Tensor Mask: A Foundation for Dense Object Segmentation

Xinlei Chen, Ross Girshick, Kaiming He, Piotr Dollar
Facebook AI Research (FAIR)



Presenter : Mahavir Dwivedi
Faculty : Yogesh Rawat

UCF

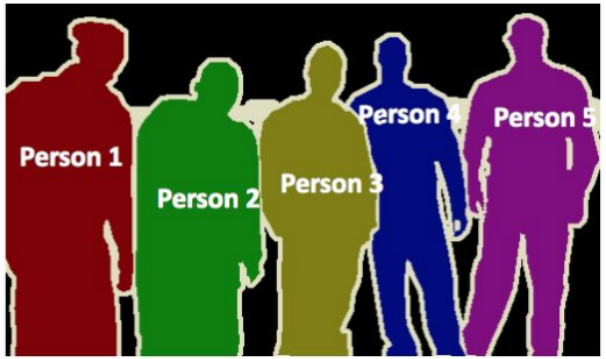
Overview

- Problem Statement
- Related Work
- Tensor Representation for Masks
- Tensor Mask Architecture
- Training
- Losses
- Results
- Implementation

Problem Statement

Instance Segmentation Task

- Label each foreground pixel with object and instance
- Object detection + semantic segmentation



Instance Segmentation

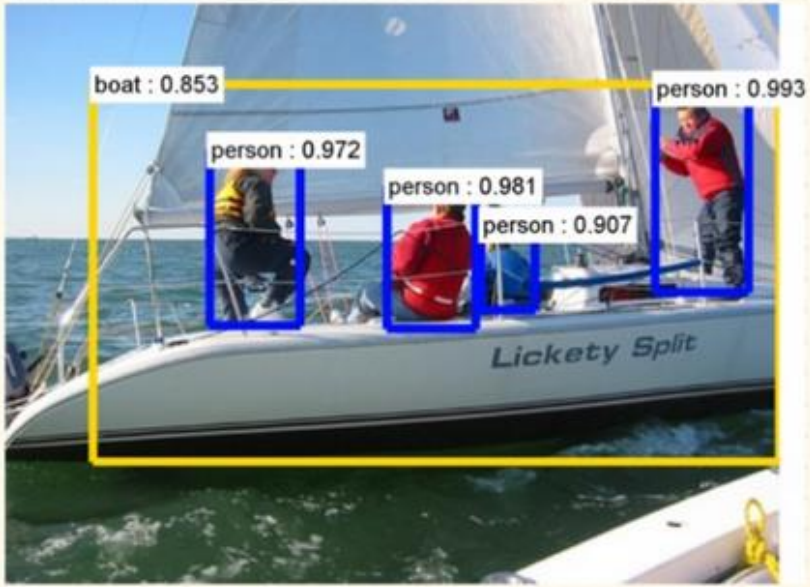
UCF



Image classification

✓boat

✓person

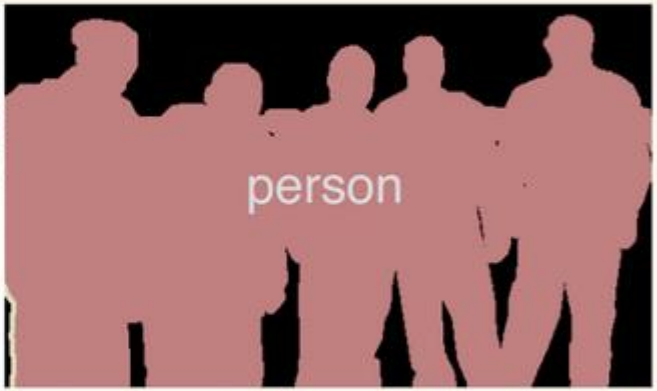


Object detection

UCF

Semantic Segmentation



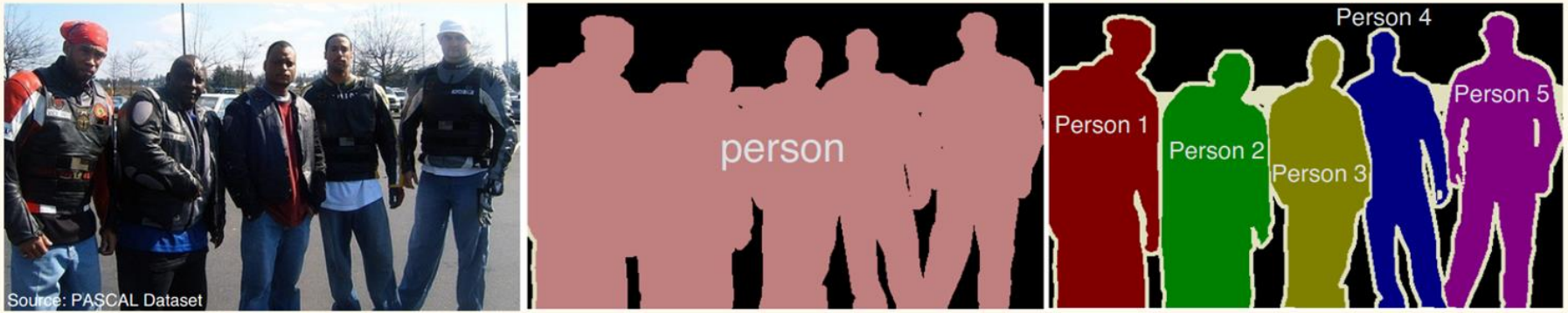


Semantic segmentation
(pixel-level classification)

UCF

The Instance Segmentation Task

Our task



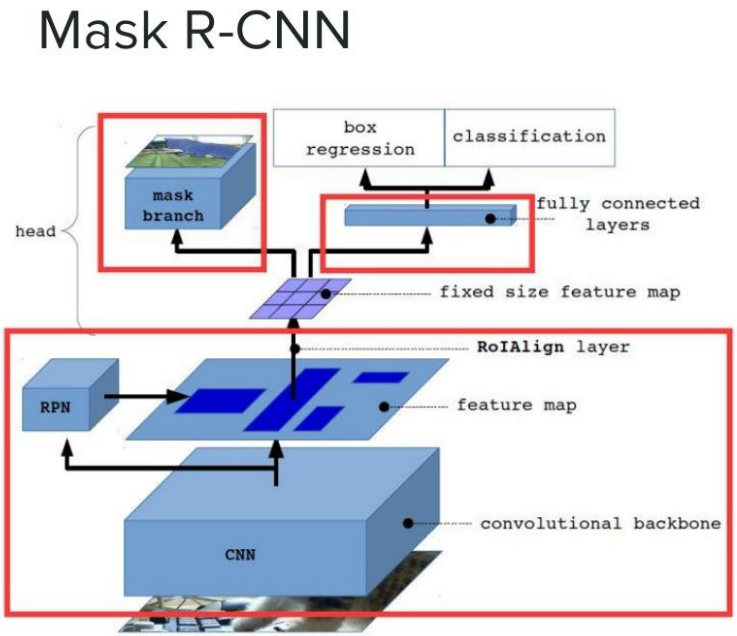
Semantic segmentation
(pixel-level classification)

Instance segmentation
(pixel-level detection)

UCF

Related Work

Detect then Segment (MaskRCNN)



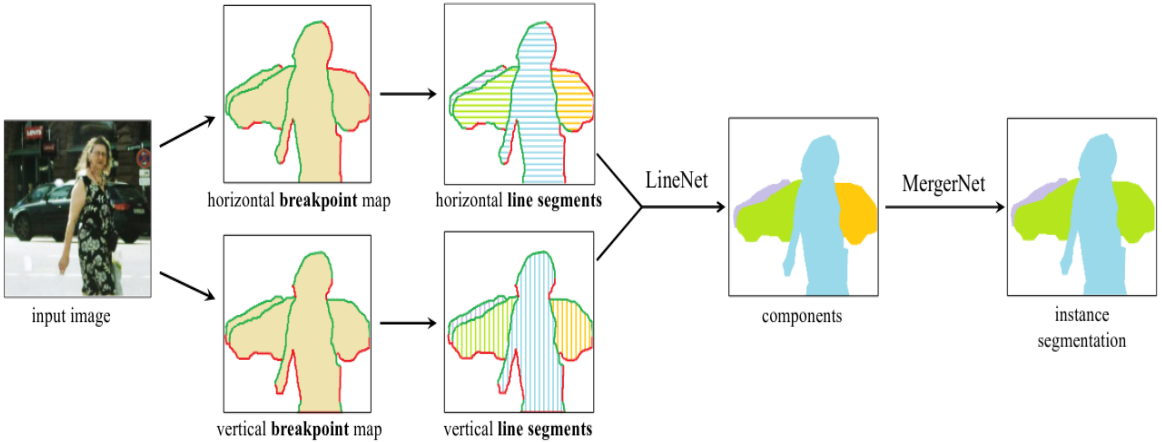
- BackBone + RPN
- Parallel heads for for box regression and classification
- RoI Align

Mask R-CNN Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick

UCF

Related Work

Label Pixels then Cluster

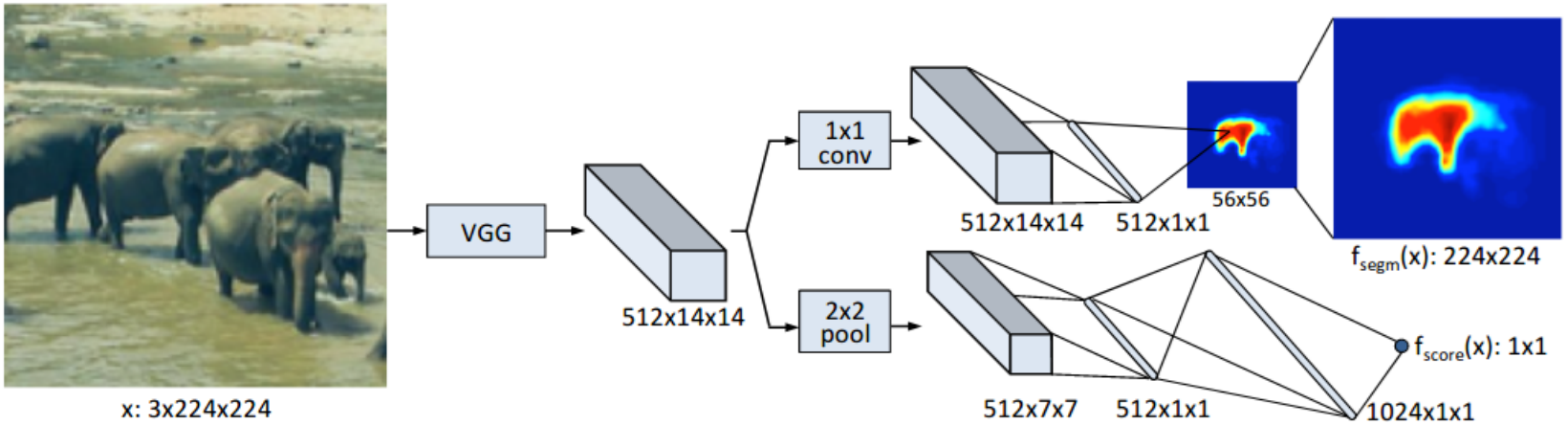


SGN: Sequential Grouping Networks for Instance Segmentation

UCF

Related Work

Dense Sliding Window methods



Learning to Segment Object Candidates Deep Mask

UCF

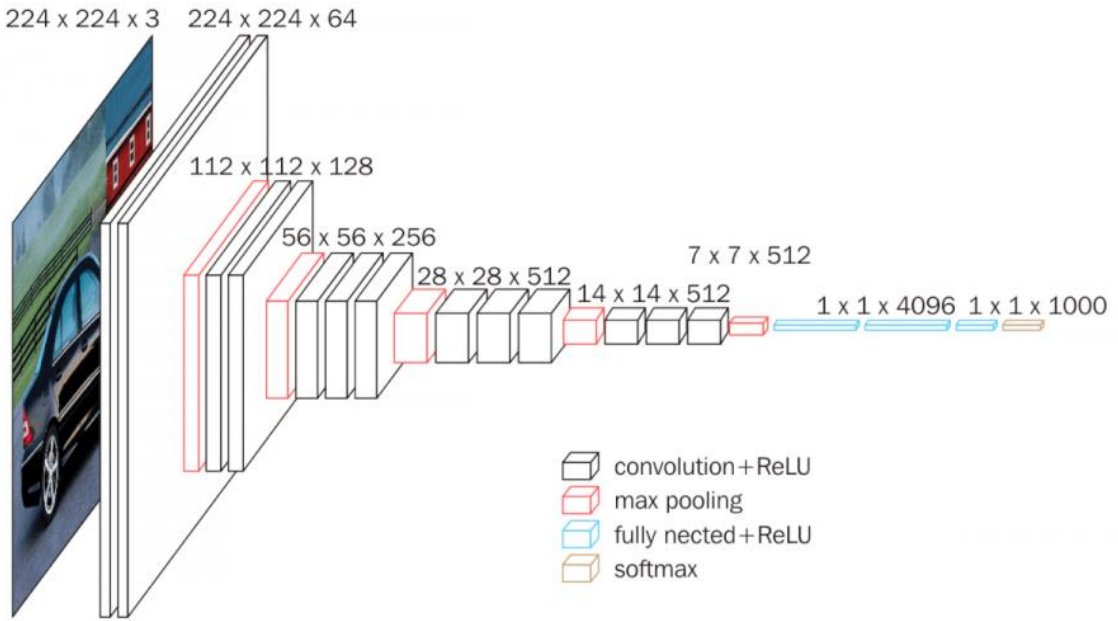
Related Work



Results from DeepMAsk

UCF

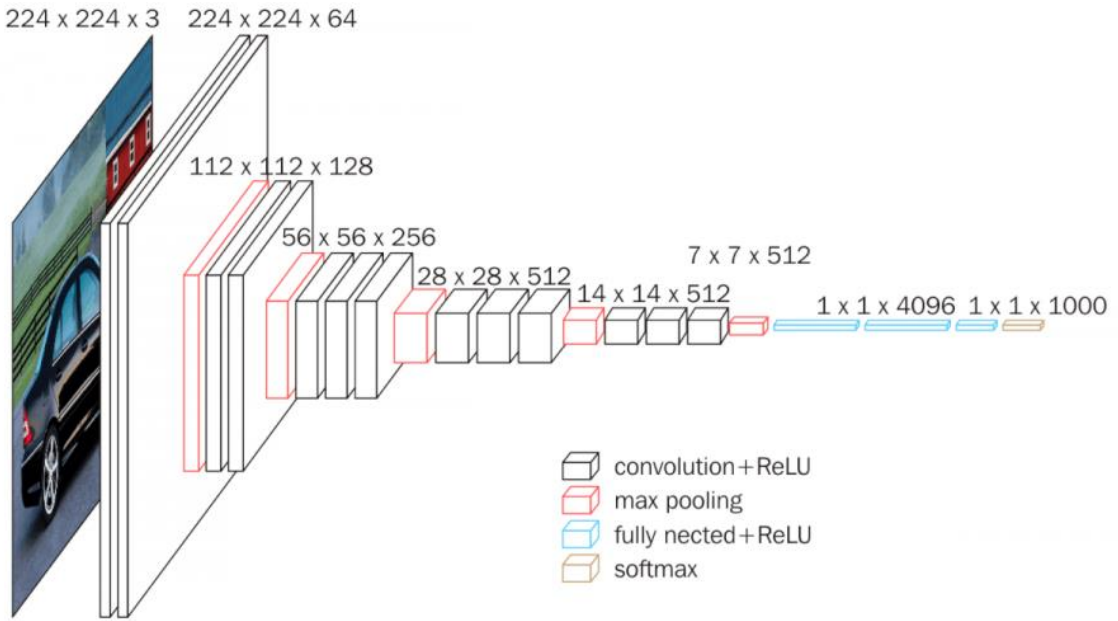
Why Tensor Mask?



- CNN's we have (C, H, W) information (3D Tensors)

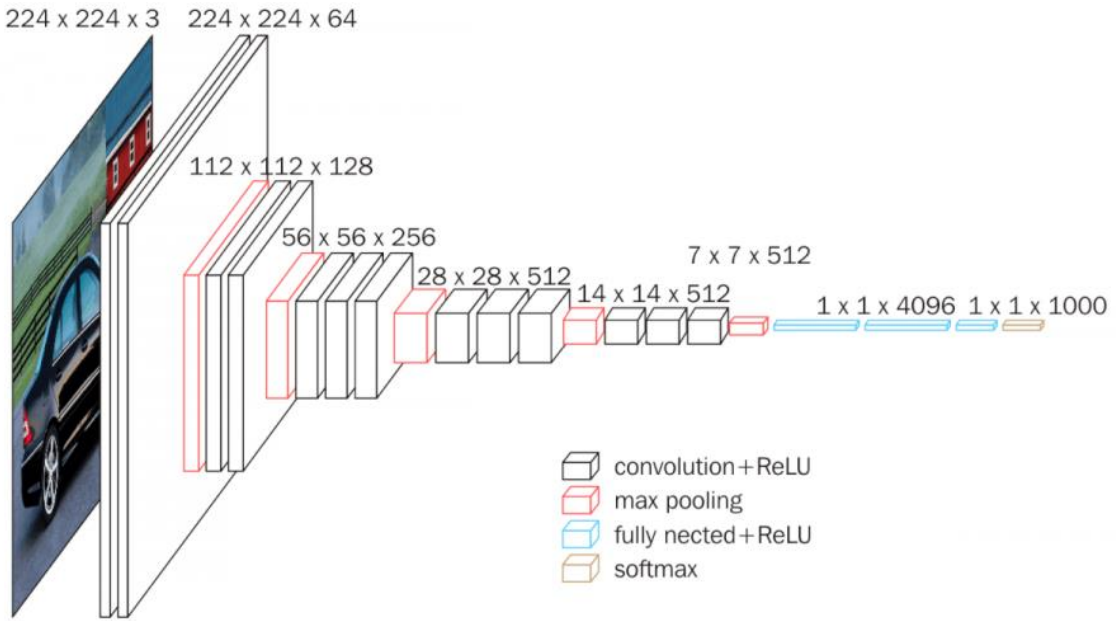
UCF

Why Tensor Mask?



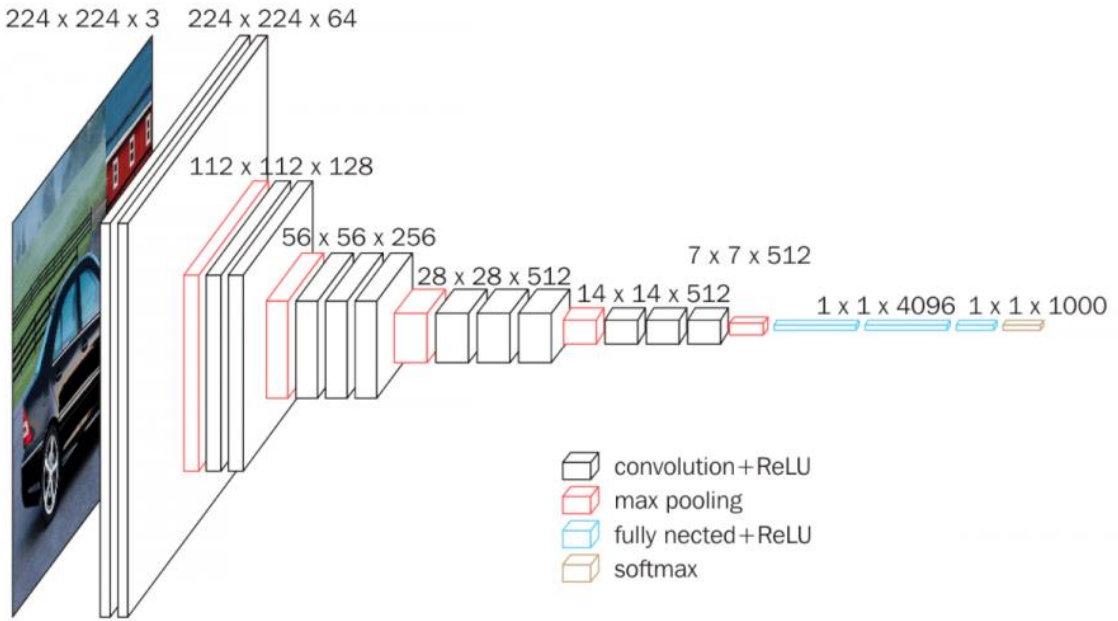
- H, W axis has a geometric meaning as it represents object position

Why Tensor Mask?



- Channel Axis does not have a geometric meaning (has semantic meaning)

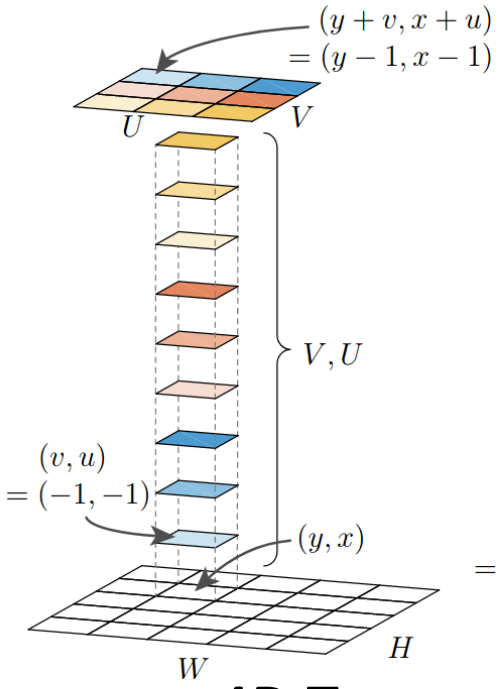
Why Tensor Mask?



- Channel Axis, Difficult to manipulate

UCF

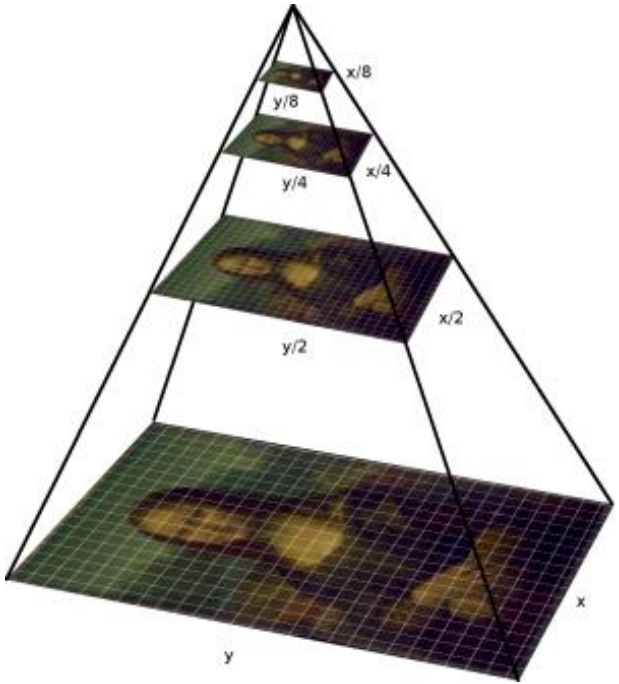
Why Tensor Mask?



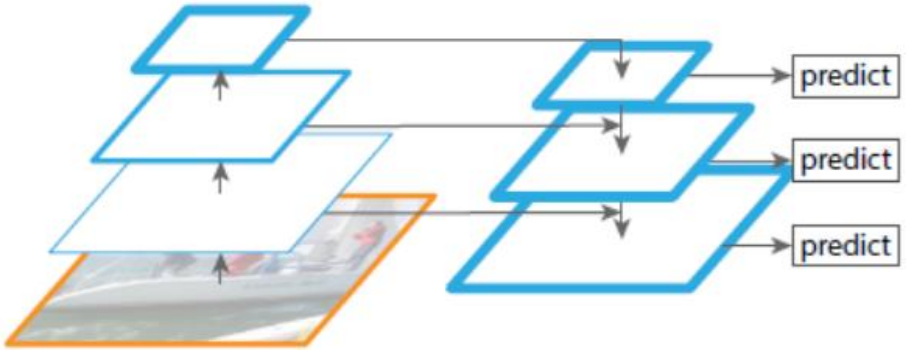
- Tensor Mask proposes 4D Tensors (V,U,H,W)
- We convert channel axes to a new dimension (V,U)

UCF

Image Pyramid



Feature Pyramid



Feature Pyramid Network

UCF

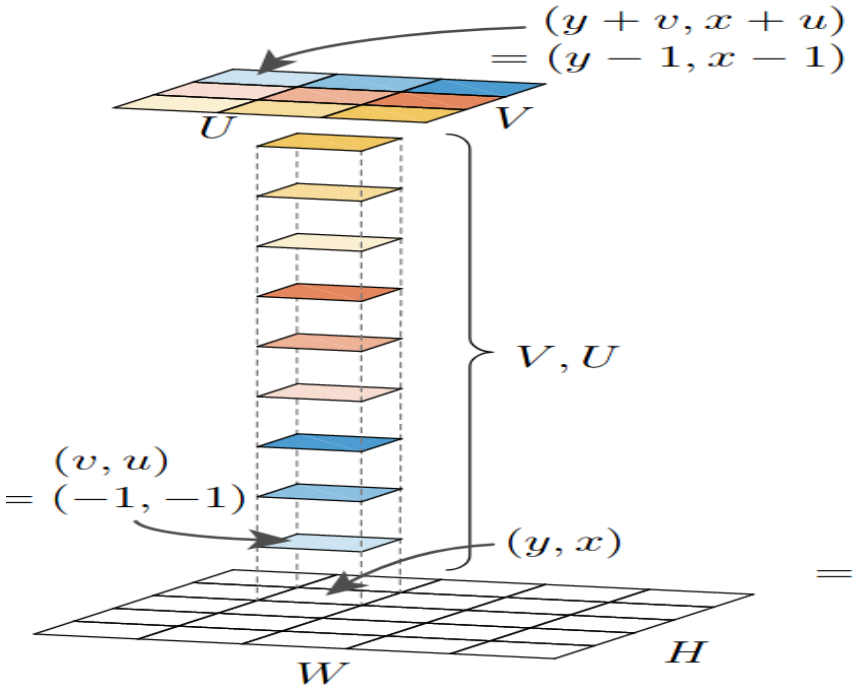
Tensor Representation for Masks

(V,U,H,W)

- Higher dimensional (4D) tensor with shape (V,U,H,W).
- Sub-tensor(V,U) represents a mask as a 2D spatial entity.
- Use these high dimensional Tensors to represent image content
- Dense set of sliding windows

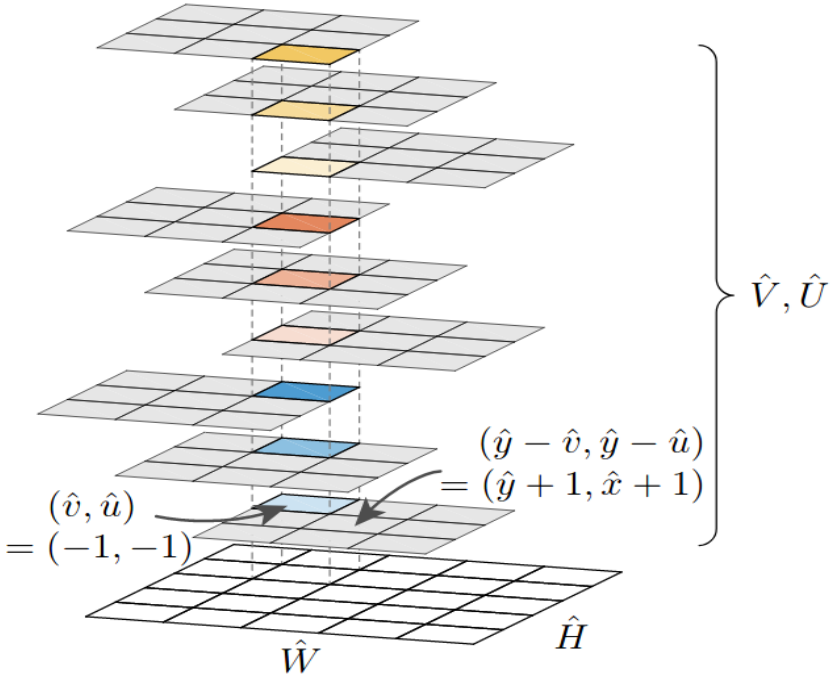
UCF

Tensor Representation for Masks



Natural Representation

Tensor Representation for Masks



Aligned Representation

Tensor Representation for Masks

$$\mathcal{F}(v, u, y, x) = \hat{\mathcal{F}}(v, u, y + \alpha v, x + \alpha u).$$

Coordinate Transformation

Tensor Representation for Masks

Upscale Transformation **(λ)**

- Coarse(ˆV ,ˆU) sub-tensors to create Finer(V,U) sub-tensors
- Generates high-resolution masks without inflating channel counts in preceding feature map.

UCF

Tensor Representation for Masks

Tensor BiPyramid

- Number of mask pixels based on scale
- List containing masks
- Shape of Tensors in the list:

$$(2^k V, 2^k U, \frac{1}{2^k} H, \frac{1}{2^k} W)$$

UCF

Tensor Mask Architecture

1. **Mask Prediction Head**
   Generate Masks in Sliding window

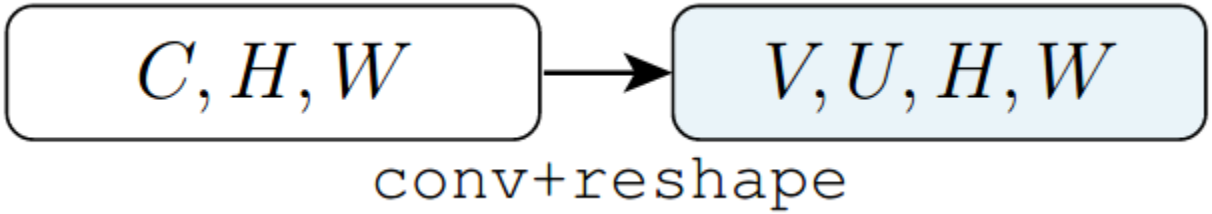
1. **Classification Head**
   Predicts object categories

   Similar to bounding box detection

Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window

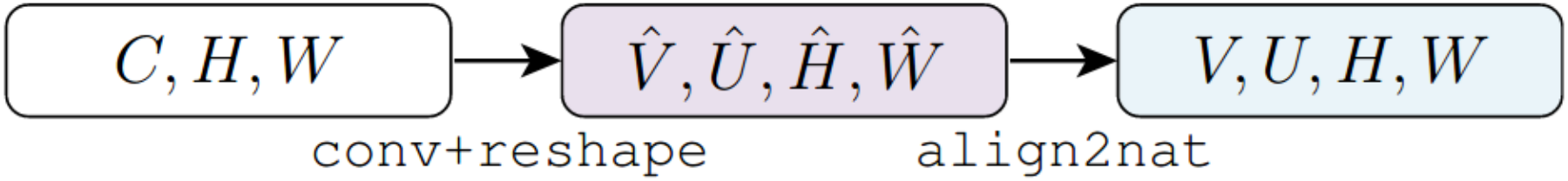


UCF

Tensor Mask Architecture

Mask Prediction Head

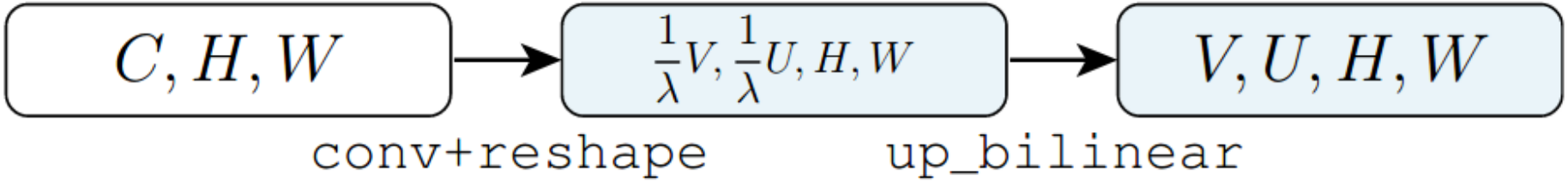
Generate Masks in Sliding window



Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window

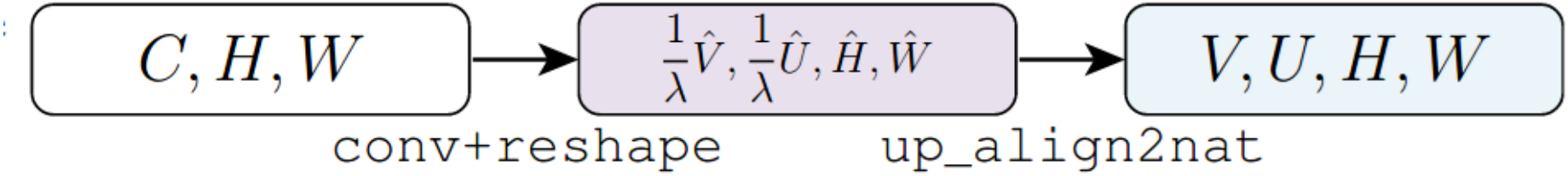


UCF

Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window

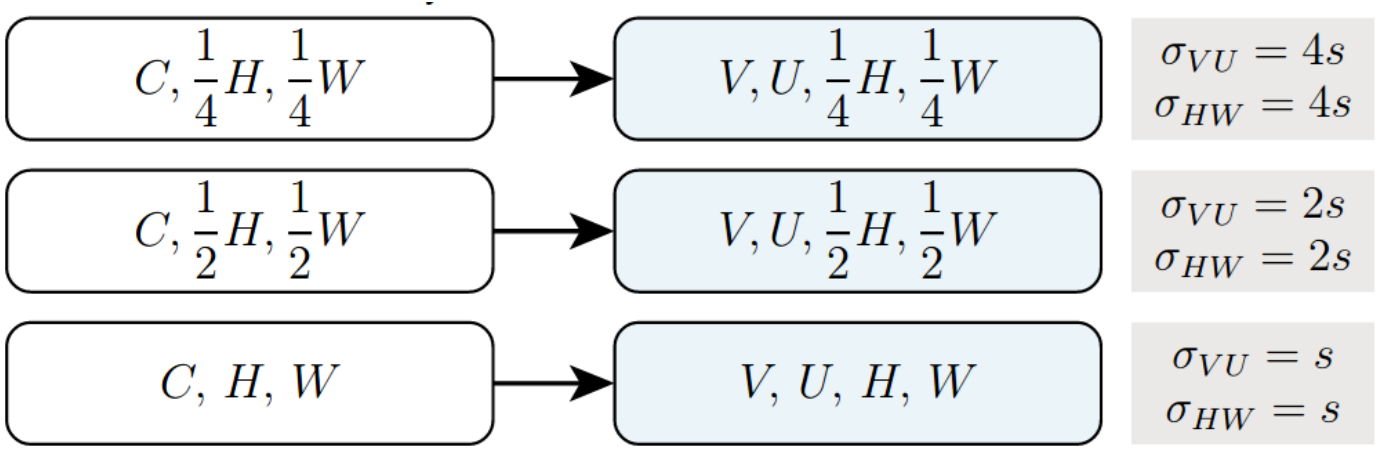


UCF

Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window

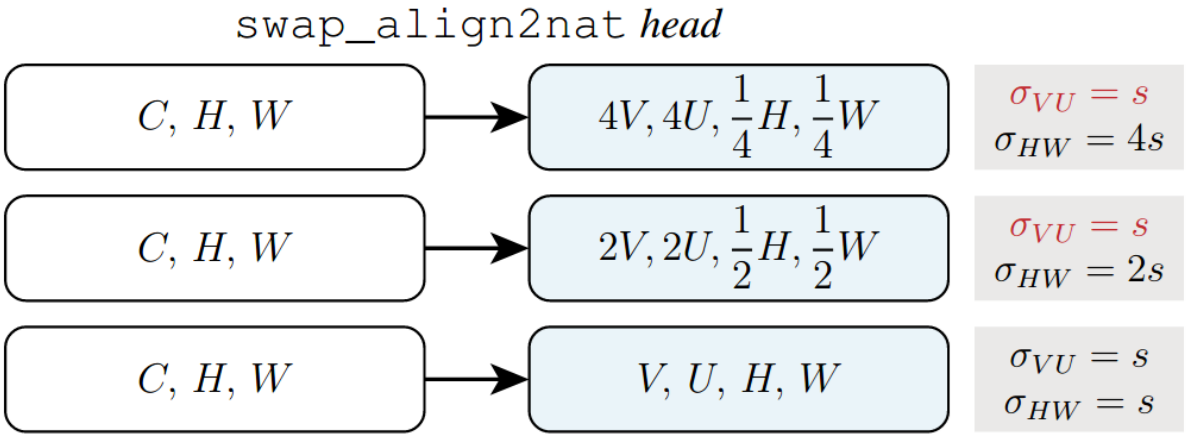


UCF

Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window

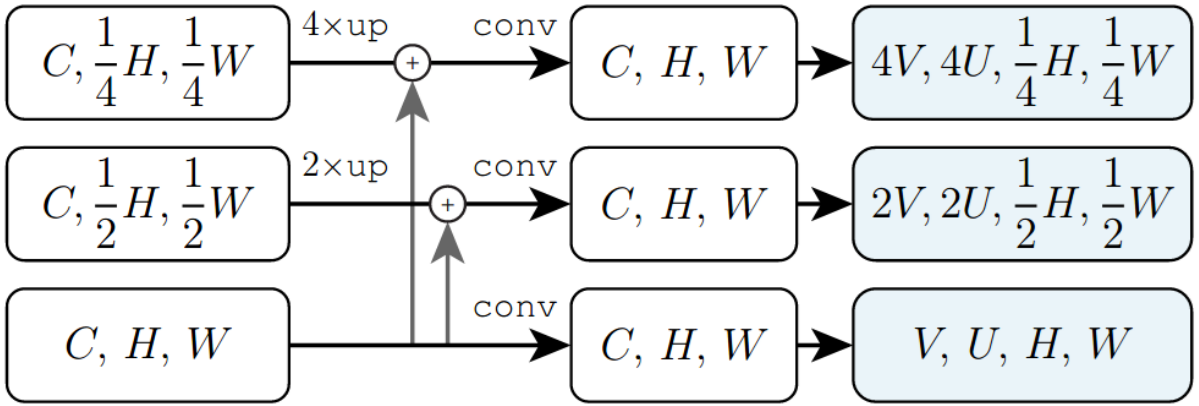


UCF

Tensor Mask Architecture

Mask Prediction Head

Generate Masks in Sliding window



UCF

Training

Label Assignment for ground truth mask(“m”):

- Containment

  Window fully contains “m” and the longer side of “m”

UCF

Training

Label Assignment:

- Centrality

Center of “m”’ bounding box is within one unit (σ_{vu}) of the window center in L2 distance

UCF

Training

Label Assignment:

- Uniqueness

There is no other mask m' = m that satisfies the other two conditions

Tensor Mask : Losses

- **For prediction of mask head:**
  Per Pixel Binary classification Loss

- **For box regression:**
  L1 Loss

- **Classification head:**
  Focal Loss

UCF

Dataset: Coco





http://cocodataset.org/#detection-eval

UCF

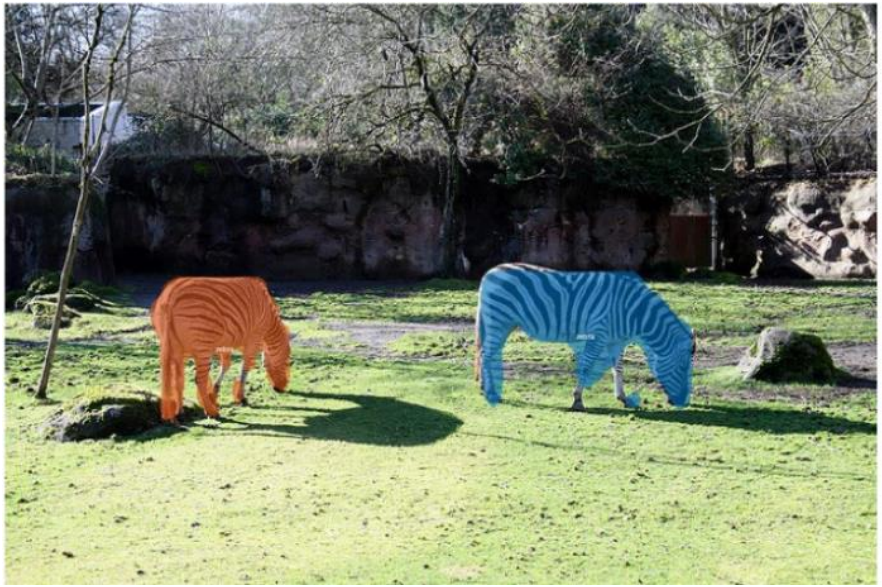
Results



Natural up-scaling head

UCF

Results





Unaligned head

UCF

Results





Aligned head

UCF

Results

head	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
natural	28.5	52.2	28.6	14.4	30.2	40.1
aligned	28.9	52.5	29.3	14.6	30.8	40.7

Simple heads: natural vs. aligned with V×U=15×15

UCF

Results

head	λ	AP	AP_{50}	AP_{75}	Δ aligned - natural		
natural	1.5	28.0	51.7	27.8	*+0.9*	*+0.7*	*+1.5*
aligned		**28.9**	**52.4**	**29.3**			
natural	3	24.7	48.4	22.7	*+4.1*	*+3.9*	*+6.4*
aligned		**28.8**	**52.3**	**29.1**			
natural	5	19.2	42.1	15.6	*+9.2*	*+9.7*	*+13.0*
aligned		**28.4**	**51.8**	**28.6**			

Upscaling heads: Natural vs. aligned heads (V×U=15×15)

UCF

Results

head	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
feature pyramid, best	28.9	52.5	29.3	14.6	30.8	40.7
tensor bipyramid	**34.0**	**55.2**	**35.8**	**15.3**	**36.3**	**48.4**
Δ	*+5.1*	*+2.7*	*+6.5*	*+0.7*	*+5.5*	*+7.7*

Beast Baseline head on Feature Pyramid vs Tensor bipyramid

UCF

Results

$V \times U$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
15×15	34.0	55.2	35.8	15.3	36.3	48.4
$15 \times 15, 11 \times 11$	**35.2**	**56.4**	**37.0**	**17.4**	**37.4**	**49.7**
Δ	*+1.2*	*+1.2*	*+1.2*	*+2.1*	*+1.1*	*+1.3*

Window Sizes

UCF

Results

method	backbone	aug	epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [13]	R-50-FPN		24	34.9	57.2	36.9	15.4	36.6	50.8
Mask R-CNN, *ours*	R-50-FPN		24	34.9	56.8	36.8	15.1	36.7	50.6
Mask R-CNN, *ours*	R-50-FPN	✓	72	**36.8**	**59.2**	**39.3**	**17.1**	**38.7**	**52.1**
TensorMask	R-50-FPN	✓	72	35.4	57.2	37.3	16.3	36.8	49.3
Mask R-CNN, *ours*	R-101-FPN	✓	72	**38.3**	**61.2**	**40.8**	**18.2**	**40.6**	**54.1**
TensorMask	R-101-FPN	✓	72	37.1	59.3	39.4	17.4	39.1	51.6

Comparison with Mask R-CNN

UCF

Results

method	aug	epochs	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RetinaNet, *ours*		24	37.1	55.0	39.9
RetinaNet, *ours*	✓	72	39.3	57.2	42.4
Faster R-CNN, *ours*	✓	72	40.6	61.4	44.2
Mask R-CNN, *ours*	✓	72	41.7	62.5	45.7
TensorMask, *box-only*	✓	72	40.8	60.4	43.9
TensorMask	✓	72	41.6	61.0	45.1

Object detection

UCF

Results

box head	NMS	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
✓	bb	35.2	56.4	37.0	41.6	60.8	44.8
✓	mask-bb	34.9	56.0	36.7	39.7	59.1	41.8
	mask-bb	34.8	56.1	36.7	39.4	58.8	41.6

Multi-task benefits

UCF

Results



Mask R-CNN (top row) and TensorMask (bottom row)

UCF

Results



Mask R-CNN (top row) and TensorMask (bottom row)

UCF

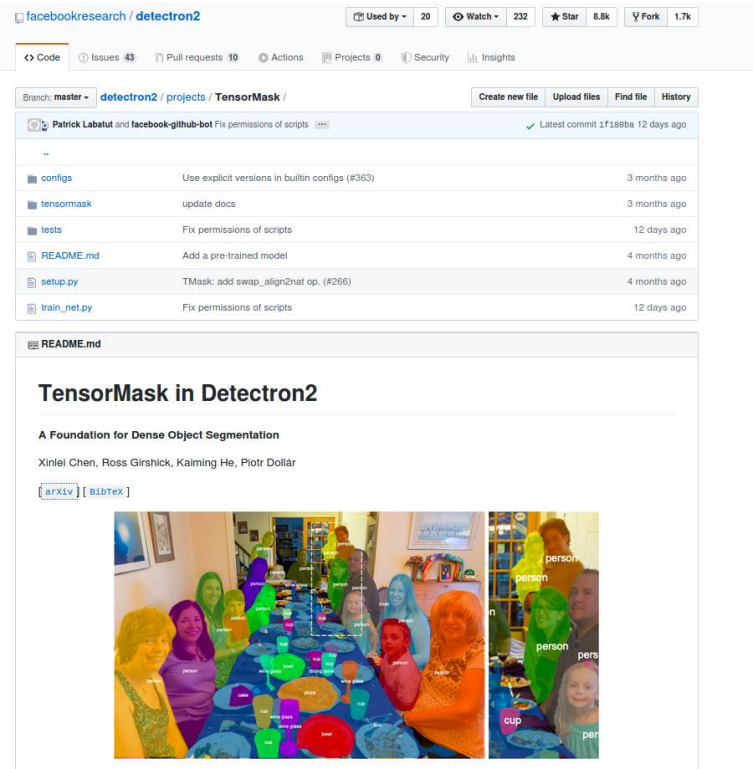
Results



Mask R-CNN (top row) and TensorMask (bottom row)

UCF

Implementation



https://github.com/facebookresearch/detectron2/tree/master/projects/TensorMask



Detectron2

A research platform and a production library
for **object detection**,
mainly built by Facebook AI Research (FAIR)

https://github.com/facebookresearch/detectron2

What's in Detectron2: model zoo

- Different settings for users to play with
- Standard baselines for researchers
- Efficient models for production (coming soon)

Faster R-CNN:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50-C4	1x	0.593	0.110	4.8	35.7	137257644	model | metrics
R50-DC5	1x	0.380	0.068	5.0	37.3	137847829	model | metrics
R50-FPN	1x	0.210	0.055	3.0	37.9	137257794	model | metrics
R50-C4	3x	0.589	0.110	4.8	38.4	137849393	model | metrics
R50-DC5	3x	0.378	0.073	5.0	39.0	137849425	model | metrics
R50-FPN	3x	0.209	0.047	3.0	40.2	137849458	model | metrics
R101-C4	3x	0.656	0.149	5.9	41.1	138204752	model | metrics
R101-DC5	3x	0.452	0.082	6.1	40.6	138204841	model | metrics
R101-FPN	3x	0.286	0.063	4.1	42.0	137851257	model | metrics
X101-FPN	3x	0.638	0.120	6.7	43.0	139173657	model | metrics

RetinaNet:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50	1x	0.200	0.062	3.9	36.5	137593951	model | metrics
R50	3x	0.201	0.063	3.9	37.9	137849486	model | metrics
R101	3x	0.280	0.080	5.1	39.9	138363263	model | metrics

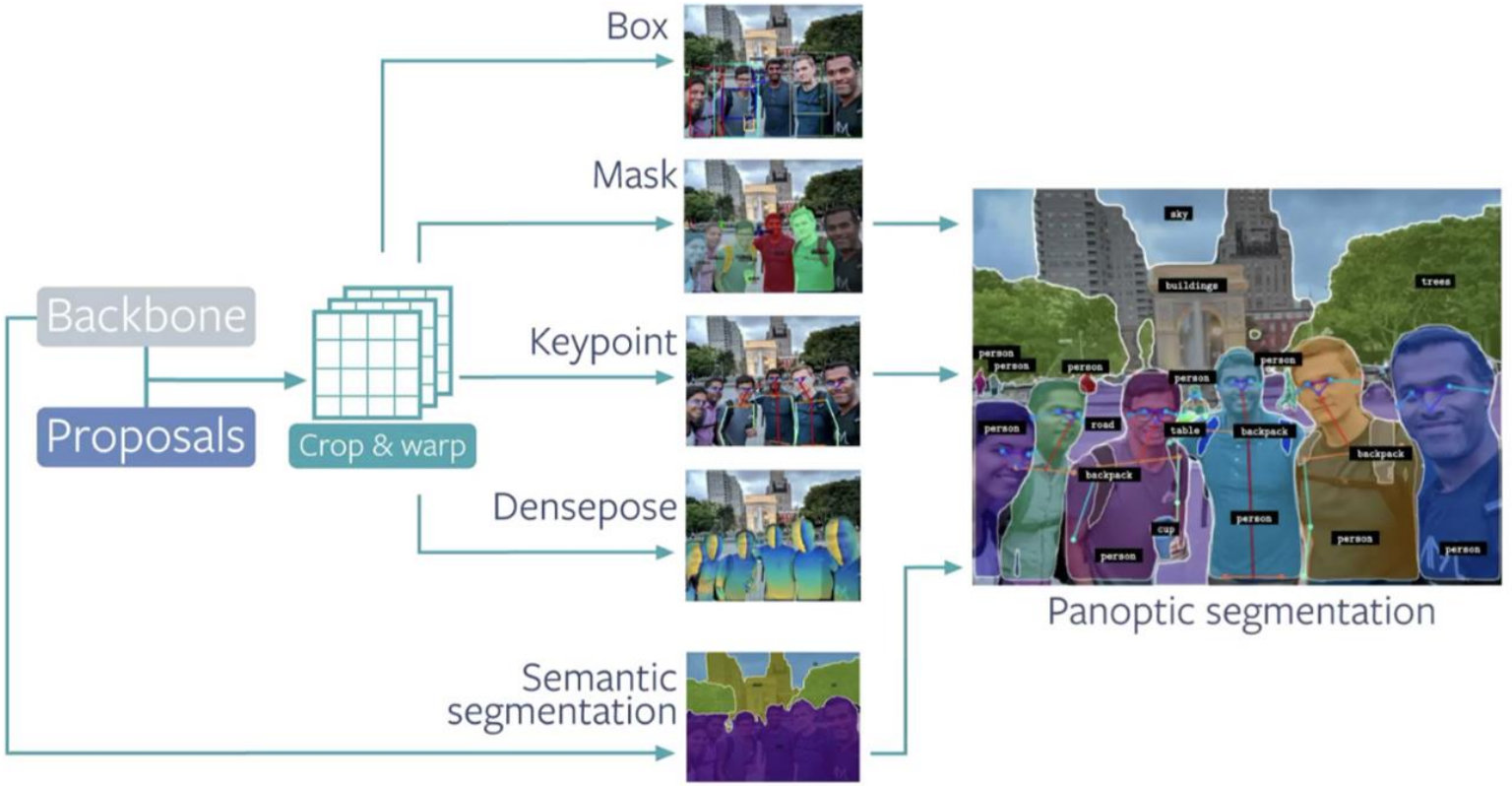
RPN & Fast R-CNN:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	prop. AR	model id	download
RPN R50-C4	1x	0.130	0.051	1.5		51.6	137258005	model | metrics
RPN R50-FPN	1x	0.186	0.045	2.7		58.0	137258492	model | metrics

https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md

What's in Detectron2: Generalized R-CNN Models

(+ a few other types of models: RetinaNet, TensorMask, etc.)




Thank You
UCF