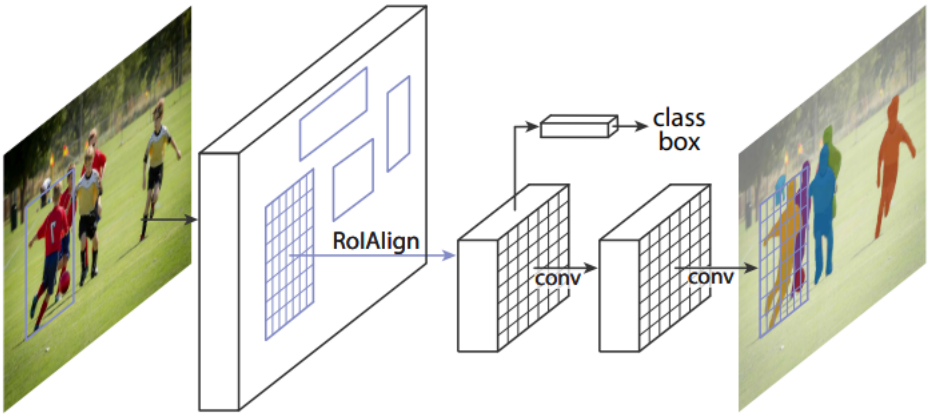
Mask RCNN

- This method has been driven by powerful baseline systems, such as the Fast/Faster RCNN [9, 29] and Fully Convolutional Network (FCN) [24] frameworks for object detection and semantic segmentation, respectively.
- Our goal in this work is to develop a comparably enabling framework for instance segmentation.
- The latest research papers tend to give results for the COCO dataset only. In COCO mAP, a 101-point interpolated AP definition is used in the calculation. For COCO, AP is the average over multiple IoUs (the minimum IoU to consider a positive match). **AP@[.5:.95]** corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05. For the COCO competition, AP is the average over 10 IoU levels on 80 categories (AP@[.50:.05:.95]: start from 0.5 to 0.95 with a step size of 0.05).

- <u>Instance Segmentation</u>

  It requires correctly detecting all objects in an image while precisely segmenting each instance. It, therefore, combines elements from the classical computer vision tasks of object detection, where the goal is to classify individual objects and localize each using a bounding box, and semantic segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.



- Mask R-CNN, extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression.
- The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner.
- Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs. Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation

- Need for developing Mask RCNN

  Faster R-CNN was not designed for pixel-to-pixel alignment between network inputs and outputs. This is most evident in how RoIPool, the de facto core operation for attending to instances, performs coarse spatial quantization for feature extraction. To fix the misalignment, we propose a simple, quantization-free layer, called RoIAlign, that faithfully preserves exact spatial locations

- Impact of Mask RCNN

  - ✓ It improves mask accuracy by a relative 10% to 50%, showing bigger gains under stricter localization metrics
  - ✓ We found it essential to decouple mask and class prediction: we predict a binary mask for each class independently, without competition among classes, and rely on the network's RoI classification branch to predict the category. (In contrast, FCNs usually perform per-pixel multi-class categorization, which couples segmentation and classification, and based on our experiments works poorly for instance segmentation)

Evolution of Mask RCNN

- The Region-based CNN (R-CNN) approach to bounding-box object detection is to attend to a manageable number of candidate object regions and evaluate convolutional networks independently on each RoI. R-CNN was extended to allow attending to RoIs on feature maps using RoIPool, leading to fast speed and better accuracy

- Faster R-CNN advanced this stream by learning the attention mechanism with a Region Proposal Network (RPN). Faster R-CNN is flexible and robust to many follow-up improvements
- Driven by the effectiveness of RCNN, many approaches to instance segmentation are based on segment proposals
- Recently, combined the segment proposal system and object detection system for “fully convolutional instance segmentation” (FCIS)
- Mask RCNN came

Structure:

Mask R-CNN is thus a natural and intuitive idea. But the additional mask output is distinct from the class and box outputs, requiring extraction of a much finer spatial layout of an object.

Faster R-CNN consists of two stages. The first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes. The second stage, which is in essence Fast R-CNN, extracts features using RoIPool from each candidate box and performs classification and bounding-box regression

- Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN) as that of Fast RCNN. In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI

- <u>Multi-task Loss</u> => $L = L_{cls} + L_{box} + L_{mask}$
  Lcls = classification loss
  Lbox = bounding box loss
  Lmask => has a Km2 - dimensional output for each RoI, which encodes K binary masks of resolution m × m, one for each of the K classes. To this, we apply a per-pixel sigmoid and define Lmask as the average binary cross-entropy loss. For an RoI associated with ground-truth class k, Lmask is only defined on the k-th mask (other mask outputs do not contribute to the loss).

- <u>Mask – representation</u>
  - ✓ A mask encodes an input object’s spatial layout. extracting the spatial structure of masks can be addressed naturally by the pixel-to-pixel correspondence provided by convolutions

- ✓ Specifically, we predict an m × m mask from each RoI using an FCN [24]. This allows each layer in the mask branch to maintain the explicit m × m object spatial layout without collapsing it into a vector representation that lacks spatial dimensions.
- ✓ This pixel-to-pixel behavior requires our RoI features, which themselves are small feature maps, to be well aligned to faithfully preserve the explicit per-pixel spatial correspondence. This motivated us to develop the following RoIAlign layer that plays a key role in mask prediction

- RoIAlign

What's the need for the RoIAlign layer….?

- ✓ RoIPool is a standard operation for extracting a small feature map (e.g., 7×7) from each RoI
- ✓ RoIPool first quantizes a floating-number RoI to the discrete granularity of the feature map, this quantized RoI is then subdivided into spatial bins which are themselves quantized, and finally feature values covered by each bin are aggregated (usually by max pooling).
- ✓ Quantization is performed, e.g., on a continuous coordinate x by computing [x/16], where 16 is a feature map stride and [·] is rounding; likewise, quantization is performed when dividing into bins (e.g., 7×7). These quantization introduce misalignments between the RoI and the extracted features. While this may not impact classification, which is robust to small translations, it has a large negative effect on predicting pixel-accurate masks.
- ✓ To address this, we propose a RoIAlign layer that removes the harsh quantization of RoIPool, properly aligning the extracted features with the input.

What's the proposed change in RoIAlign from RoIPool…?

- ✓ we avoid any quantization of the RoI boundaries or bins
- ✓ We use bilinear interpolation to compute the exact values of the input features at four regularly sampled locations in each RoI bin and aggregate the result (using max or average).

The bilinear method is also prevalent in RoIWarp. Then why we are not using it?

- ✓ Unlike RoIAlign, RoIWarp overlooked the alignment issue and was implemented as quantizing RoI just like RoIPool. So even though RoIWarp also adopts bilinear resampling, it performs on par with RoIPool as shown by experiments

- <u>Network Architecture</u>

  mask prediction that is applied separately to each RoI:

  I. The convolutional backbone architecture used for feature extraction over an entire image
  II. the network head for bounding-box recognition (classification and regression)

Backbone Architecture / Network-depth-features:

- ✓ For ResNet and ResNeXt networks of depth 50 or 101 layers, the original implementation of Faster R-CNN with ResNets extracted features from the final convolutional layer of the 4th stage, which we call C4. This backbone with ResNet-50, for example, is denoted by ResNet-50-C4
- ✓ FPN(Feature Pyramid Network) uses a top-down architecture with lateral connections to build an in-network feature pyramid from a single-scale input. Faster R-CNN with an FPN backbone extracts RoI features from different levels of the feature pyramid according to their scale, but otherwise, the rest of the approach is similar to vanilla ResNet - More suitable for Mask RCNN

Network Head:

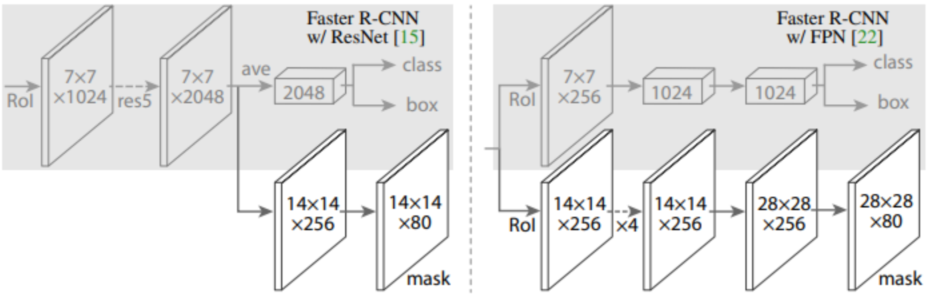
- ✓ We closely follow architectures presented in previous work to which we add a fully convolutional mask prediction branch. Specifically, we extend the Faster R-CNN box heads from the ResNet and FPN papers
- ✓ The head on the ResNet-C4 backbone includes the 5-th stage of ResNet (namely, the 9-layer 'res5'), which is compute-intensive

- <u>Implementation Details</u>

  Training:

  - ✓ As in Fast R-CNN, an RoI is considered positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise
  - ✓ The mask loss Lmask is defined only on positive RoIs. The mask target is the intersection between an RoI and its associated ground-truth mask.
  - ✓ We adopt image-centric training. Images are resized such that their scale (shorter edge) is 800 pixels.
  - ✓ Each mini-batch has 2 images per GPU and each image has N sampled RoIs, with a ratio of 1:3 of positive to negatives

- N is 64 for the C4 backbone (512 for FPN). We train on 8 GPUs (so the effective mini-batch size is 16) for 160k iterations, with a learning rate of 0.02 which is decreased by 10 at the 120k iteration. We use a weight decay of 0.0001 and a momentum of 0.9.
- For convenient ablation, RPN is trained separately and does not share features with Mask R-CNN, unless specified. For every entry in this paper, RPN and Mask R-CNN have the same backbones and so they are shareable.



Inference:

- We run the box prediction branch on the proposals, followed by non-maximum suppression
- The mask branch is then applied to the highest-scoring 100 detection boxes. Although this differs from the parallel computation used in training, it speeds up inference and improves accuracy (due to the use of fewer, more accurate RoIs).
- The mask branch can predict K masks per RoI, but we only use the k-th mask, where k is the predicted class by the classification branch.
- The m×m floating-number mask output is then resized to the RoI size and binarized at a threshold of 0.5.

- <u>Instance Segmentation</u>

  - We perform a thorough comparison of Mask R-CNN to the state of the art along with comprehensive ablation experiments by using the COCO dataset
  - the standard COCO metrics including AP (averaged over IoU thresholds), AP50, AP75, and APS, APM, APL (AP at different scales).

- RoIAlign:

  RoIAlign improves AP by about 3 points over RoIPool, with much of the gain coming at high IoU (AP75). RoIAlign is insensitive to the max/average pool; we use average in the rest of the paper.

- Mask Branch:

  Segmentation is a pixel-to-pixel task and we exploit the spatial layout of masks by using an FCN

Advantages of Mask RCNN:

- ✓ Multinomial vs. Independent Masks

  Mask R-CNN decouples mask and class prediction: as the existing box branch predicts the class label, we generate a mask for each class without competition among classes (by a per-pixel sigmoid and a binary loss). When we compare this to using a per-pixel softmax and a multinomial loss (as commonly used in FCN). This alternative couples the tasks of mask and class prediction, and results in a severe loss in mask AP (5.5 points). This suggests that once the instance has been classified as a whole (by the box branch), it is sufficient to predict a binary mask without concern for the categories, which makes the model easier to train
- ✓ Class-Specific vs. Class-Agnostic Masks:

  Mask R-CNN with class agnostic masks (i.e., predicting a single m×m output regardless of class) is near as effective

Disadvantages:

- ✓ It works on still images, so cannot explore temporal information of the object of interest such as dynamic hand gestures.
- ✓ Mask R-CNN usually fails to detect object suffered from motion blur at low resolution as hand