Mask R-CNN

MLRG 2021 @ UBC
Victor Sanches Portella
The task: **instance segmentation**

**Classification**
- **CAT**
  - No spatial extent

**Semantic Segmentation**
- **GRASS, CAT, TREE, SKY**
  - No objects, just pixels

**Object Detection**
- **DOG, DOG, CAT**
  - Multiple Object

**Instance Segmentation**
- **DOG, DOG, CAT**

[Source](http://cs231n.stanford.edu/slides/2020/lecture_12.pdf)
The full story

Looking **only** at the Mask R-CNN paper is not helpful, looks like magics

Looking at the **series** of work leading-up to Mask R-CNN is more interesting

- Region CNN (R-CNN)
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN

**Object detection**

**Instance Segmentation**
Region CNN

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik
Object classification vs Object detection

Are the results from image classification transferable to image detection?

Fixed # of outputs VS Varying # of outputs

Given a region/box of interest, we could run classification

How to propose regions?
Selective Search

In the original paper, it proposes around **2k regions per image**

For each region, we can run classification (with a CNN)!

R-CNN: Regions with CNN features

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Improving bounding boxes

Proposed boxes may not be well-fitted to the object

We can tighten these boxes using linear regression (*details skipped*)
Three models to be trained

SVM vs Softmax

Features are extracted for each RoI

SLOW

Fast R-CNN

Ross Girshick
Key insights to speed-up R-CNN

- Extract features first, select regions of interest later
  - A lot of proposed regions for a image overlap
  - Use RoIPool to share features!

- One network to rule them all
  - Instead of stacking models, make one network to do everything
Region of Interest (RoI) Pooling
Region of Interest (RoI) Pooling
Region of Interest (RoI) Pooling
Putting everything together into a NN

https://arxiv.org/abs/1504.08083
Performance gains

**Training time (Hours)**

- R-CNN: 84 hours
- SPP-Net: 25.5 hours
- Fast R-CNN: 8.75 hours

**Test time (seconds)**

- R-CNN: 49 seconds (Including Region proposal) 47 seconds (Excluding Region proposal)
- SPP-Net: 4.3 seconds (Including Region proposal) 2.3 seconds (Excluding Region proposal)
- Fast R-CNN: 2.3 seconds (Including Region proposal) 0.32 seconds (Excluding Region proposal)

**Remark:** The efficiency bottleneck of Fast R-CNN is region proposal via Selective Search
Faster R-CNN

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun
Region proposal in Fast R-CNN

- Selective Search became the main bottleneck for prediction

- Rol selection depends on features computed by a CNN

- **Idea:** Pass features through yet another NN, the Region Proposal Network
Region Proposal Network (RPN)

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

CNN

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

Region Proposal Network (RPN)

Slide an anchor box to generate candidates

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

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

Region Proposal Network (RPN)

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

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

Slide an anchor box to generate candidates

Is box an object?

Box warping

Region Proposal Network (RPN)

Slide an anchor box to generate candidates

Is box an object?

Box warping

Use K different anchor boxes at each point!

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

CNN

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

Conv

Region Proposal Network (RPN)

2k scores

4k coordinates

cls layer

reg layer

256-d

k anchor boxes

sliding window

conv feature map

How to train Faster R-CNN?

- **Option 1:** Alternating training (used in the paper)
  - Train RPN, then train Fast R-CNN, then fix the shared CNN, train RPN again, and then train Fast R-CNN again

- **Option 2:** Train the whole network simultaneously
  - By ignoring the derivative of the box coordinates, one can (approximately) train the whole network at once. Apparently it works without affecting efficiency by much.
Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick
Adapt Faster R-CNN to do segmentation?

- Can we in some way adapt Faster R-CNN to do segmentation?

- **Idea:** For each RoI box, have a separate network to predict pixel mask

- Add this as a branch to Faster R-CNN and perform end-to-end training

- Some tweaks are needed to the Faster R-CNN architecture
Classification loss

Bounding-box regression loss

Region Proposal Network

Proposals

Feature map

CNN

Roi pooling

Mask prediction NN

Alignment problem
Rol Pool vs Rol Align

Rol Pool

Truncation
+ MaxPool
Rol Pool vs Rol Align

Rol Align

Bilinear Interpolation
+ MaxPool
Or Avg
Mask prediction branch

- Fully convolutional network (2 or 4 layers depending on the backbone)

- Outputs, **for each class**, a small binary mask (14x14 ou 28x28)
  - In the end uses only one of these masks depending on the class prediction

- Mask loss is given by cross-entropy

- Upsampling technique of the mask not clearly stated (I think)
Network architecture

Segmentation examples

Using Mask R-CNN for pose estimation

- Task: for each region, predict $K$ keypoints types (left shoulder, right elbow, etc.)
- Each keypoint is represented by a 1-hot bitmap
- Cross-entropy loss
Using Mask R-CNN for pose estimation