most similar regions based on similarity.
2. Update similarities between the new region and its neighbors.
3. Go back to step 1. until the whole image is a single region.

[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]
Region proposals: Selective search

Take bounding boxes of all generated regions and treat them as possible object locations.

[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]
Region proposals: Selective search

[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]
Object detection: CNN-based methods
R-CNN: Region-based CNN

Apply bounding-box regressors
Classify regions with SVMs
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image
Post hoc component

Girshick et al. CVPR14.

R-CNN Training

**Step 1:** Train (or download) a classification model for ImageNet (AlexNet)
**R-CNN Training**

**Step 2: Extract features**
- Extract region proposals for all images
- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk
- Have a big hard drive: features are ~200GB for PASCAL dataset!
Step 3: Fine-tune model for detection

- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images
Step 4: Train one binary SVM per class to classify region features

Training image regions

Cached region features

+ Positive samples for cat SVM

- Negative samples for cat SVM
R-CNN Training

**Step 4:** Train one binary SVM per class to classify region features

- **Training image regions**
  - [Images of cats and a dog]

- **Cached region features**
  - **- Negative** samples for dog SVM
  - **+ Positive** samples for dog SVM
**Step 5 (bbox regression):** For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for “slightly wrong” proposals.

- **Training image regions**
- **Cached region features**
- **Regression targets**
  - (dx, dy, dw, dh)
  - Normalized coordinates
- **Proposal is good**: (0, 0, 0, 0)
- **Proposal too far to left**: (.25, 0, 0, 0)
- **Proposal too wide**: (0, 0, -0.125, 0)
R-CNN Results

Big improvement compared to pre-CNN methods

Mean Average Precision (mAP)

Wang et al, "Regionlets for Generic Object Detection", ICCV 2013
R-CNN Results

Bounding box regression helps a bit

![Graph showing R-CNN Results with mean average precision (mAP)]
Features from a deeper network help a lot.

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM (2011)</td>
<td>33.7</td>
<td>23.6</td>
</tr>
<tr>
<td>Regionlets (2013)</td>
<td>41.7</td>
<td>39.7</td>
</tr>
<tr>
<td>R-CNN (2014, AlexNet)</td>
<td>54.2</td>
<td>50.2</td>
</tr>
<tr>
<td>R-CNN + bbox reg (AlexNet)</td>
<td>58.3</td>
<td>53.7</td>
</tr>
<tr>
<td>R-CNN (VGG-16)</td>
<td>66</td>
<td>62.9</td>
</tr>
</tbody>
</table>
R-CNN [CVPR 2014] Summary

Two-stage detector

- Propose large number of regions potentially with objects

- Classify each proposed region
R-CNN [CVPR 2014] Limitations

Two-stage detector

- Propose large number of regions potentially with objects

1. **Slow at test-time:** need to run full forward pass of CNN for each region proposal
2. **Not end-to-end:** SVMs and regressors are post-hoc, CNN features not updated in response to SVMs and regressors
3. **Complex** multistage training pipeline

---

Fast R-CNN [ICCV 2015]

- Small accuracy improvement
- Timing excluding region proposal
  ~10x faster for training
  ~100x faster for testing (< 1 sec / image)

R. Girshick, Fast R-CNN, ICCV 2015

---

Input: an image

Region proposal:
selective search, edge box

Proposals/Candidates
Region of Interests (RoI)

Class Probability

BBox Regression

ConvNet
Multilayer Perceptron (MLP)

Feature map for an image

Rol-Pool

Similar to Crop & Resize

Feature map for a RoI
Problem #1: Slow at test-time due to independent forward passes of the CNN

Solution: Share computation of convolutional layers between proposals for an image

Girshick et al. ICCV 2015.
R-CNN Problems

**Problem #2**: Post-hoc training: **CNN not updated** in response to final classifiers and regressors.

**Problem #3**: Complex training pipeline.

**Solution**: Just train the whole system end-to-end all at once!

---

Fast R-CNN Solutions

**Girshick et al. ICCV 2015.**
Fast R-CNN [ICCV 2015]

Region proposal is still independent and can be slow (1-2 sec)

Input: an image

Region proposal: selective search, edge box

Class Probability
BBox Regression

ConvNet
Multilayer Perceptron (MLP)

Feature map for an image
Similar to Crop & Resize

Feature map for a RoI
### Faster R-CNN [NeurIPS 2015]

- Clear boost in performance
- \(~ 0.2\) sec / image
- End-to-end trainable
- Today, still reference for detection

---

**Input:** an image

**Region Proposal Network (RPN)**

**Region of Interests (RoI)**

**Class Probability**

**BBox Regression**

**ConvNet Multilayer Perceptron (MLP)**

**Feature map for an image**

**Feature map for a RoI**

**Similar to Crop & Resize**

---


Slides modified from Ross Girshick tutorial at CVPR 2019
Faster R-CNN

- Insert a Region Proposal Network (RPN) after the last convolutional layer.

- RPN trained to produce region proposals directly; no need for external region proposals!

- After RPN, use “RoI Pooling” and an upstream classifier and bbox regressor just like Fast R-CNN.
RPN: Region Proposal Network

Slide a small window on the feature map

Build a small network for:
• classifying object or not-object, and
• regressing bbox locations

Position of the sliding window provides localization information with reference to the image

Box regression provides finer localization information with reference to this sliding window

Class-agnostic!

Slide credit: Kaiming He
RPN: Region Proposal Network

Use **N anchor boxes** at each location.

Anchors are **translation invariant**: use the same ones at every location.

Regression gives offsets from anchor boxes.

Classification gives the probability that each (regressed) anchor shows an object.
RPN: Region Proposal Network

\[ f_I = \text{FCN}(I) \]

Feature Map: 16 x 16 x 256

Conv feature map

Slide credit: Ross Girshick
RPN: Region Proposal Network

\[ f_I = \text{FCN}(I) \]

3x3 “sliding window”
Scans the feature map looking for objects

Feature Map : 16 x 16 x 256

Conv feature map

Slide credit: Ross Girshick
RPN: Anchor Box

Anchor box: predictions are w.r.t. this box, not the 3x3 sliding window

$f_I = \text{FCN}(I)$

3x3 “sliding window”
Scans the feature map looking for objects

Feature Map: 16 x 16 x 256

Slide credit: Ross Girshick
RPN: Anchor Box

$ f_I = \text{FCN}(I) $  

3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

Conv feature map

Feature Map: 16 x 16 x 256

Anchor box: predictions are w.r.t. this box, not the 3x3 sliding window

Slide credit: Ross Girshick
RPN: Prediction (on object)

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

3x3 “sliding window”

Objectness score

P(object) = 0.94

the probability that each (regressed) anchor shows an object

Slide credit: Ross Girshick
RPN: Prediction (on object)

- **Objectness classifier [0, 1]**
- **Box regressor predicting (dx, dy, dh, dw)**

Anchor box: transformed by box regressor

P(object) = 0.94

3x3 “sliding window”

Slide credit: Ross Girshick
RPN: Prediction \textbf{(off object)}

Objectness score

3x3 “sliding window”
- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)

P(object) = 0.02

Anchor box: transformed by box regressor

Slide credit: Ross Girshick
RPN: Multiple Anchors

- 3x3 "sliding window"
  - \( n \) objectness classifiers
  - \( n \) box regressors

**Conv feature map**

Feature Map: 16 x 16 x 256

**Anchor boxes**: \( n \) anchors per location with different scales and aspect ratios

\[ f_I = \text{FCN}(I) \]

\[ \hat{Y} \text{ (output): 16 x 16 x } n \times (1+4) \]

\[ \mathbf{Y} \text{ (ground truth): } [1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0] \]

\[ M \text{ (mask): } [1 \\ dx \\ dy \\ dh \\ dw \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0] \]

Loss: \[ \sum_i M_i \cdot \mathcal{L}(\hat{Y}_i, Y_i) \]

- Objectness
- Box regression
- #anchors
- box regression
Faster R-CNN

- Still two-stage


Slides modified from Ross Girshick tutorial at CVPR 2019
Object detection: 2-stage vs 1-stage

CNN Detectors

2-stage detectors
- Region Proposal Network
- Feature map
- Classifier
- RoI pooling
- Proposals

Fast(er) RCNN, Mask-RCNN, SNIPPER, PANet, TridentNet

1-stage detectors
- No object proposals
- Use anchors
- Faster but less accurate

YOLO, SSD, RetinaNet, EfficientDet

Anchor-free detectors
- Use points

CornerNet, CenterNet, FCOS, ExtremeNet
1-stage object detection: YOLO/SSD

Detection without proposals

Input image
3 x H x W

Divide image into grid
7 x 7

Image a set of base boxes
centered at each grid cell
Here B = 3

Redmon et al, “You Only Look Once:
Unified, Real-Time Object Detection”, CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
1-stage object detection: YOLO/SSD

**Detection without proposals**

Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: 
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:

\[ 7 \times 7 \times (5 \times B + C) \]

Divide image into grid

\[ 7 \times 7 \]

Image a set of base boxes centered at each grid cell

Here \( B = 3 \)

From input image to scores with a single network. **Faster but not as accurate as RCNN.**

See also: Lin et al., Focal loss for dense object detection, ICCV 2017.

Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/
Yolo v2 Demo video

http://pureddie.com/yolo

[Redmon et al., CVPR’17]
Scale in object detection

**Problem with YOLO:** Single cell can corresponds to multiple objects, even with multiple anchors, still **too coarse.**
1-stage object detection: **RetinaNet**

- Pre-define anchor boxes on **multiple scales**, e.g., Feature Pyramid Networks (FPNs).
- 6 anchors per location, 100 - 200k anchor boxes to classify per image (dense detection).
- Focal loss for soft-version hard sample mining.
1-stage object detection: **CenterNet** (anchor-free)

- Represent objects by a single point + (width, height)
- Regress other parameters such as:
  - Bounding box
  - 3D box
  - Human pose
  - ...  

Output: $64 \times 64 \times 1 \times (C+2+2)$

[Objects as Points, X. Zhou, D. Wang and P. Krahenbuhl, 2019]
State-of-the-art comparison: MS COCO

1-stage detectors

- CenterNet-HD
- CenterNet-OLA
- ExtremeNet (CVPR19)
- CornerNet (ECCV18)
- FSAF (CVPR19)
- YOLOv3 (arXiv18)

2-stage detectors

- MaskRCNN (ICCV17)
- SNI/PER (NIPS18)
- PANet (CVPR18)
- TridentNet (CVPR19)
Object detection: Transformer-based methods
DETR: Object detection with transformers

N. Carion et al., End-to-end object detection with transformers, ECCV 2020
DETR: Object detection with transformers

N. Carion et al., End-to-end object detection with transformers, ECCV 2020
DETR: Object detection with transformers

N. Carion et al., End-to-end object detection with transformers, ECCV 2020
DETR: Object detection with transformers

N. Carion et al., End-to-end object detection with transformers, ECCV 2020
DETR: Training

- Match each box proposal to ground truth
- Use Hungarian algorithm to find permutation minimizing matching loss
## DETR: Results COCO Val

<table>
<thead>
<tr>
<th>Model</th>
<th>Epochs</th>
<th>mAP</th>
<th>mAP (small)</th>
<th>mAP (medium)</th>
<th>mAP (large)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster RCNN-FPN</td>
<td>109</td>
<td>42.0</td>
<td>26.6</td>
<td>45.4</td>
<td>53.4</td>
</tr>
<tr>
<td>DETR</td>
<td>500</td>
<td>42.0</td>
<td>20.5</td>
<td>45.8</td>
<td>61.1</td>
</tr>
</tbody>
</table>

N. Carion et al., End-to-end object detection with transformers, ECCV 2020
Agenda

• 0. Intro to structured outputs

• 1. Object detection (localization)

• 2. Segmentation

• 3. Human pose estimation
Semantic segmentation

- Label each pixel in the image with a category label
- Don’t differentiate instances, only care about pixels
Semantic segmentation: sliding window

Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al., “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling,” ICML 2014
Semantic segmentation: fully convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!

Input: $3 \times H \times W$

Convolutions: $D \times H \times W$

Scores: $C \times H \times W$

Predictions: $H \times W$

Problem: convolutions at original image resolution will be very expensive ...
Semantic segmentation: fully convolutional

**Downsampling:**
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

**Upsampling:**
???

Input: $3 \times H \times W$

High-res: $D_1 \times H/2 \times W/2$

High-res: $D_1 \times H/2 \times W/2$

Predictions: $H \times W$

Med-res: $D_2 \times H/4 \times W/4$

Med-res: $D_2 \times H/4 \times W/4$

Low-res: $D_3 \times H/4 \times W/4$

---


Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/
In-network upsampling: “Unpooling”

Max Pooling
Remember which element was max!

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>6</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Input: 4 x 4

Output: 2 x 2

Max Unpooling
Use positions from pooling layer

<table>
<thead>
<tr>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Rest of the network

Input: 2 x 2

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers

Slide credit: L. Fei-Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/
Learnable upsampling

Other names: Deconvolution (bad); Upconvolution; Fractionally strided convolution; Backward strided convolution; Transpose convolution

Input  Kernel  
1. Calculate parameters $z$ and $p'$  
2. Insert $z$ zeros between the rows and columns

Output

- $s$ (stride)
- $p$ (padding)
- $z = s - 1$
- $p' = k - p - 1$
- $s' = 1$

upsampled
Learnable upsampling

Other names: Deconvolution (bad); Upconvolution; Fractionally strided convolution; Backward strided convolution; Transpose convolution
**Semantic segmentation: fully convolutional**

**Downsampling:**
Pooling, strided convolution

**Upsampling:**
Unpooling or strided transpose convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input: $3 \times H \times W$

High-res: $D_1 \times H/2 \times W/2$

Low-res: $D_3 \times H/4 \times W/4$

Med-res: $D_2 \times H/4 \times W/4$

Med-res: $D_2 \times H/4 \times W/4$

High-res: $D_1 \times H/2 \times W/2$

Predictions: $H \times W$

---


Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/
Semantic segmentation: Auto-encoder

Why is this a bad idea for segmentation?
Semantic segmentation: U-Net or “Hourglass”

Ronneberger et al., MICCAI 2015.

Newell et al., ECCV 2016.

Fig: Nushaine Ferdinand
Semantic segmentation: Segmenter

Transformer architecture for image segmentation

Input Image

Transformer Encoder

Mask Transformer

Scalar Product

Upsample and Argmax

Class Masks

Segmentation Map

R. Strudel et al., Segmenter: Transformer for Semantic Segmentation, ICCV 2021
(Object) Instance segmentation

- Differentiate instances
- Object detection + segmentation
Object detection using Fast(er) R-CNN

Mask R-CNN

Object detection and segmentation

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
1. Object detector using Faster RCNN +
2. Fully convolutional network (FCN) on region of interest (RoI)

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
Mask R-CNN

Combining loss functions

- Easy, fast to implement and train

(slow) R-CNN

Fast/er R-CNN

Mask R-CNN

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io
Mask R-CNN

Example results

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io
Mask R-CNN

Example results

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
Mask R-CNN

Example results

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
Mask R-CNN

Example results

Mask R-CNN results on CityScapes

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
Mask R-CNN

Example failures: recognition

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]
Segment Anything

Alexander Kirillov\textsuperscript{1,2,4} Eric Mintun\textsuperscript{2} Nikhila Ravi\textsuperscript{1,2} Hanzi Mao\textsuperscript{2} Chloe Rolland\textsuperscript{3} Laura Gustafson\textsuperscript{3}
Tete Xiao\textsuperscript{3} Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollar\textsuperscript{4} Ross Girshick\textsuperscript{4}
\textsuperscript{1}project lead \hspace{0.5cm} \textsuperscript{2}joint first author \hspace{0.5cm} \textsuperscript{3}equal contribution \\
\textsuperscript{4}directional lead

Meta AI Research, FAIR

(a) Task: promptable segmentation

(b) Model: Segment Anything Model (SAM)

(c) Data: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation task, a segmentation model (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a data engine for collecting SA-1B, our dataset of over 1 billion masks.

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]
Segment Anything Model (SAM)  

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

Prompting with a point  

Prompting with a dense grid of points

https://segment-anything.com/demo
Segment Anything Model (SAM)

Not semantic segmentation (no category)

Could be used for instance segmentation by integrating an object detector

Prompting with detected boxes

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]
Segment Anything Dataset (SA-1B)

- 11M images
- 1B+ masks (99.1% of masks fully automatic)
- Collected through interactive interface

3-stage annotation:
- Assisted-manual stage (+30sec/image to annotate, reduced to 14sec after 6 x retraining, 4.3M masks from 12K images)
- Semi-automatic stage (bbox for less prominent objects, up to 34sec. 5 x retraining, 5.9M masks in 180K images)
- Fully-automatic stage.

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]
Segment Anything

- Spatial distribution of object centers
- Common photographer bias
- Greater coverage of image corners in SA-1B

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]
Agenda

• 0. Intro to structured outputs
• 1. Object detection (localization)
• 2. Segmentation
• 3. Human pose estimation
2D Human pose estimation

Source: https://www.youtube.com/watch?v=2DiQUX11YaY

[Cao, Simon, Wei and Sheikh CVPR 2017]
DeepPose: Human Pose Estimation via Deep Neural Networks

Trains CNN to regress locations \((x_i, y_i)\) for each joint \(i\)

[Toshev and Szegedy, CVPR 2014]
DeepPose: Human Pose Estimation via Deep Neural Networks

Cascade regressor:
Stage $s$ improves output of the previous stage $s-1$ using higher resolution sub-image

3 stages in practice

[Toshev and Szegedy, CVPR 2014]
DeepPose: Human Pose Estimation via Deep Neural Networks

[Toshev and Szegedy, CVPR 2014]
Convolutional Pose Machines

- Regression to joint “heatmaps”: 2D gaussians around joint coordinates
- Heatmaps enable to handle spatial ambiguity

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]
Convolutional Pose Machines

- Regression to joint “heatmaps”: 2D gaussians around joint coordinates
- Heatmaps enable to handle spatial ambiguity
- Multi-stage refinement

[Image of input image and stages 1, 2, 3 with heatmaps] [Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]
Convolutional Pose Machines

- Intermediate supervision at every stage; Increasing context
Convolutional Pose Machines

Qualitative results

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]
Convolutional Pose Machines

Qualitative results

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]
Convolutional Pose Machines

Quantitative comparison

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]
Stacked Hourglass Networks

Remember U-Net

• Also heatmap regression
• Also multi-stage refinement - but full context (receptive field = entire image)

[Newell, Yang, Deng, ECCV 2016]
OpenPose: Multi-person pose estimation

Novelty: Jointly Learning Parts Detection and Parts Association

[Z. Cao, T. Simon, S. Wei, and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, CVPR 2017]
OpenPose

Part-Person Association for Multi-Person Pose Estimation

Part detections

Elbow
Wrist
OpenPose

Part-Person Association for Multi-Person Pose Estimation

Part detections

Pose
OpenPose

Figure 2. Overall pipeline. Our method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, shown in (b), and part affinity fields for parts association, shown in (c). The parsing step performs a set of bipartite matchings to associate body parts candidates (d). We finally assemble them into full body poses for all people in the image (e).

Key Idea: Encode the Part Affinity Score on the Image Plane
=> Part Affinity Fields encode direction and position
Dense pose estimation aims at mapping all human pixels of an RGB image to the 3D surface of the human body.
DensePose

Guler et al. DensePose, CVPR 2018
Human pose estimation beyond 2D keypoints

Human body analysis

Input image

2D pose

3D pose

Body parts

Body depth

Body shape
Challenges

How to model the body shape?
(a) Skeleton representation
(b) Parametric representation
(c) Point cloud representation
(d) Voxel representation

[1] Loper et al. SMPL: A Skinned Multi-Person Linear Model, SIGGRAPH Asia 2015
SMPL parametric body model: surface & joints

pose variation

shape (identity) variation

[Loper et al. 2015]
Human pose estimation beyond 2D keypoints

• A rich literature also on 3D human pose & motion estimation

VIBE [Kocabas et al. CVPR 2020]
Human pose estimation beyond 2D keypoints

- A rich literature also on 3D human pose & motion estimation

4D Humans [Goel et al. ICCV 2023]
Agenda

• 0. Intro to structured outputs
• 1. Object detection (localization)
• 2. Segmentation
• 3. Human pose estimation
Feedback welcome throughout the course (anonymous)
Can fill the form multiple times