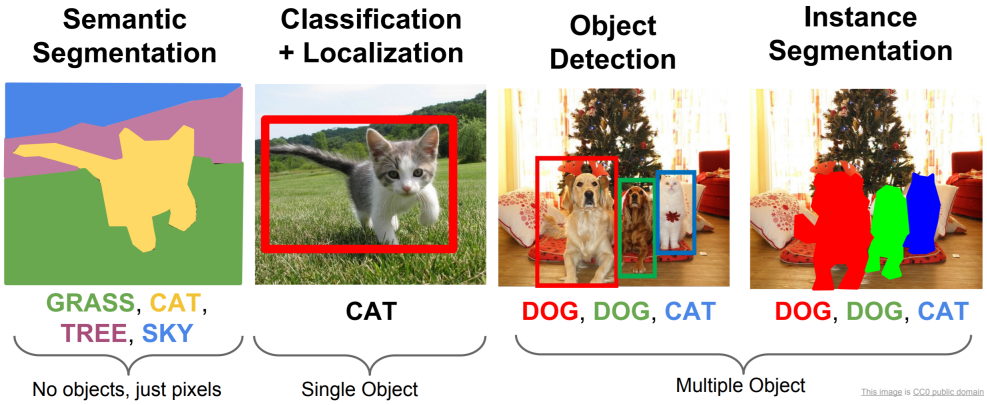
Instance Segmentation

Nikhil Sardana

January 2018

1 Introduction

Instance Segmentation is one of the most difficult image-based computer vision tasks. It combines elements of semantic segmentation (pixel-level classification) and object detection (instance recognition). Essentially, at every pixel, we wish to classify not only the type of object (or background) the pixel is part of, but also determine which instance the pixel is part of.



Not unsurprisingly, instance segmentation networks rely heavily on existing object detection networks. This lecture will cover one particular instance segmentation network, called Mask R-CNN. Mask R-CNN modifies the Faster R-CNN architecture and adapts it for instance segmentation with minimal overhead. Mask R-CNN is the current leader in instance segmentation performance.

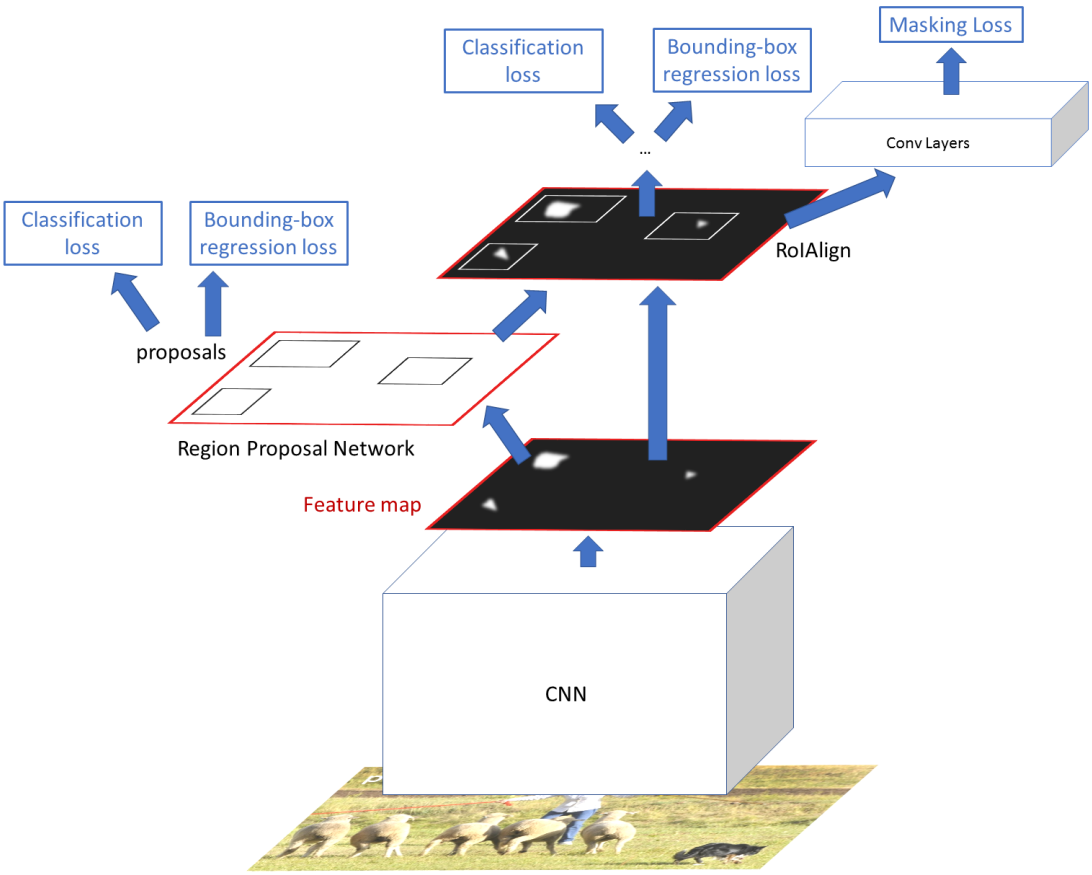
This lecture is designed for students with understanding of object detection networks. If you are unfamiliar with R-CNN, Fast R-CNN, or Faster R-CNN, read our lecture "Object Detection" before proceeding.

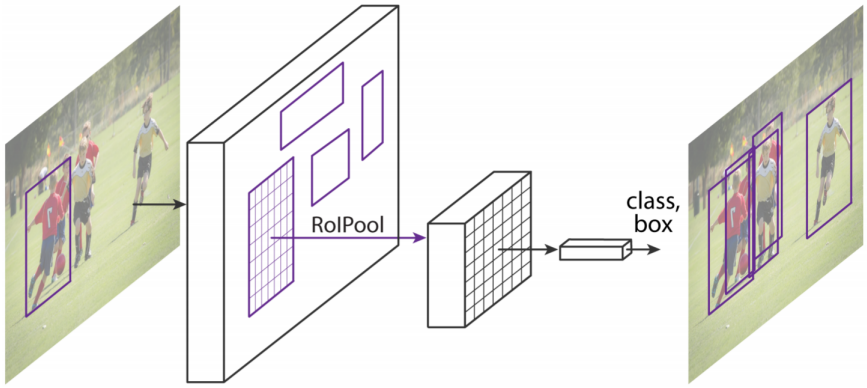
2 Mask R-CNN

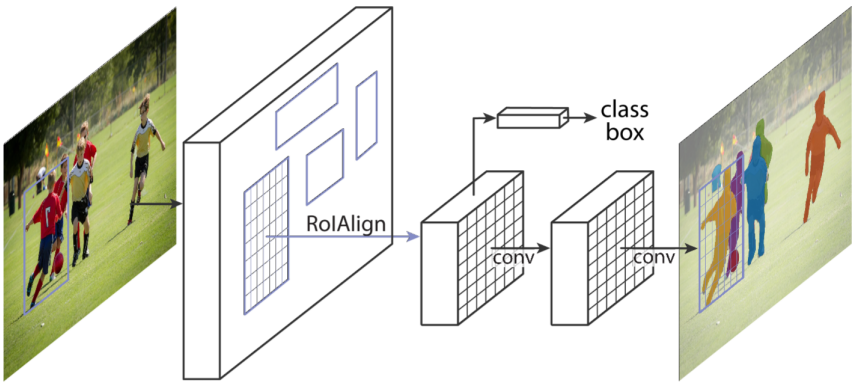
Mask R-CNN is a modification of the Faster R-CNN architecture. The authors noticed that a previous "fully convolutional instance segmentation" (FCIS) solutions, which performed segmentation, classification, and bounding-box regression simultaneously, although they ran fast, exhibited low segmentation accuracy, especially on overlapping objects. Mask R-CNN therefore takes a different approach, *decoupling* segmentation from classification and bounding-box regression. Mask R-CNN thus adds a separate mask "head" to the Faster R-CNN network. This is shown in the diagram below.

The mask "head" is simply a small fully convolutional network that outputs an $m \times m$ mask for each region proposal. We use a fully convolutional network rather than fully connected layers so we do not lose spatial information. A fully convolutional solution requires fewer parameters than previous fully connected solutions while simultaneously increasing accuracy.

The two diagrams at the bottom of Page 2 show a different visualization of Faster R-CNN and Mask R-CNN. Besides the additional "mask" head of Mask R-CNN, you may have noticed RoIAlign replacing RoI Pooling. We will cover this in the next section.


Classification loss
Bounding-box regression loss
Masking Loss
Conv Layers
...
Classification loss
Bounding-box regression loss
RoIAlign
proposals
Region Proposal Network
Feature map
CNN


RoIPool
class, box


RoIAlign
class box
conv
conv

2.1 RoIAlign

If you are unfamiliar or have forgotten RoI Pooling, please refer back to Section 3.2 in our lecture "Object Detection". RoAlign is simply a more precise version of RoI Pooling.

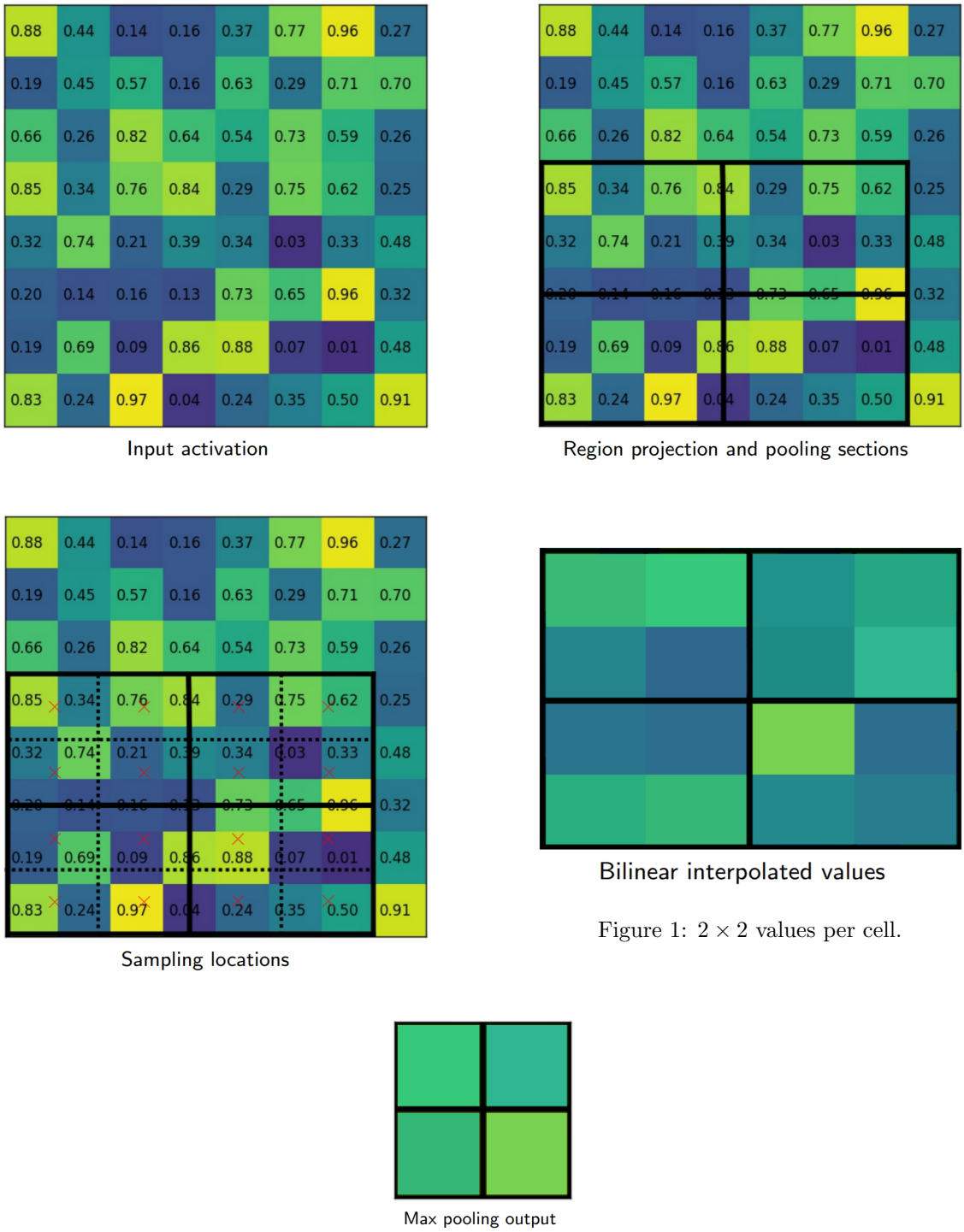


Figure 1: 2×2 values per cell.

Simply put, if an $N \times N$ output is desired, the proposed region (black rectangle in the upper-right image) is divided into an $N \times N$ grid. Unlike RoI Pooling, these regions will contain the exact same number of pixels, so we will often have fractional pixels. From each grid cell, we sample four regions as shown by the red $\times$ marks in the third image. We then subdivide each grid cell into four subcells, each centered on

an $\times$. We perform bilinear interpolation to get a single value for each subregion, or four values for each cell. These values are shown in the fourth image. Finally, we perform a simple max pooling on the bilinear interpolated values, taking the maximum value per cell to reach an $N \times N$ output. This output is then passed through the fully connected layers for bounding-box regression and classification, and through the small Fully Convolutional Network (FCN) that makes up our masking head.

Of course, you should have one question remaining: What exactly is bilinear interpolation?

2.1.1 Bilinear Interpolation

Bilinear interpolation is fairly trivial. It is best understood visually. Simply put, the bilinearly interpolated value at the black spot is the sum of the values of each of the four colors multiplied by the areas of their respective rectangles, divided by the total area.

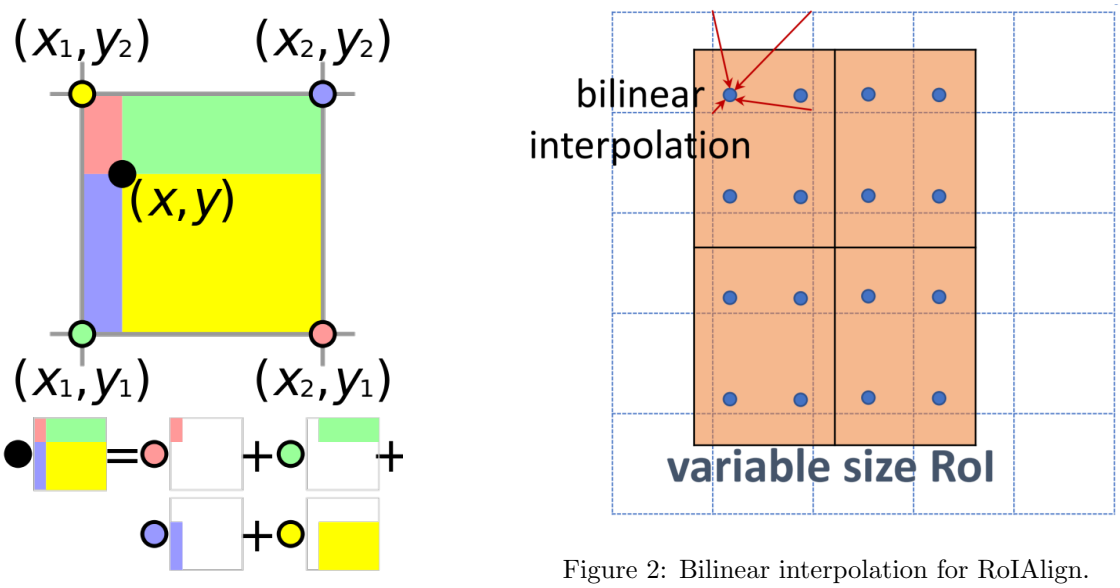


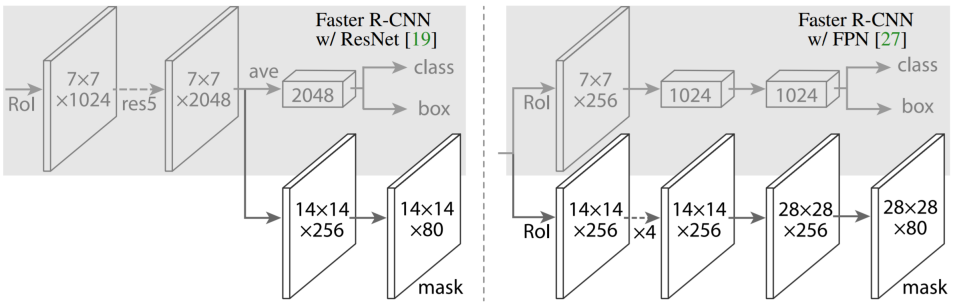
Figure 2: Bilinear interpolation for RoIAlign.

Note how in the figure on the left, the red pixel value corresponds to the smaller area opposite the pixel. This is because closer pixels (like the yellow one) have greater weighting. The figure on the right makes it clear how bilinear interpolation is implementaed in RoIAlign. At each blue dot (represented with a red $\times$ in the figures on the previous page), we take the closest 4 pixel values and multiply them by the respective areas.

And that's all RoIAlign is. It achieves the same goal as RoI Pooling, which is to take a region of any shape and create a fixed output. However, because we are using fractional pixels, we can get much better alignment. This simple change resulted in considerable accuracy improvements for Mask R-CNN.

2.2 Mask Head

Depending on the network backbone, the mask head differs for Mask R-CNN. Below is a look at the two different heads. Both are trivial FCNs.



In the diagram above, FPN standds for "Feature Pyramid Network". You should already be familiar with ResNet.

There are a few important things to know about this mask head. First, like we said earlier, our output is an $m \times m$ mask. However, the authors found it beneficial to have binary masks. In other words, we predict K $m \times m$ masks for each RoI, where K is the number of classes. One mask per class. Thus, the mask branch has a Km^2-dimensional output for each region of interest.

Our loss function is now different. Previously, we had $L = L_{boundingbox} + L_{classification}$. We've covered the details of classification and bounding-box loss in our object detection, for both the region proposal network (RPN) and the network itself. For Mask R-CNN, we add another loss, L_{mask}. For some region r, if the ground truth class is k, we apply a per-pixel sigmoid on *only* the kth mask. This allows us to define L_{mask} as the average binary cross-entropy loss. Thus, the masks for classes that don't correspond to the ground truth aren't calculated. (remember, we have one mask per class for every region).

By computing one mask per class, we are decoupling classification and segmentation. We simply don't care what class the object is when we segment it. Previous practices, like FCNs for semantic segmentation, use multi-class cross-entropy losses and per-pixel softmax. These allows for competition between classes, which Mask R-CNN eliminates.

2.3 Training and Testing

When training, Mask R-CNN shares similarities with its object detection cousins. Hyperparameters were set to the same values. Positive RoIs have IoU of at least 0.5 with the ground truth box. In addition, L_{mask} is defined only on positive RoIs. The mask target is the intersection between an RoI and its associated ground-truth mask.

At test time, after non-maximum supression is applied, the masking branch is applied on only the top 100 RoIs. If an region was classified into class k, we simply choose the kth mask. The mask is then resized to the size of the region of interest. By reducing segmentation computation to only 100 regions, we dramatically decrease the amount of overhead. In fact, Mask R-CNN runs at 5 fps, compared to Faster R-CNN's 7 fps.

2.4 Model Performance

The Mask R-CNN paper not only provides evidence that their model outperforms all previous models, but also conducted various ablation experiments to show that RoIAlign, segmentation decoupling, and fully convolutional mask heads each individually improve accuracy. The results are shown in the tables below.

	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*

	mask branch	AP	AP_{50}	AP_{75}
MLP	fc: 1024→1024→80·28^2	31.5	53.7	32.8
MLP	fc: 1024→1024→1024→80·28^2	31.5	54.0	32.6
FCN	conv: 256→256→256→256→256→80	**33.6**	**55.2**	**35.3**

In addition, Mask R-CNN performs better with a deeper backbone CNN. However, it should be noted that the 5 fps speed was achieved using the shallow ResNet-50 network as a backbone.

net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

The results on the COCO and Cityscapes benchmarks are shown below. Mask R-CNN performs with state-of-the-art accuracy on both.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Figure 3: COCO results.

	training data	AP [val]	AP	AP_{50}	person	rider	car	truck	bus	train	mcycle	bicycle
InstanceCut [23]	fine + coarse	15.8	13.0	27.9	10.0	8.0	23.7	14.0	19.5	15.2	9.3	4.7
DWT [4]	fine	19.8	15.6	30.0	15.1	11.7	32.9	17.1	20.4	15.0	7.9	4.9
SAIS [17]	fine	-	17.4	36.7	14.6	12.9	35.7	16.0	23.2	19.0	10.3	7.8
DIN [3]	fine + coarse	-	20.0	38.8	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1
Mask R-CNN	fine	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
Mask R-CNN	fine + COCO	**36.4**	**32.0**	**58.1**	**34.8**	**27.0**	**49.1**	**30.1**	**40.9**	**30.9**	**24.1**	**18.7**

Figure 4: Cityscapes results.

Some segmentation examples from the two benchmarks are provided below. Segmentation examples from COCO are on the left; Cityscapes is on the right.

Mask R-CNN can also be sued for human pose estimation. We refer readers to the Mask R-CNN paper.

3 Conclusion

Mask R-CNN leaps ahead of the competition in terms of pure instance segmentation performance. However, no current instance segmentation method can achieve great results while operating in real-time (60 FPS). In the future, look for networks which dramatically improve segmentation speed as well as accuracy.