Mask R-CNN

presented by Jiageng Zhang, Jingyao Zhan, Yunhan Ma

Mask R-CNN

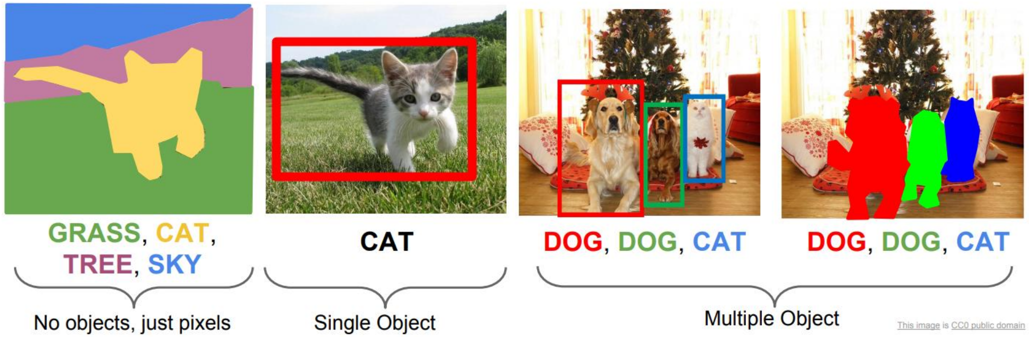
- Background
- Related Work
- Architecture
- Experiment

Mask R-CNN

- Background
- Related Work
- Architecture
- Experiment

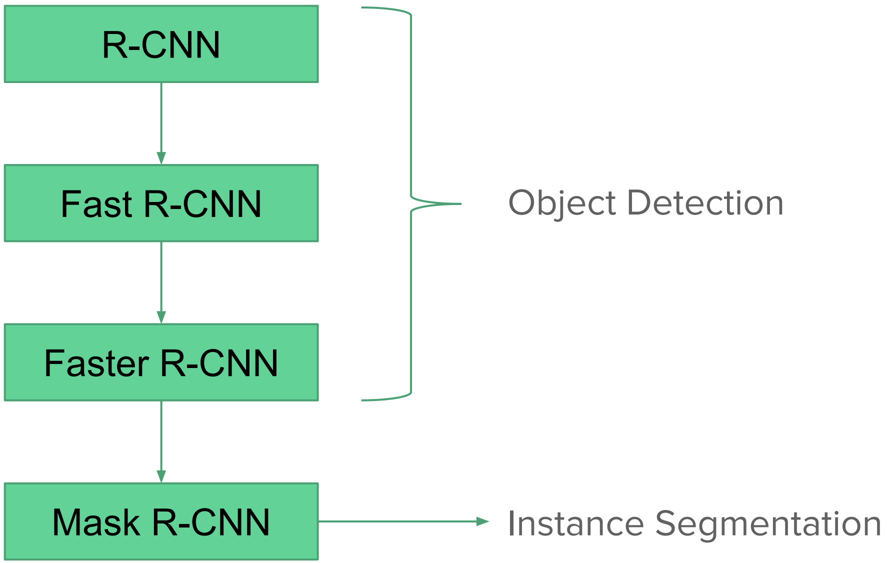
Background

- From left to right
  - Semantic segmentation
  - Single object detection
  - Multiple objects detection
  - Instance segmentation
- Video Demo: https://www.youtube.com/watch?v=OOT3UIXZztE&t=410s



Background

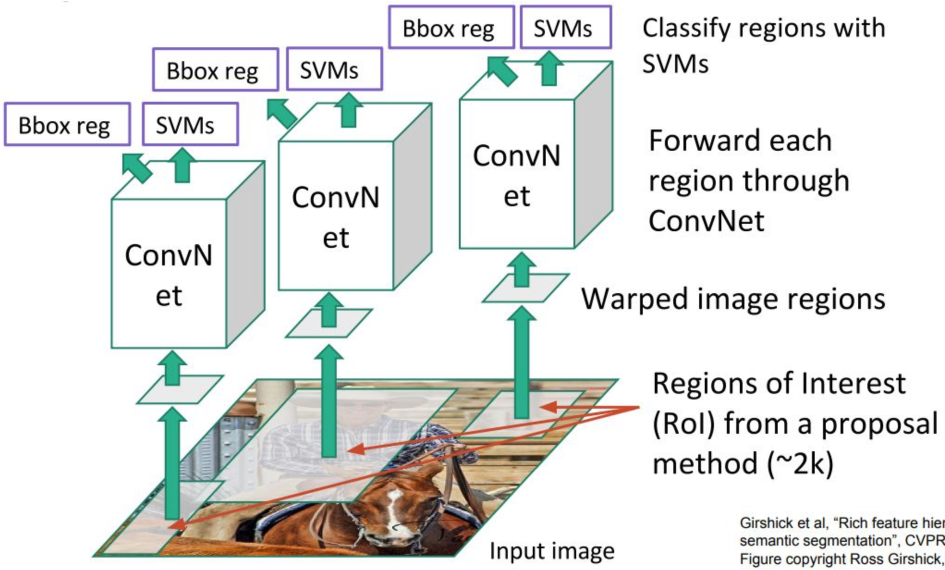
- The R-CNN family



Mask R-CNN

- Background
- Related Work
- Architecture
- Experiment

Region-based CNN (RCNN)



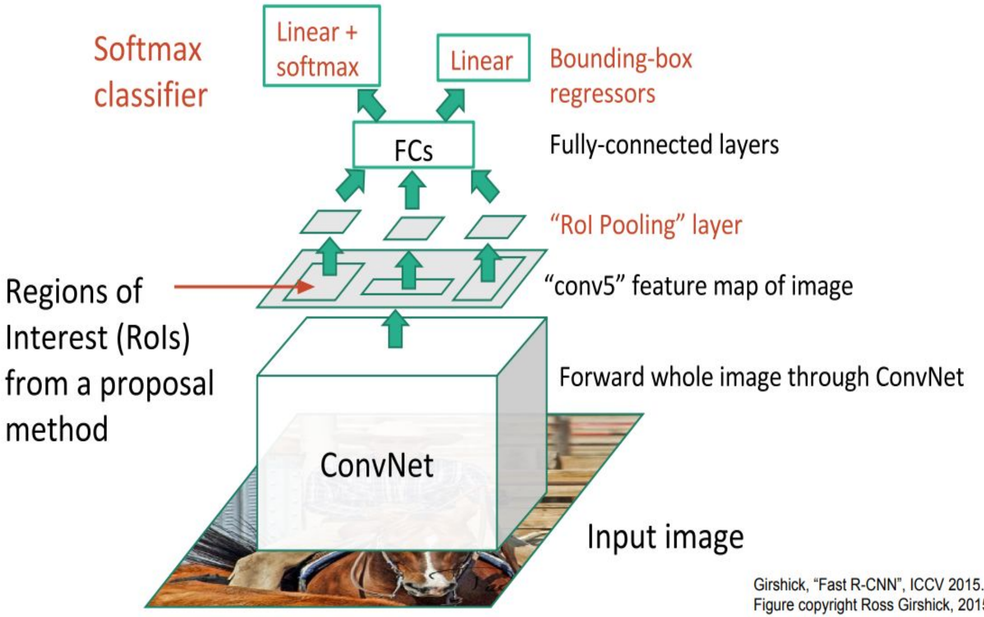
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

- Selective Search for region of interests
- Extracts CNN features from each region independently for classification

Limitations

- Training is expensive and slow because of selective search and lack of shared computation
- Slow detection speed which is not suitable for real-time usage

Fast R-CNN



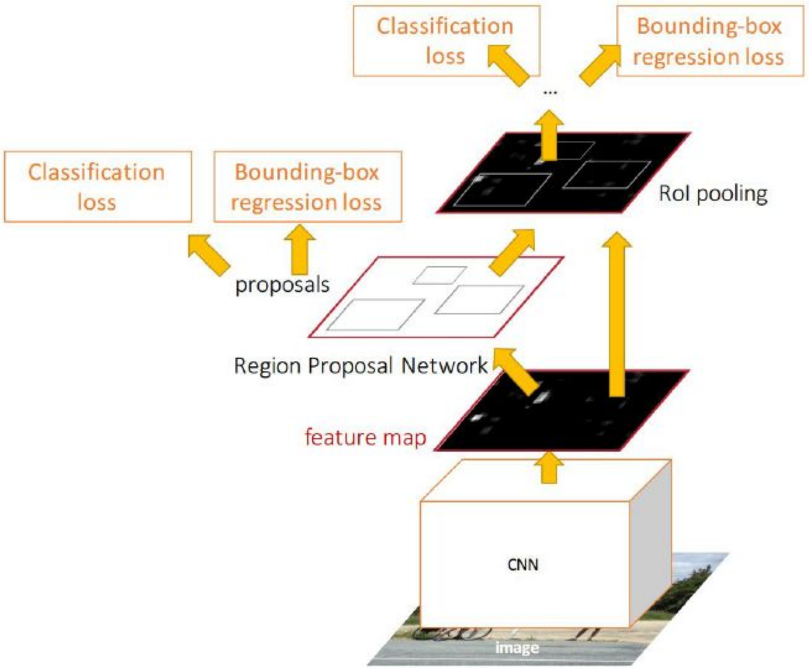
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

- Share computation of convolutional layers between proposals as a result of RoI Pooling
- The system is trained end-to-end

Limitations

- The improvement in speed is not large because the region proposals are generated separately by another model

Faster R-CNN: Fast R-CNN + RPN

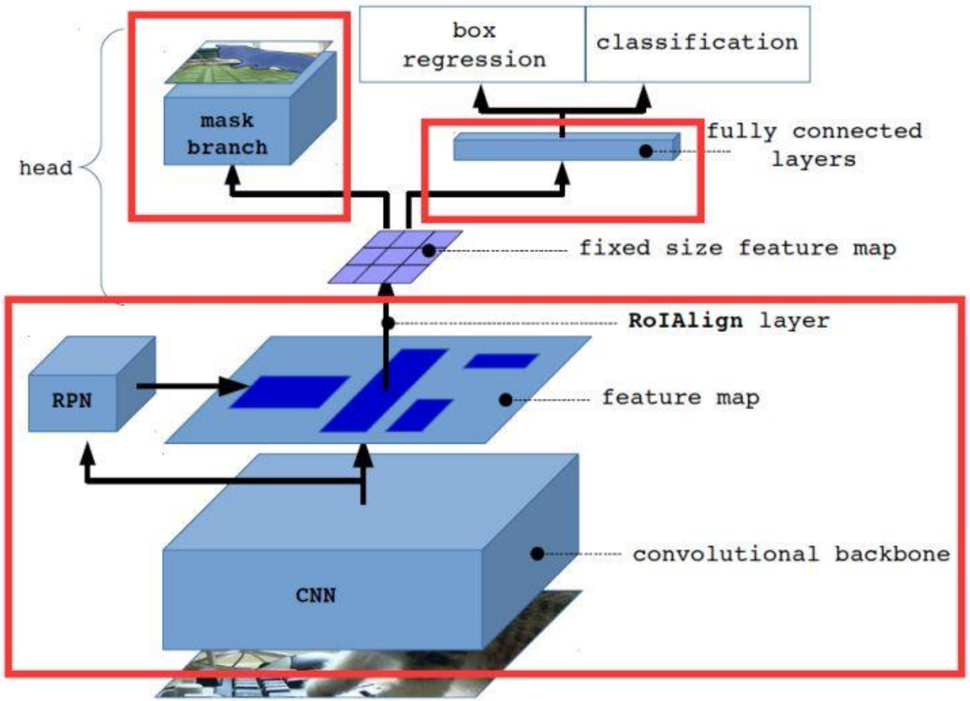


- **Region Proposal Network (RPN)** after last convolutional layer
- RPN produces region proposals directly
- Can be extended for Instance Segmentation

Limitations

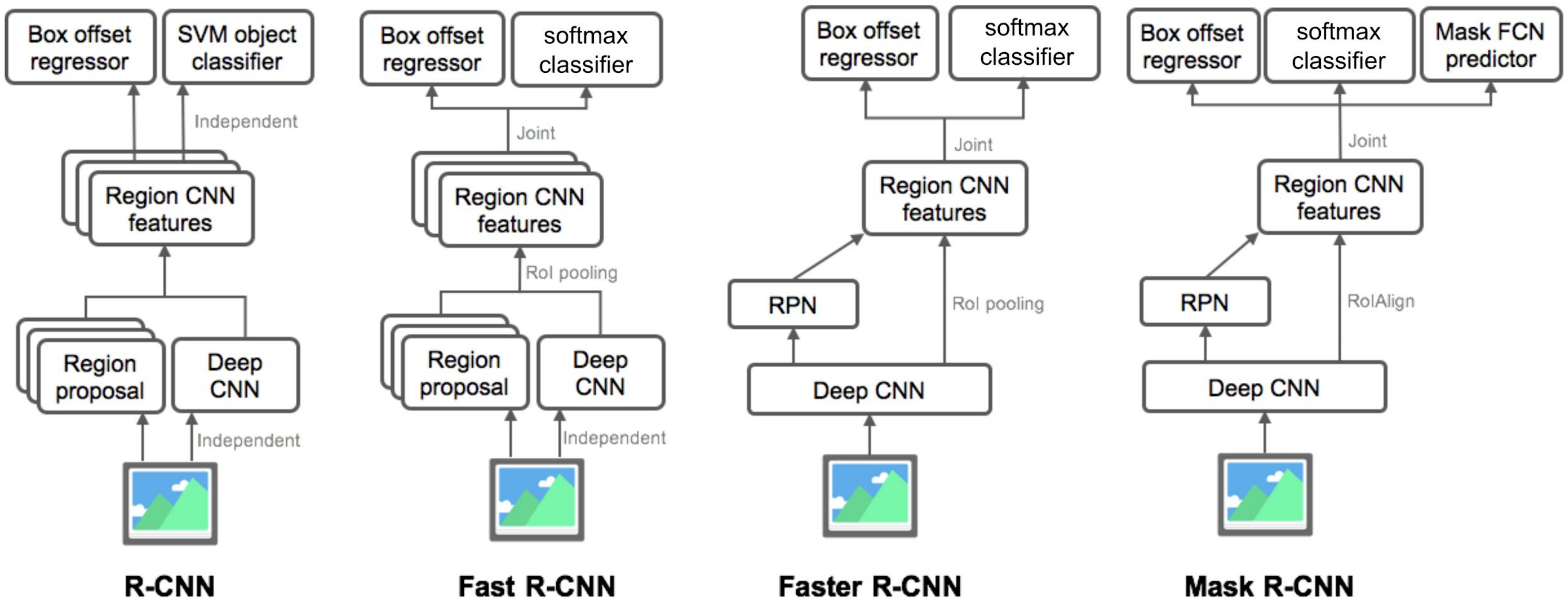
- Box Classification and Regression are being done 2 times. The two stage object detection is time-consuming.

Mask R-CNN



- backbone+RPN
- Parallel heads for box regression and classification
- RoIAlign

Summary of R-CNN family



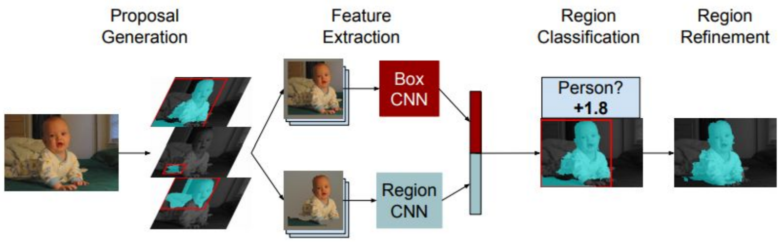
Instance Segmentation

Methods driven by R-CNN

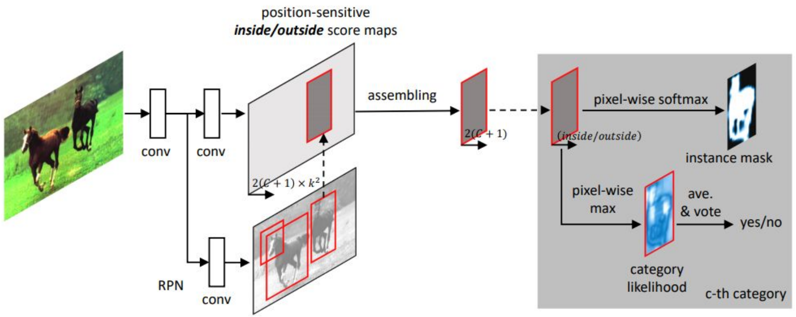
- SDS [Hariharan et al, ECCV'14]
- HyperCol [Hariharan et al, CVPR'15]
- Convolutional Feature Masking (CFM) [Dai et al,CVPR'15]
- Multi-task Network Cascades (MNCs) [Dai et al,CVPR'16]

Methods driven by FCN

- InstanceCut [Kirillov et al, CVPR'17]
- Fully Convolutional Instance-aware Semantic Segmentation (FCIS) [Li et al, CVPR'17]
- Dynamically Instantiated Network (DIN) [Arnab & Torr, CVPR'17]



SDS

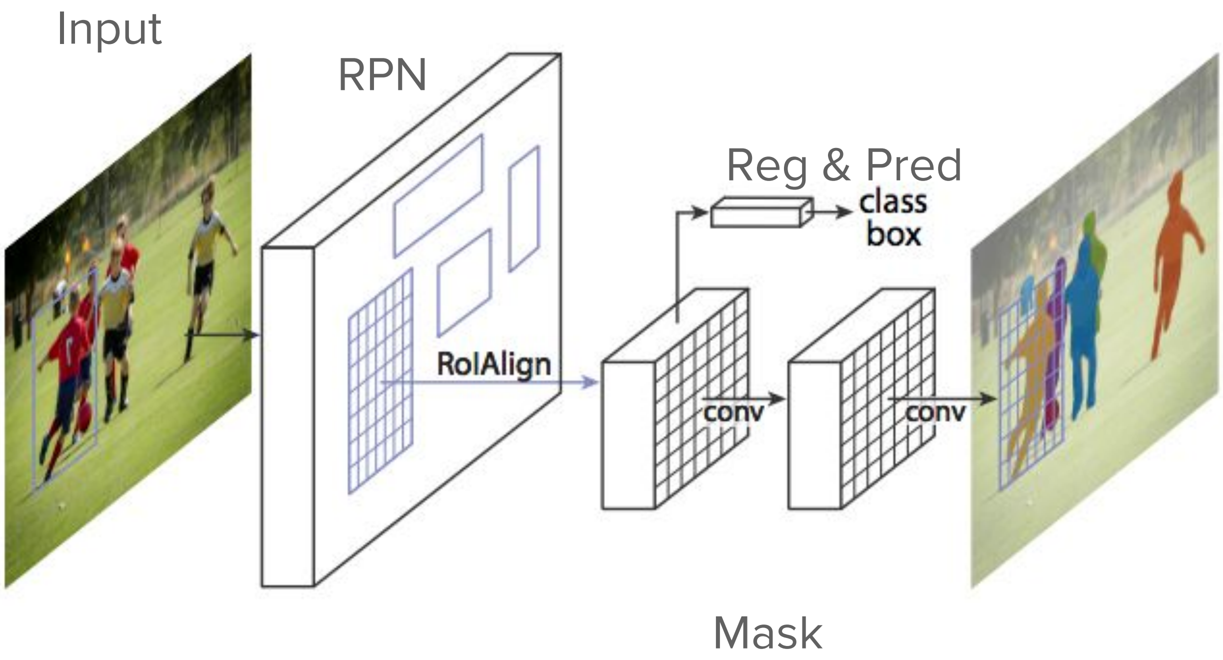


FCIS

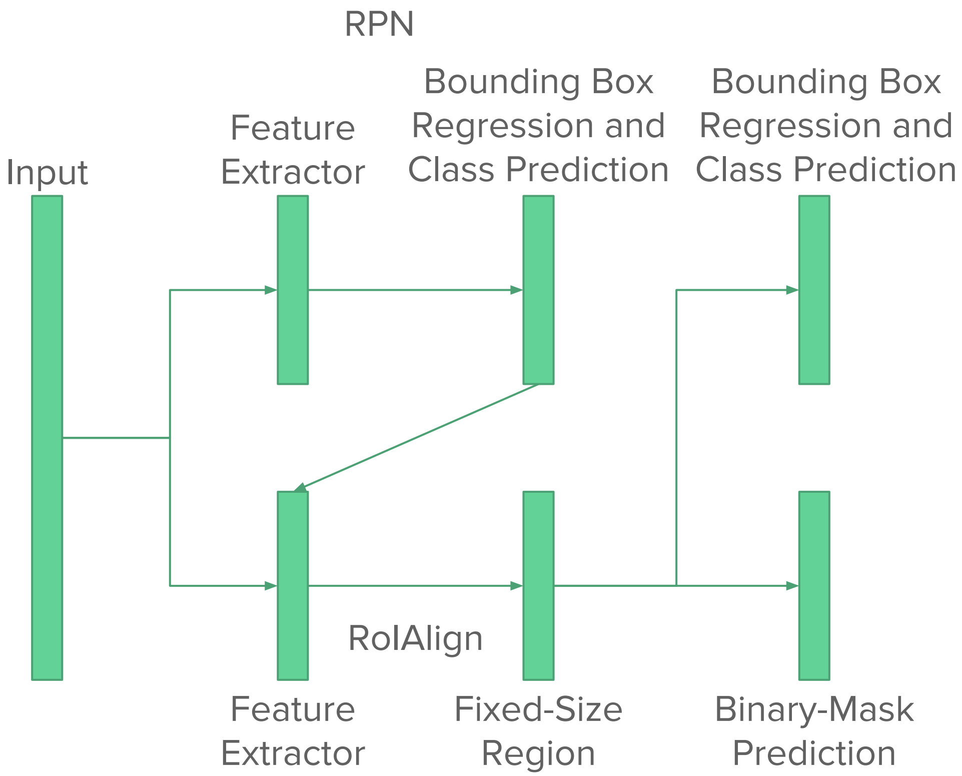
Mask R-CNN

- Background
- Related Work
- Architecture
- Experiment

Architecture



Architecture



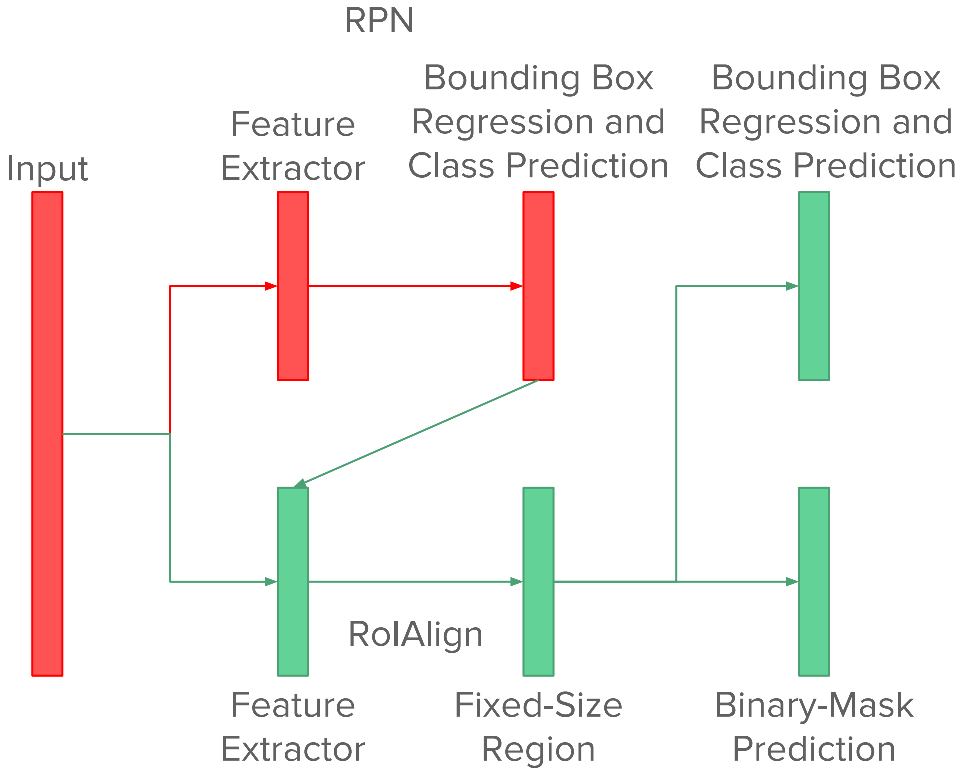
Architecture

- Stage I
  - Region Proposal Network
- Stage II
  - Bounding Box Regression
  - Class Prediction
  - Binary Mask Prediction

Architecture

- Stage I
  - Region Proposal Network
- Stage II
  - Bounding Box Regression
  - Class Prediction
  - Binary Mask Prediction

Region Proposal Network

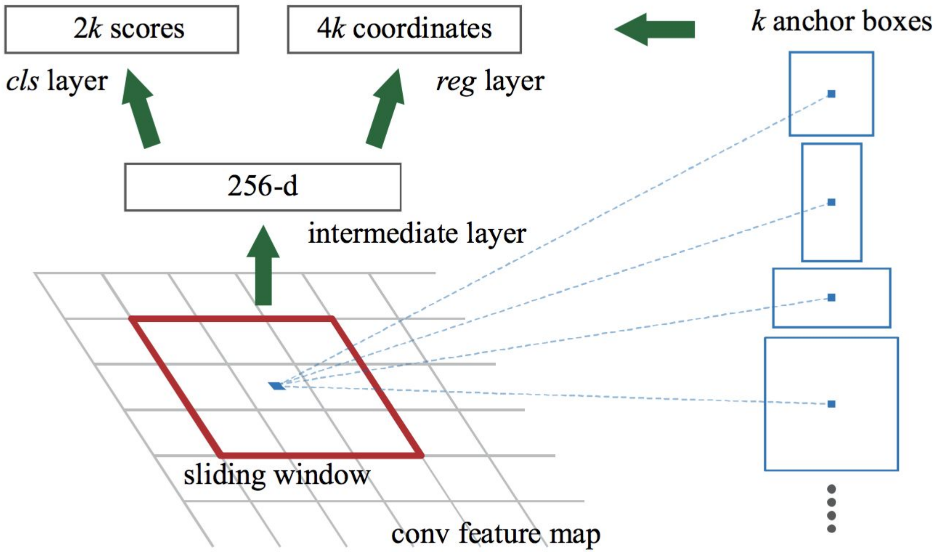


Region Proposal Network

- Feature Extractor
  - Used to extract high-level features from a input image
  - End up with MxNxC
    - M and N are related to the size of the image
    - C is the number of kernel used
    - Note that M and N are odd numbers
- Region Proposal
  - In the last layer of feature extractor, use a 3x3 sliding window to traverse the whole image
  - While traversing, try k anchor boxes
    - Anchor boxes can have different ratios and areas
    - Thus, the size of bounding-box regression will be 4k, and the size of class prediction will be 2k

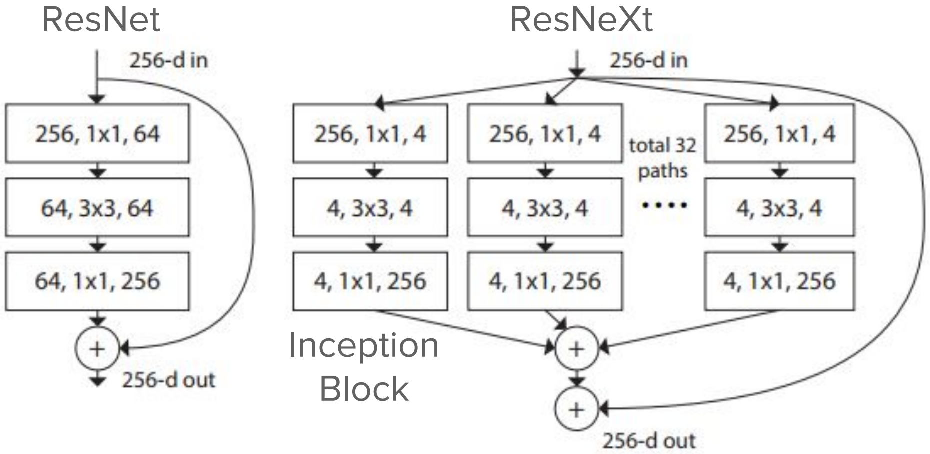
Region Proposal Network

- Why 3x3 and odd MxN?
  - Don't want to miss information from center
- Why k anchor boxes?
  - Objects can be in different shapes, and thus by using different anchor settings we can find a better bounding-box match to the objects
- Why 2k scores and 4k coordinates?
  - 2 refers to whether background or not.
  - 4 refers to (x, y, w, h)
  - For each anchors, we predict scores and offsets, that counts for k.



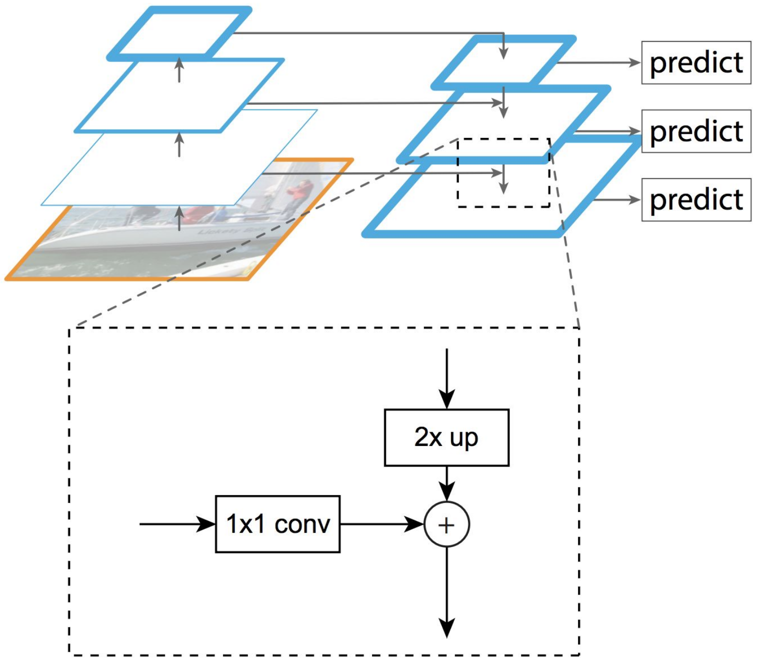
Feature Extractor

- Before jumping to stage II, let's talk about feature extractor
- In this paper, the authors try two network settings (backbone)
  - ResNet
  - Feature Pyramid Network with ResNet and ResNeXt



Feature Pyramid Network

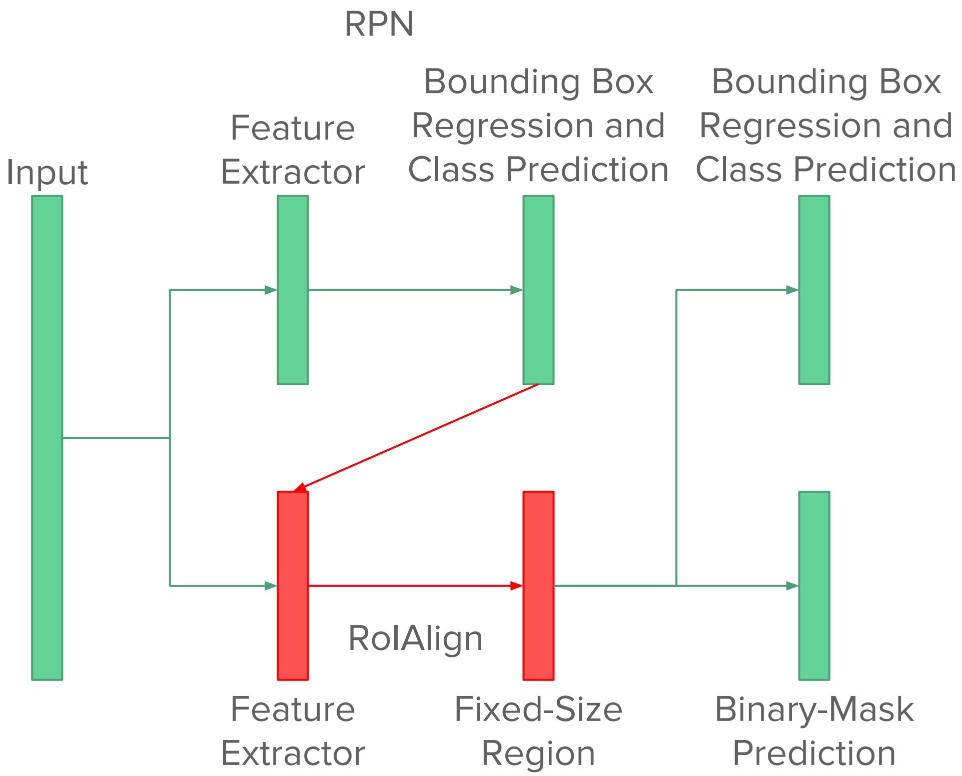
- Why pyramid?
  - Objects with different scales can fall naturally into one of the levels in the pyramid
- Why skip connections?
  - All levels are semantically strong
- Why not using it?
  - Computation cost and memory cost
- Solution
  - 1x1 convolution
- Why prediction is made in every scale?
  - To handle objects in multi-scale



Architecture

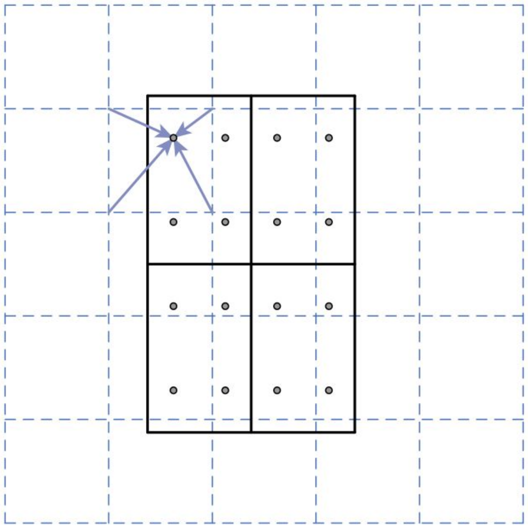
- Stage I
  - Region Proposal Network
- Stage II
  - Bounding Box Regression
  - Class Prediction
  - Binary Mask Prediction

RoIAlign


RPN
Input
Feature Extractor
Bounding Box Regression and Class Prediction
Bounding Box Regression and Class Prediction
RoIAlign
Feature Extractor
Fixed-Size Region
Binary-Mask Prediction

RoIAlign

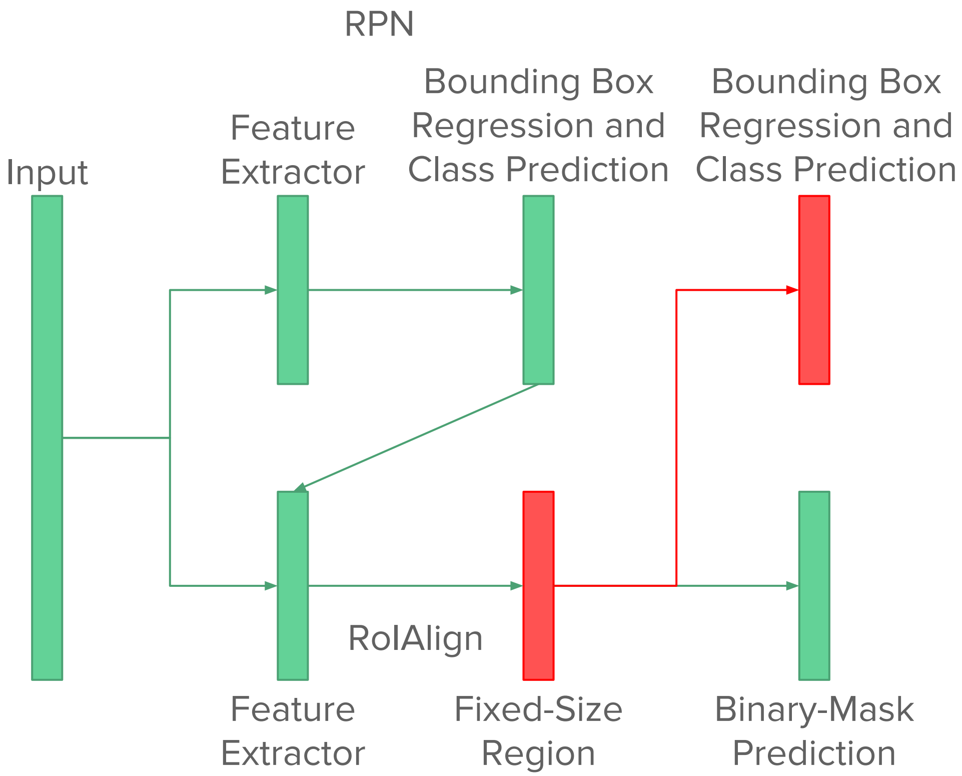
- Very first step in Stage II
- Why use RoIAlign?
  - Keep the size of the feature map the same so that the same sub-network can be used to predict class, mask and regress bounding box
  - Focus on translation variance - the location of the object matters
  - Avoid quantization that causes misalignment
- RoIAlign
  - Use bilinear interpolation to sample points in one bin
    - In the right image, 4 points are sampled for each bin
  - Then, perform max pooling to select a point to represent that bin



RoIAlign

- RoIAlign is performed on feature extractor for Stage II
- How does this feature extractor differ from the one in Stage I
  - They have exactly same structure
  - They can share weights to decrease training time
  - In this paper, the authors compare performance between model that share weights and model that does not share
    - Sharing features improves accuracy by a small margin
    - Feature sharing also reduces the testing time

Bounding Box Regression & Class Prediction



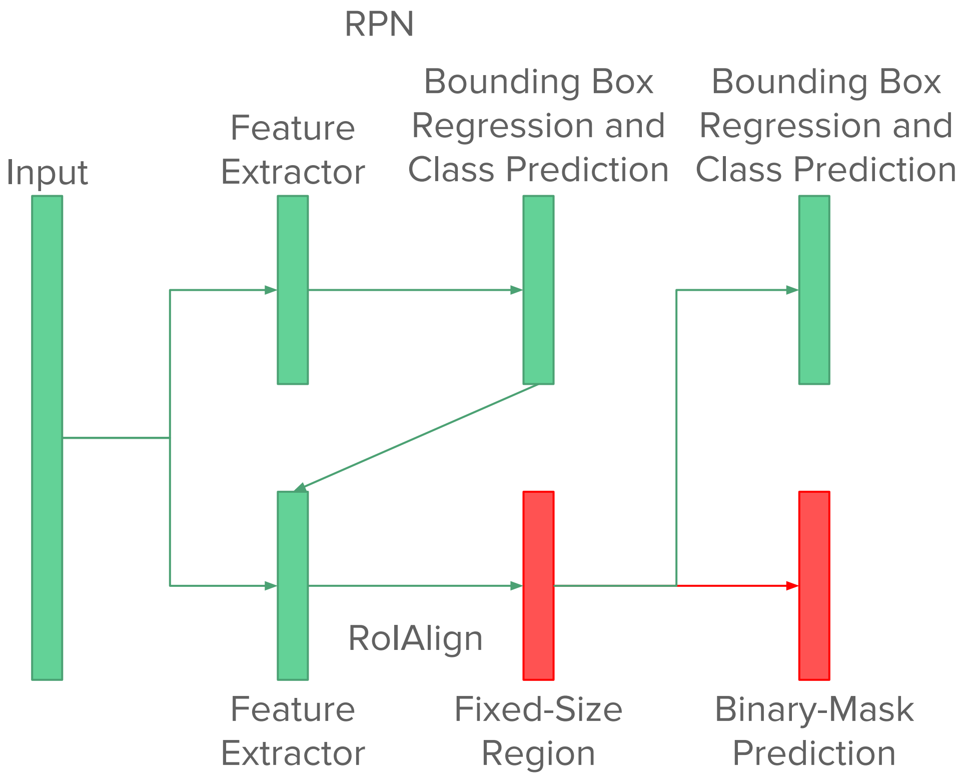
Bounding Box Regression

- Further refine the bounding box offsets to get a more accurate bounding box
- Produce 4 values
  - Top left x value
  - Top left y value
  - Width of bounding box
  - Height of bounding box

Class Prediction

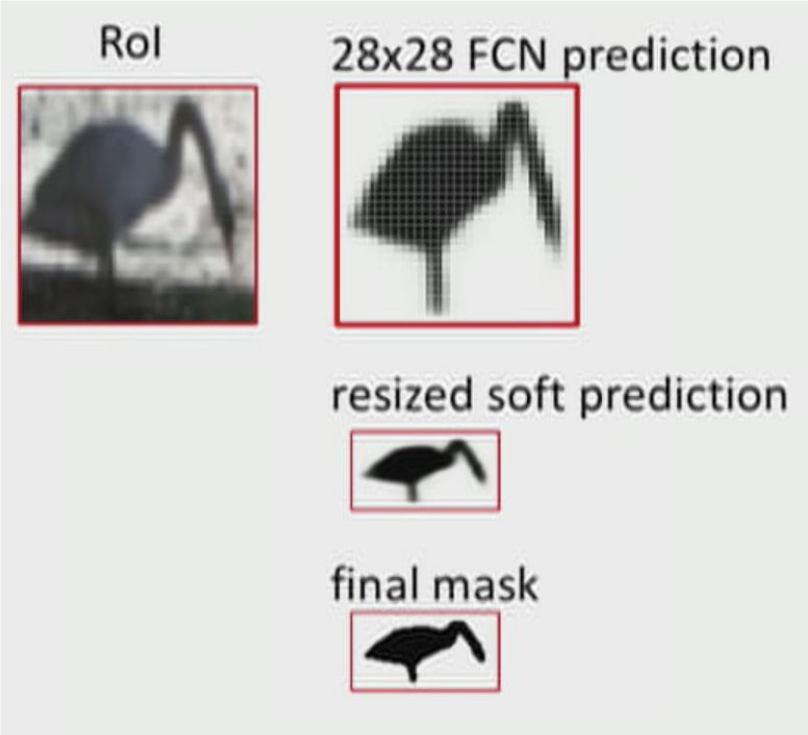
- Predict label from the label set

Binary Mask Prediction



Binary Mask Prediction

- Dimension: Km^2
  - K represents for class number
  - m^2 represents the spatial resolution
- Mask is resized to fit the shape of the original objects



Loss Function

$$L_{\mathrm{loc}}(t^u, v) = \sum_{i \in \{\mathrm{x,y,w,h}\}} \mathrm{smooth}_{L_1}(t_i^u - v_i),$$

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

- $L = L_{cls} + L_{box} + L_{mask}$
- Classification Loss
  - Multiclass cross-entropy loss
- Bounding Box Loss
  - Smooth L1 loss between ground-truth bounding boxes and predicted bounding boxes
  - Smooth L1 loss is a robust L1 loss that is less sensitive to outliers than the L2 loss
    - Prevent gradient explosion
- Mask Loss
  - Only defined at K^{th} class, where K is the class with ground-truth label
  - Defined as average binary cross-entropy loss
  - Thus, masks acrosses classes do not compete
    - Good for instance segmentation

Mask R-CNN

- Background
- Related Work
- Architecture
- Experiment

Highlights from Main Results

- Major Experiments: Microsoft COCO Dataset:
  - A Dataset aims to address the question of scene understanding
  - Depict complex everyday scenes of common objects in their natural context. (91 Categories)
  - Instance segmentation with Mask
  - Object Detection Challenge (**object segmentation** and **bounding box** output)
  - Human Body Key Point Challenge
- Outperforms winners of COCO 2015 and 2016 segmentation challenges
  - FCIS and MNC
  - considered to the state-of-the-art methods
- Eliminates artifacts on overlapping instances
- Fast training and inference speed
- Can be extended to other problems:
  - Human Pose Estimation as a example

Figure 5. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

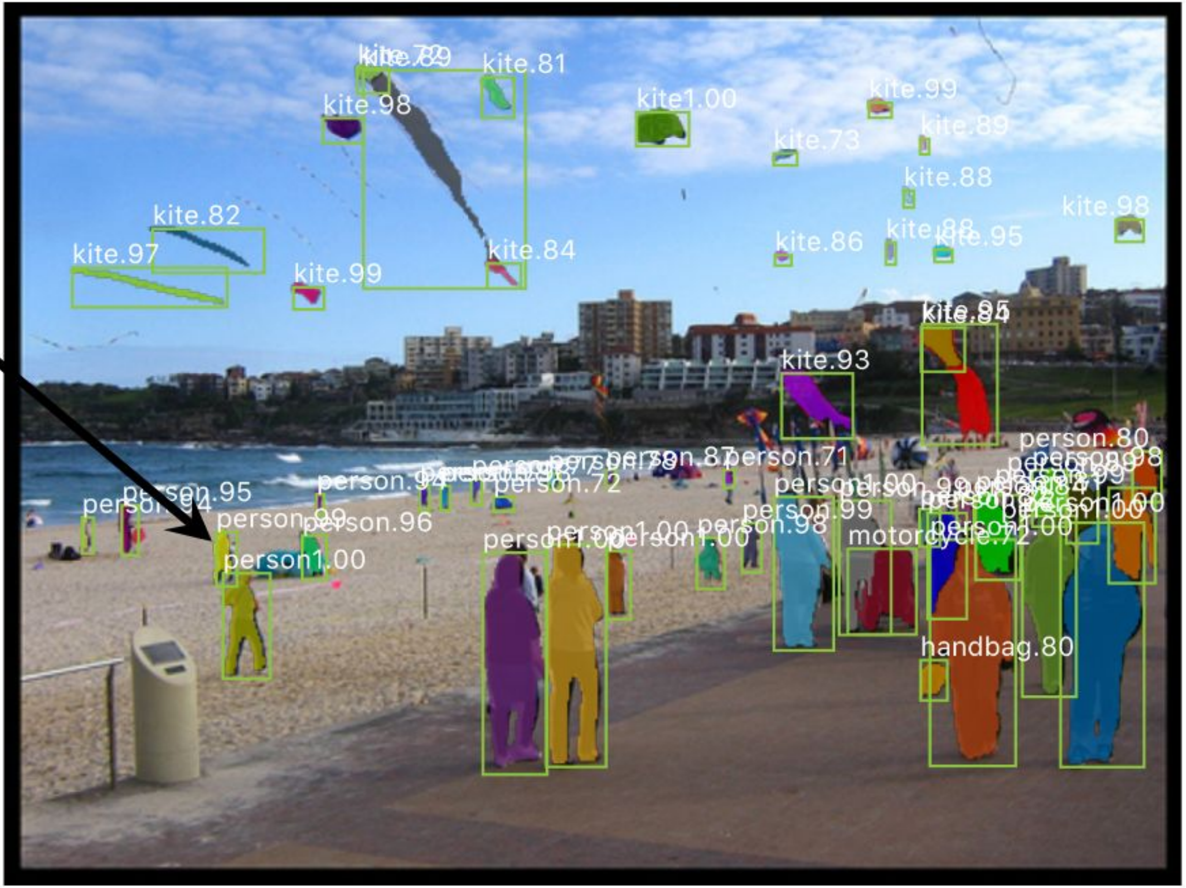

person1.00
person1.00
person1.00
person1.00
person1.00
person1.00
person1.00
baseball bat.99
baseball bat.85
baseball bat.98dog1.00
HarD
VISITORS


traffic light.99
traffic light.71
person.99
person1.00
person.92
person.74
tie.85
handbag.88
person.95
handbag.99
person.99
handbag.88

disconnected
object

Mask R-CNN results on COCO

small
objects

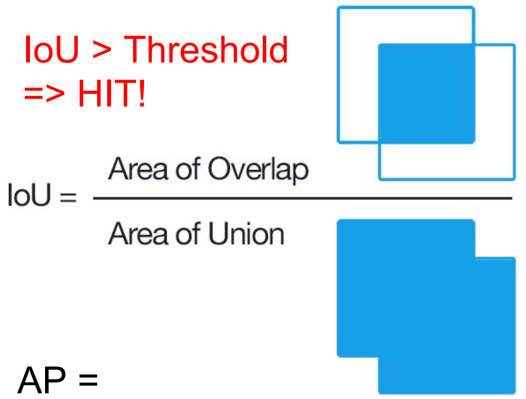


Mask R-CNN results on COCO

False Alerts

Metrics from COCO Dataset (Average Precision)

Average Precision (AP):	
AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)
AP Across Scales:	
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2
Average Recall (AR):	
$AR^{max=1}$	% AR given 1 detection per image
$AR^{max=10}$	% AR given 10 detections per image
$AR^{max=100}$	% AR given 100 detections per image
AR Across Scales:	
AR^{small}	% AR for small objects: area < 32^2
AR^{medium}	% AR for medium objects: 32^2 < area < 96^2
AR^{large}	% AR for large objects: area > 96^2



$$AP = \frac{1}{|classes|} \sum_{c \in classes} \frac{\#TP(c)}{\#TP(c) + \#FP(c)}$$

Majorly used for experiment analysis

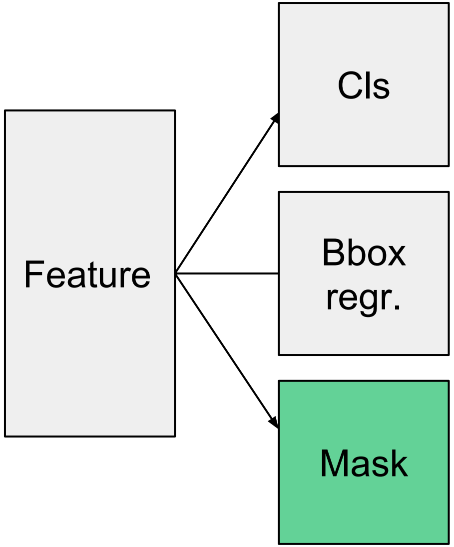
Instance Segmentation Results on COCO(test-dev)

	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**

- Mask R-CNN outperforms “state-of-the-art” FCIS+++ (bells and whistles)
- Bell and Whistles: multi-scale train/test, horizontal flip test, and online hard example mining (OHEM)

Ablation Experiments

- Change of the backbone networks structures
  - various ResNet CNN + (Conv4 or FPN)
  - Best AP result with ResNeXt
- Class-Specific vs. Class-Agnostic Masks
  - Nearly as effective for agnostic mask
- Multinomial vs. Independent Masks
  - Multinomial Masks raises a severe loss
  - Enough to use the result from cls layer for class labeling
- RoI Pooling vs. RoI align
  - RoI align reduces the information loss in resizing and significantly improves AP
- MLP vs FCN
  - MLP cannot perform as good to capture the spatial layout of the mask



Ablation: Backbone and Mask size per class

- Backbone
  - Not all frameworks automatically benefit from deeper or advanced networks

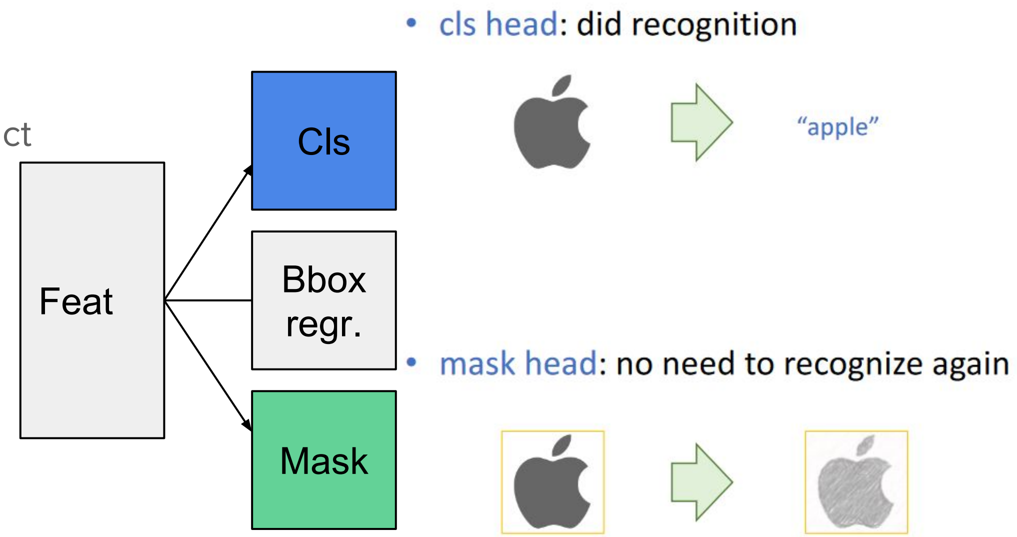
net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

- Class-Specific vs. Class-Agnostic Masks
  - Default instantiation predicts specific masks (m x m per class)
  - Agnostic mask: (m x m in **regardless** of the class)
  - As effectiveness for agnostic mask
  - On ResNet-50-C4:
    - **29.7** (agnostic) vs **30.3** (specific)
    - Nearly as effective

Ablation: Multinomial vs Independent Mask

- Multinomial vs. Independent Masks
  - multinomial: mask competing among classes (softmax)
  - box classification is sufficient to predict binary mask (sigmoid)

	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*



Ablation: RoI Pooling vs RoI Align

RoIPool vs. RoIAlign (one of the distinguished contribution of the paper)

baseline: ResNet-50-Conv5 backbone, **stride=32**

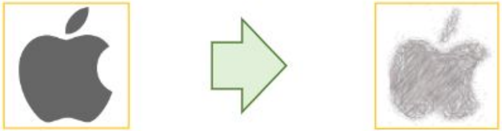
	mask AP			box AP		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	$\boxed{\textbf{34.0}}$	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

- nice box AP without dilation/upsampling

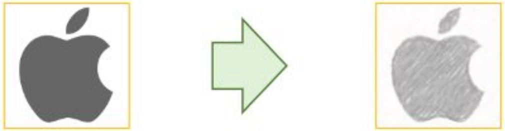
Ablation: MLP vs FCN

- MultiLayer Perceptron vs. Fully Convolutional Network
  - Fully Convolutional Networks improves results as they take advantage of explicit encoding spatial layout

MLP: lose "place-coded" info, too abstract

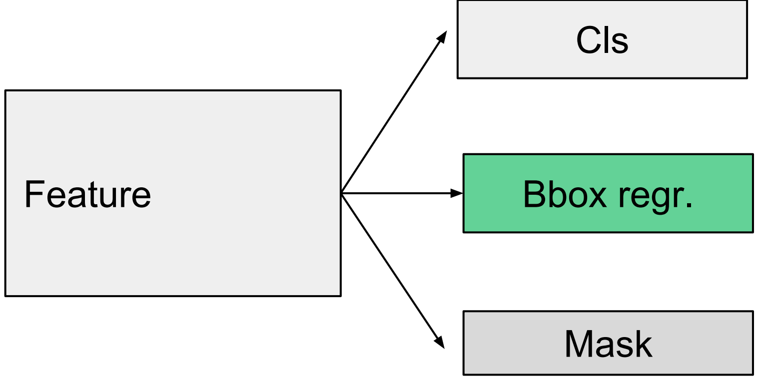


FCN: translation-equivariant



	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**

Bounding Box Results



	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [39]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

RoI Align

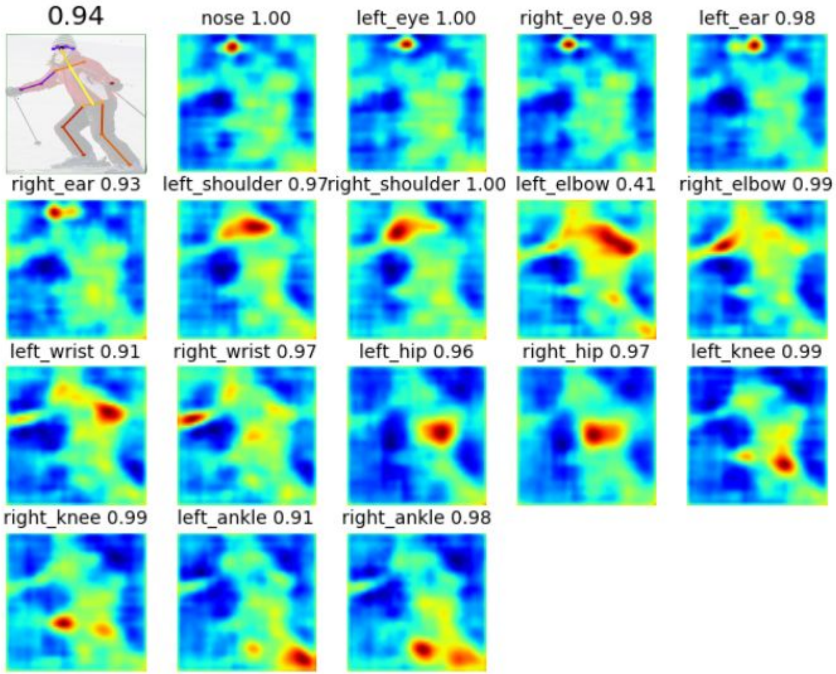
RoI Align + Multi-task training w/ mask

Timing (ResNet-101-FPN model)

- Inference:
  - 4-step training of Faster R-CNN
  - **195** ms per image on single Nvidia Tesla M40 GPU
  - **15** ms per image for CPU resizing
  - Not the most optimized method in regard of inference speed, but still very fast
  - (resizing images, toggling the number of proposed regions)
- Training:
  - Fast to train
  - COCO trainval35k
  - Synchronized 8-GPU configuration
  - 44 hours of training time

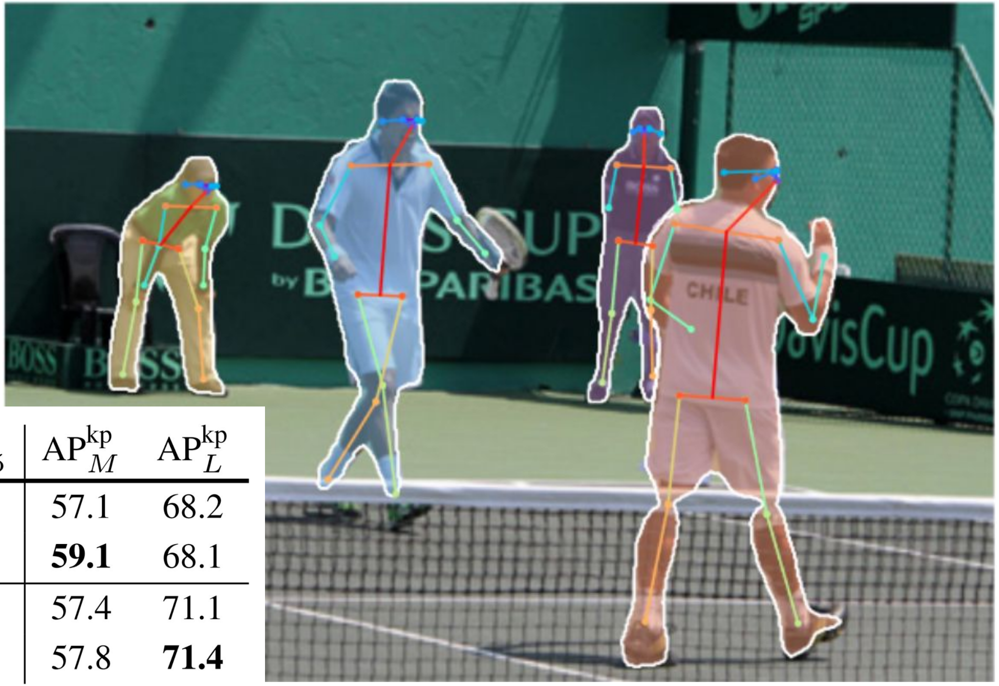
Extension: Human Keypoint Detection

- COCO Keypoint Detection (2nd Challeng from COCO dataset)
  - localization of person keypoints in challenging, uncontrolled conditions
  - simultaneously detect body location and keypoi
- Implementation of Mask R-CNN
  - 1 keypoint = 1 ‘hot’ mask (m x m)
  - Human pose (17 keypoints) => 17 Masks
  - Training:
    - m^2 softmax over spatial location
    - encourage 1 point detection



Extension: Human Keypoint Detection Result

- CMU-Pose+++(COCO 2015 Winner)
- G-RMI (COCO 2016 Winner)



	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose+++ [6]	61.8	84.9	67.5	57.1	68.2
G-RMI [32]†	62.4	84.0	68.5	**59.1**	68.1
Mask R-CNN, keypoint-only	62.7	87.0	68.4	57.4	71.1
Mask R-CNN, keypoint & mask	**63.1**	**87.3**	**68.7**	57.8	**71.4**

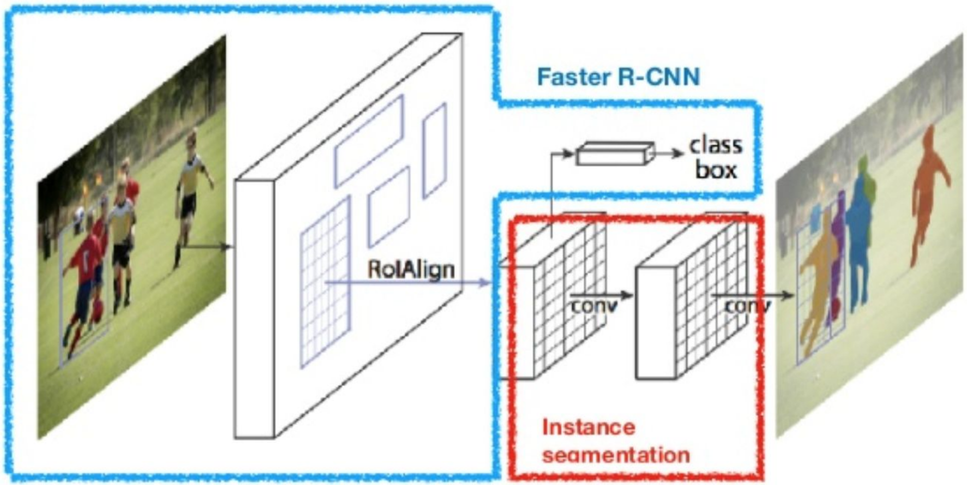
Experiments on Cityscapes

- Mask R-CNN with ResNet-FPN-50 backbone
- Better result is achieved with the pre-trained model on COCO and then fine-tuned for the Cityscapes data
- Demonstrate the real world application effectiveness

person	rider	car	truck	bus	train	mcycle	bicycle
17.9k	1.8k	26.9k	0.5k	0.4k	0.2k	0.7k	3.7k

Summary

- Mask R-CNN Advantages
  - Good Inference Speed
  - Good Accuracy
  - Intuitive and easy to implement
  - Extension Capability
- Limitations:
  - False alerts
  - Missing labels



Questions?

Thank you very much!

Reference

- S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In TPAMI, 2017.
- R. Girshick, “Fast R-CNN,” in IEEE International Conference on Computer Vision (ICCV), 2015.
- Stanford slides - Fei Fei Li & Andrej Karpathy & Justin Johnson
- https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html
- http://kaiminghe.com/iccv17tutorial/maskrcnn_iccv2017_tutorial_kaiminghe.pdf
- K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask ´ R-CNN. arXiv:1703.06870, 2017.
- S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.