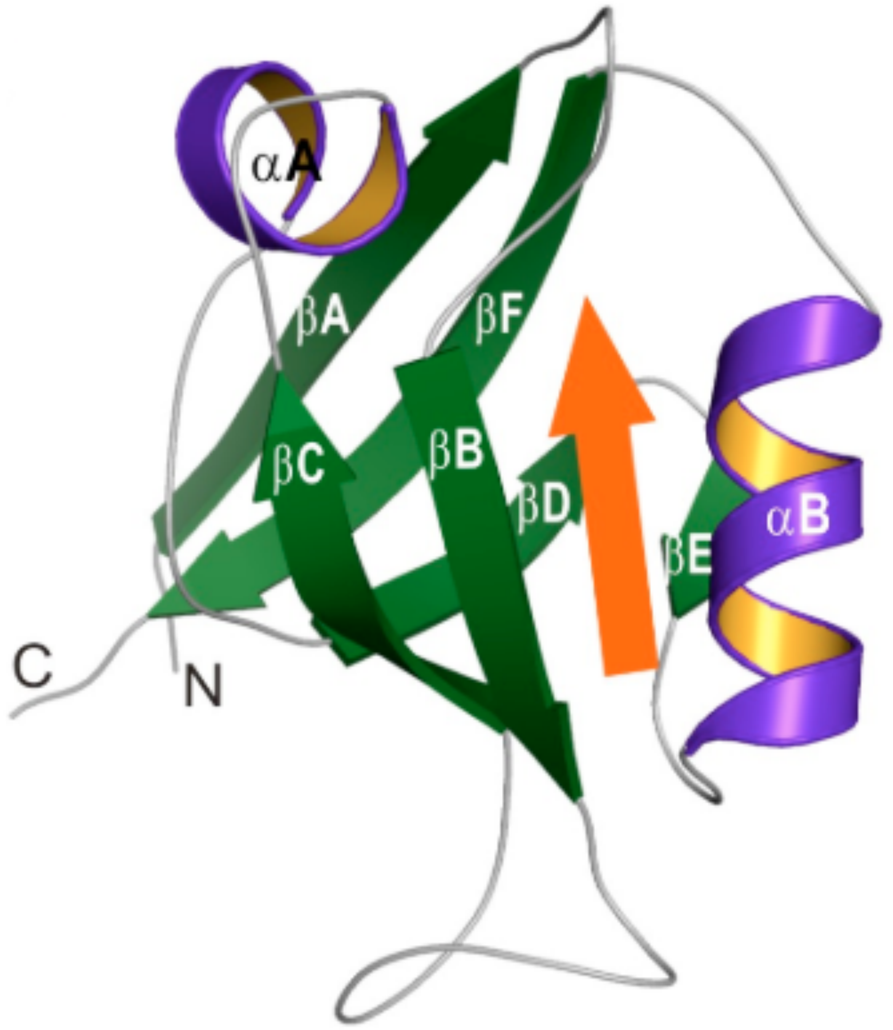
Instance Segmentation

Models

Josh Myers-Dean (he/him) - CSCI 7000 - 9/27/2021

About Me

- 1st year PhD Student
- Image and Video Computing Group
- Undergrad at Western Washington University
- Interned for a year at PNNL
- Features are cool!
- Skiing, running, etc.

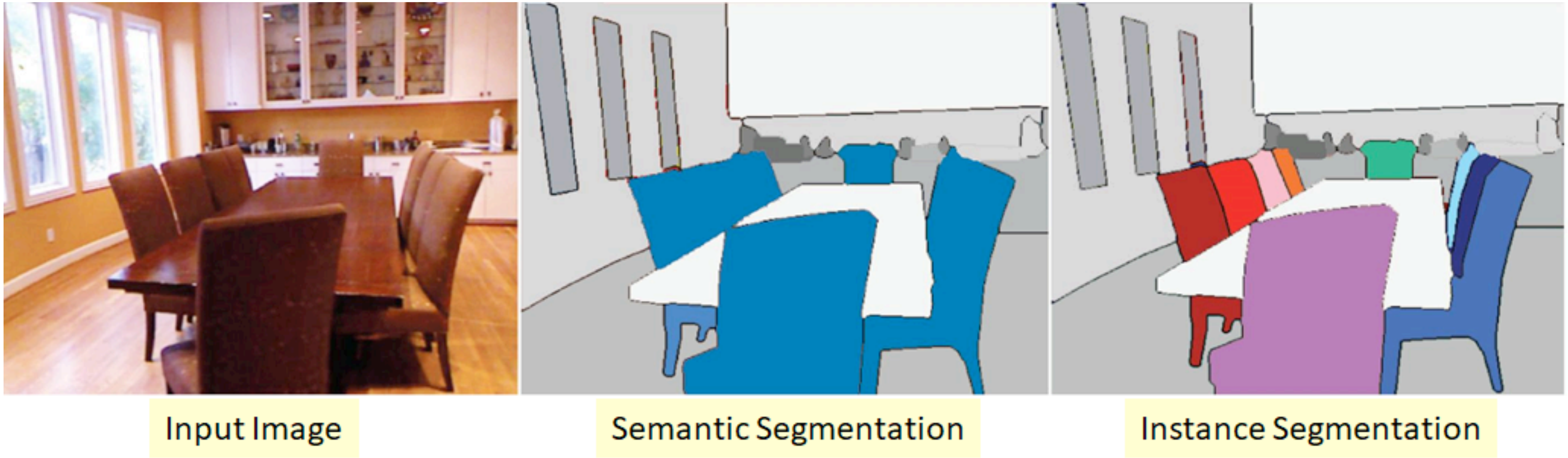


Upper Left: credit Morgan Sjogren Lower image: https://facultyweb.cs.wwu.edu/~wehrwes/semantic_pixels/ Upper Right: https://facultyweb.cs.wwu.edu/~jagodzf/publications.html

Review: What is Instance Segmentation?

Detect instances, give category, label pixels

"Simultaneous detection and segmentation"



Slides adapted from Standford CS231n Lecture 11, image from https://towardsdatascience.com/review-deepmask-instance-segmentation-30327a072339

Overview

- Faster R-CNN Recap
- Mask R-CNN
- SWIN Transformer
- Discussion

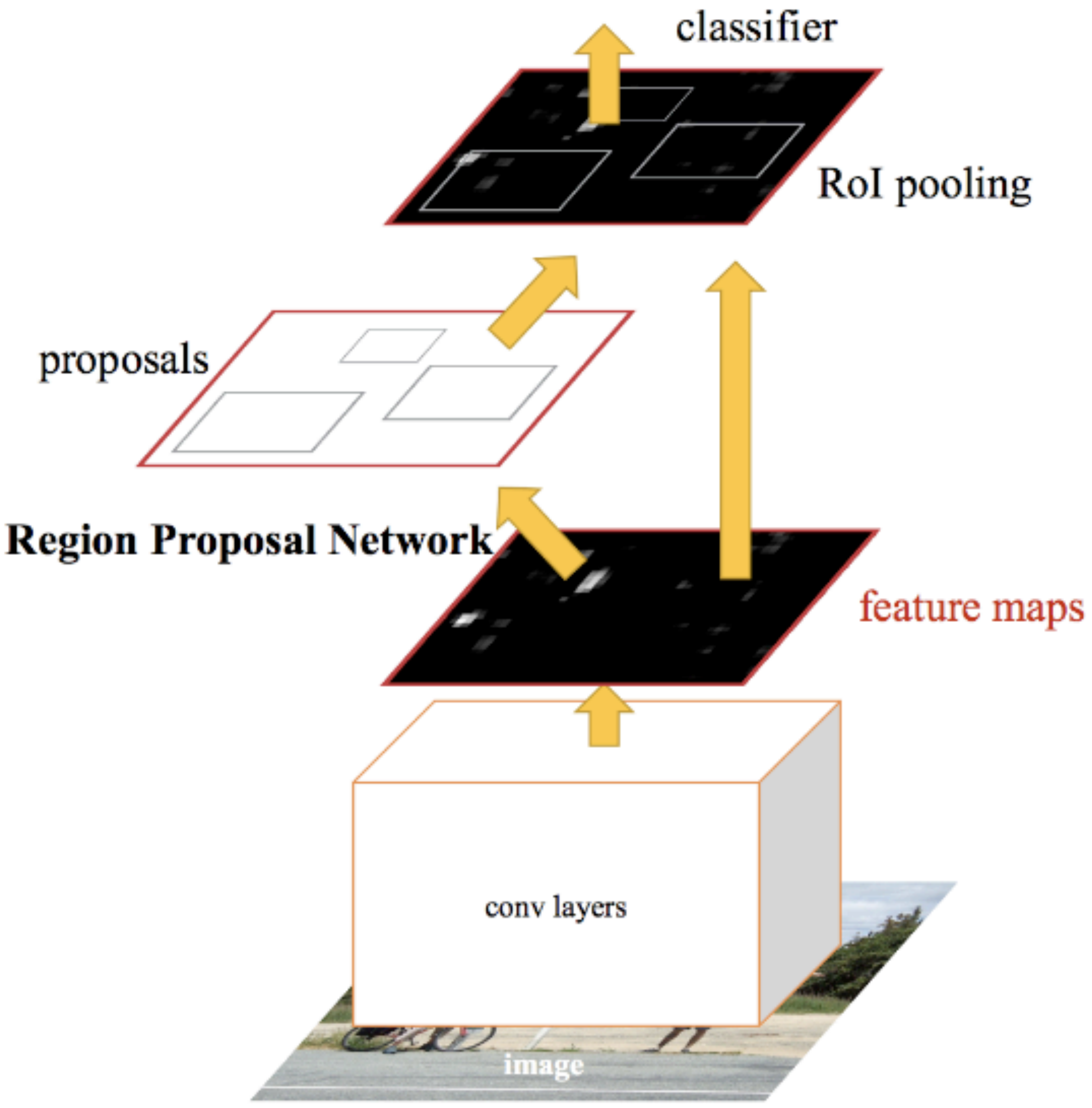
Overview

- Faster R-CNN Recap
- Mask R-CNN
- SWIN Transformer
- Discussion

Faster R-CNN: Architecture Recap

Composed of a region proposal network and a Fast R-CNN classifier

Main novelty was using RPN instead of selective search



Ren Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Neurips 2015.

Faster R-CNN: ROI Pooling Motivation

Want: Fixed sized feature maps for classification

Challenge: Proposals can be of different sizes

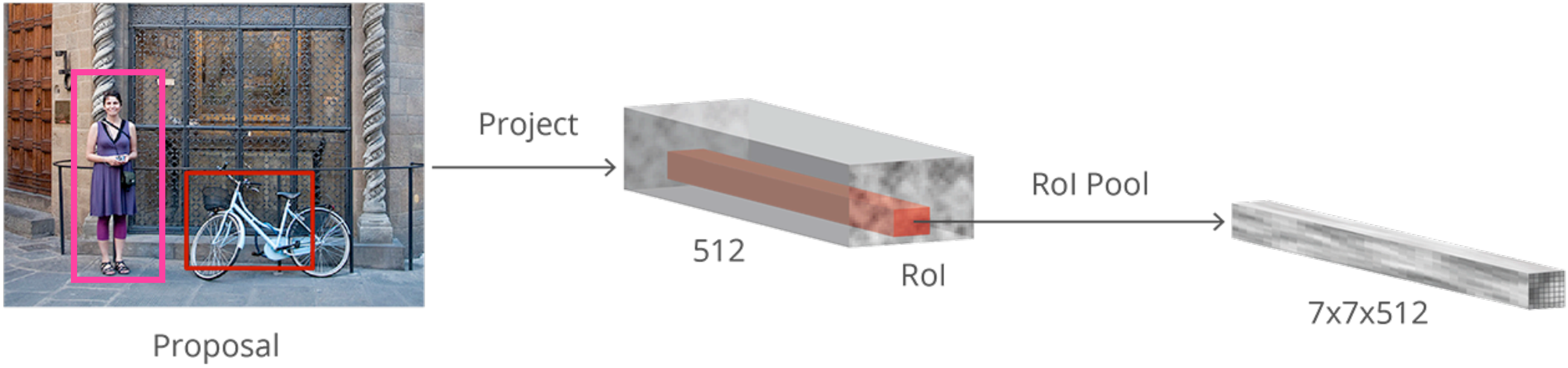


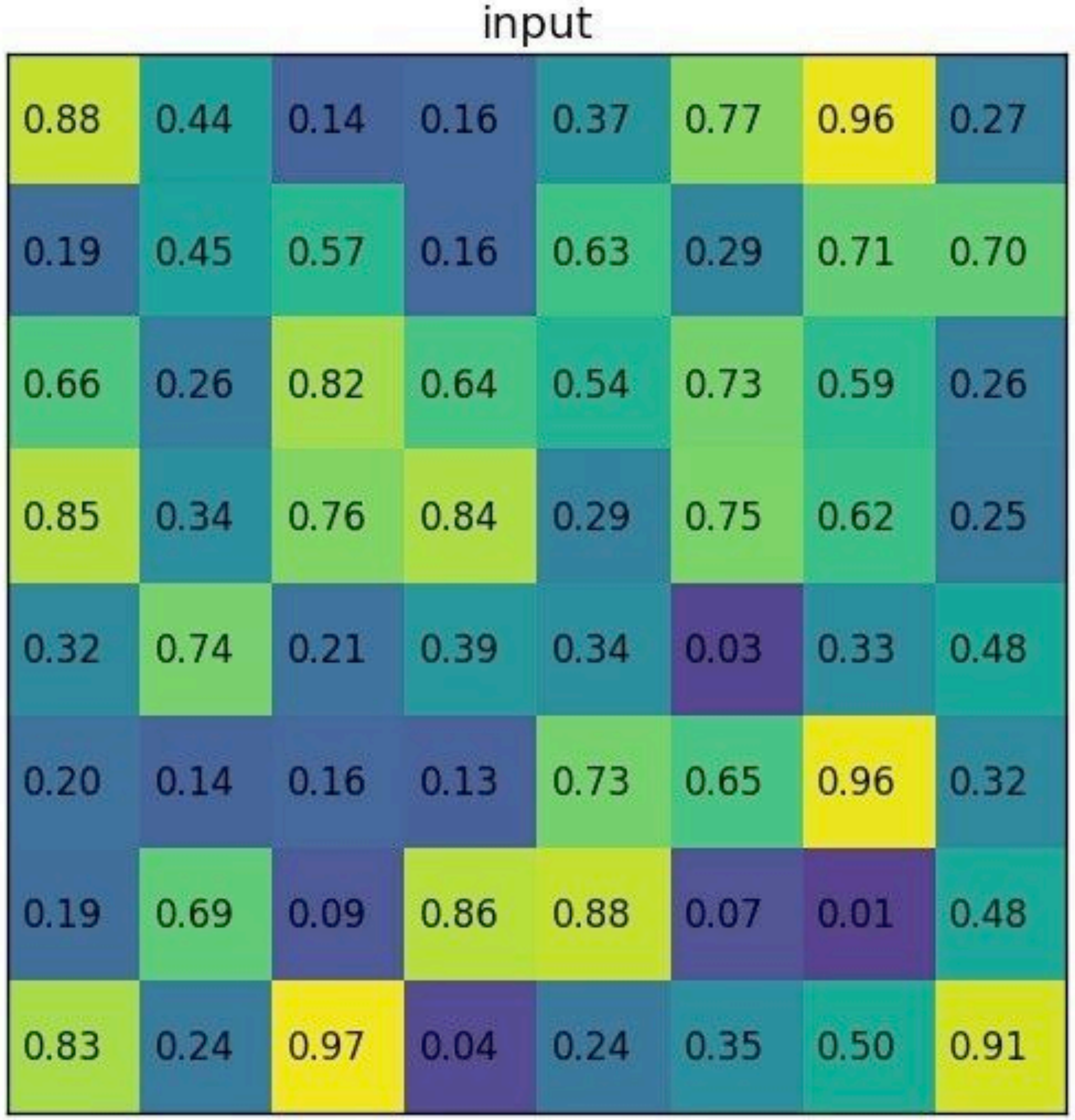
Image: https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

Faster R-CNN: ROI Pooling Mechanics

Divide the feature map into **a quantized grid** of kxk pixels then do max pool

Quantized: $\left\lfloor \frac{height/width}{k} \right\rfloor$

This will be the size of our grid cell



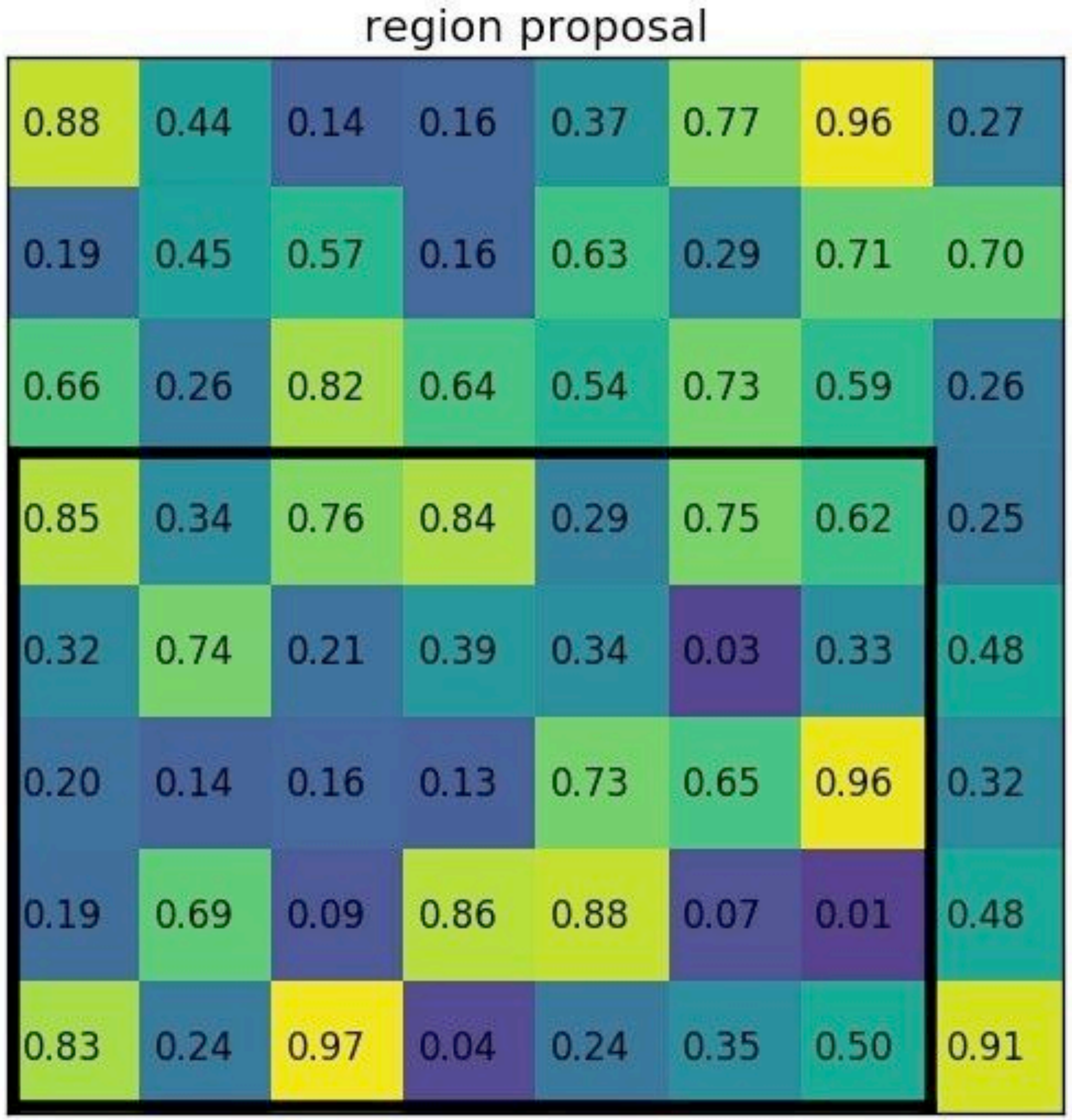
Source: https://deepsense.ai/region-of-interest-pooling-explained/

Faster R-CNN: ROI Pooling Mechanics

Example: We have a $5x7$ region and want a $2x2$ output

Dimensions of each grid cell?

$$\left\lfloor \frac{height/width}{k} \right\rfloor$$



Source: https://deepsense.ai/region-of-interest-pooling-explained/

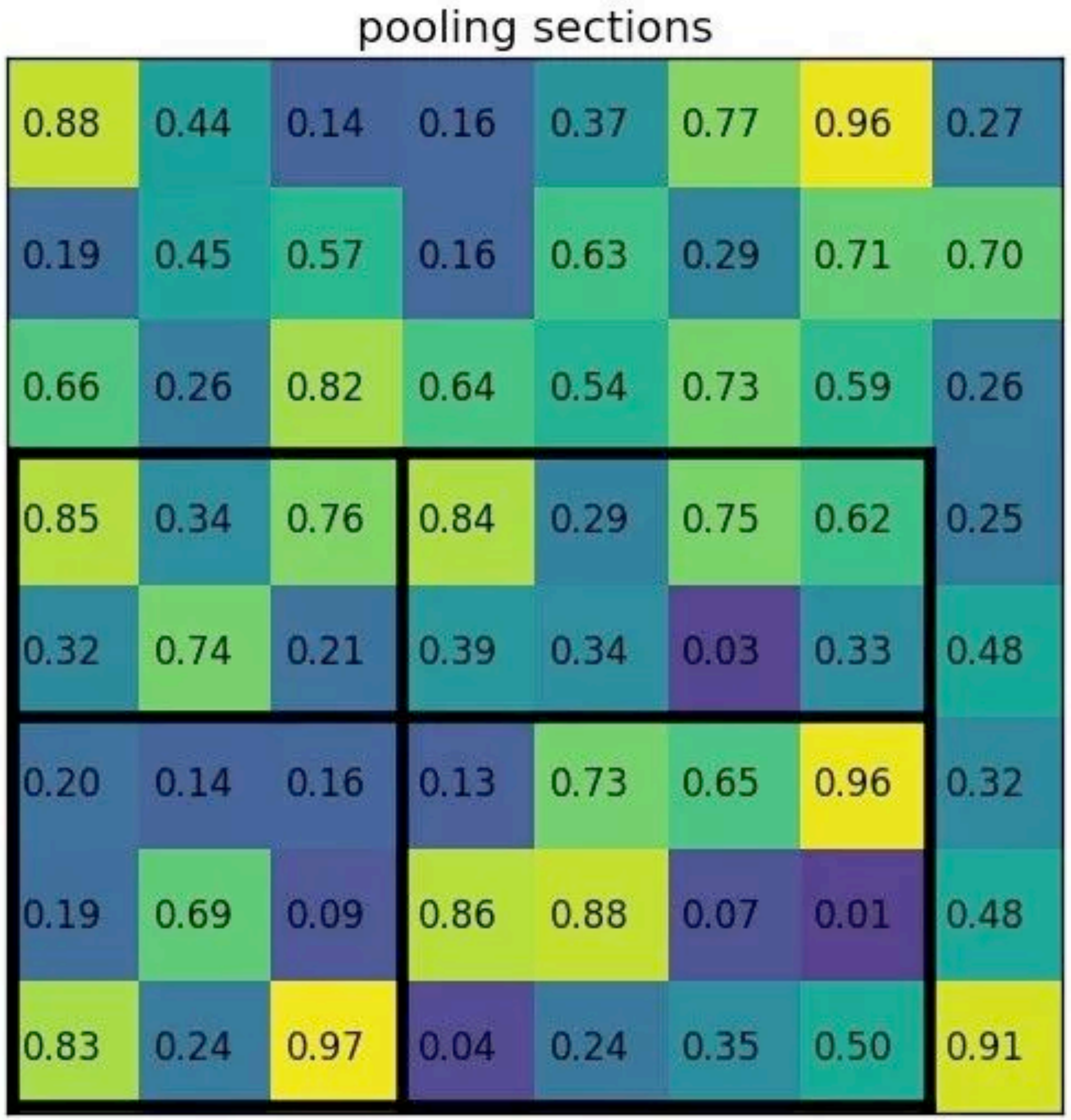
Faster R-CNN: ROI Pooling Mechanics

Example: We have a $5x7$ region and want a $2x2$ output

Dimension of each grid cell?

$$\left\lfloor \frac{5}{2} \right\rfloor, \left\lfloor \frac{7}{2} \right\rfloor = (2,3)$$

Have to adjust if proposal dimensions are uneven

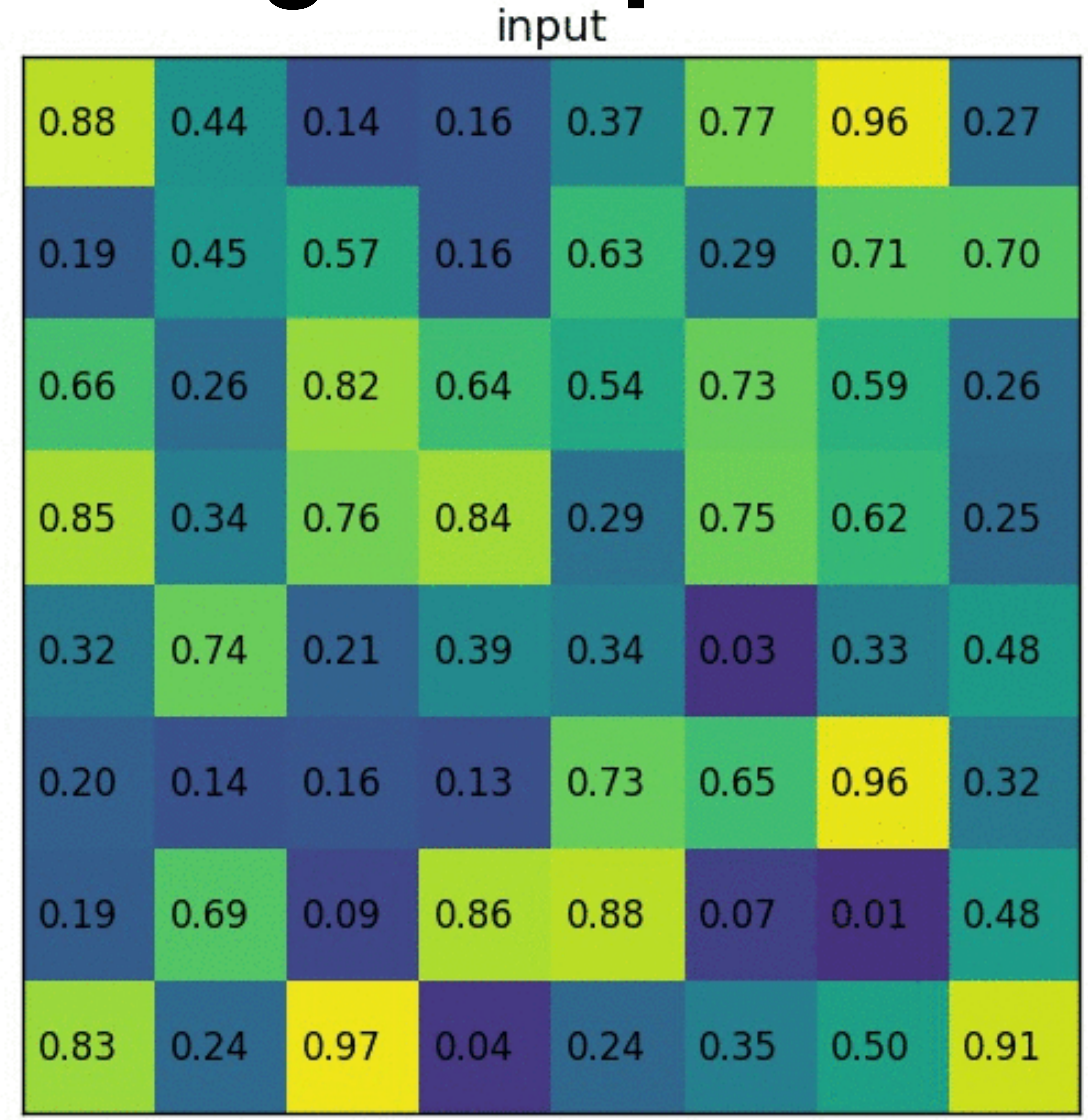


Source: https://deepsense.ai/region-of-interest-pooling-explained/

Faster R-CNN: ROI Pooling Recap

Full example

- Quantize
- Divide into $2x2$ grid
- Pool



Source: https://deepsense.ai/region-of-interest-pooling-explained/

Faster R-CNN: Limitation

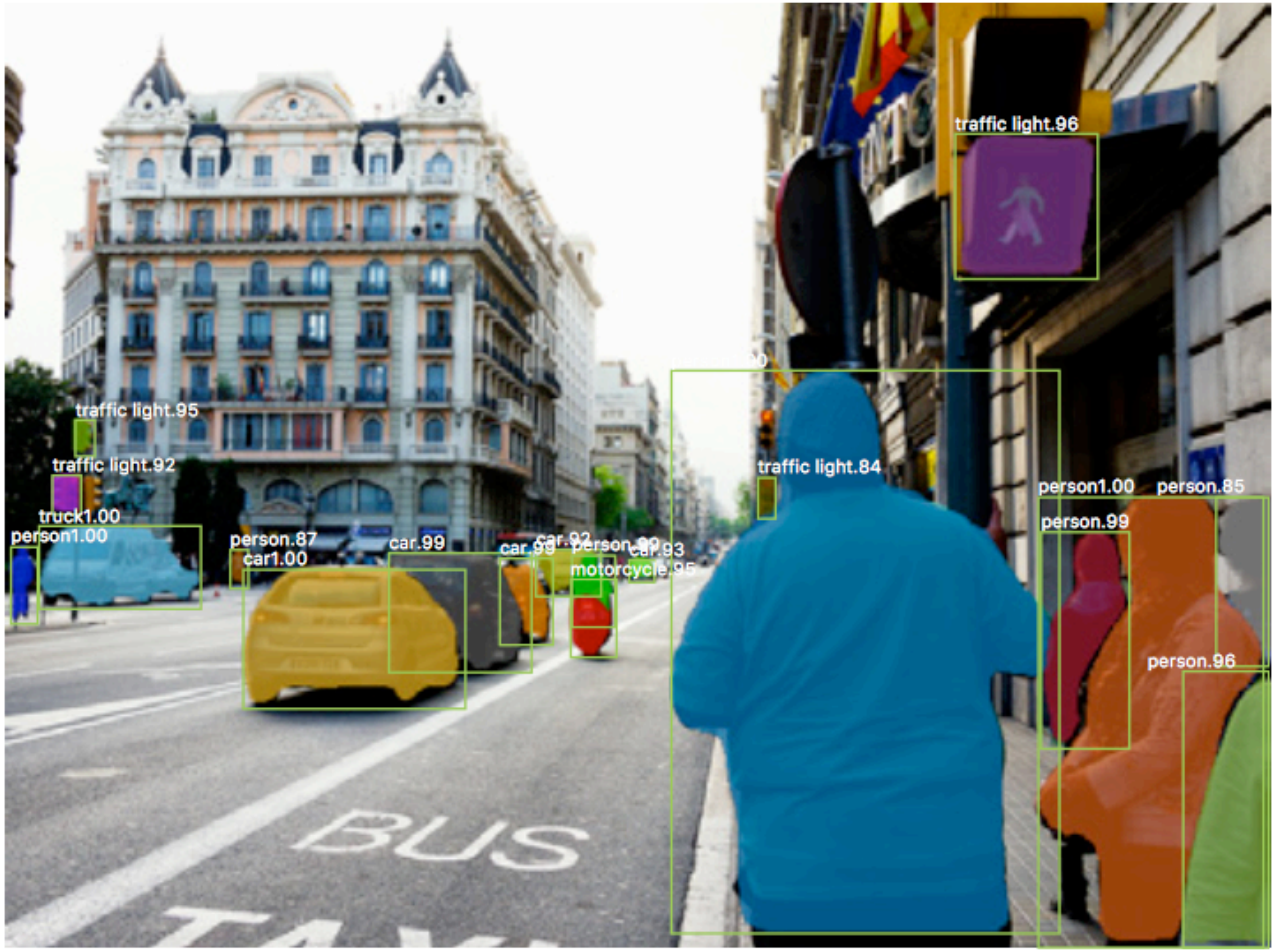
What if we want pixel-wise annotations?

Overview

- Faster R-CNN Recap
- Mask R-CNN
- SWIN Transformer
- Discussion

Mask R-CNN

Faster R-CNN but make it pixel accurate



Source: https://medium.com/@umerfarooq_26378/from-r-cnn-to-mask-r-cnn-d6367b196cfd

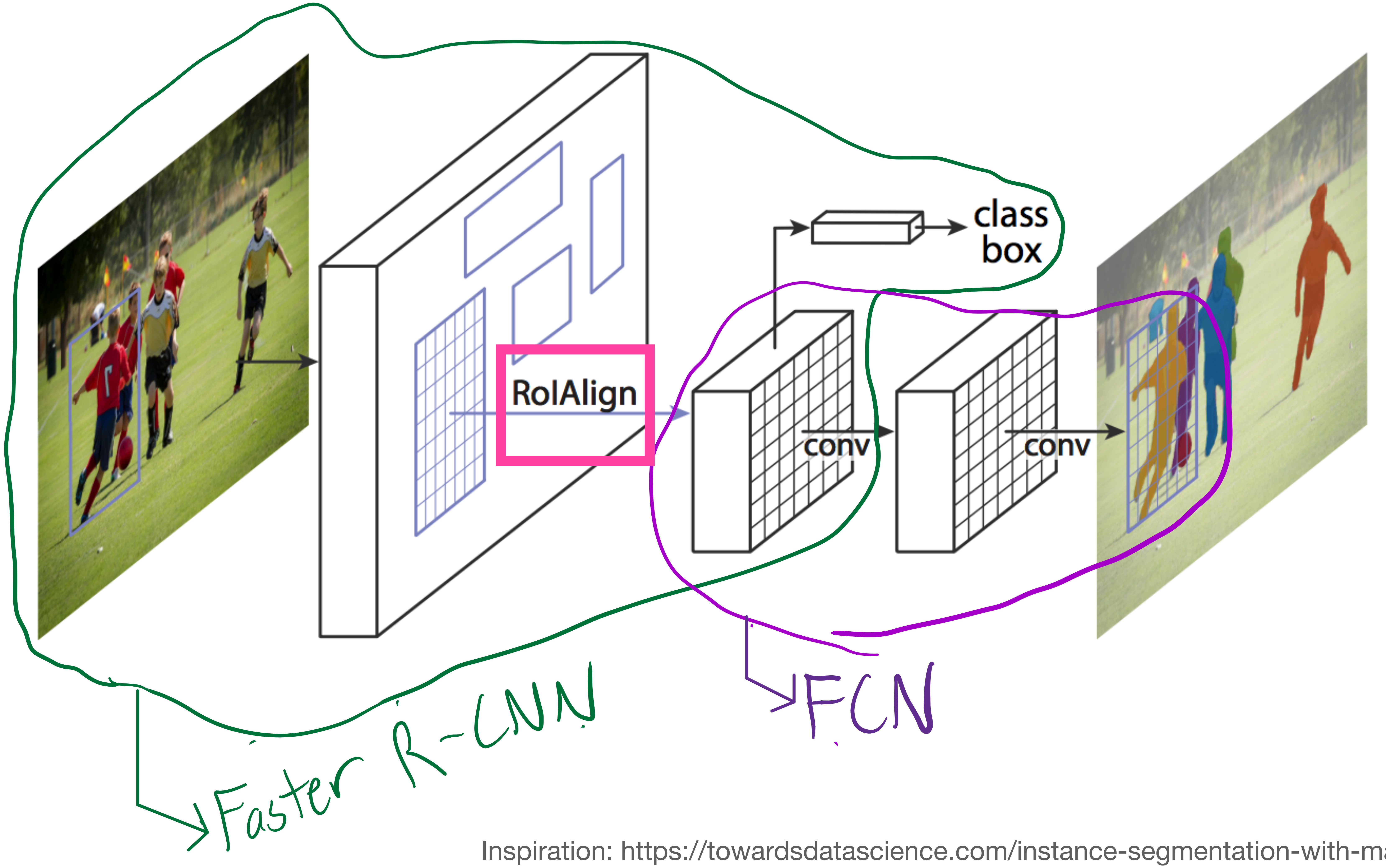
Mask R-CNN: Main Contributions

1. Add an additional branch to Faster R-CNN
2. ROI Align
3. Learn mask in parallel
4. Simple and end-to-end

Source: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

Mask R-CNN: A Natural Extension

Mask R-CNN = Faster R-CNN + FCN



Inspiration: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

Mask R-CNN: ROI Alignment Motivation

We do not want quantization for pixel-wise accuracy

Why?

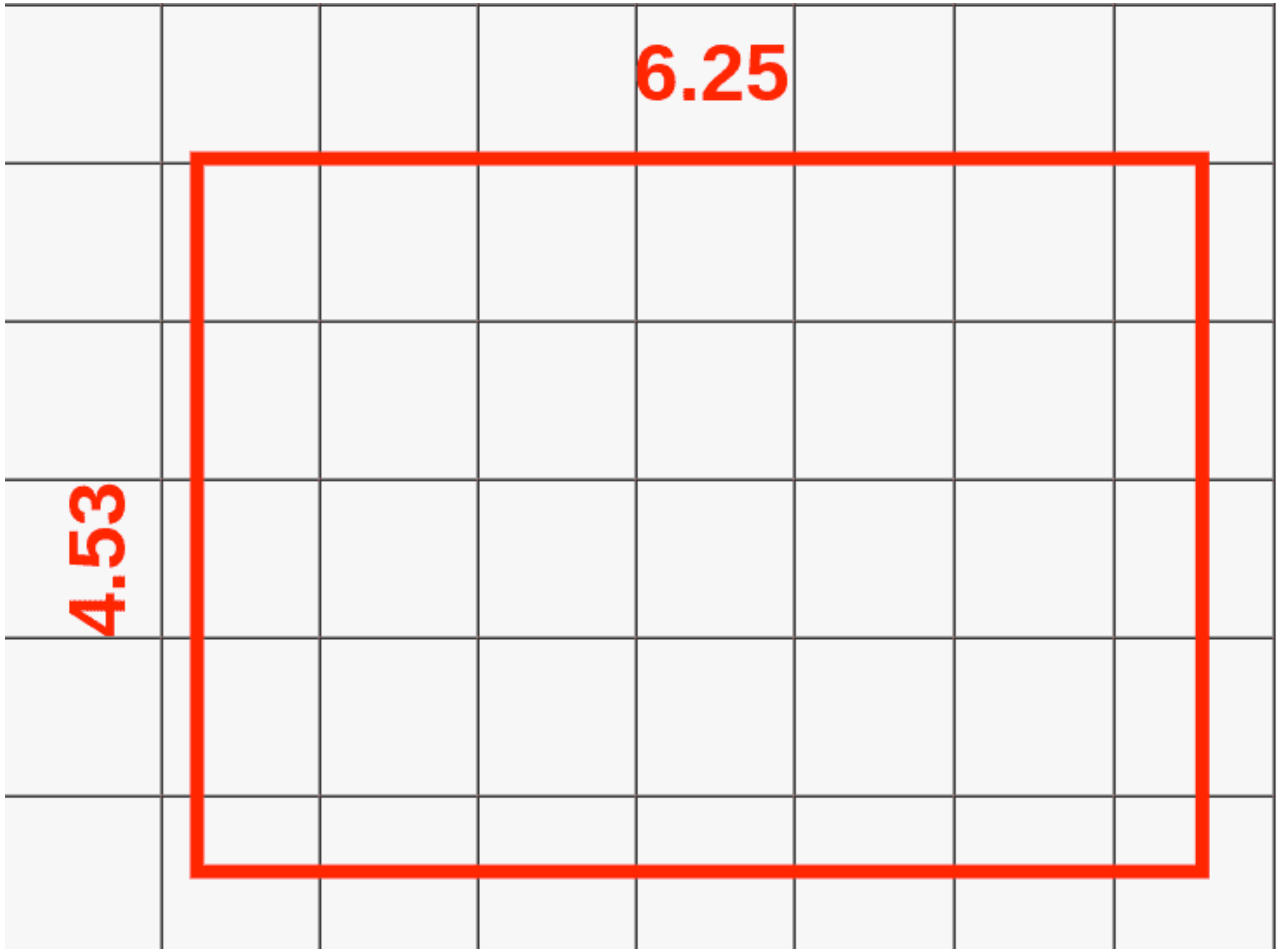
More sensitive to perturbations

RPN regress a bounding box

Proposal can have floating point coordinates

We don't know the value at floating point coordinates

Losing data —> losing precision



Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

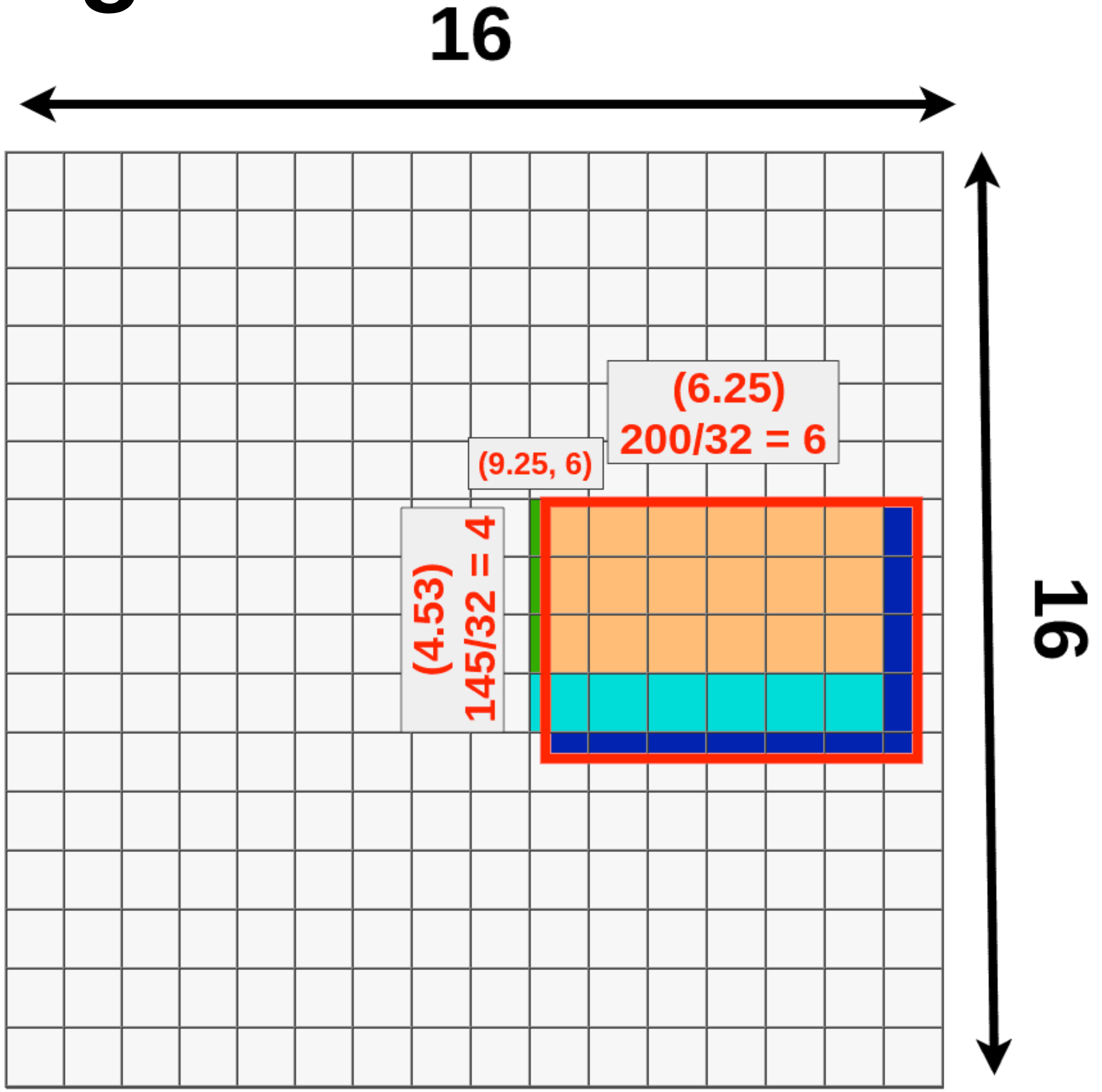
Mask R-CNN: ROI Alignment Motivation

Suppose we map a image to a 16x16xd feature map and the proposal has floating point coordinates, want a 3x3 output

Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

Mask R-CNN: ROI Alignment Motivation

Quantization causes us to lose data!



Green = info gained

Dark Blue = info lost

Light Blue = info lost if enforcing equal partitions

Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

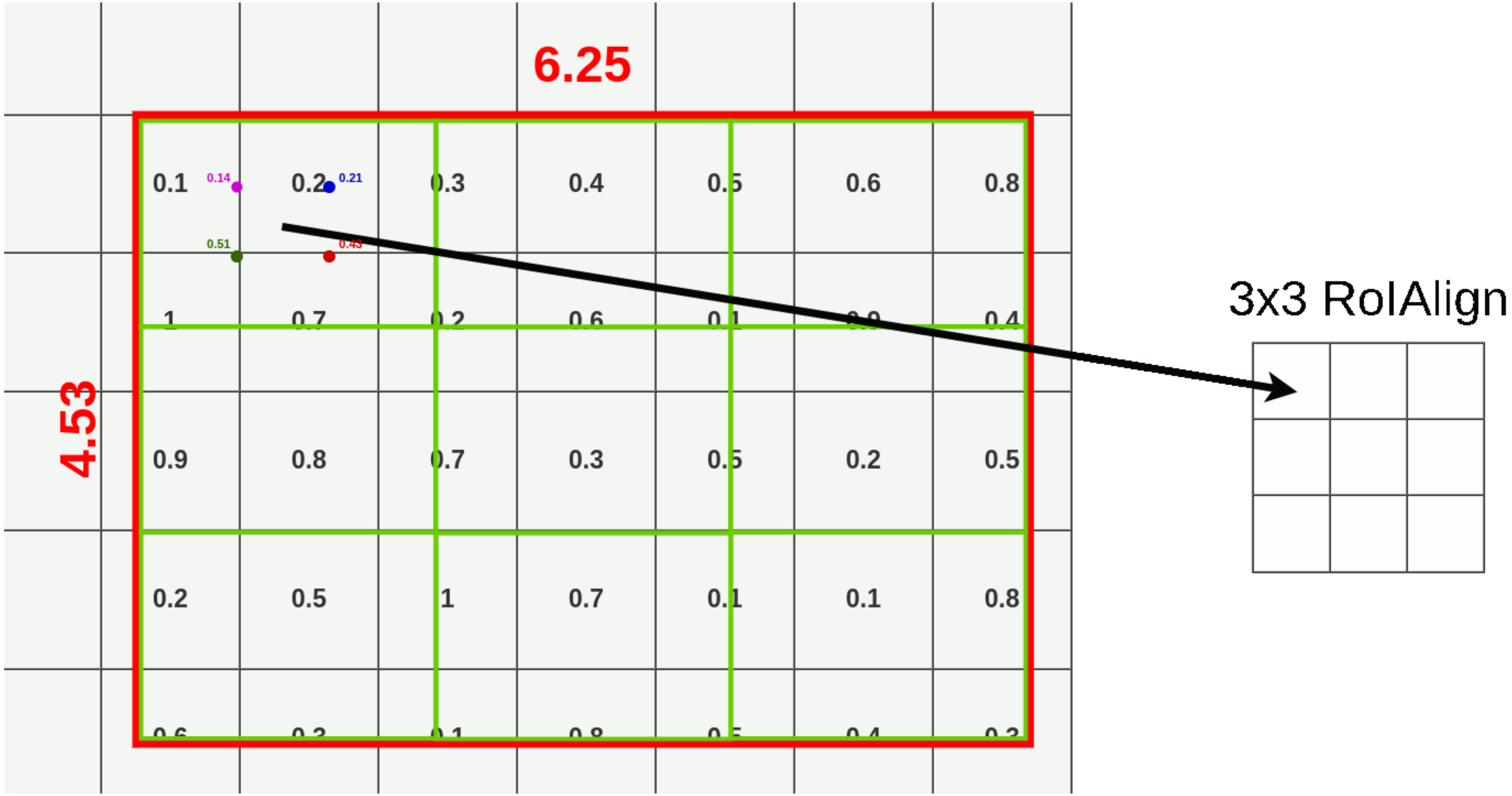
Mask R-CNN: ROI Alignment Mechanics

- Quantize without floor
- Divide into 3x3 grid
- Sample *n* points in each cell
- Bilinear interpolation
- Repeat

Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

Mask R-CNN: ROI Alignment Mechanics

Max pool into output map from newly computed values



Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

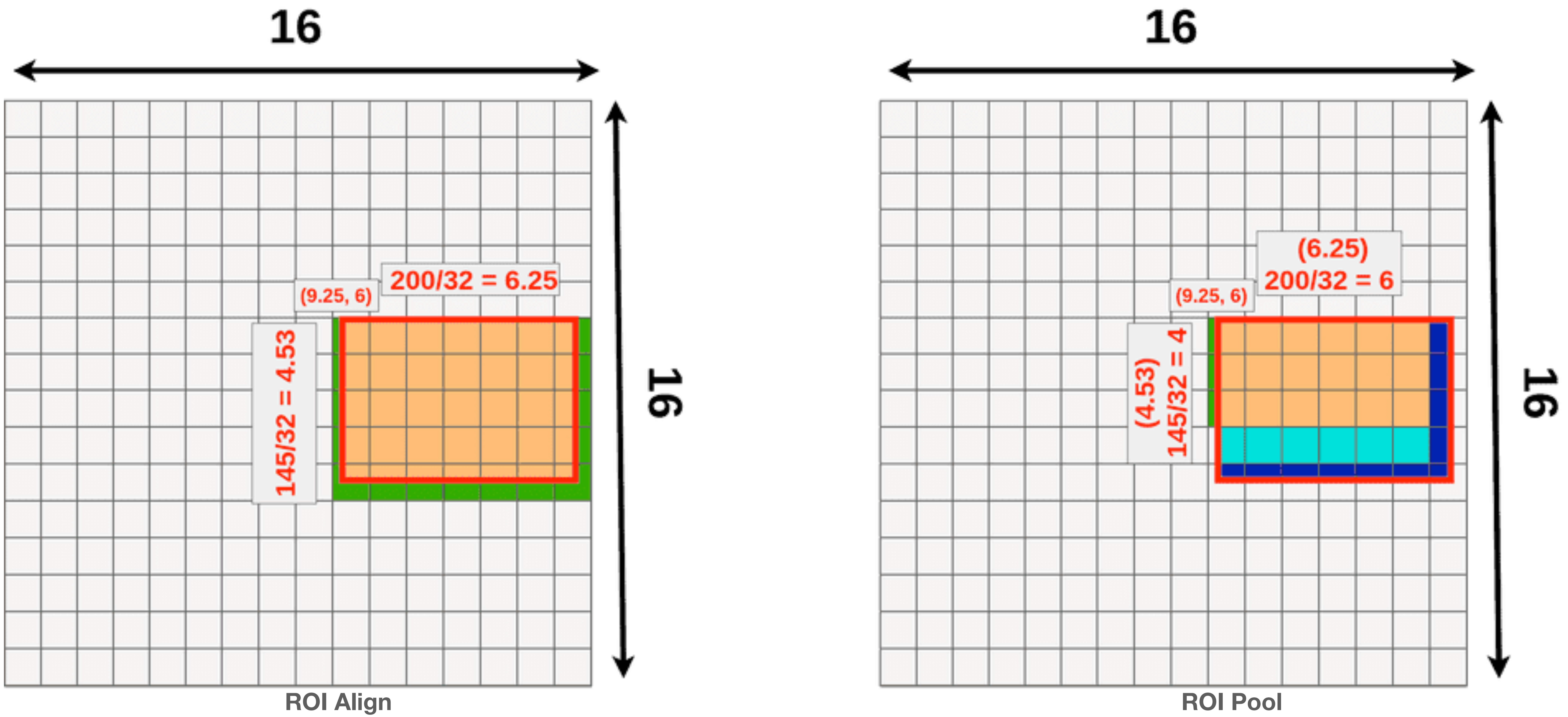
Mask R-CNN: ROI Alignment Mechanics

Now we do not lose data but we do pick up extra information

Green = info gained

Dark Blue = info lost

Light Blue = info lost if enforcing equal partitions



Slides adopted from https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

Mask R-CNN: ROI Alignment Results

ROI Align gives gains to bounding box AP when used with Faster R-CNN

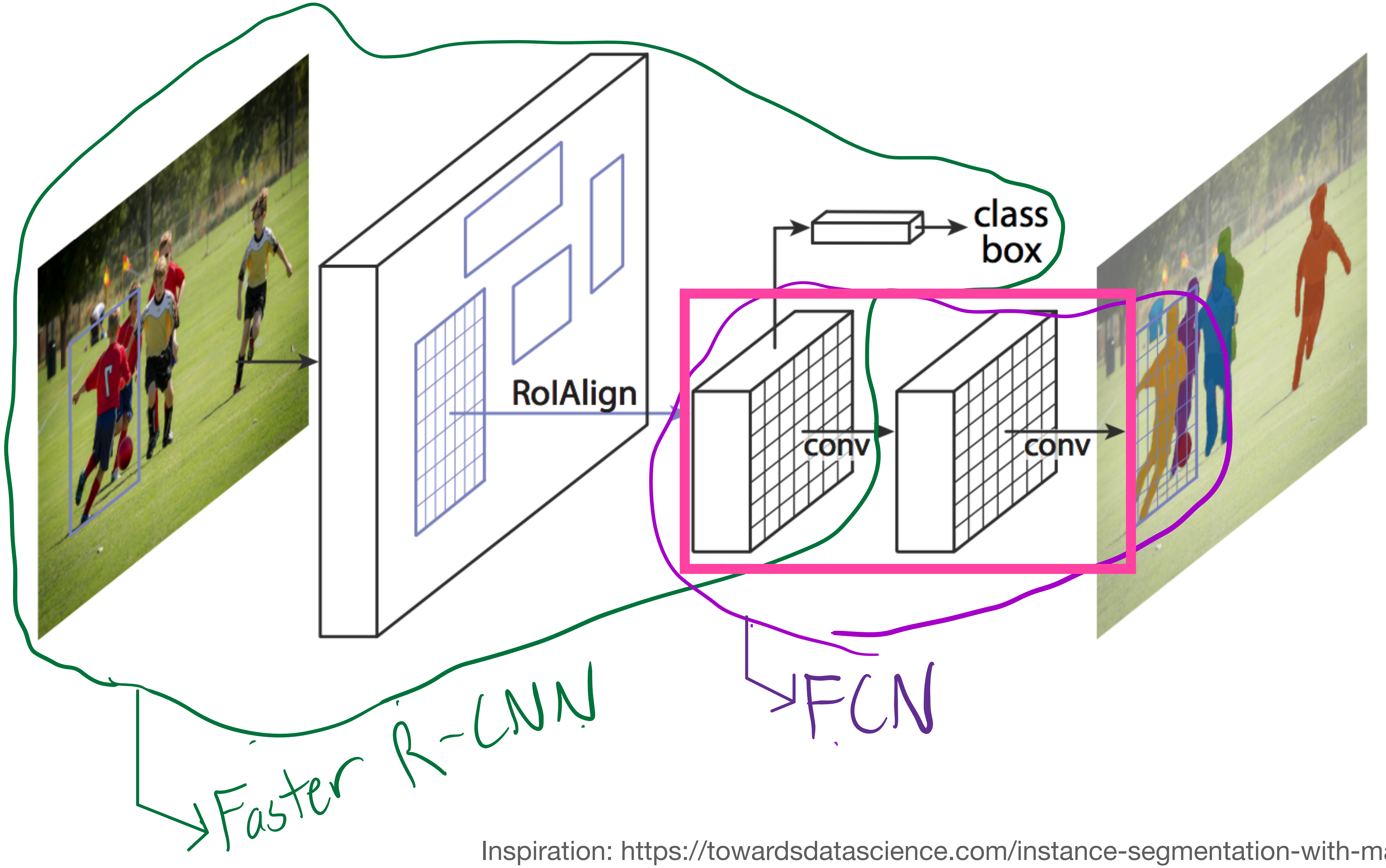
Mask R-CNN sees further benefits from MTL and backbone

	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

He, Kaiming, et al. “Mask R-CNN.” *ArXiv:1703.06870 [Cs]*, Jan. 2018. *arXiv.org*, http://arxiv.org/abs/1703.06870.

Mask R-CNN: Mask Branch

Mask R-CNN = Faster R-CNN + FCN



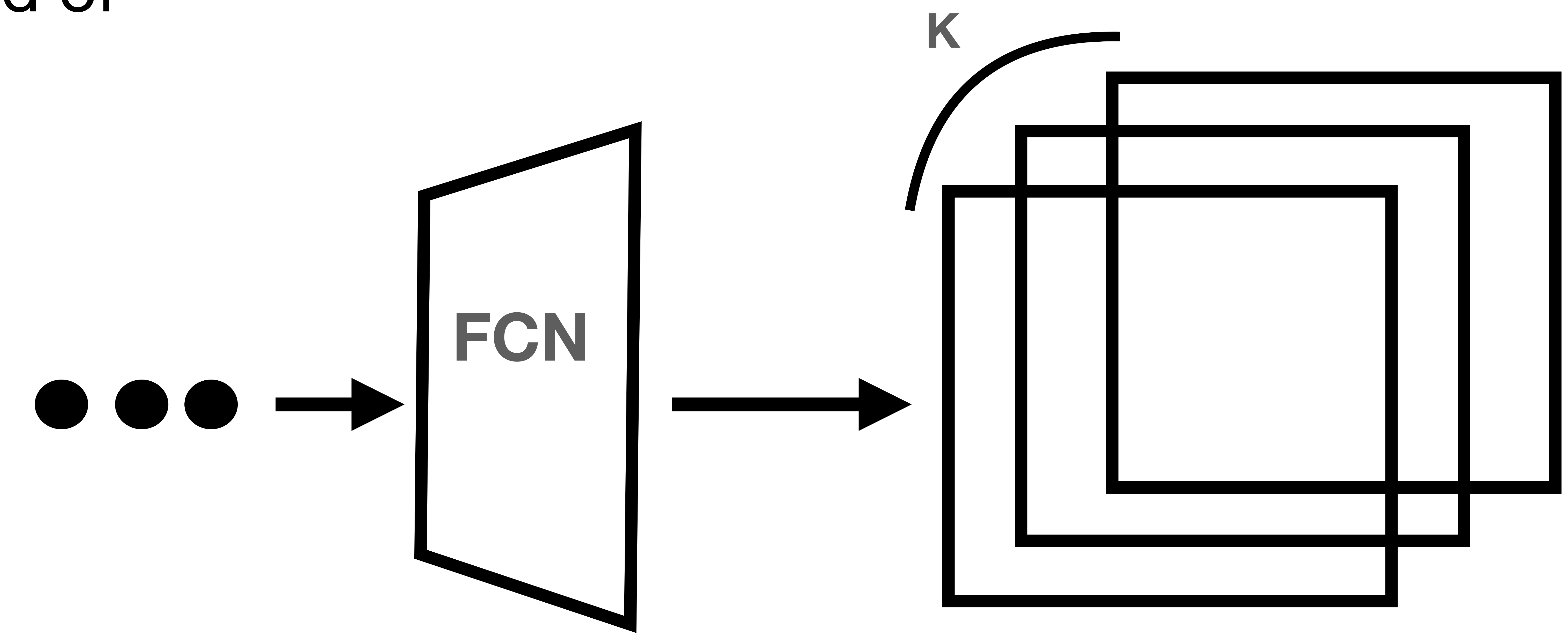
Inspiration: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

Mask R-CNN: Mask Head

Head outputs *K* m*x*m binary masks

Use **sigmoid** output layer instead of softmax

Multi-class vs multi-label



Inspiration: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

Mask R-CNN: Sigmoid Activation

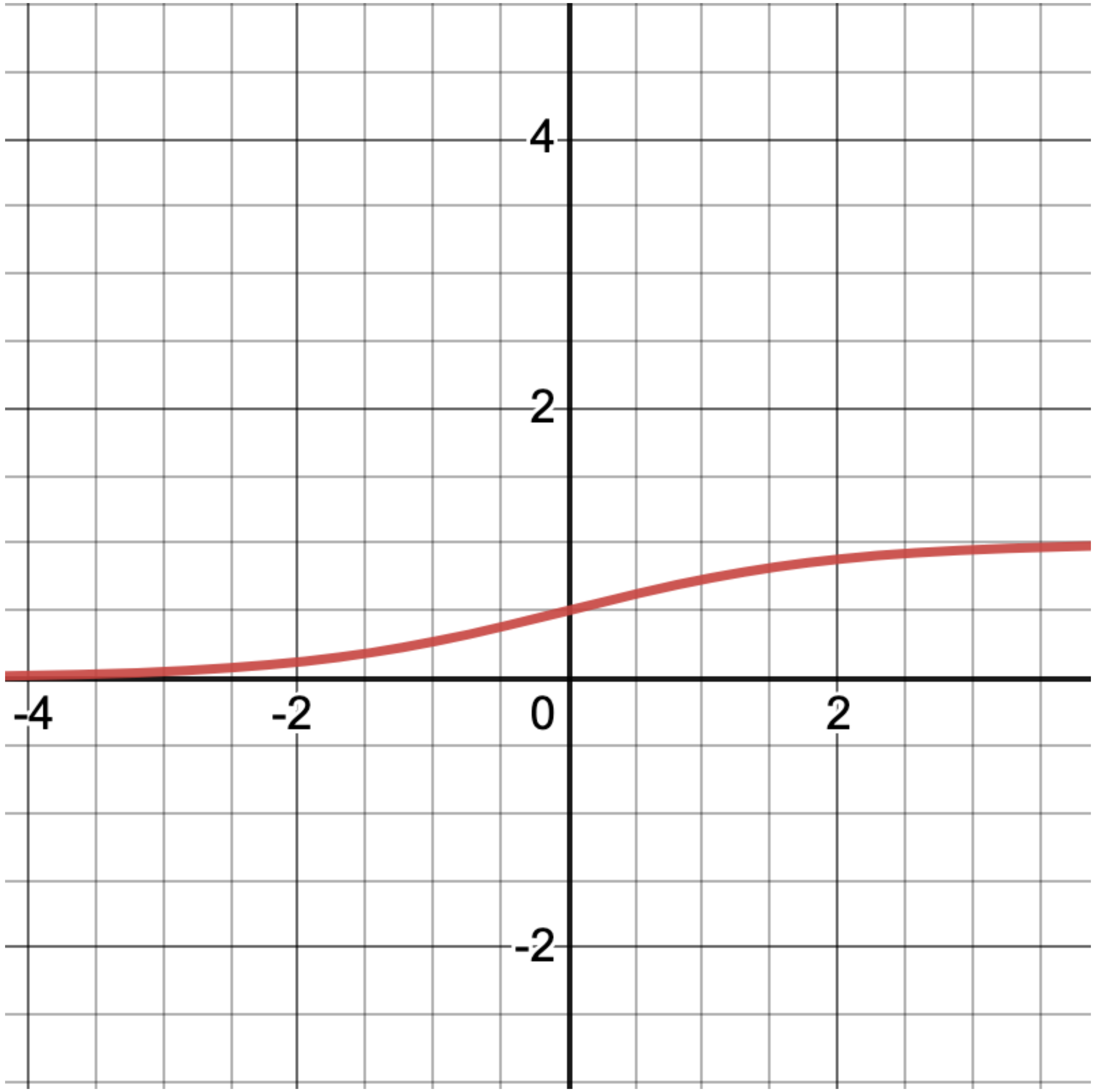
$$sigmoid(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

As $x \rightarrow \infty$, $\sigma(x) \rightarrow 1$

As $x \rightarrow -\infty$, $\sigma(x) \rightarrow 0$

$\sigma(x) \in [0,1]$

Sigmoid can be thought of as the **binary version** of softmax / softmax is a **generalization** of sigmoid for more than 2 classes



Graph from desmos.com

Mask R-CNN: Multi-label vs Multi-Class

Multi-class: Each sample has only one label

Multi-label: Each sample can have multiple labels

How does this fit in to the instance and semantic segmentation?

Image: https://theailearner.com/2019/07/15/multi-label-classification/

Mask R-CNN: Multi-label vs Multi-Class

In **semantic segmentation** we want to get the most probable class for a pixel, so we use **softmax** to create a probability distribution over **all classes for that pixel**

In **instance segmentation** the mask is not responsible for the classification so every pixel in every mask *can* have an object in it so we use a **sigmoid** to create a distribution for **each pixel**

$$pixel_{i,j} = [x_1, x_2, x_3, x_4, x_5]$$

Softmax on whole vector

$$[.2, .1, .3, .15, .25]$$

Semantic

Sigmoid on each element of the vector

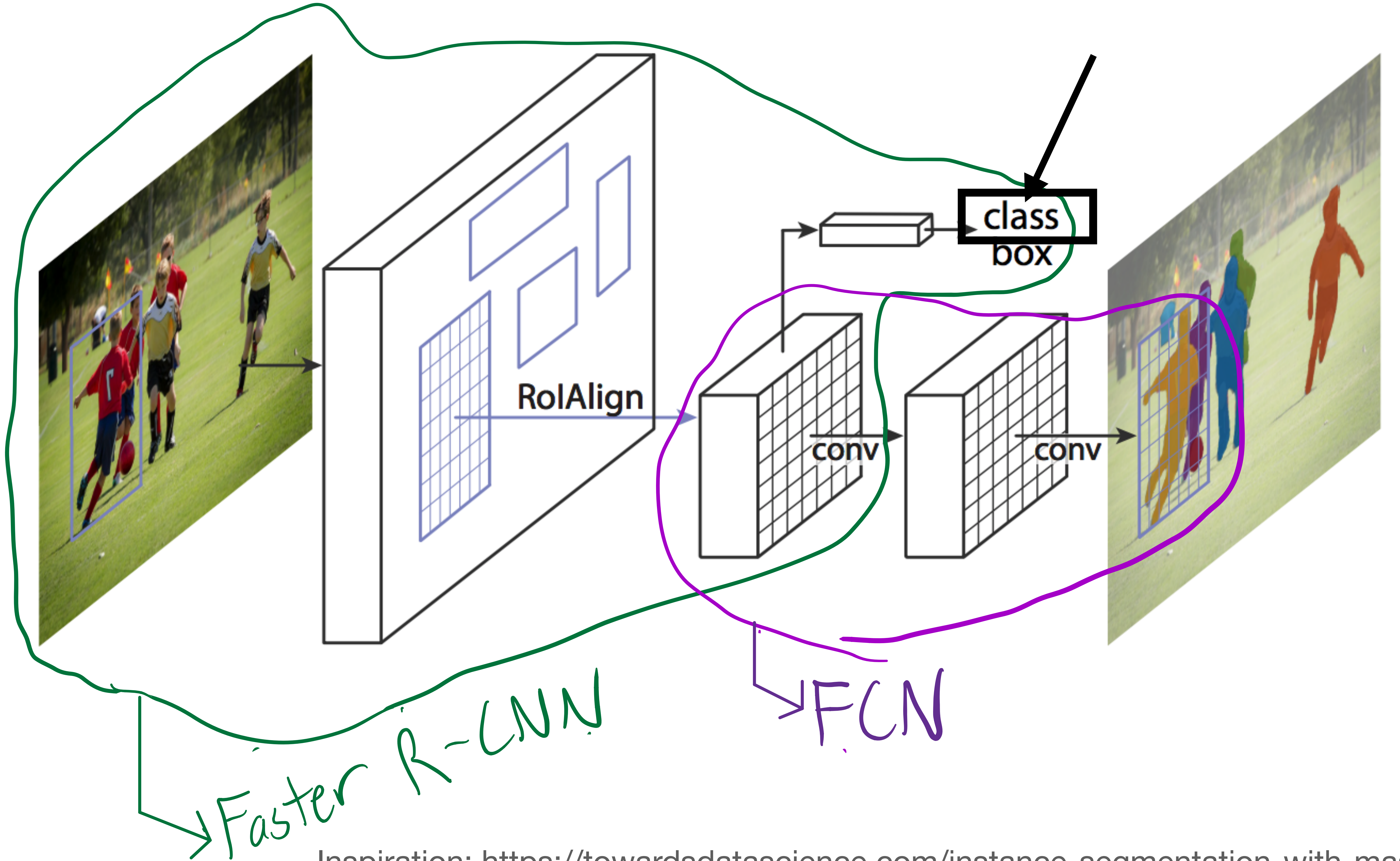
$$[1, 0, 0, 1, 1]$$

Instance

Mask R-CNN: Mask Head

We only care about the binary mask corresponding to the predicted class

Separates mask and class prediction

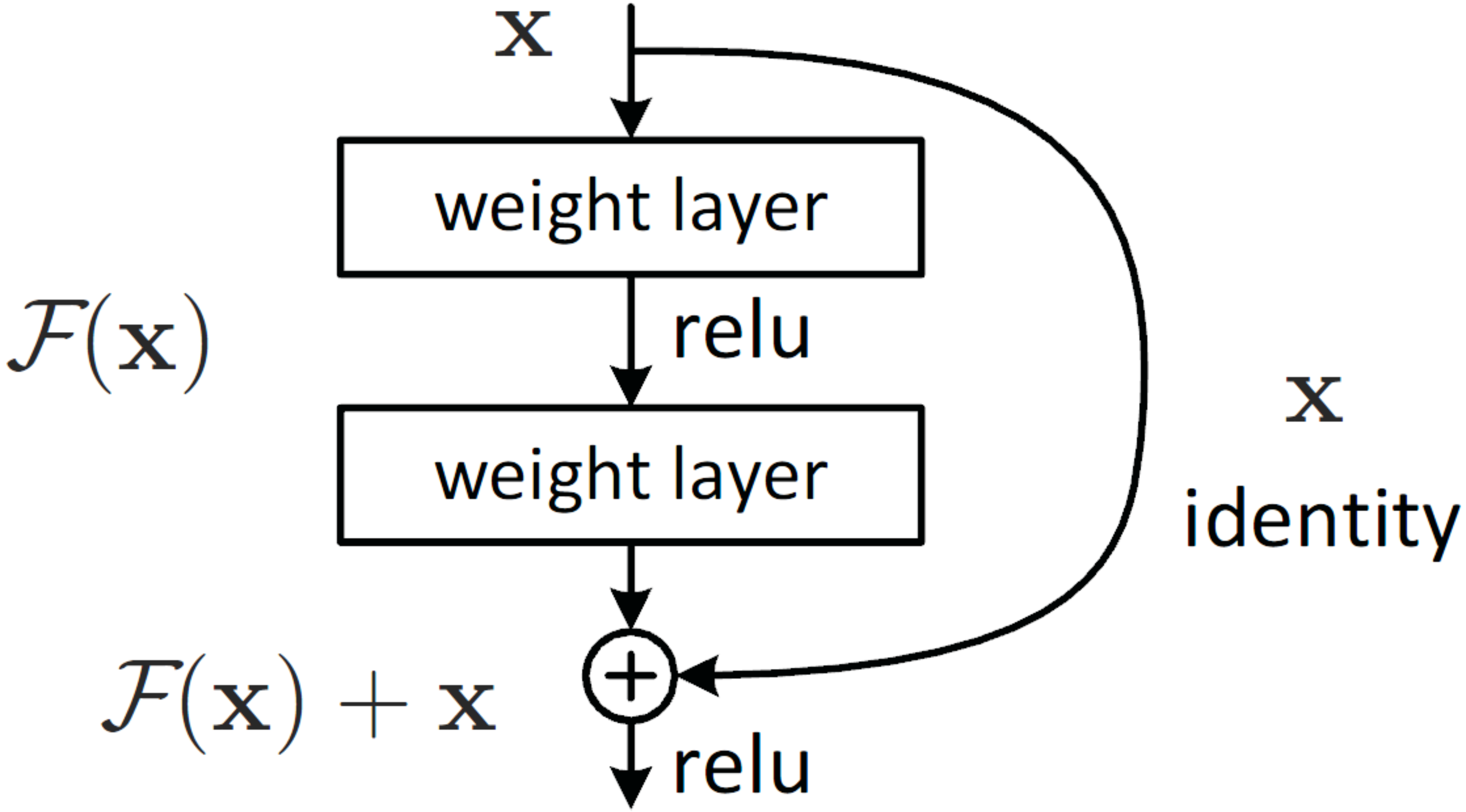


Inspiration: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

Mask R-CNN: Backbones Recap

- ResNet
- Feature Pyramid Network

Mask R-CNN: Backbones - ResNet



Source https://neurohive.io/en/popular-networks/resnet/

Mask R-CNN: Backbones - ResNet

- Only uses ResNet up to stage 4
- Why would you want earlier features?

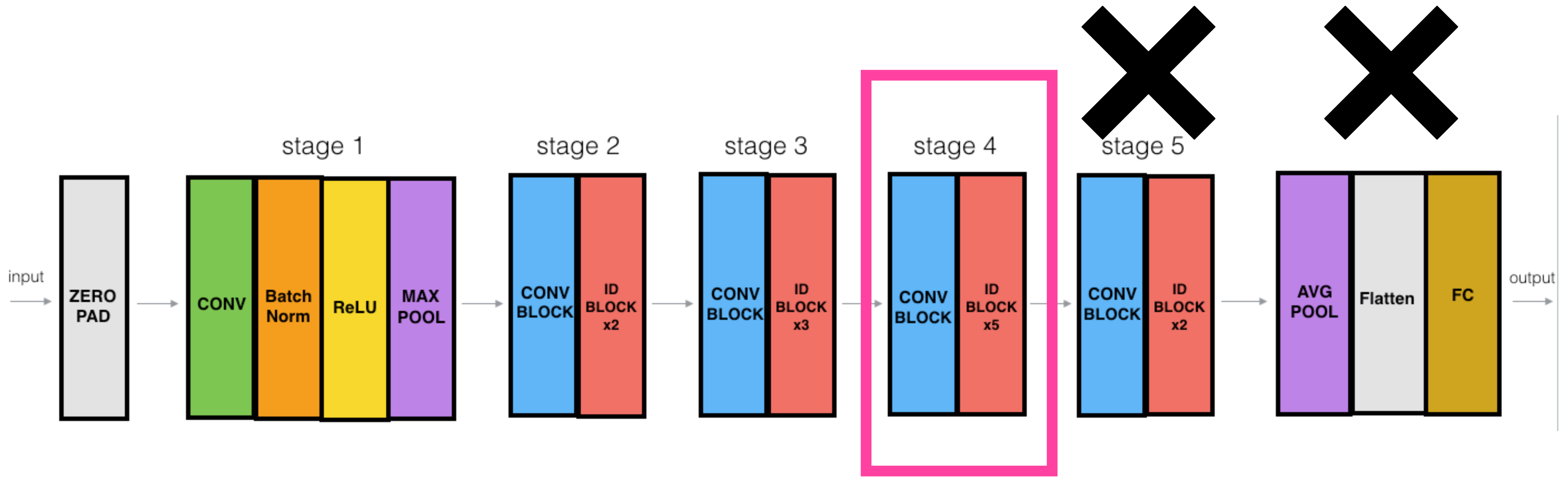
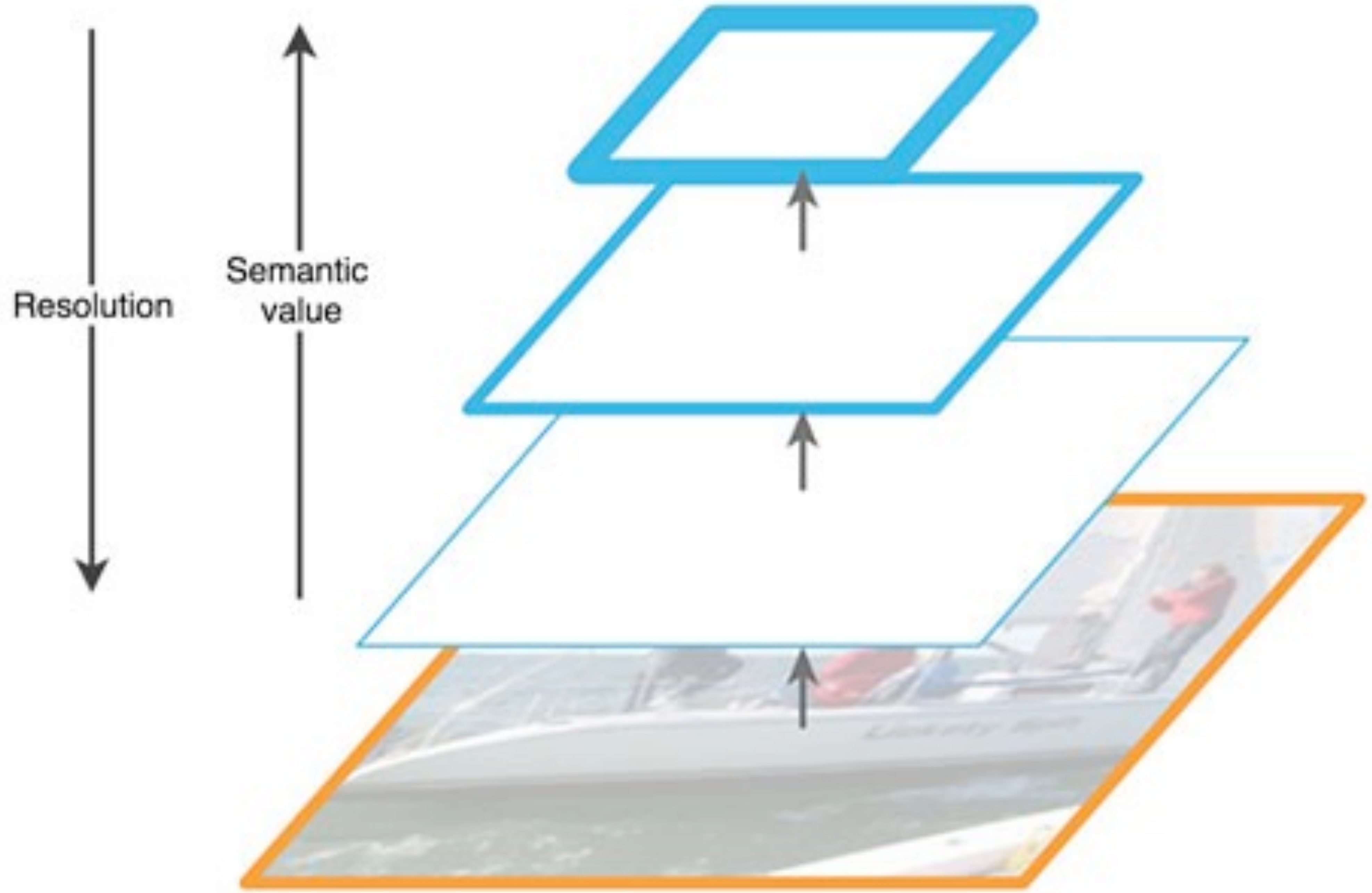


Image http://vernlium.github.io/2018/10/25/coursera-deeplearning-ai-c4-week2/

Mask R-CNN: Backbones - FPN

Allows different ROI scales to help with scale invariance



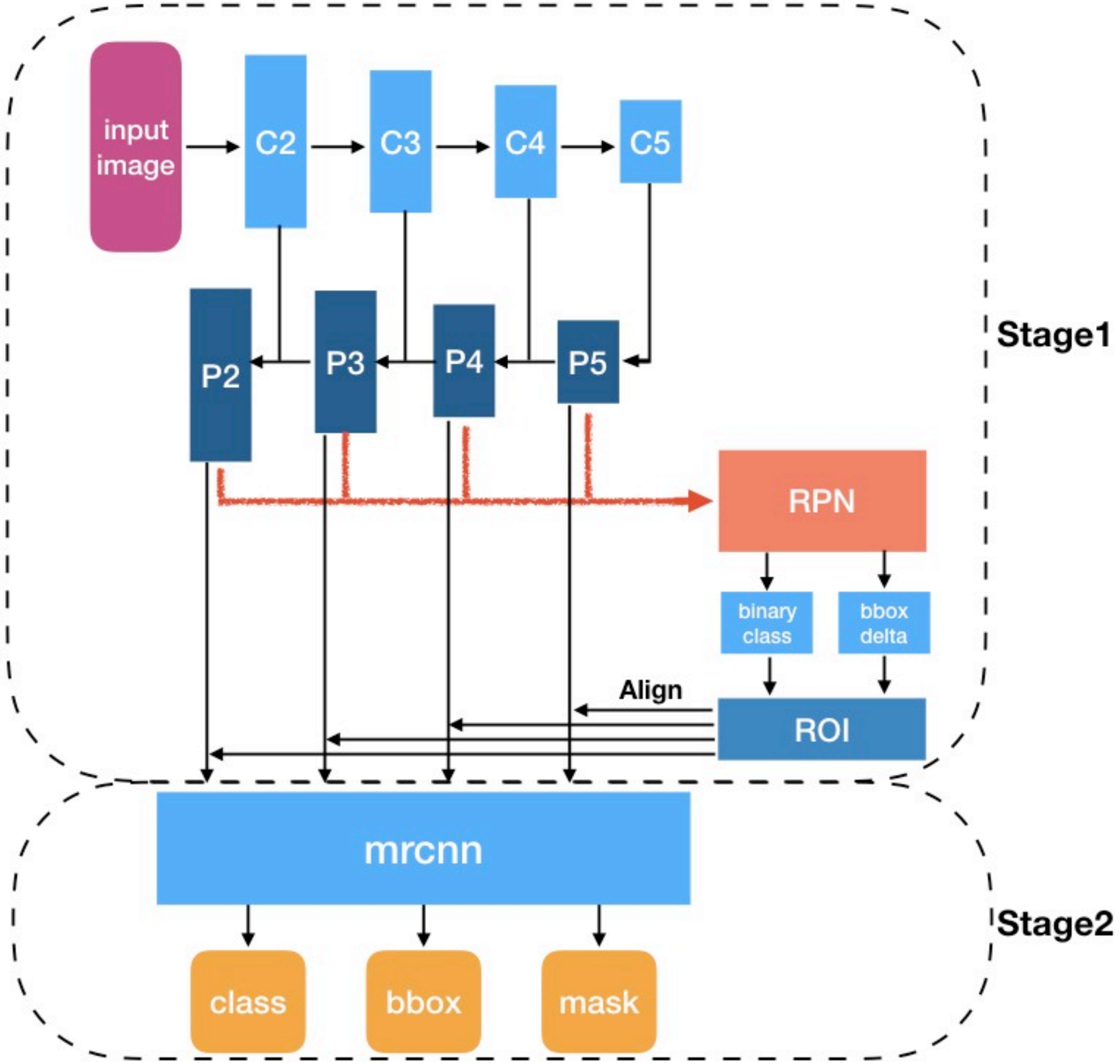
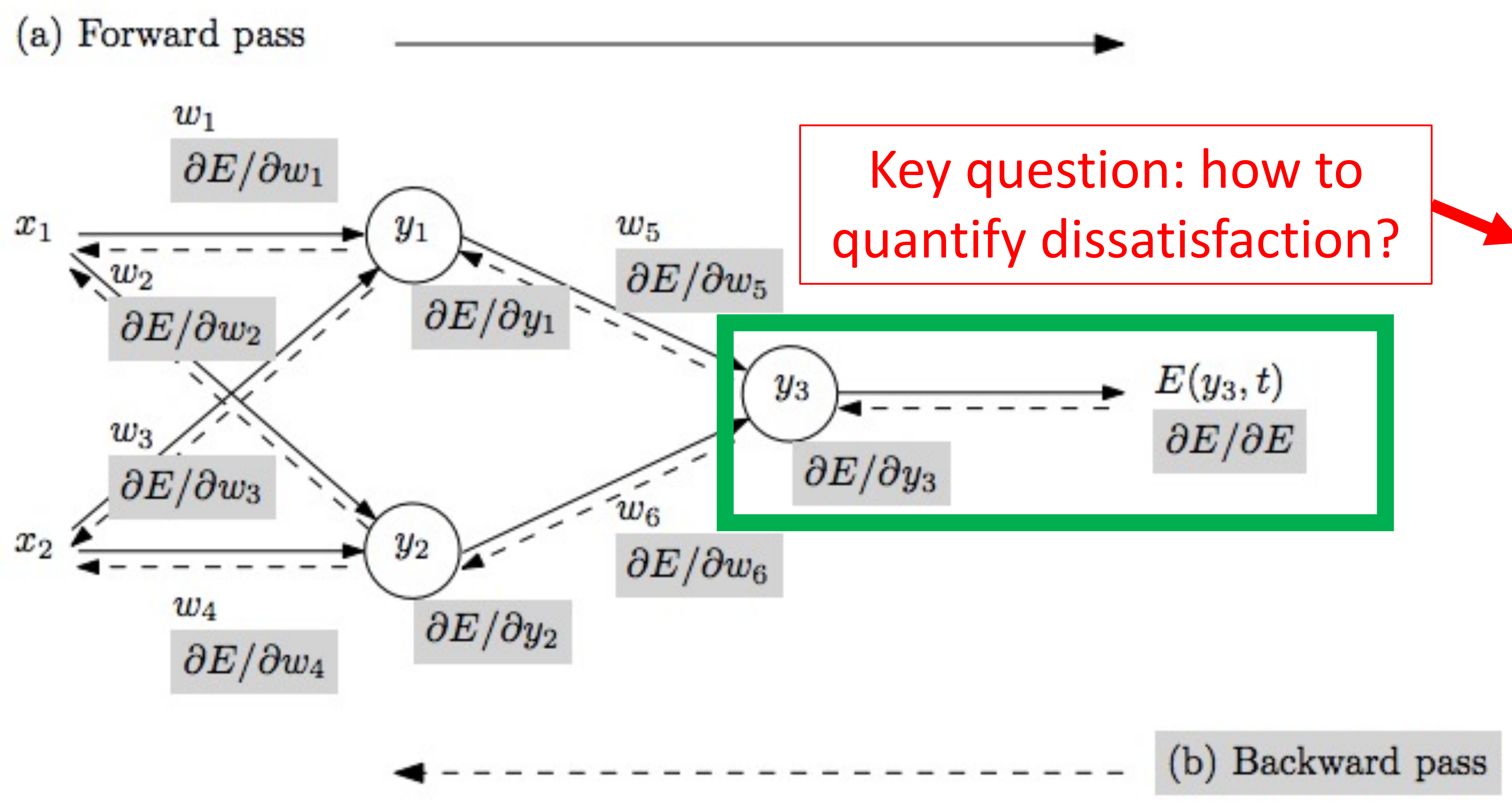


Image: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

Mask R-CNN: Training

Algorithm Training



- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make prediction
  2. Quantify the dissatisfaction with a model's results on the training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

Mask R-CNN: Loss

Multi-task loss

$$L = L_{class} + L_{box} + L_{mask}$$

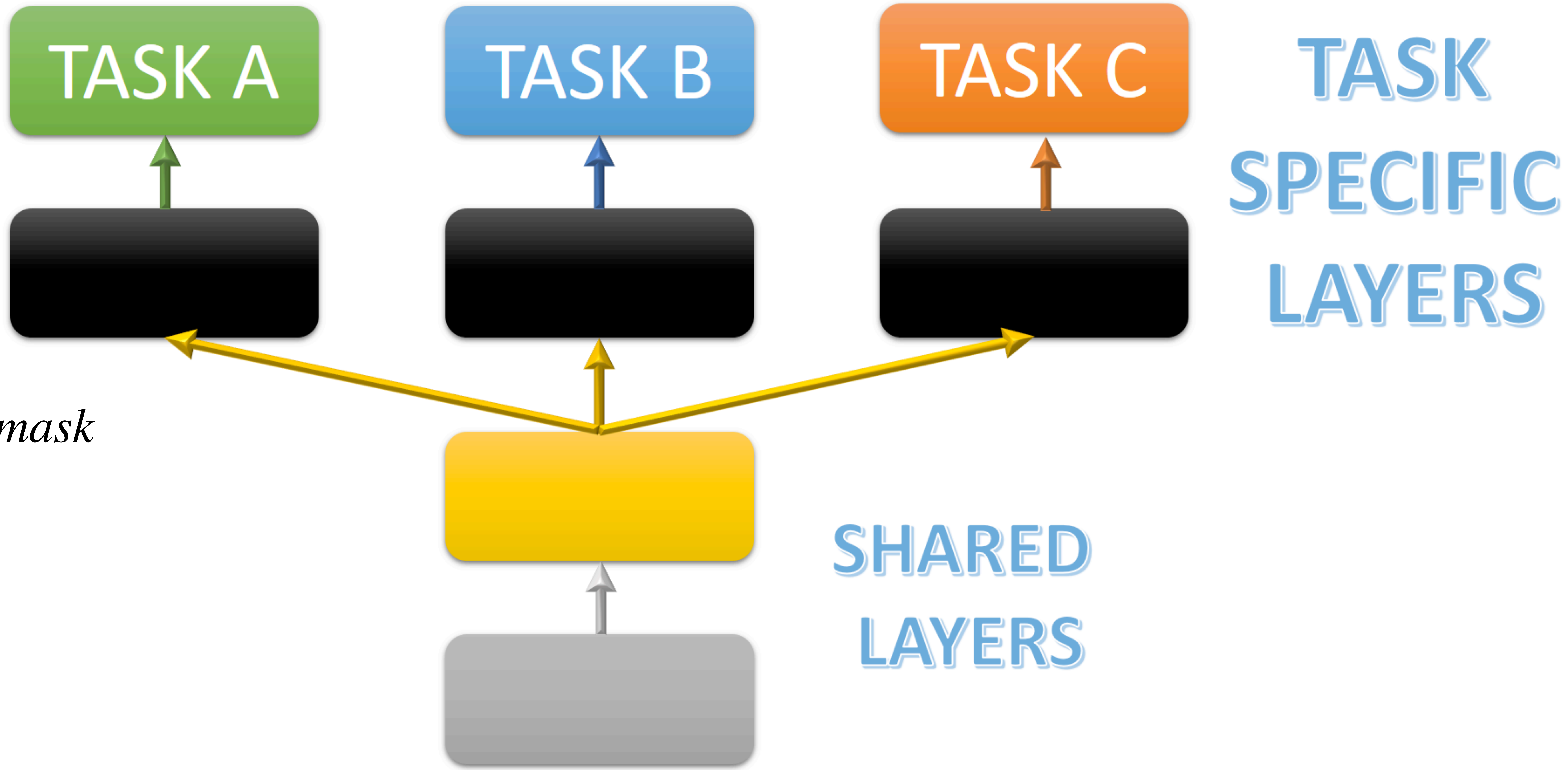
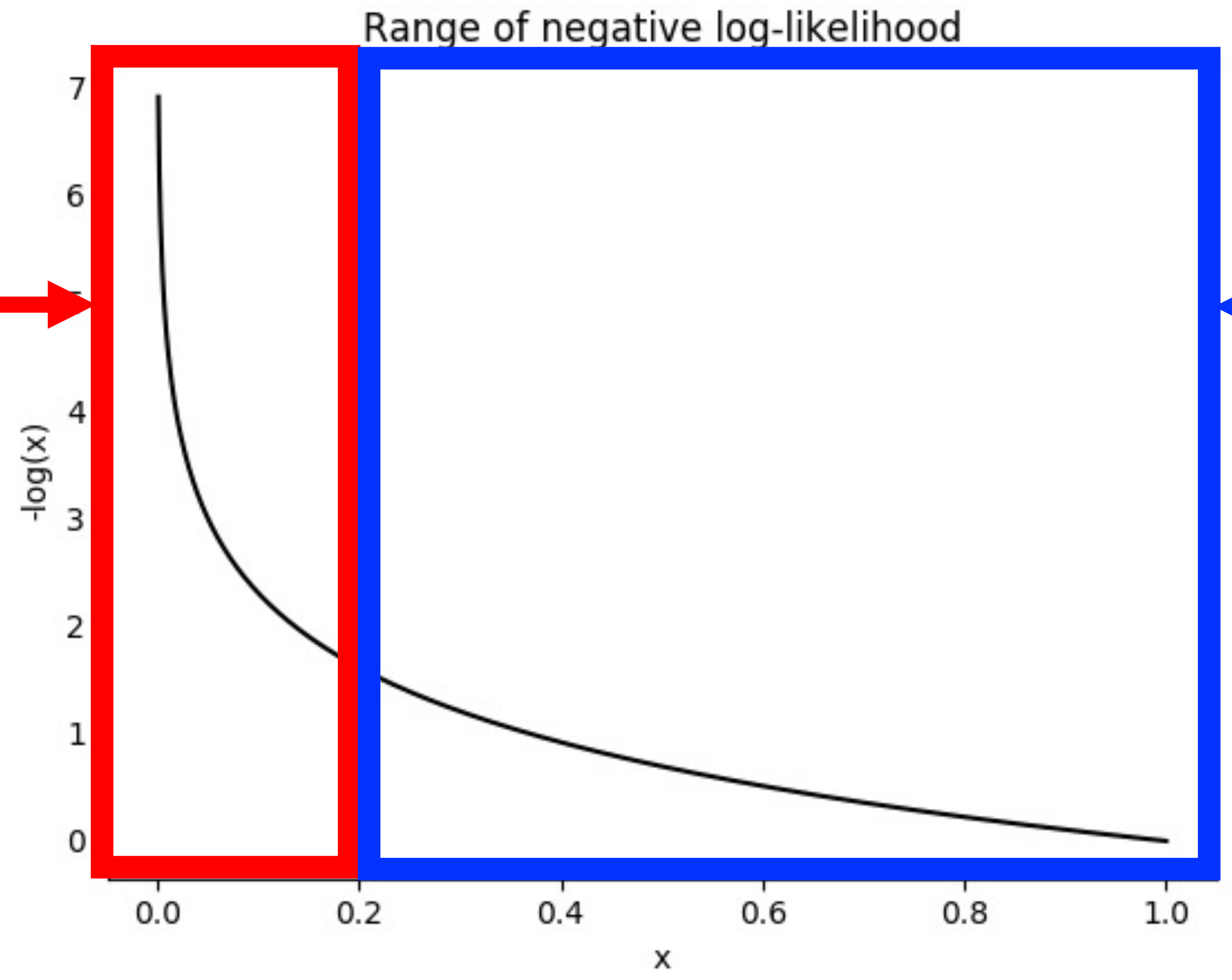


Image: https://blog.manash.io/multi-task-learning-in-keras-implementation-of-multi-task-classification-loss-f1d42da5c3f6

Mask R-CNN: Class Loss

- $L = \boxed{L_{class}} + L_{box} + L_{mask}$
- Negative log likelihood/cross entropy

Greater penalty when predicted probability of true class is confidently wrong



Lesser penalty otherwise

Softmax

What is the range of possible values?

- Minimum: 0 (negative log of 1)
- Maximum: Infinity (negative log of 0)

$$= -\log \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} \exp(w_j \cdot x + b_j)}$$

Mask R-CNN: Box Loss

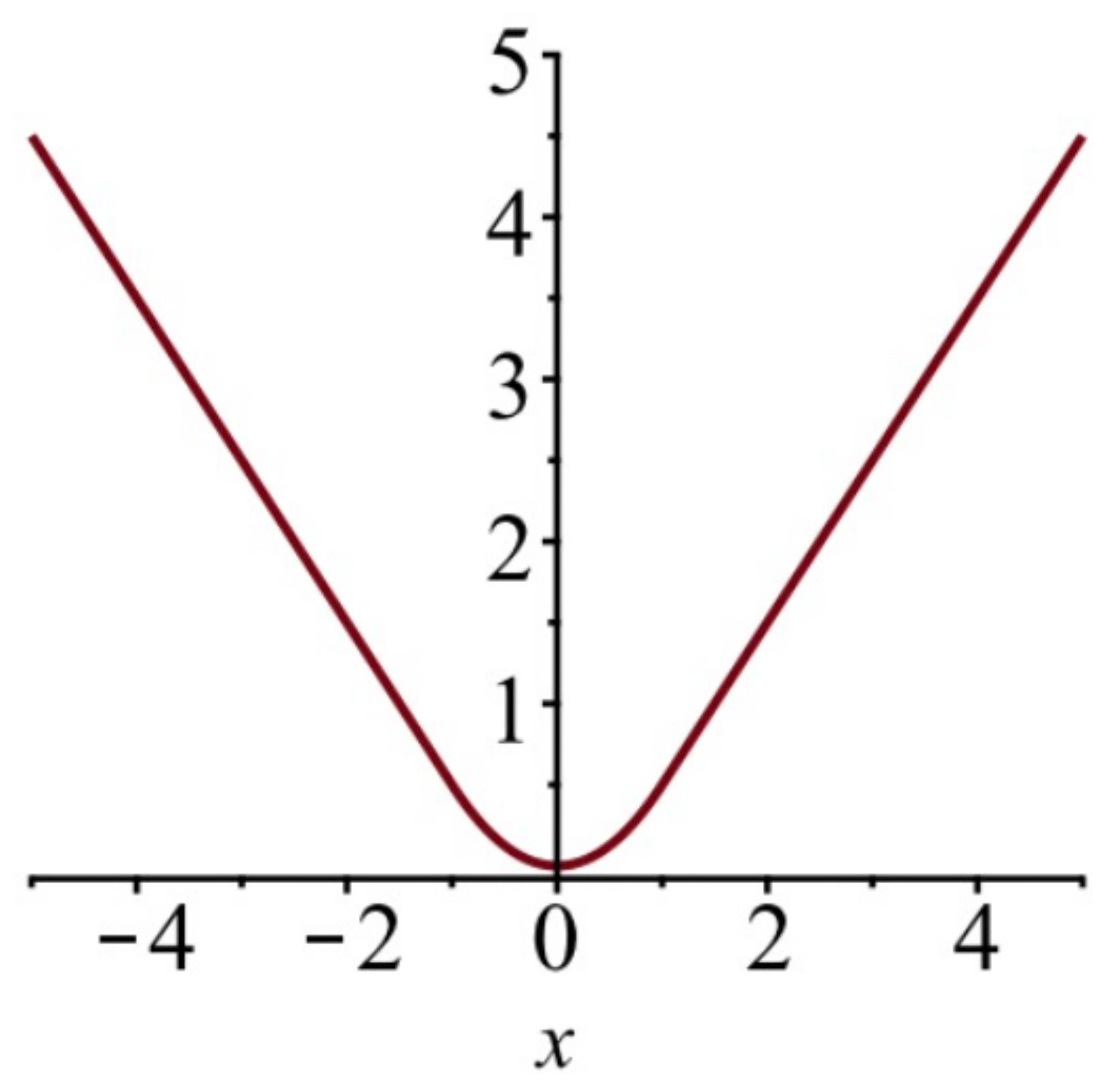
- $L = L_{class} + \boxed{L_{box}} + L_{mask}$
- Smooth ℓ_1 loss

$$\mathcal{L}_{\text{box}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{\text{smooth}}(t_i^u - v_i) \rightarrow L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

True location for true class "u"

Predicted location for class u

Less sensitive to outliers than SSE



Slide from Danna Gurari: https://home.cs.colorado.edu/~DrG/Courses/RecentAdvancesInComputerVision/Lectures/07-ObjectDetection-Part2.pdf

Mask R-CNN: Mask Loss

- $L = L_{class} + L_{box} + \boxed{L_{mask}}$
- Binary cross entropy for the k^{th} mask corresponding to ground truth class *k*
- $\hat{y}_{ij}^{k} = \sigma(\hat{y}_{ij}^{k}) \in [0,1]$ $\qquad y_{ij} \in \{0,1\}$

$$L_{mask} = -\frac{1}{\boxed{m^2}} \sum_{1 \leq i,j \leq m} \boxed{[y_{ij} log(\hat{y}_{ij}^{k}) + (1 - y_{ij}) log(1 - \hat{y}_{ij}^{k})]}$$

Total number of pixels

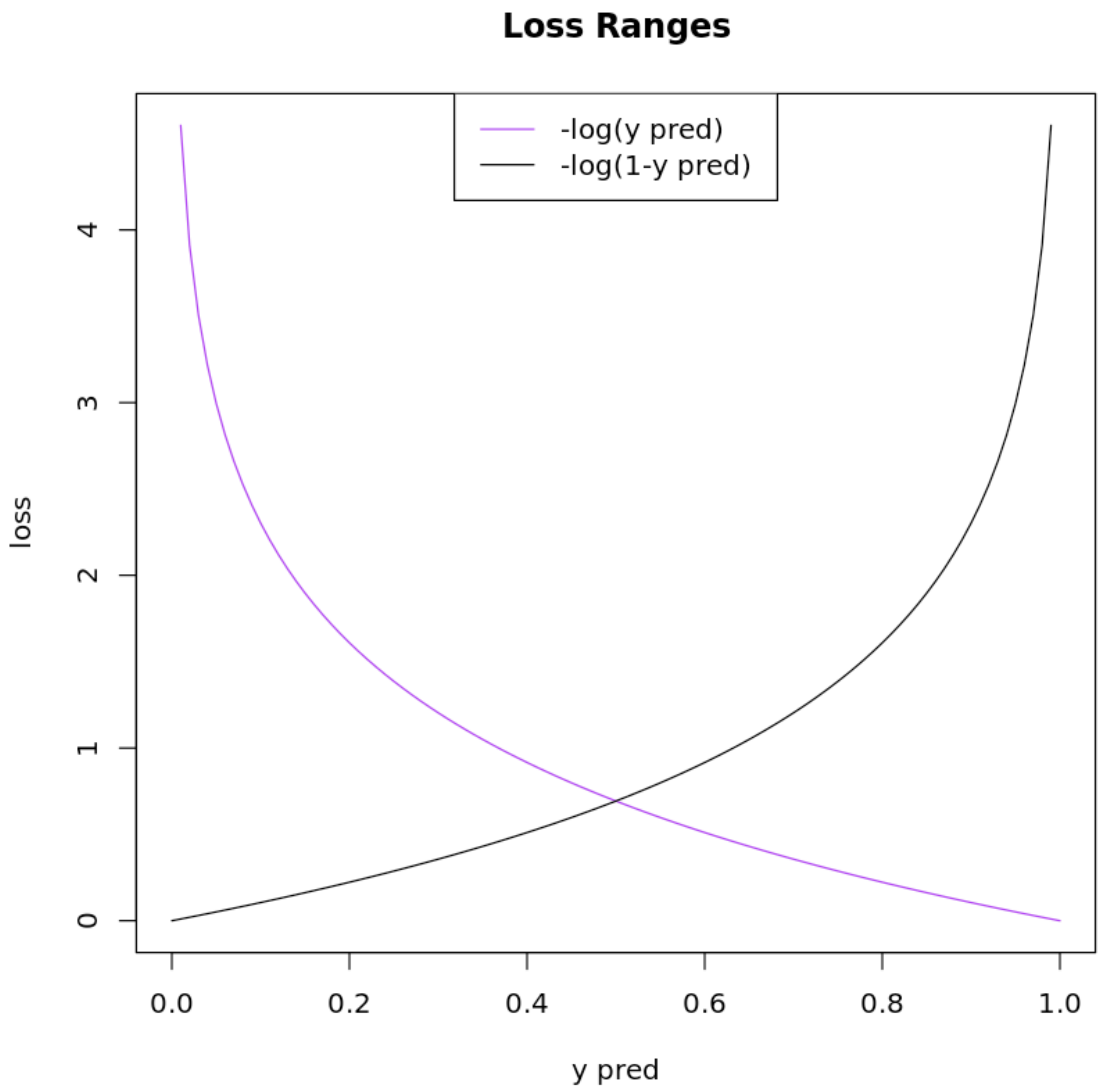
Binary Cross entropy per class!

He, Kaiming, et al. “Mask R-CNN.” *ArXiv:1703.06870 [Cs]*, Jan. 2018. *arXiv.org*, http://arxiv.org/abs/1703.06870.

Mask R-CNN: Mask Loss Simplified

Loss simplifies depending on the ground truth value

$$Loss(\hat{y}_{ij}^{k}) = \begin{cases} -log(\hat{y}_{ij}^{k}), & if\ y_{ij} = 1 \\ -log(1 - \hat{y}_{ij}^{k}), & if\ y_{ij} = 0 \end{cases}$$



Mask R-CNN: Results

We can achieve SOTA results with a simple extension

	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**

K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

Mask R-CNN: Bonus!

We get pose estimation for free!

K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

Overview

- Faster R-CNN
- Mask R-CNN
- SWIN Transformer
- Discussion

Swin Transformer: Motivation

- Previous work (ViT) great for image recognition not as a general backbone
- Motivation: General purpose backbone from a transformer

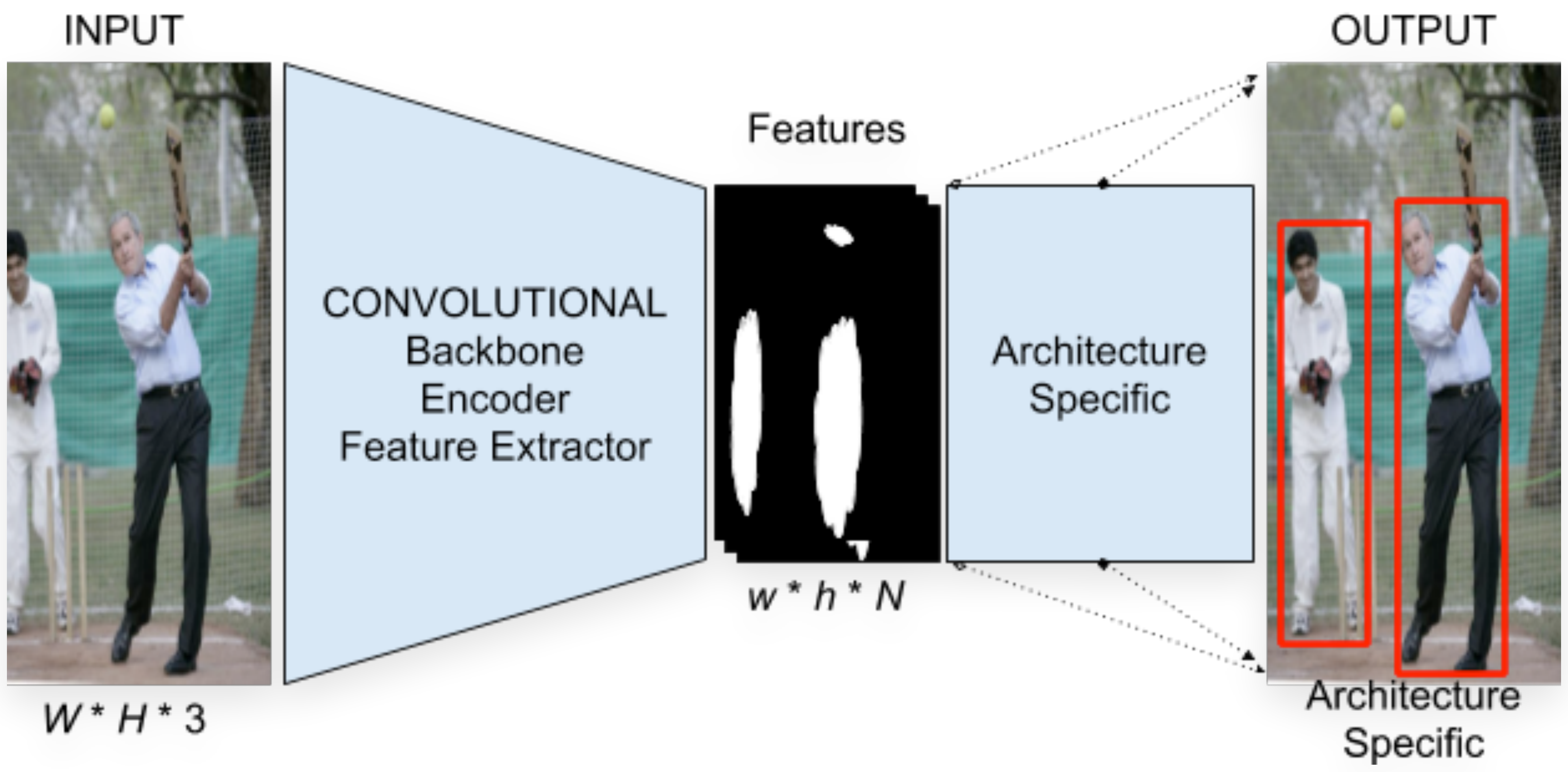


Image: https://medium.com/analytics-vidhya/how-to-select-the-perfect-cnn-back-bone-for-object-detection-a-simple-test-b3f9e9519174

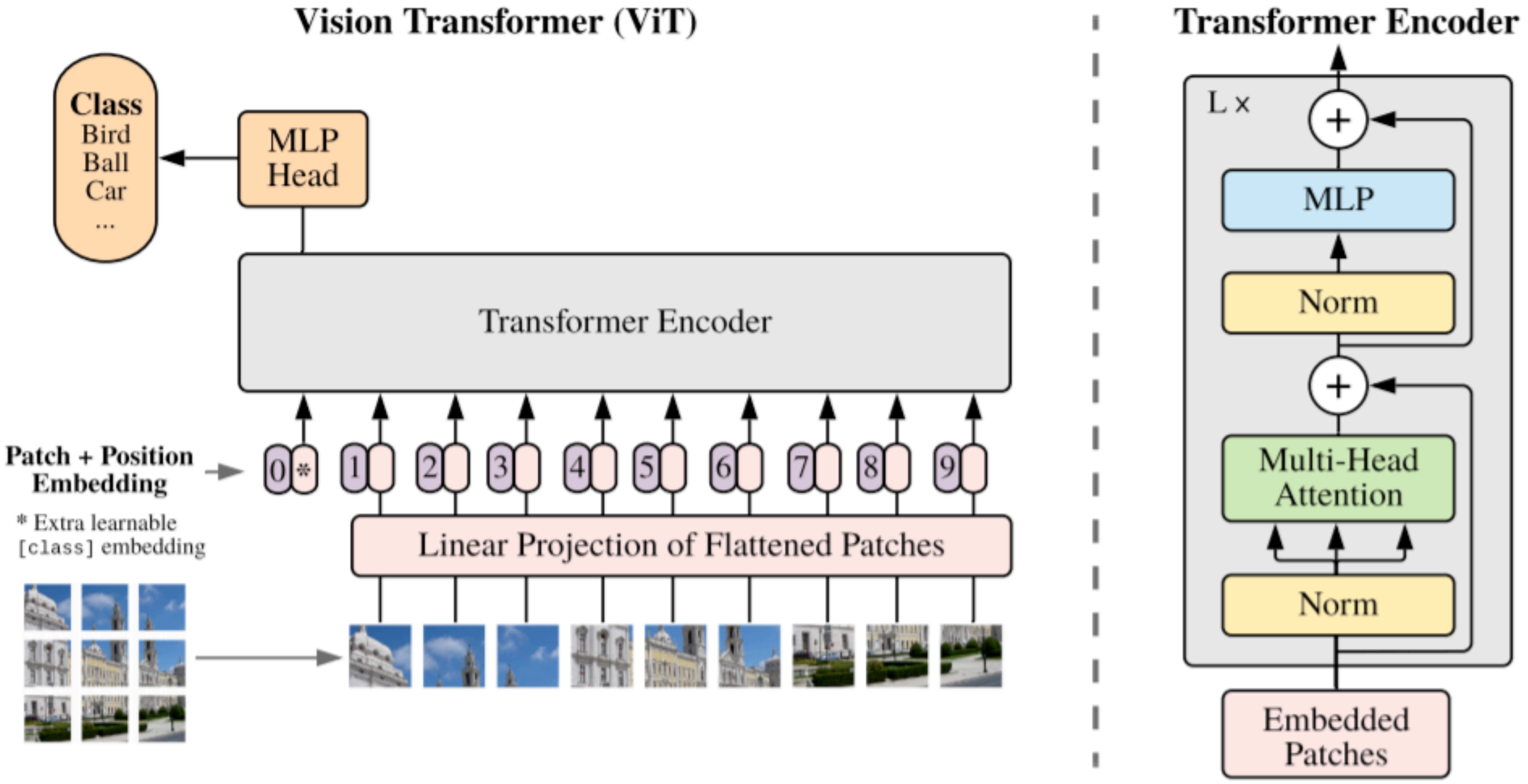


Image: https://medium.com/analytics-vidhya/vision-transformers-bye-bye-convolutions-e929d022e4ab

Swin Transformer: Motivation Cont.

- Transformers are inefficient for image data due to multiple scales
- Self-attention runs in $O(n^2)$

Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Contributions

- General purpose backbone for vision transformers
- More sensible attention algorithm
- Hierarchical patch embeddings

Liu, Ze, et al. “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows.” *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Architecture

Composed of stages of patch merging and successive transformer blocks

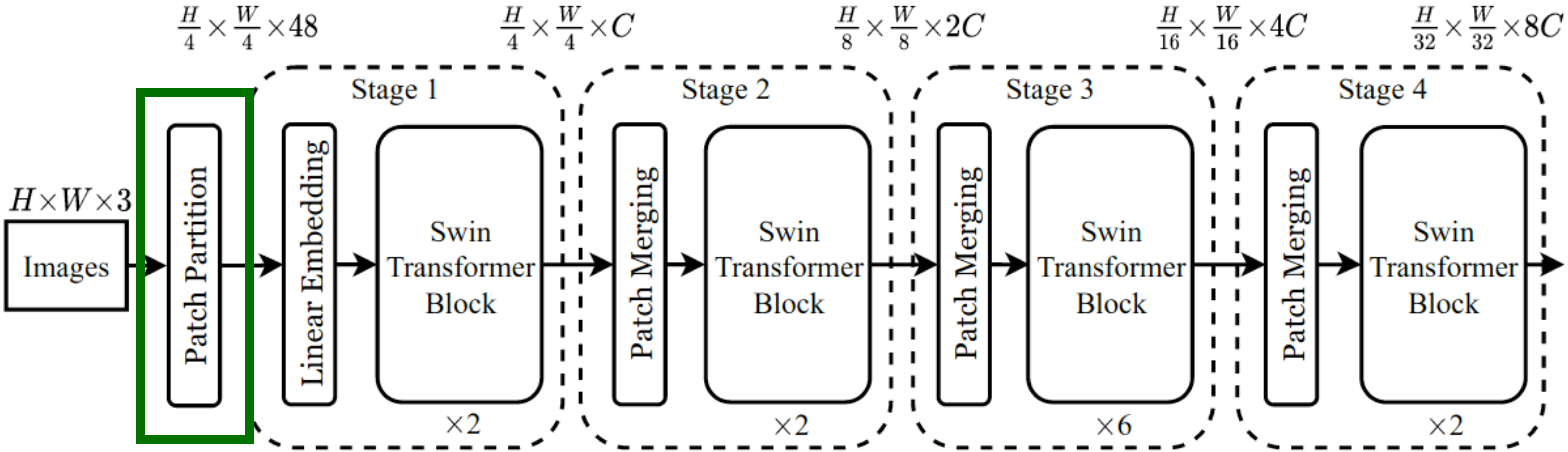
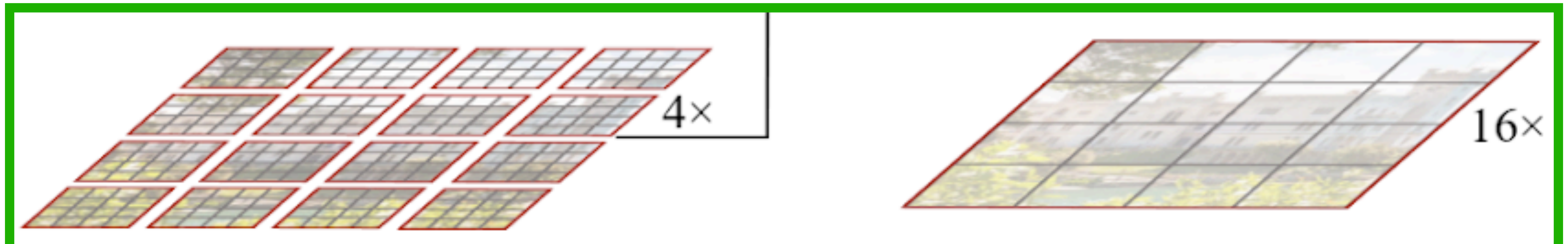


Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Patch Partition

Let's just start with a finer resolution and get more coarse as we go further

Start with $\frac{H/W}{4}$ resolution instead of $\frac{H/W}{16}$ resolution



(a) Swin Transformer (ours) (b) ViT

Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Architecture

Composed of stages of patch merging and successive transformer blocks

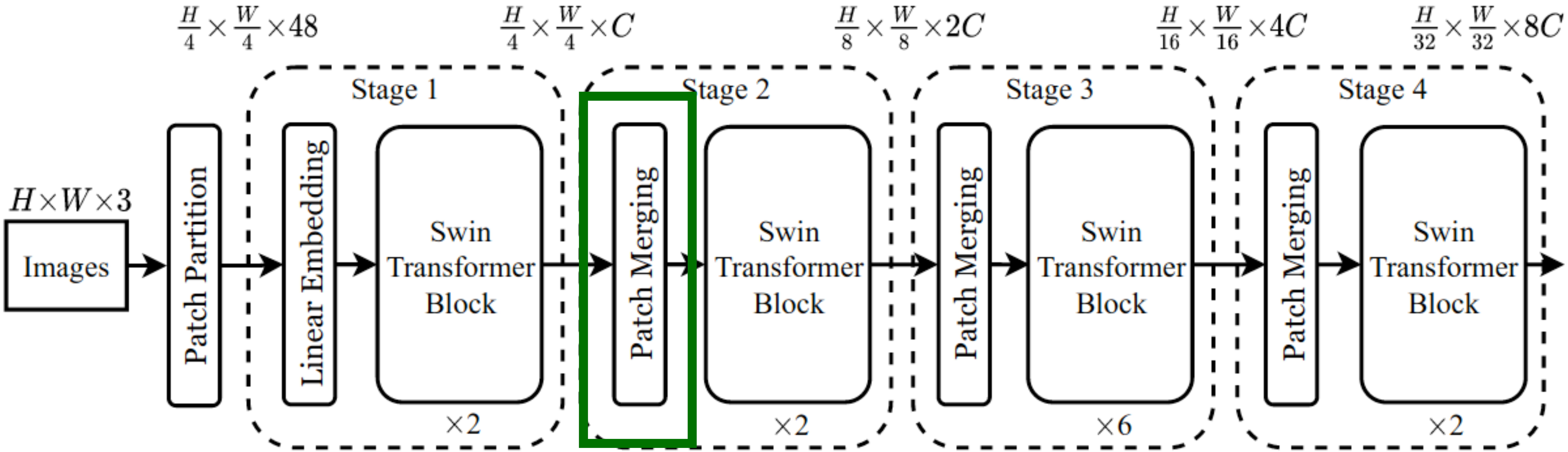
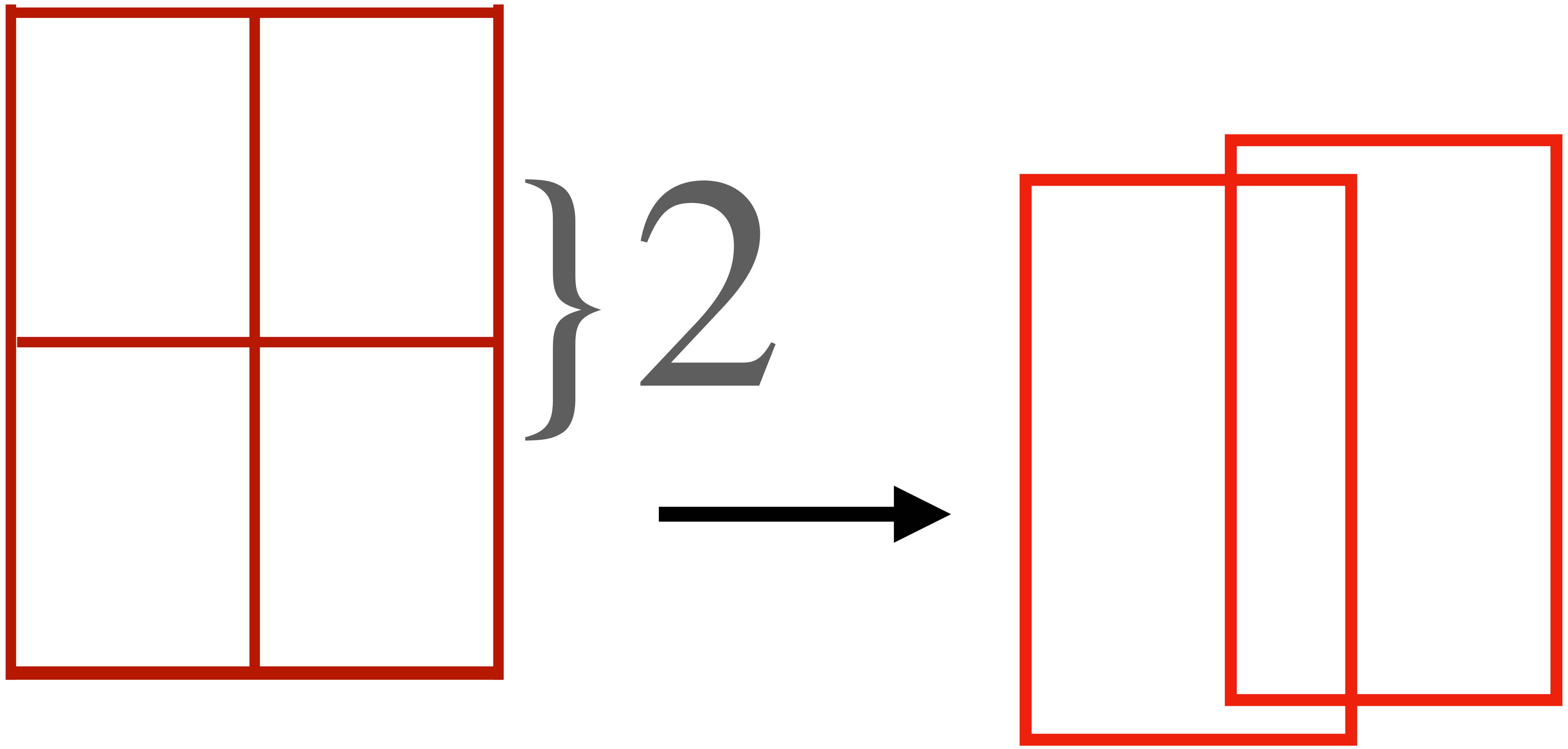


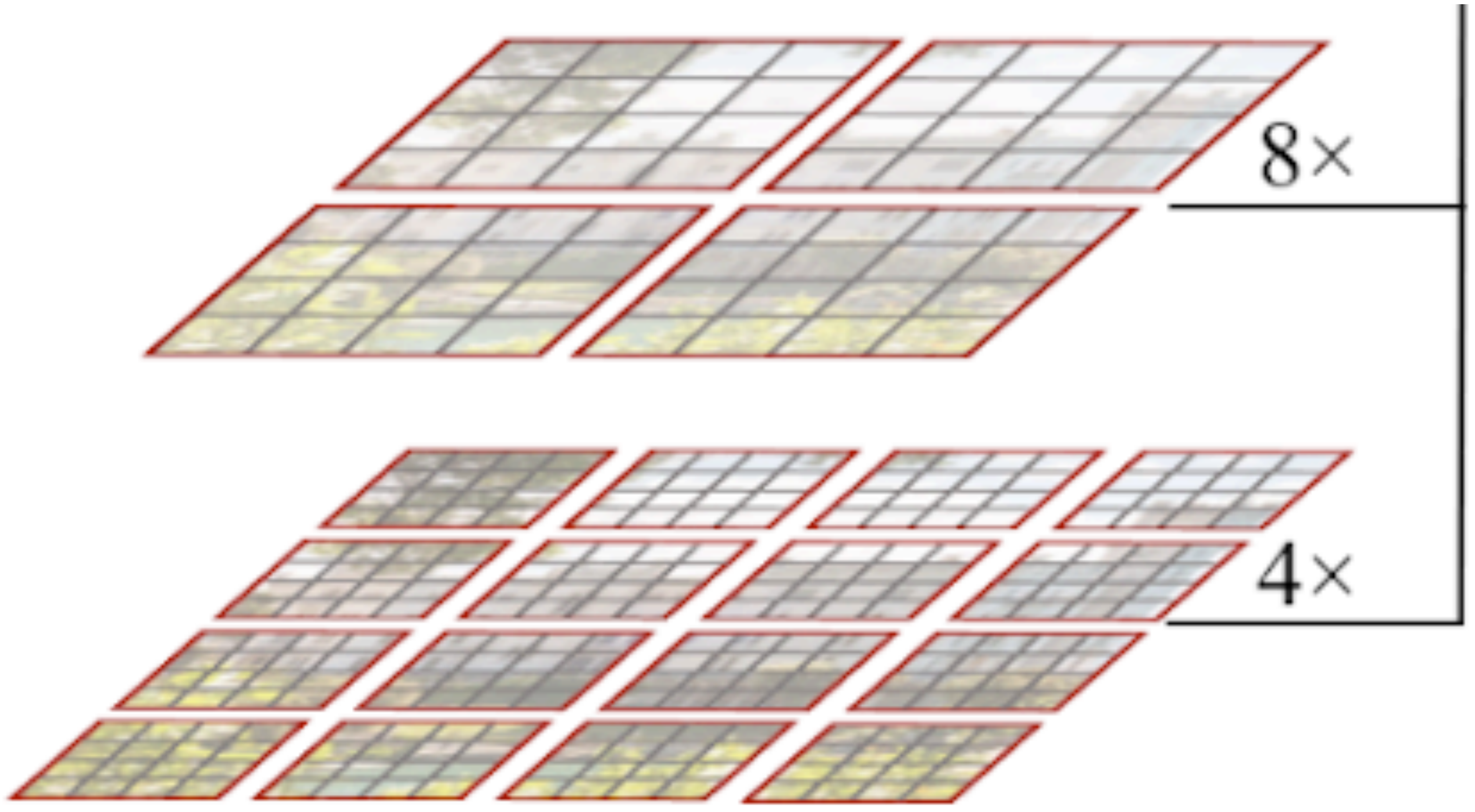
Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Patch Merging

Merge 2x2 grid patches and then double the channels through a linear layer

Allows representations at different resolutions like FPNs





Images: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Architecture

Composed of stages of patch merging and successive transformer blocks

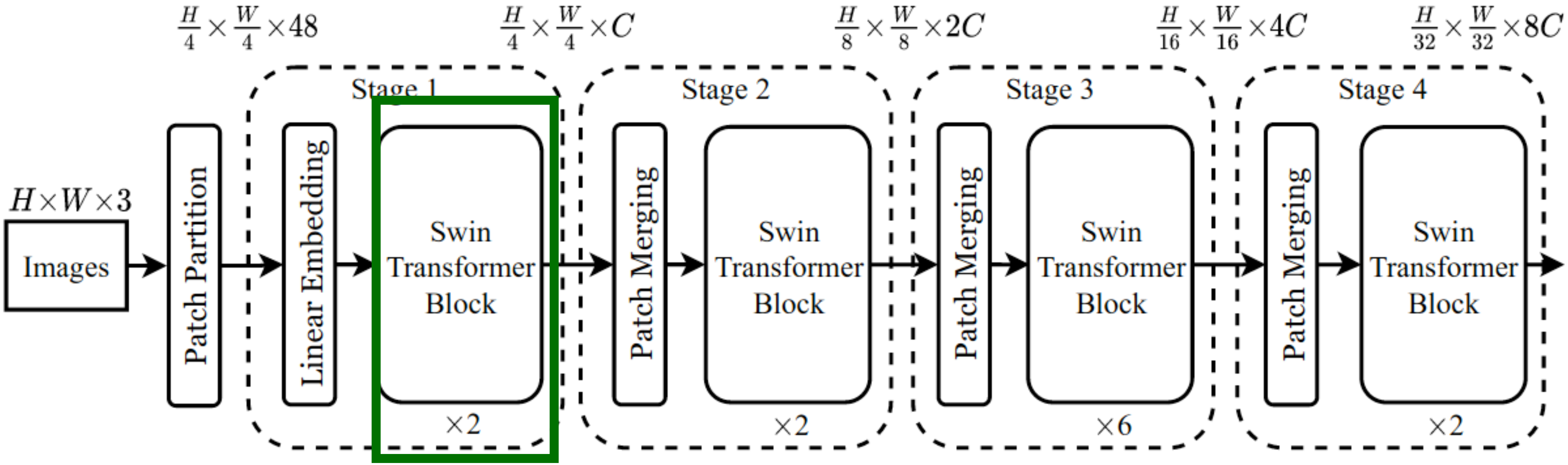
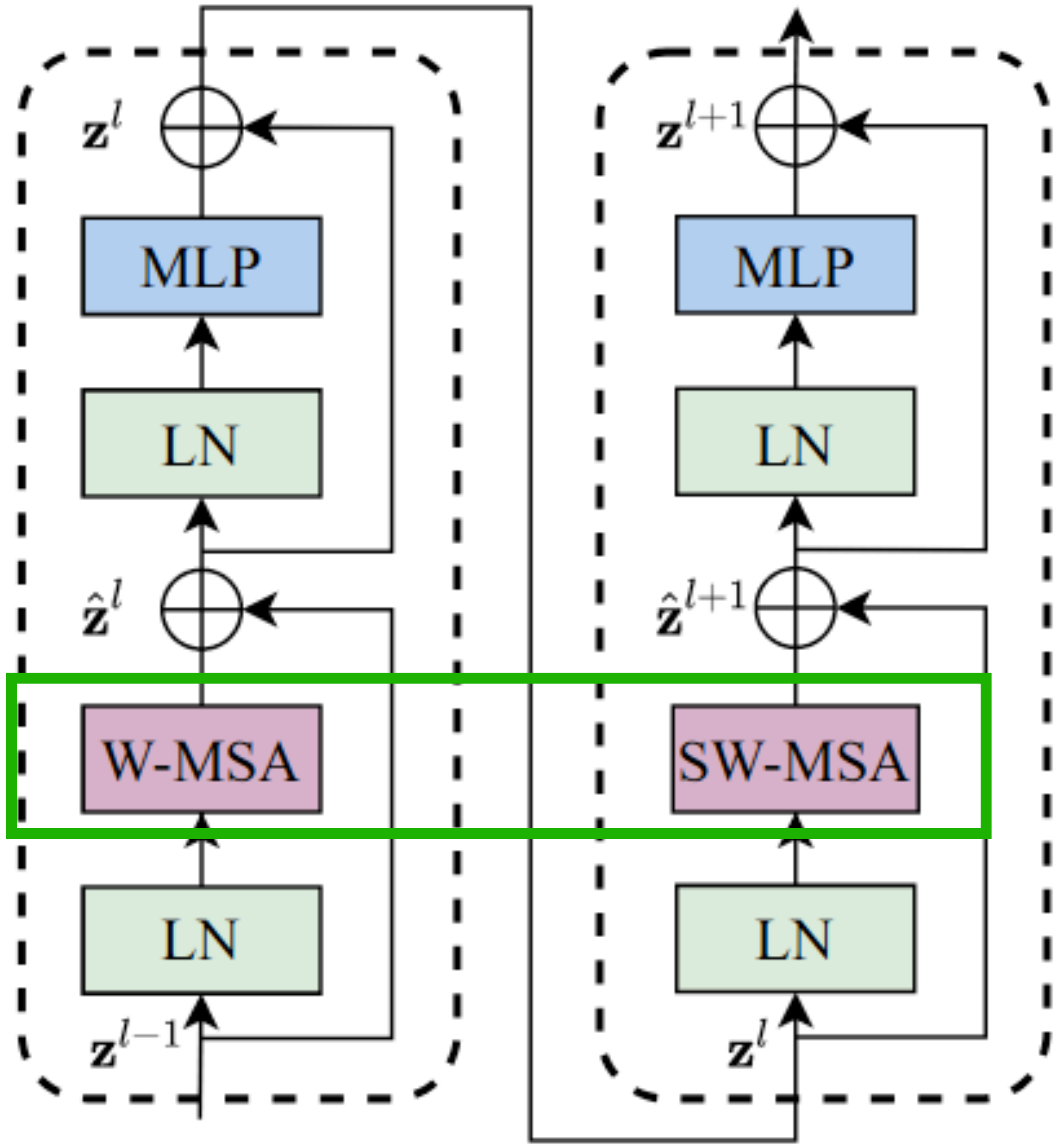


Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Architecture

Composed of

1. Layer norm
2. {Window | Sliding Window} MSA
3. Skip Connections
4. MLP



Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: W-MSA

Typical MSA Approach: For every pixel, attend to every other pixel in the image - **expensive!**

Window-MSA Approach: For every pixel in a local window, attend to every other pixel in that window

Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: SW-MSA

- Take prior windows, shift by $\left\lfloor \frac{M}{2} \right\rfloor$, where M = size of window
- What to do with the empty space?

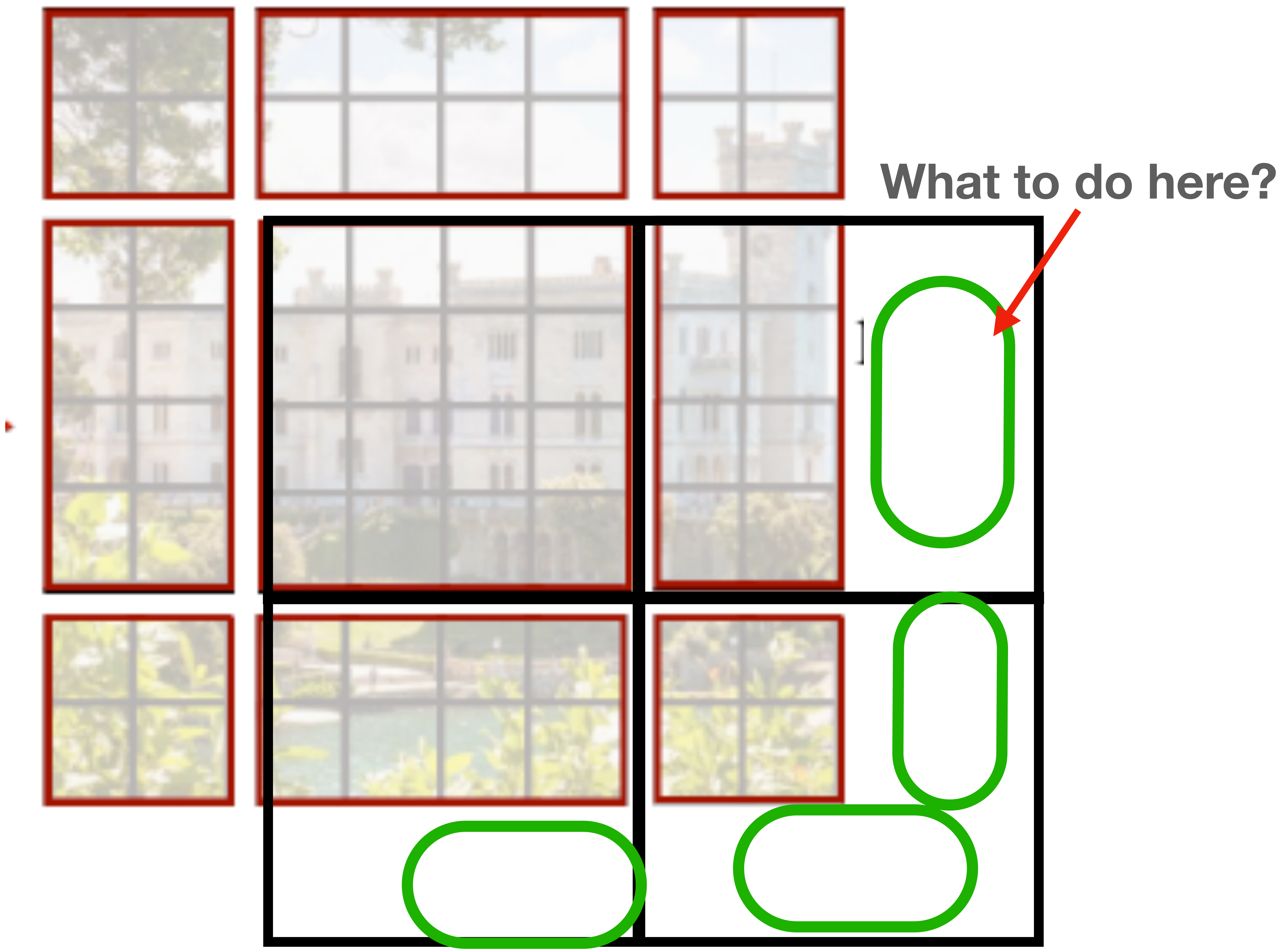


Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: SW-MSA

- Take prior windows, shift by $\left\lfloor \frac{M}{2} \right\rfloor$, where M = size of window
- Non MxM patches stitched together cyclically
- **S**liding **Win**dow transformer
- Convolution-esque

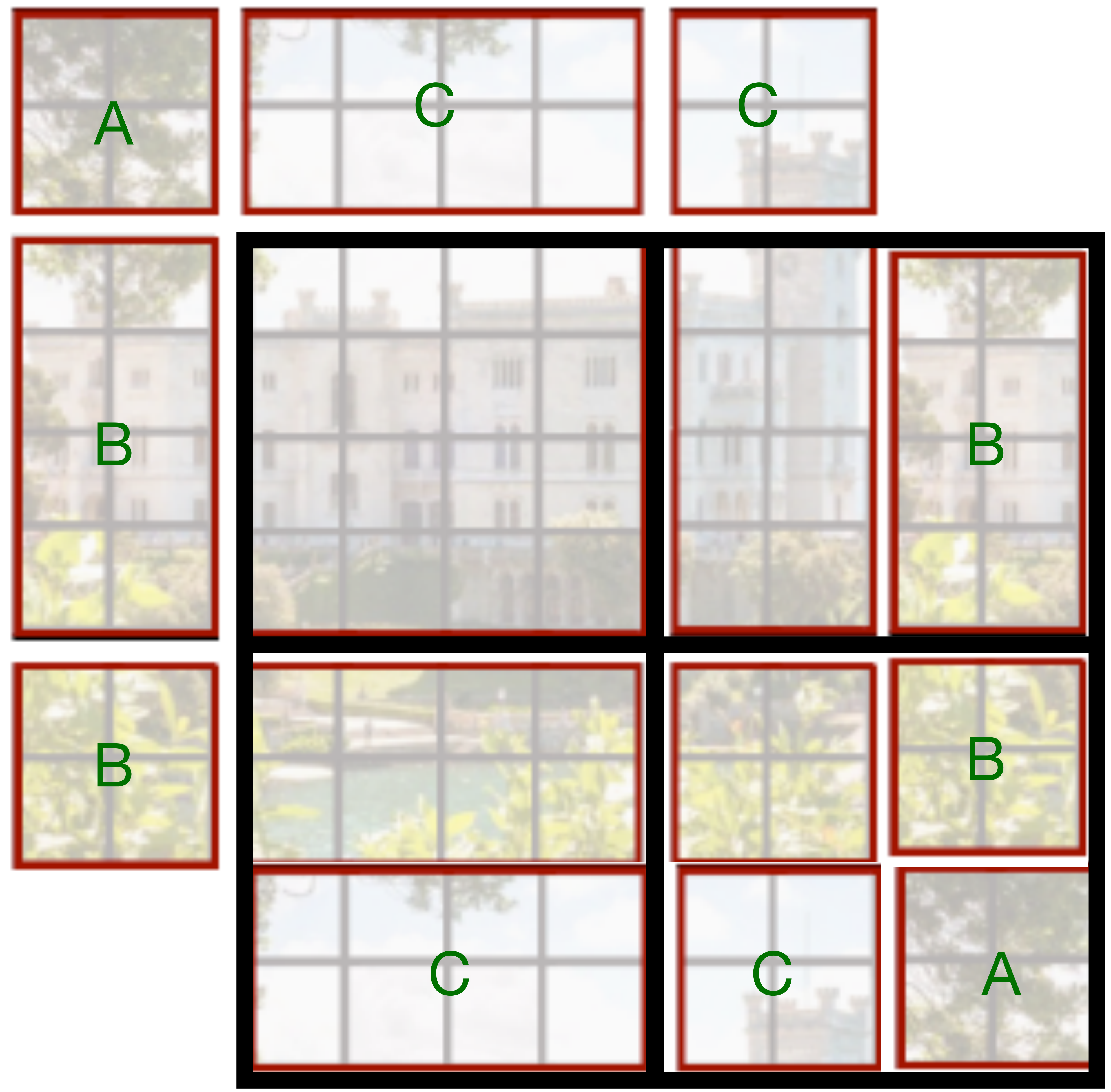


Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Architecture

Composed of stages of patch merging and successive transformer blocks

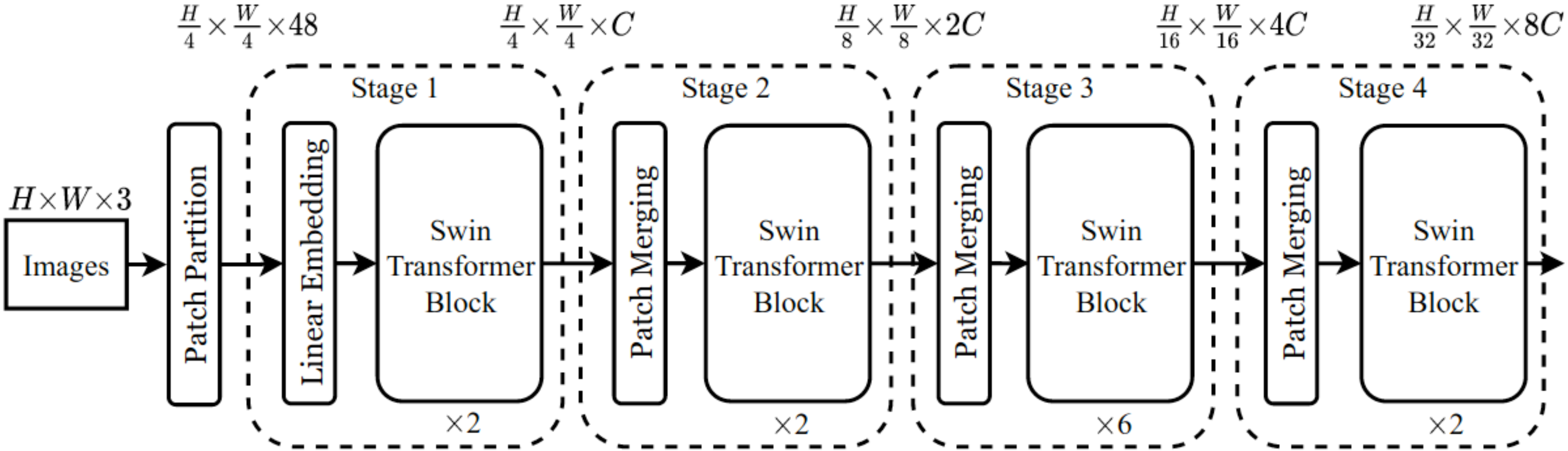


Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Swin Transformer: Instance Segmentation

Experiments ran on COCO 2017

SOTA on detection and segmentation

(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

Image: Liu, Ze, et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." *ArXiv:2103.14030 [Cs]*, Aug. 2021. *arXiv.org*, http://arxiv.org/abs/2103.14030.

Overview

- Faster R-CNN
- Mask R-CNN
- Transformer Background
- SWIN Transformer
- Discussion (and Questions?)