Machine Learning for Image Analysis

Lecture 7: CNNs for Object Detection

Dagmar Kainmueller, 11.06.2024

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

MAX DELBRÜCK CENTER

Course Outline

- Introduction to Image Analysis
- Machine Learning Basics: Linear Regression, Basic Classifiers
- Neural Networks
- **Convolutional Neural Networks**
- Recurrent Neural Networks, Transformers, Diffusion
- Model Interpretability
- Probabilistic Machine Learning
- Generative Models

Convolutional Neural Networks (CNNs)

- A bit of CNN history
- CNN Layers
    - Convolutions, Pooling, Fully Connected Layers, Normalization
- Training CNNs
    - Weight initialization
    - Advanced training ingredients: Regularization: Dropout; Optimization: Adam
    - Extended training data: Augmentation, Transfer Learning
    - Sanity-checking the learning process; Hyperparameter tuning strategies
- Famous CNN architectures
- CNNs for image segmentation; CNNs for object detection

Recap: Sliding Window

Input image

Extract patch

Classify center pixel with ConvNet

cow

cow

grass

Ciresan et al, Deep neural networks segment neuronal membranes in electron microscopy, NIPS 2012
Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Credits to Fei-Fei Li, Jiajun Wu and Ruohan Gao (Stanford, Deep Learning for Computer Vision, 2022, cs231n)

Recap: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network

Downsampling:
Pooling, strided convolution

Upsampling:
Interpolation, transpose convolution

Long, Shelhamer and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al., “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

Recap: Back to full detail: The U-Net

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Recap: U-Net Summary

U-Net advantages

- Flexible use for any pixel-wise prediction task
- Competitive accuracy (given proper training, dataset and training time)
- Doesn’t contain any fully connected layers
- Faster than sliding-window
- Powerful segmentation tool also in scenarios with limited data
- Achieves best performance in various biomedical applications

U-Net disadvantages

- Larger images need lots of GPU memory
- Takes significant amount of time to train
- Pre-trained models not widely available (but coming up)

Recap: CNNs for Instance Segmentation

Assign one instance label to each pixel in an image

Recap: CNN-based “Proposal-free” Instance Segmentation

Lead on ISBI 2012 EM Segmentation Challenge in 2015:

U-Net for boundary pixel classification

Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. MICCAI 2015 (pp. 234-241).

Recap: CNN-based Proposal-free Instance Segmentation

Instances with Background:

Two labels: Fg, Bg

Three labels: Fg, Bg, Boundary

Problem:

Tiny change in prediction / loss can cause huge (topological) change in segmentation

Recap: CNN-based Proposal-free Instance Segmentation

Hirsch, Kainmueller: *An Auxiliary Task for Learning Nuclei Segmentation in 3D Microscopy Images.* MIDL 2020

Recap: CNN-based Proposal-free Instance Segmentation

a, Procedure for transforming manually annotated masks into a vector flow representation that can be predicted by a neural network. A simulated diffusion process starting from the center of the mask is used to derive spatial gradients that point toward the center of the cell, potentially indirectly around corners. The *x* and *y* gradients are combined in a single direction from 0° to 360°. **b**, Example spatial gradients for cells from the training dataset. **c**, A neural network is trained to predict the horizontal and vertical gradients, as well as whether a pixel belongs to any cell. The three predicted maps are combined into a gradient vector field. **d**, The details of the neural network that contains a standard backbone U-Net[3] to downsample and then upsample the feature maps, with skip connections between layers of the same size and global skip connections from the image styles, computed at the lowest resolution, to all successive computations. **e**, At test time, the predicted gradient vector fields are used to construct a dynamical system with fixed points whose basins of attraction represent the predicted masks. Informally, every pixel 'tracks the gradients' toward their eventual fixed point. **f**, All the pixels that converge to the same fixed point are assigned to the same mask.

Convolutional Neural Networks (CNNs)

- A bit of CNN history
- CNN Layers
    - Convolutions, Pooling, Fully Connected Layers, Normalization
- Training CNNs
    - Weight initialization
    - Advanced training ingredients: Regularization: Dropout; Optimization: Adam
    - Extended training data: Augmentation, Transfer Learning
    - Sanity-checking the learning process; Hyperparameter tuning strategies
- Famous CNN architectures
- CNNs for image segmentation; CNNs for object detection

Object Detection: Task Definition

Input: Single RGB Image

Output: A <u>set</u> of detected objects;
For each object predict:

1. Category label (from fixed, known set of categories)
2. Bounding box (four numbers: x, y, width, height)

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Object Detection: Challenges

- **Multiple outputs**: Need to output variable numbers of objects per image
- **Multiple types of output**: Need to predict ”what” (category label) as well as “where” (bounding box)
- **Large images**: Classification works at 224x224; need higher resolution for detection, often ~800x600

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Bounding Boxes

Bounding boxes are typically *axis-aligned*

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Bounding Boxes

Bounding boxes are typically *axis-aligned*

Oriented boxes are much less common

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Object Detection: Modal vs Amodal Boxes

Bounding boxes (usually) cover only the visible portion of the object

Zhu et al, ”Semantic Amodal Segmentation”, CVPR 2017

This image is CC0 Public Domain

Detecting a single object

This image is CC0 public domain

Vector:
4096

Treat localization as a regression problem!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting a single object "What"

Correct label:
Cat

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Vector:
4096

This image is CC0 public domain

Treat localization as a regression problem!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting a single object

"What"

Correct label:
Cat

This image is CC0 public domain

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

"Where"

Treat localization as a regression problem!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting a single object

This image is CC0 public domain

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

"What"

Correct label:
Cat

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

"Where"

L2 Loss

Correct box:
(x', y', w', h')

Weighted Sum

Loss

Multitask Loss

$$L = L_{cls} + \lambda L_{reg}$$

Treat localization as a regression problem!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting a single object

This image is CC0 public domain

Often pretrained on ImageNet (Transfer learning)

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Multitask Loss

Weighted Sum → **Loss**

$$L = L_{cls} + \lambda L_{reg}$$

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

"What"

"Where"

Treat localization as a regression problem!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting a single object

"What"

Correct label:
Cat

Often pretrained on ImageNet (Transfer learning)

This image is CC0 public domain

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Multitask Loss

Weighted Sum

Loss

$$L = L_{cls} + \lambda L_{reg}$$

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

"Where"

Treat localization as a regression problem!

Problem: Images can have more than one object!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects

Need different numbers of outputs per image

CAT: (x, y, w, h)

4 numbers

DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 numbers

DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Duck image is free to use under the Pixabay license

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Dog? NO
Cat? NO
Background? YES

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Dog? YES
Cat? NO
Background? NO

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Dog? YES
Cat? NO
Background? NO

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Dog? NO
Cat? YES
Background? NO

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

$$=\frac{H(H+1)}{2}\frac{W(W+1)}{2}$$

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

800 x 600 image has ~58M boxes! No way we can evaluate them all

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

$$=\frac{H(H+1)}{2}\frac{W(W+1)}{2}$$

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposals

- Find a small set of boxes that are likely to cover all objects
- Often based on heuristics: e.g. look for “blob-like” image regions
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU

Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Region-Based CNN

Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN: Region-Based CNN

Input image

Regions of Interest (RoI) from a proposal method (~2k)

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Region-Based CNN

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Region-Based CNN

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Region-Based CNN

Classify each region

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Region-Based CNN

Classify each region

Bounding box regression:
Predict "transform" to correct the RoI: 4 numbers (t_x, t_y, t_h, t_w)

Forward each region through ConvNet

Warped image regions (224x224)

Regions of Interest (RoI) from a proposal method (~2k)

Input image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Box Regression

Consider a region proposal with center (p_x, p_y), width p_w, height p_h

Model predicts a transform (t_x, t_y, t_w, t_h) to correct the region proposal

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Box Regression

Consider a region proposal with center (p_x, p_y), width p_w, height p_h

Model predicts a transform (t_x, t_y, t_w, t_h) to correct the region proposal

The output box is defined by:

$$b_x = p_x + p_w t_x$$
$$b_y = p_y + p_h t_y$$

Shift center by amount relative to proposal size

$$b_w = p_w \exp(t_w)$$
$$b_h = p_h \exp(t_h)$$

Scale proposal; exp ensures that scaling factor is > 0

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Box Regression

Consider a region proposal with center (p_x, p_y), width p_w, height p_h

Model predicts a transform (t_x, t_y, t_w, t_h) to correct the region proposal

The output box is:

$$b_x = p_x + p_w t_x$$
$$b_y = p_y + p_h t_y$$
$$b_w = p_w \exp(t_w)$$
$$b_h = p_h \exp(t_h)$$

When transform is 0, output = proposal

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Box Regression

Consider a region proposal with center (p_x, p_y), width p_w, height p_h

Model predicts a transform (t_x, t_y, t_w, t_h) to correct the region proposal

The output box is:

$$b_x = p_x + p_w t_x$$
$$b_y = p_y + p_h t_y$$
$$b_w = p_w \exp(t_w)$$
$$b_h = p_h \exp(t_h)$$

Scale / Translation invariance: Transform encodes *relative* difference between proposal and output; important since CNN doesn't see absolute size or position after cropping

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN: Box Regression

Consider a region proposal with center (p_x, p_y), width p_w, height p_h

Model predicts a transform (t_x, t_y, t_w, t_h) to correct the region proposal

The output box is:

$$b_x = p_x + p_w t_x$$
$$b_y = p_y + p_h t_y$$
$$b_w = p_w \exp(t_w)$$
$$b_h = p_h \exp(t_h)$$

Given proposal and target output, we can solve for the transform the network should output:

$$t_x = (b_x - p_x)/p_w$$
$$t_y = (b_y - p_y)/p_h$$
$$t_w = \log(b_w/p_w)$$
$$t_h = \log(b_h/p_h)$$

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Training

Input Image

Ground-Truth boxes

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Training

Input Image

Ground-Truth boxes

Region Proposals

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Training

Input Image

GT Boxes | Positive

Neutral | Negative

Intersection over union (IoU):

Categorize each region proposal as positive, negative, or neutral based on overlap with ground-truth boxes:

Positive: > 0.5 IoU with a GT box

Negative: < 0.3 IoU with all GT boxes

Neutral: between 0.3 and 0.5 IoU with GT boxes

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Training

Input Image

GT Boxes

Positive

Neutral

Negative

Crop pixels from each positive and negative proposal, resize to 224 x 224

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Training

Run each region through CNN
Positive regions: predict class and transform
Negative regions: just predict class

Input Image

GT Boxes | Positive | Neutral | Negative

Class target: Dog
Box target: →

Class target: Cat
Box target: →

Class target: Dog
Box target: →

Class target: Background
Box target: None

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

R-CNN Test-Time

Input Image

Region Proposals

1. Run proposal method
2. Run CNN on each proposal to get class scores, transforms
3. Threshold class scores to get a set of detections

2 problems:

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Overlapping Boxes

Problem: Object detectors often output many overlapping detections:

Puppy image is CC0 Public Domain

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Overlapping Boxes: Non-Max Suppression (NMS)

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

Puppy image is CC0 Public Domain

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Overlapping Boxes: Non-Max Suppression (NMS)

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.78**
IoU(■, ■) = 0.05
IoU(■, ■) = 0.07

Puppy image is CC0 Public Domain

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Overlapping Boxes: Non-Max Suppression (NMS)

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.74**

Puppy image is CC0 Public Domain

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Overlapping Boxes: Non-Max Suppression (NMS)

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

Puppy image is CC0 Public Domain

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

“Slow” R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Input image

Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

“Slow” R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

"Backbone" network: AlexNet, VGG, ResNet, etc

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

"Slow" R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Regions of Interest (RoIs) from a proposal method

Image features

"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Input image

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

"Slow" R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Regions of Interest (RoIs) from a proposal method

"Backbone" network: AlexNet, VGG, ResNet, etc

Crop + Resize features

Image features

Run whole image through ConvNet

Input image

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

"Slow" R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Regions of Interest (RoIs) from a proposal method

Per-Region Network

Crop + Resize features

Image features

"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Input image

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

"Slow" R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

"Slow" R-CNN

Process each region independently

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Category and box transform per region

Per-Region network is relatively lightweight

Per-Region Network

Regions of Interest (RoIs) from a proposal method

Crop + Resize features

Image features

"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Most of the computation happens in backbone network; this saves work for overlapping region proposals

Input image

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Category and box transform per region

Per-Region Network

Crop + Resize features

Image features

Run whole image through ConvNet

Input image

Regions of Interest (RoIs) from a proposal method

"Backbone" network: AlexNet, VGG, ResNet, etc

Example:
For ResNet, last stage is used as per-region network; the rest of the network is used as backbone

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN

Category and box transform per region

Per-Region Network

Regions of Interest (RoIs) from a proposal method

Crop + Resize features

How to crop features?

Image features

"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Input image

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Cropping Features: RoI Pool

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the box of a fixed size (2x2 in this example, 7x7 or 14x14 in practice)

Girshick, "Fast R-CNN", ICCV 2015.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Cropping Features: RoI Pool

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the box of a fixed size (2x2 in this example, 7x7 or 14x14 in practice)

Girshick, "Fast R-CNN", ICCV 2015.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Cropping Features: RoI Pool

Divide into 2x2 grid of (roughly) equal subregions

"Snap" to grid cells

Project proposal onto features

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 20 x 15)

Want features for the box of a fixed size (2x2 in this example, 7x7 or 14x14 in practice)

Girshick, "Fast R-CNN", ICCV 2015.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Cropping Features: RoI Pool

Input Image
(e.g. 3 x 640 x 480)

Project proposal onto features

CNN

Image features
(e.g. 512 x 20 x 15)

"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions

Max-pool within each subregion

Region features
(here 512 x 2 x 2;
In practice 512x7x7)

Region features always the same size even if input regions have different sizes!

Girshick, "Fast R-CNN", ICCV 2015.

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN vs “Slow” R-CNN

Fast R-CNN:

“Slow” R-CNN:

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN vs “Slow” R-CNN

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN vs “Slow” R-CNN

Problem: Runtime dominated by region proposals!

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast R-CNN vs “Slow” R-CNN

Problem: Runtime dominated by region proposals!

Recall: Region proposals computed by heuristic ”Selective Search” algorithm on CPU -- let’s learn them with a CNN instead!

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast<u>er</u> R-CNN: Learnable Region Proposals

Insert **Region Proposal Network (RPN)** to predict proposals from features

Otherwise same as Fast R-CNN: Crop features for each proposal, classify each one

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Run backbone CNN to get
features aligned to input image

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Predict object vs not object scores for all anchors with a conv layer (512 input filters, 2 output filters)

Anchor is object?
1 x 5 x 6

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

For positive anchors, also predict a transform that converting the anchor to the GT box (like R-CNN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Anchor is object?
1 x 5 x 6

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

For positive anchors, also predict a transform that converting the anchor to the GT box (like R-CNN)
Predict transforms with conv

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Anchor is object?
K x 5 x 6

Anchor transforms
4K x 5 x 6

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 5 x 6)

Conv

Anchor is object?
K x 5 x 6

Anchor transforms
4K x 5 x 6

During training, supervised positive / negative anchors and box transforms like R-CNN

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 5 x 6)

Conv

Anchor is object?
K x 5 x 6

Anchor transforms
4K x 5 x 6

Positive anchors: >= 0.7 IoU with some GT box (plus highest IoU to each GT)

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Negative anchors: < 0.3 IoU with all GT boxes. Don't supervised transforms for negative boxes.

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 5 x 6)

Conv

Anchor is object?
K x 5 x 6

Anchor transforms
4K x 5 x 6

Neutral anchors: between 0.3 and 0.7 IoU with all GT boxes; ignored during training

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

At test-time, sort all K*5*6 boxes by their positive score, take top 300 as our region proposals

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast<u>er</u> R-CNN: Learnable Region Proposals

Jointly train with 4 losses:

1. **RPN classification**: anchor box is object / not an object
2. **RPN regression**: predict transform from anchor box to proposal box
3. **Object classification**: classify proposals as background / object class
4. **Object regression**: predict transform from proposal box to object box

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Fast<u>er</u> R-CNN: Learnable Region Proposals

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Faster R-CNN: Learnable Region Proposals

Faster R-CNN is a
Two-stage object detector

First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale

We need to detect objects of many different scales.
How to improve *scale invariance* of the detector?

This image is free for commercial use under the Pixabay license

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Image Pyramid

Classic idea: build an *image pyramid* by resizing the image to different scales, then process each image scale independently.

Lin et al, “Feature Pyramid Networks for Object Detection”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Efficient multiscale features where all levels benefit from the whole backbone! Widely used in practice

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Dealing with Scale: Feature Pyramid Network

Add *top down connections* that feed information from high level features back down to lower level features

Efficient multiscale features where all levels benefit from the whole backbone! Widely used in practice

Faster R-CNN with RPN: Detector at each level gets its own RPN to produce proposals; proposals from all levels route to a shared second stage

Lin et al, "Feature Pyramid Networks for Object Detection", ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Faster R-CNN: Learnable Region Proposals

Question: Do we really need the second stage?

Faster R-CNN is a
Two-stage object detector

First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: RetinaNet

Similar to RPN – but rather than classify anchors as object/no object, directly predict object category (among C categories) or background

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Lin et al, “Focal Loss for Dense Object Detection”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: RetinaNet

In practice, RetinaNet also uses Feature Pyramid Network to handle multiscale

Lin et al, “Focal Loss for Dense Object Detection”, ICCV 2017

Figure credit: Lin et al, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

YOLO: Training

1 - pedestrian

2 - car

3 - motorcycle

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

3 × 3 × 16

y is $\underbrace{3 \times 3}_{19 \times 19 \times 16 \;\; 19 \times 19 \times 40} \times \underset{\#\text{anchors}}{2} \times \underset{5 + \#\text{classes}}{8}$

100 × 100 × 3 → ConvNet →

3 × 3 × 16

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

Andrew Ng

YOLO: Making Predictions

Andrew Ng

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.

Single-Stage Detectors: FCOS

"Anchor-free" detector

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

Classify points as positive if they fall into a GT box, or negative if they don't

Train independent per-category logistic regressors

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

→ Class scores
C x 5 x 6

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

For positive points, also regress distance to left, right, top, and bottom of ground-truth box (with L2 loss)

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

For positive points, also regress distance to left, right, top, and bottom of ground-truth box (with L2 loss)

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

Finally, predict "centerness" for all positive points (using logistic regression loss)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 5 x 6)

CNN

Class scores
C x 5 x 6

Box edges
4 x 5 x 6

Centerness
1 x 5 x 6

$$centerness = \sqrt{\frac{\min(L,R)}{\max(L,R)} \cdot \frac{\min(T,B)}{\max(T,B)}}$$

Ranges from 1 at box center to 0 at box edge

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

Test-time: predicted "confidence" for the box from each point is product of its class score and centerness

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

CNN

Image features
(e.g. 512 x 5 x 6)

CNN

Class scores
C x 5 x 6

Box edges
4 x 5 x 6

Centerness
1 x 5 x 6

$$centerness = \sqrt{\frac{\min(L,R)}{\max(L,R)} \cdot \frac{\min(T,B)}{\max(T,B)}}$$

Ranges from 1 at box center to 0 at box edge

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Single-Stage Detectors: FCOS

"Anchor-free" detector

FCOS also uses a Feature Pyramid Network with heads shared across stages

Tian et al, "FCOS: Fully Convolutional One-Stage Object Detection", ICCV 2019

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Summary

"Slow" R-CNN: Run CNN independently for each region

Fast R-CNN: Apply differentiable cropping to shared image features

Faster R-CNN: Compute proposals with CNN

Single-Stage: Fully convolutional detector

With anchors: RetinaNet
Anchor-Free: FCOS

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Object Detection: Faster R-CNN

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NeurIPS 2015

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Instance Segmentation: **Mask R-CNN**

He et al, “Mask R-CNN”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN

He et al, “Mask R-CNN”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Example Training Targets

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Example Training Targets

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Example Training Targets

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Example Training Targets

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Very Good Results!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Keypoint Estimation

Keypoint estimation

He et al, “Mask R-CNN”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Mask R-CNN: Keypoints

Classification Scores: C
Box coordinates (per class): 4 * C
Segmentation mask: C x 28 x 28

CNN
+RPN

RoI Align

256 x 14 x 14

Conv…

One mask for each of
the K different keypoints

Keypoint masks:
K x 56 x 56

Ground-truth has one “pixel” turned on per keypoint. Train with softmax loss

He et al, “Mask R-CNN”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Joint Instance Segmentation and Pose Estimation

He et al, “Mask R-CNN”, ICCV 2017

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2022

Course Outline

- Introduction to Image Analysis
- Machine Learning Basics: Linear Regression, Basic Classifiers
- Neural Networks
- Convolutional Networks
- **Recurrent Neural Networks, Transformers, Diffusion**
- Model Interpretability
- Probabilistic Machine Learning
- Generative Models

StarDist

Fig. 1: *(a)* Potential segmentation errors for images with crowded nuclei: Merging of touching cells (upper right) or suppression of valid cell instances due to large overlap of bounding box localization (lower right). *(b)* The proposed StarDist method predicts object probabilities $d_{i,j}$ and star-convex polygons parameterized by the radial distances $r^k_{i,j}$. *(c)* We densely predict $r^k_{i,j}$ and $d_{i,j}$ using a simple U-Net architecture [15] and then select the final instances via non-maximum suppression (NMS).

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G.: Cell Detection with Star-convex Polygons. In *MICCAI 2018* (pp. 265-273).