Different Tasks:

Semantic segmentation (UNet / DeepLab)

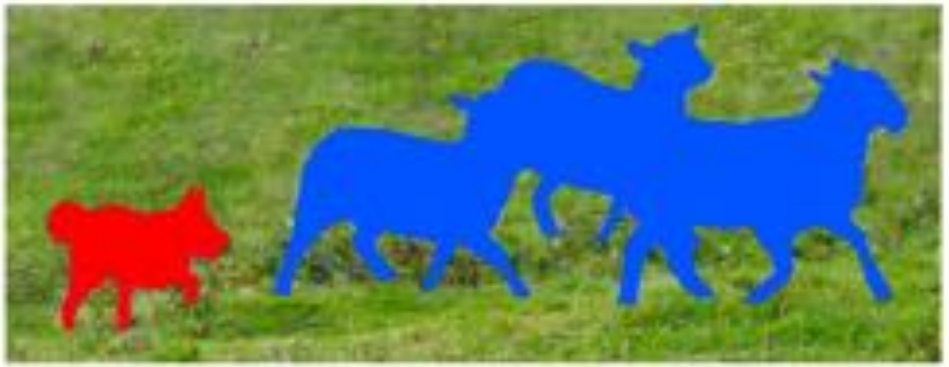
Detection / Instance Segmentation (RCNN)

Five Key Components

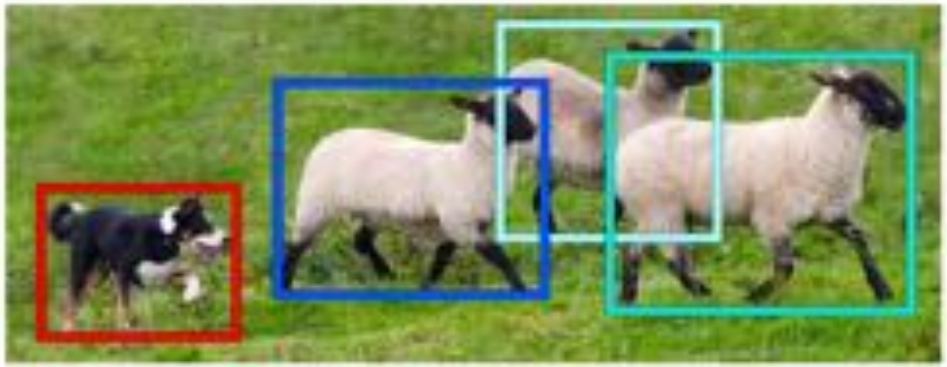
Task:	Data	Models	Metrics	Loss

Consider Different Tasks…..

Image Recognition



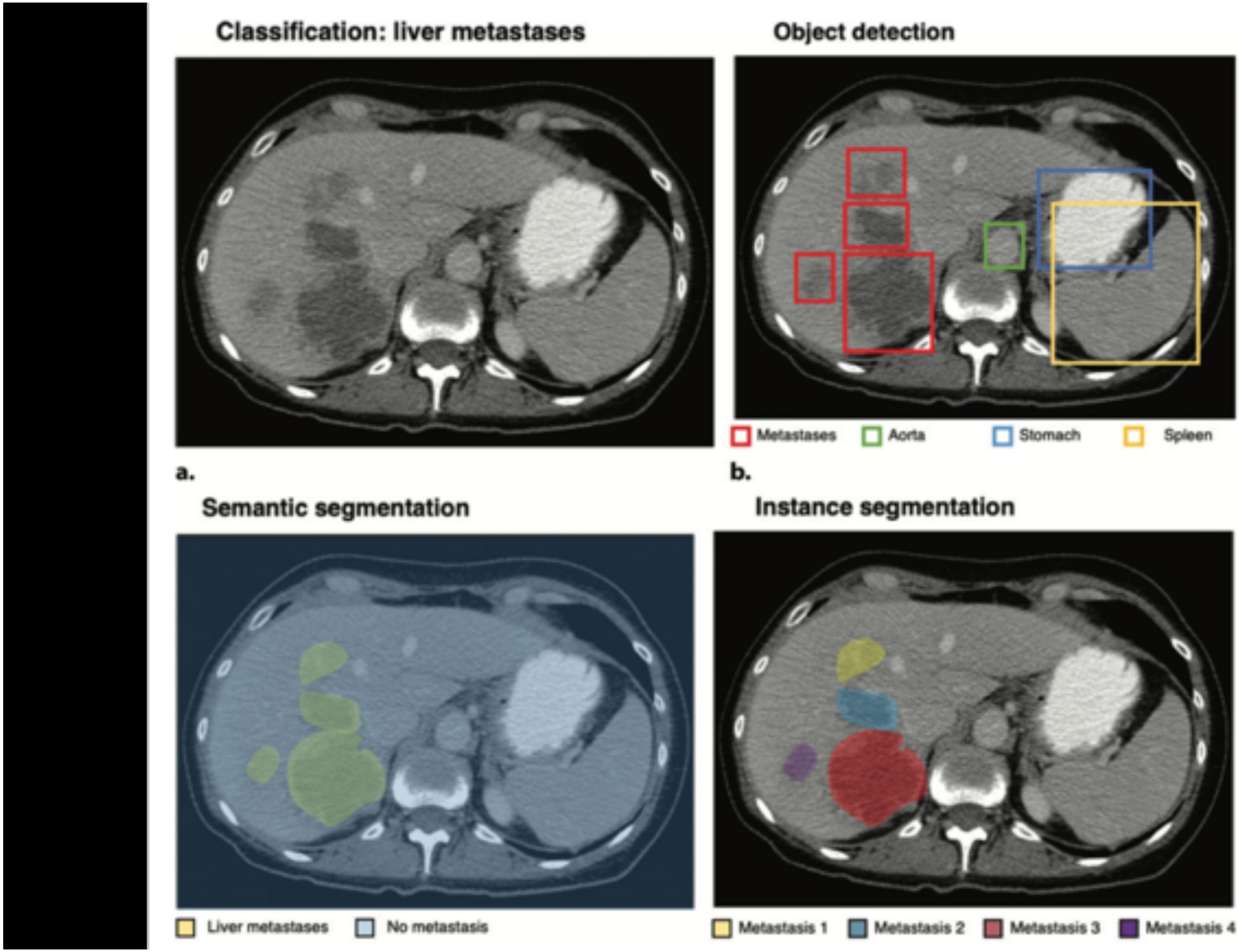
Semantic Segmentation



Object Detection

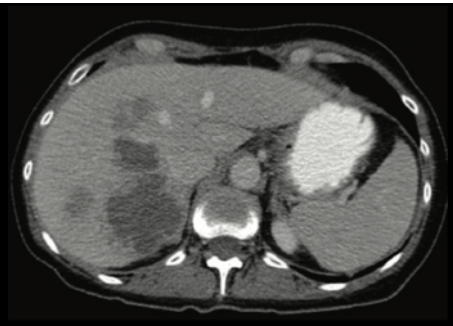
Instance Segmentation

Instance Segmentation is kind of Semantic segmentation + detection

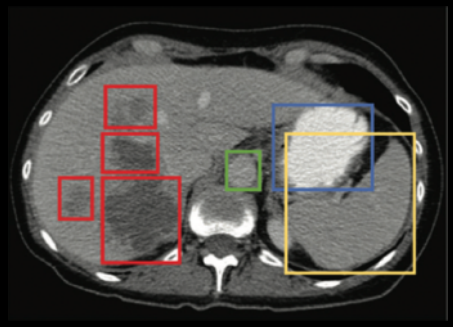

Classification: liver metastases
Object detection
Metastases
Aorta
Stomach
Spleen
a.
b.
Semantic segmentation
Instance segmentation
Liver metastases
No metastasis
Metastasis 1
Metastasis 2
Metastasis 3
Metastasis 4

When to do *what*?

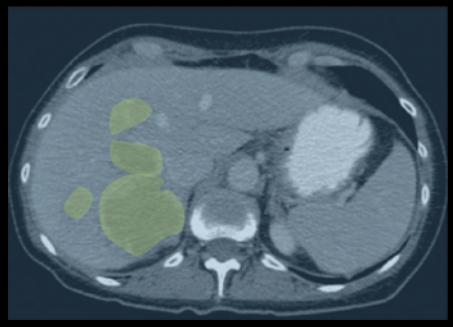
Classification



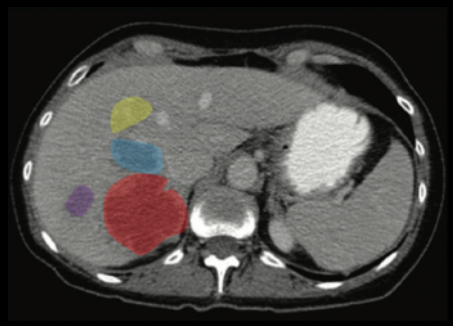
Detection



Semantic Segmentation



Instance Segmentation



No annotation

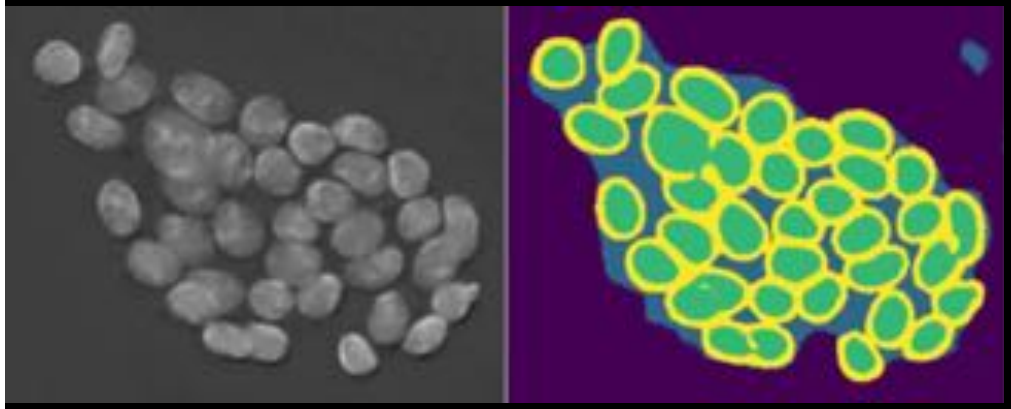
But may need more data

More manual cost

More Information

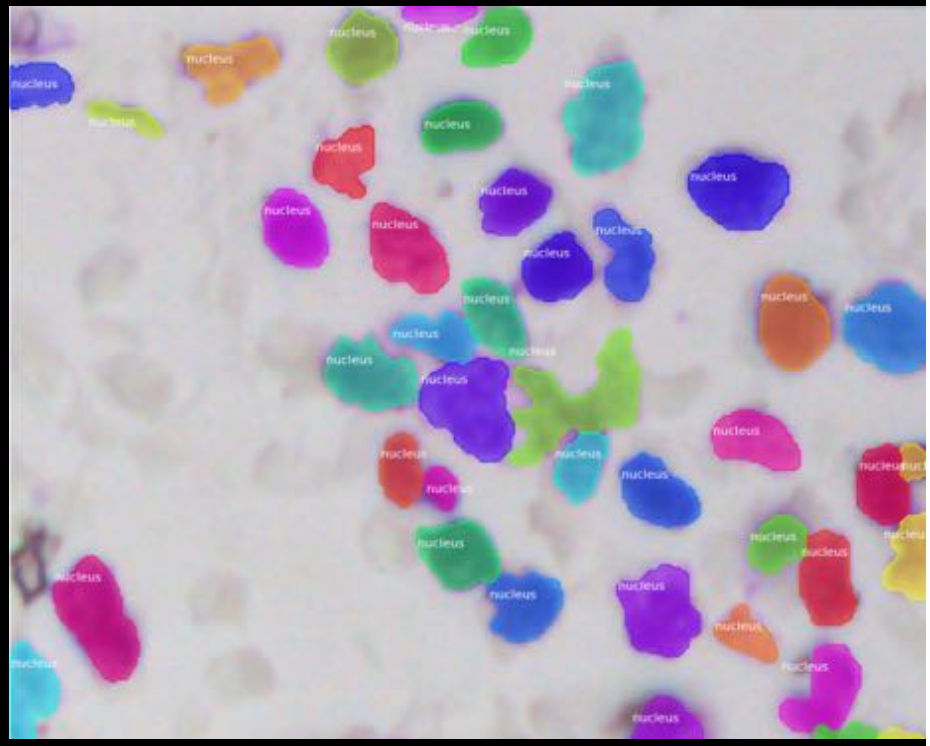
Cell semantic segmentation

(cell + boundaries

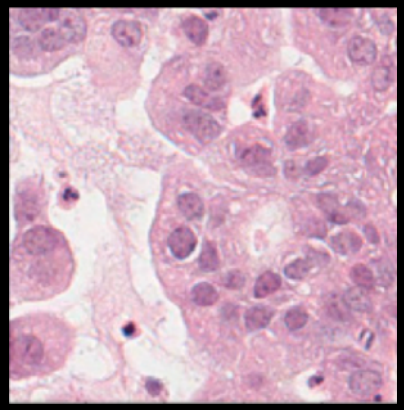


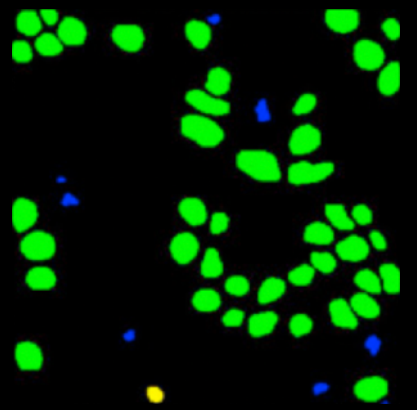
Cell instance segmentation

(cell by cell)

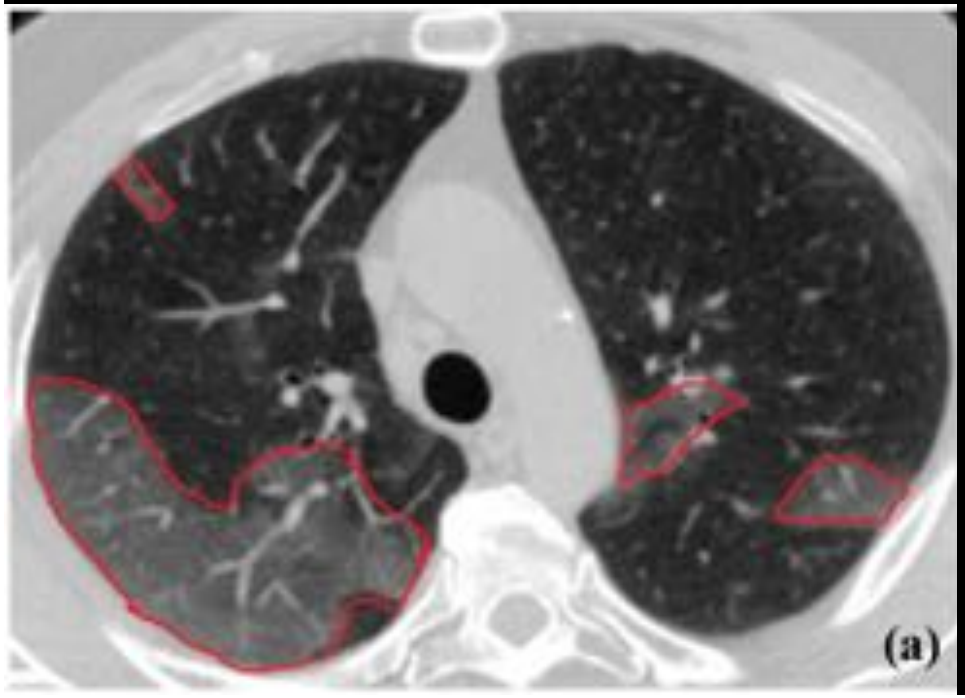


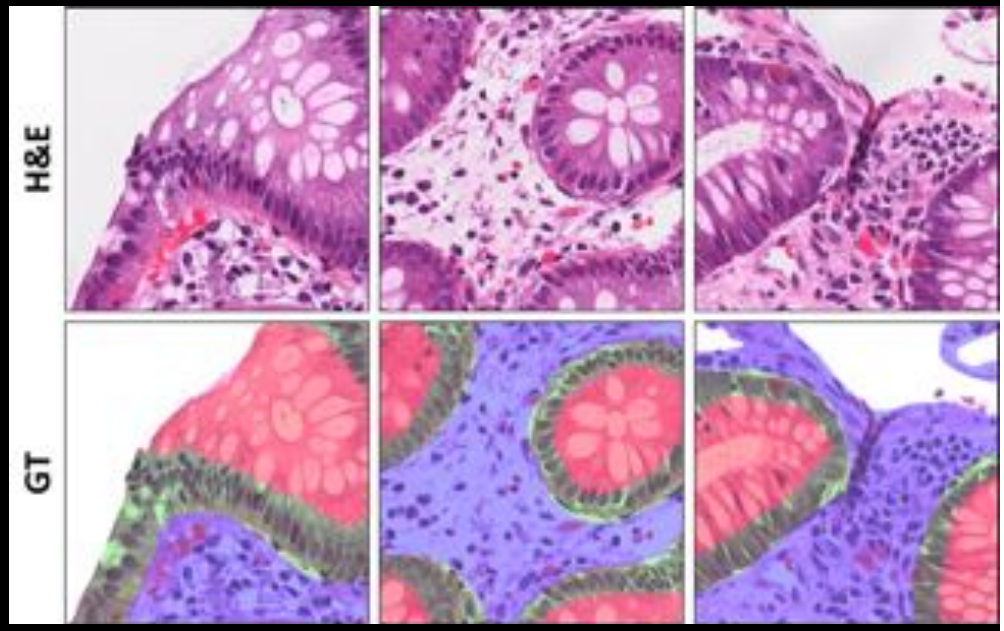
Some instances are clearly defined



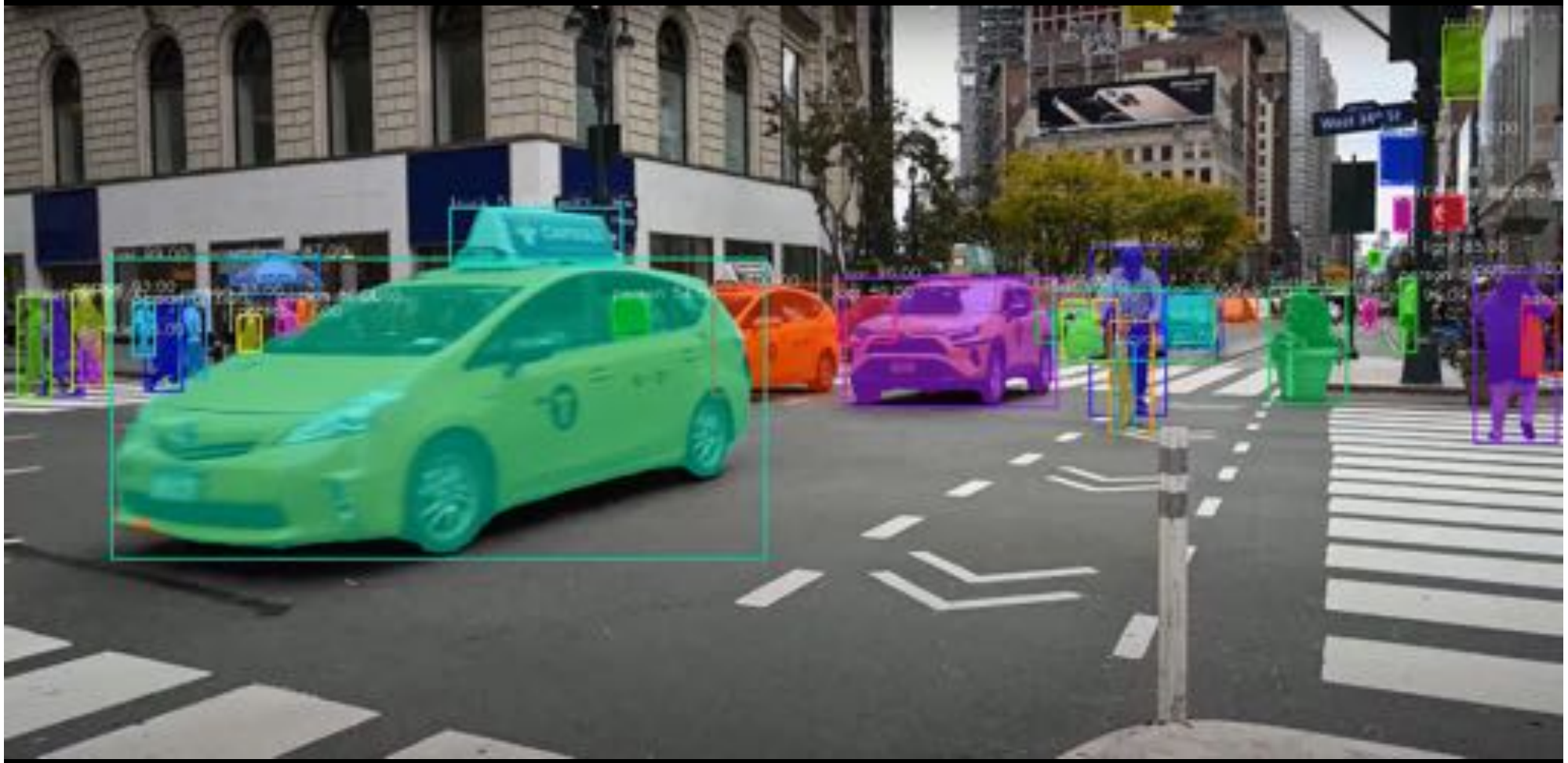


Some are not





Roughly $1 per object / How much per object for your application?



{motorbike,person}

{motorbike (point), person (point)}

{motorbike (b-box), person (b-box)}

{motorbike (pixel labels), person (pixel labels)}

1 sec per class

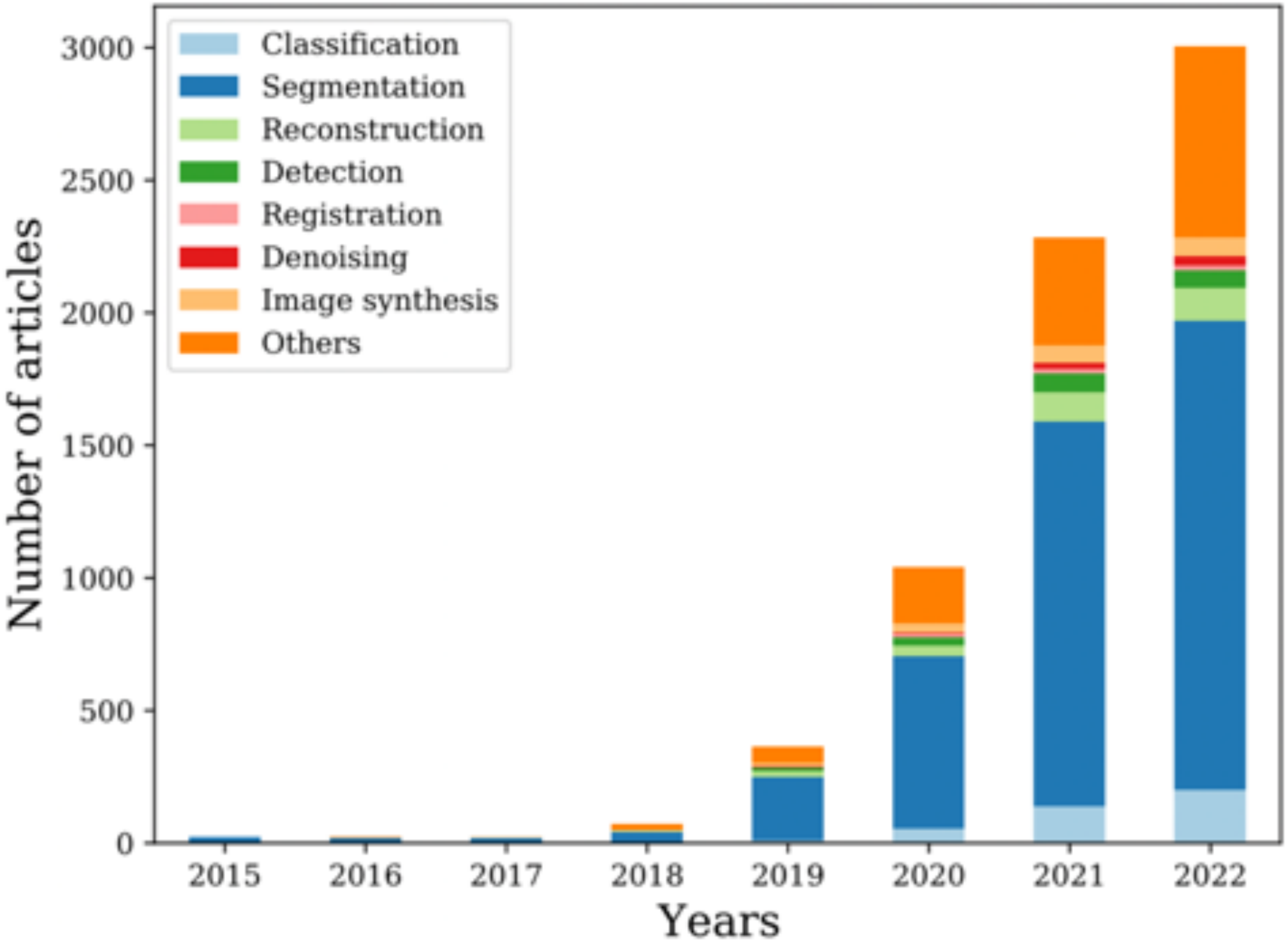
2.4 sec per instance

10 sec per instance

78 sec per instance

annotation time

Berman et al., What's the Point: Semantic Segmentation with Point Supervision, ECCV 16


Classification
Segmentation
Reconstruction
Detection
Registration
Denoising
Image synthesis
Others
Number of articles
3000
2500
2000
1500
1000
500
0
2015
2016
2017
2018
2019
2020
2021
2022
Years

Encoder only segmentation (FCN)

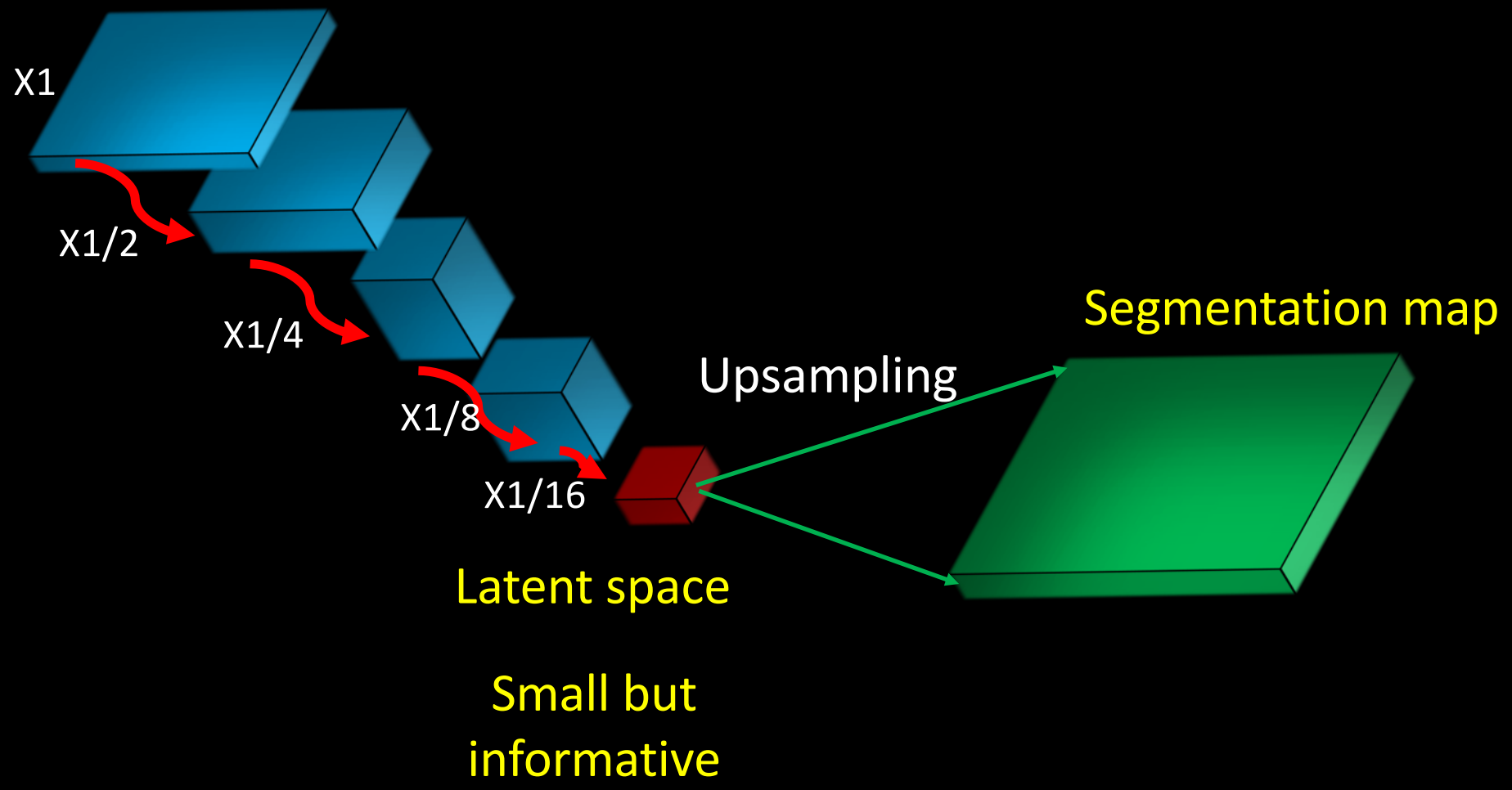
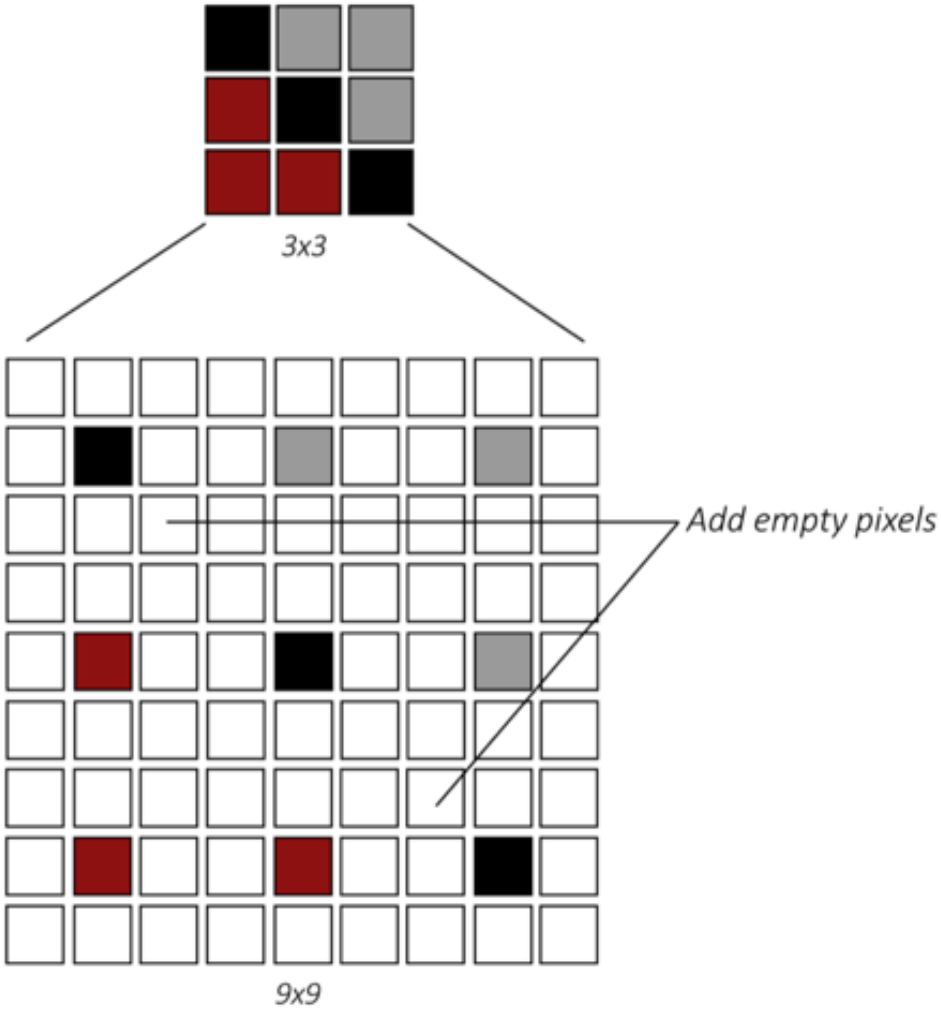
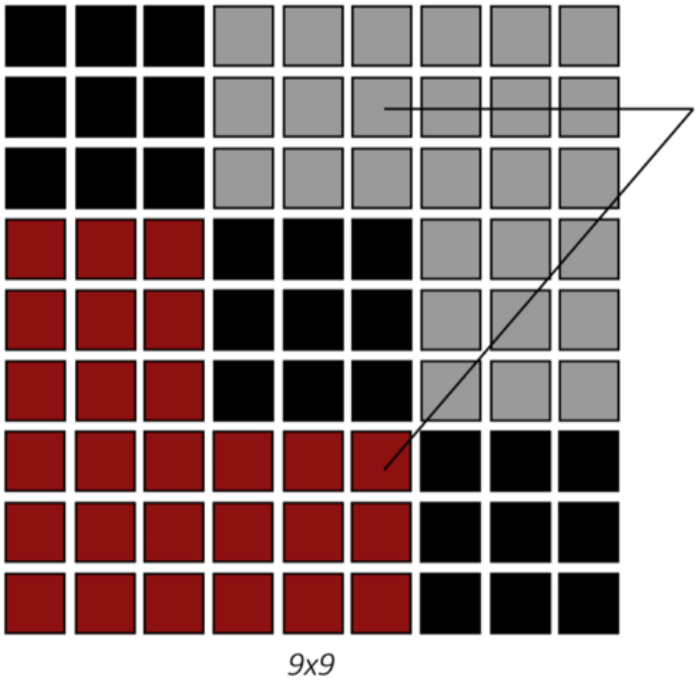


Image Up-sampling Problem



Fill by Nearest Neighbors



Low Resolution

Nearest Neighbor Interpolation

Bilinear Interpolation

Super-Resolution

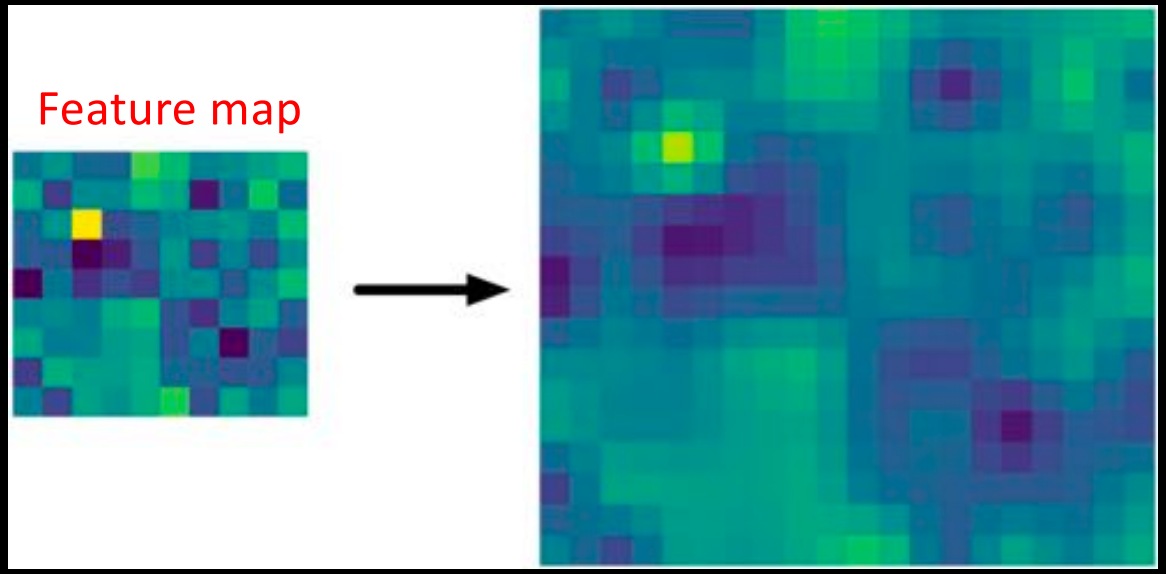
Accurate but not smart

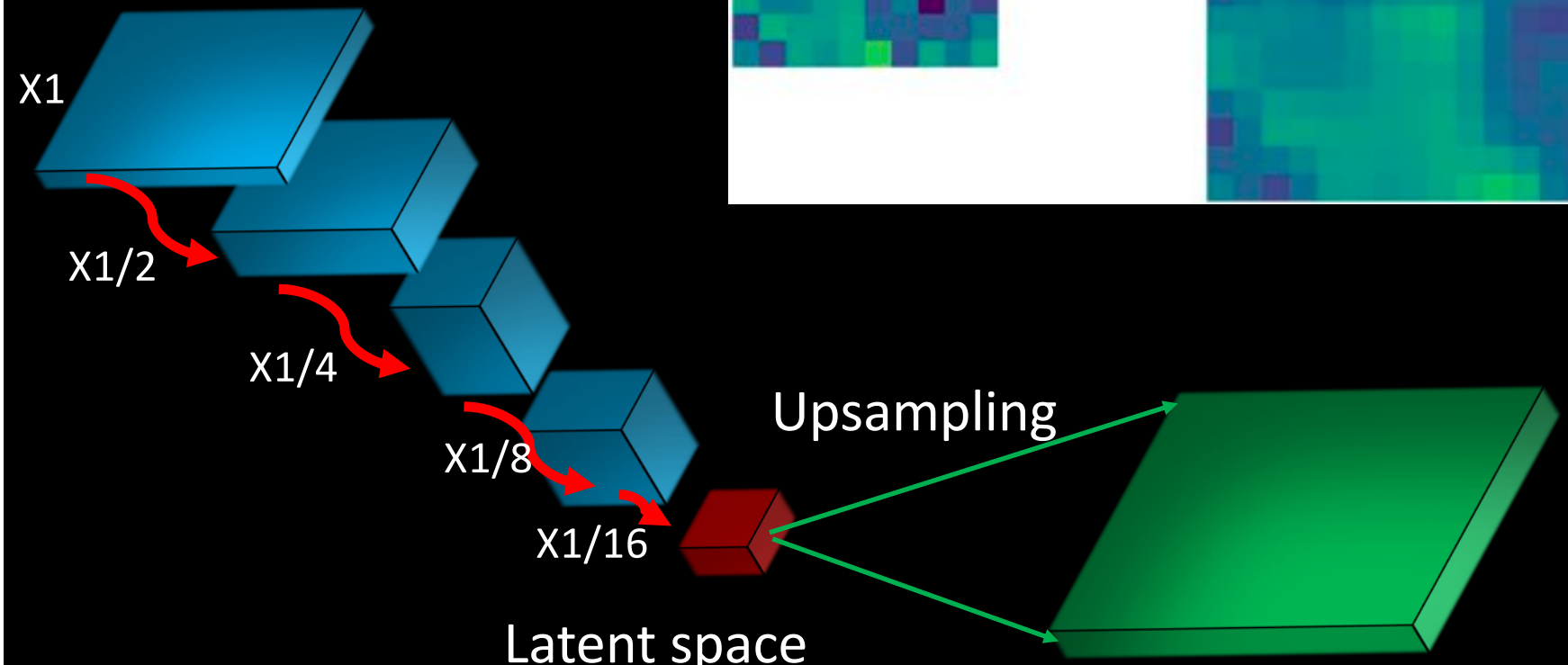
Vague Boundaries

“smart” interpolation

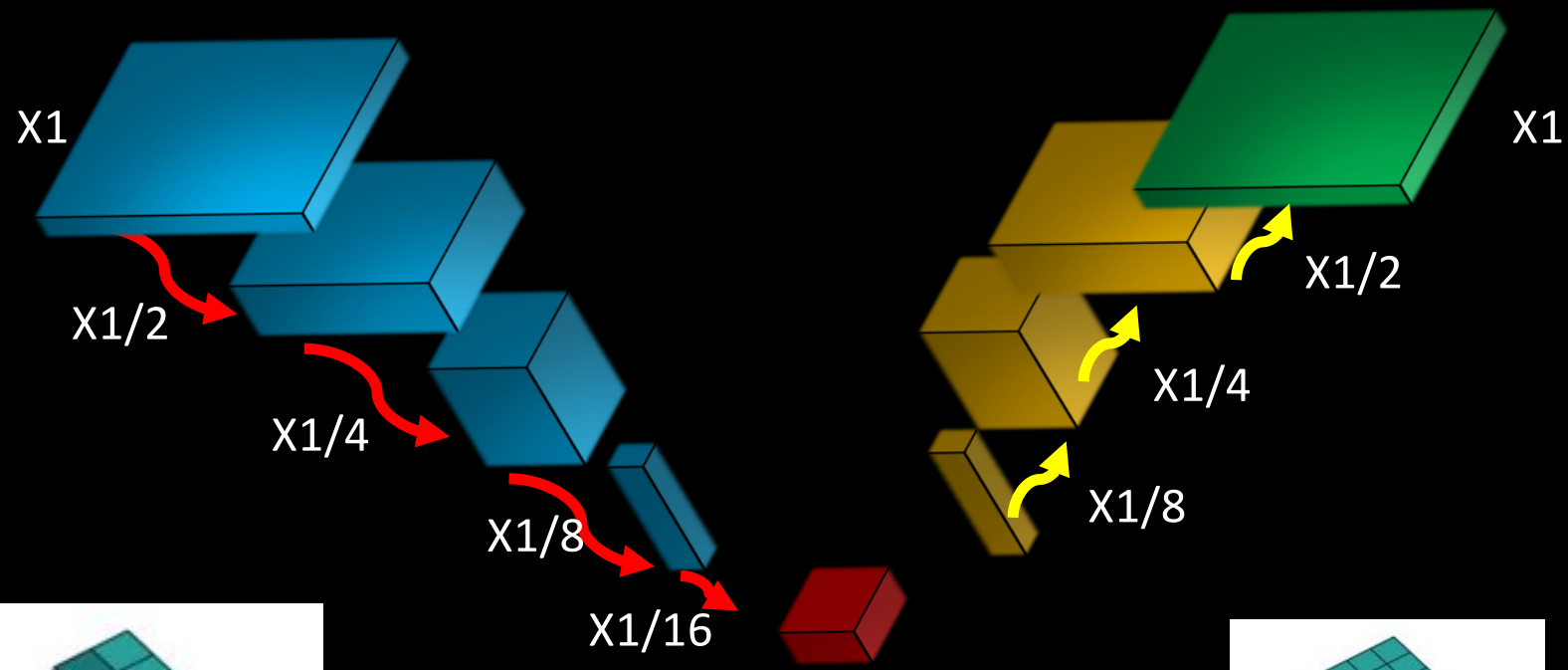
Encoder only segmentation (FCN)

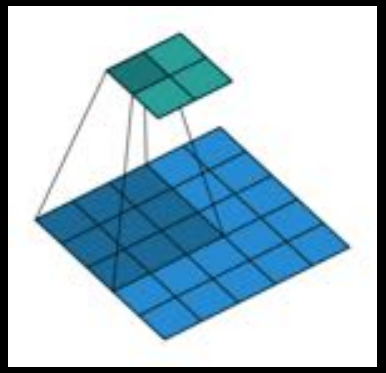
Upsampled (not very smart!)

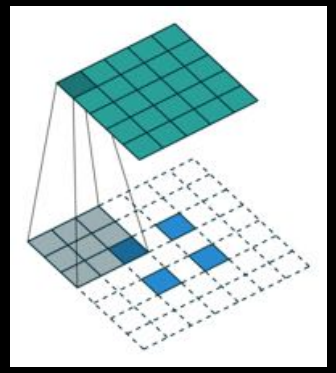




Decoding branch for
processing while upsampling

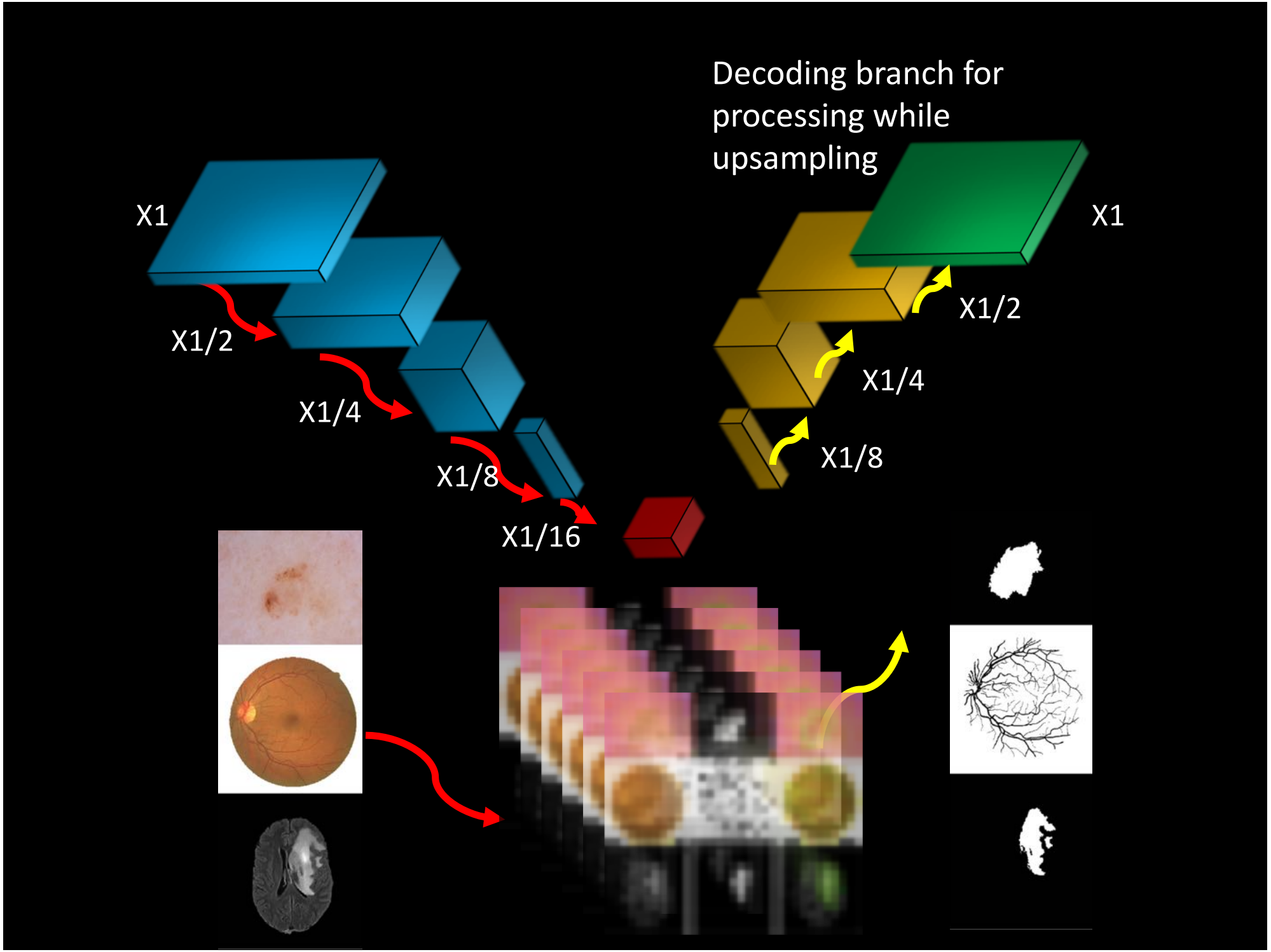


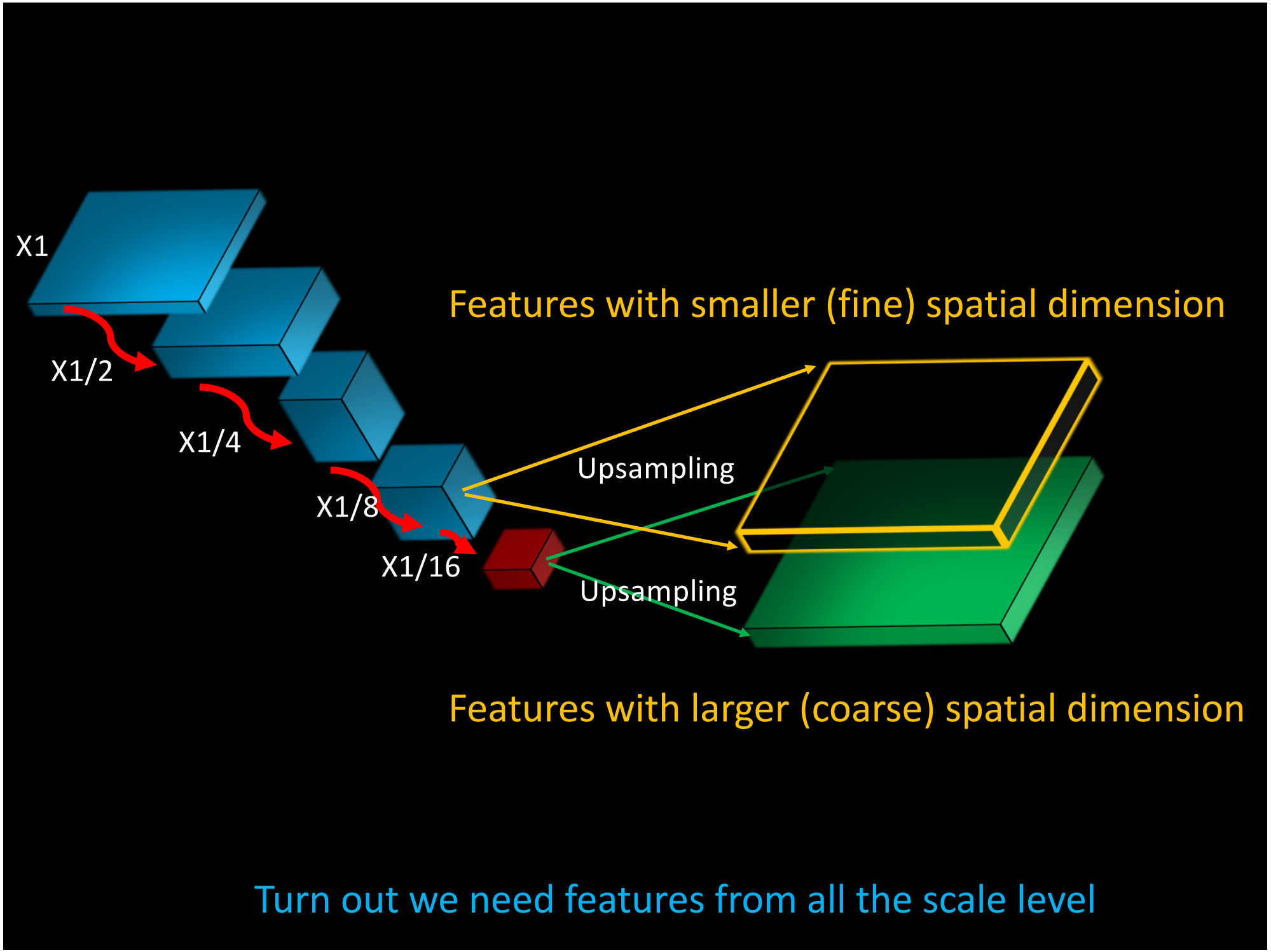




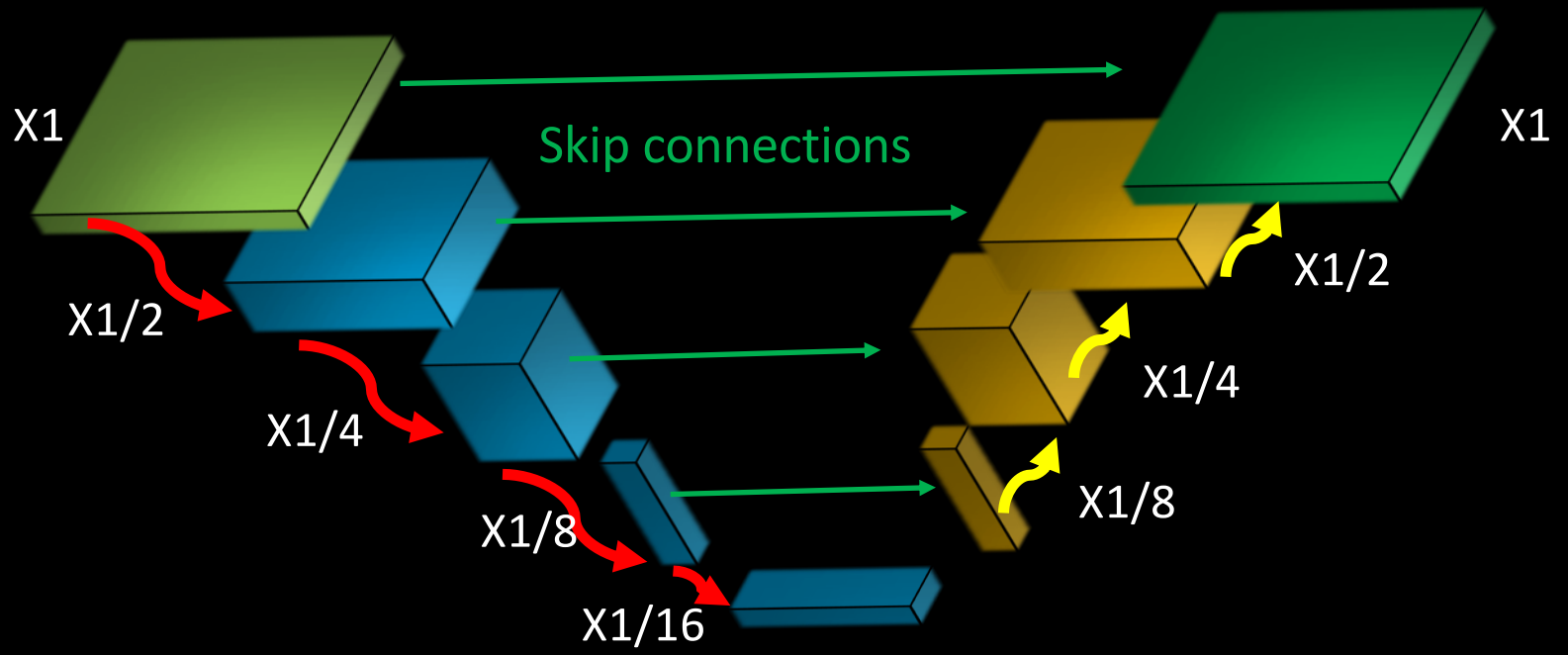
Convolution
(increase features, decrease spatial)

Deconvolution
(decrease features, increase spatial)

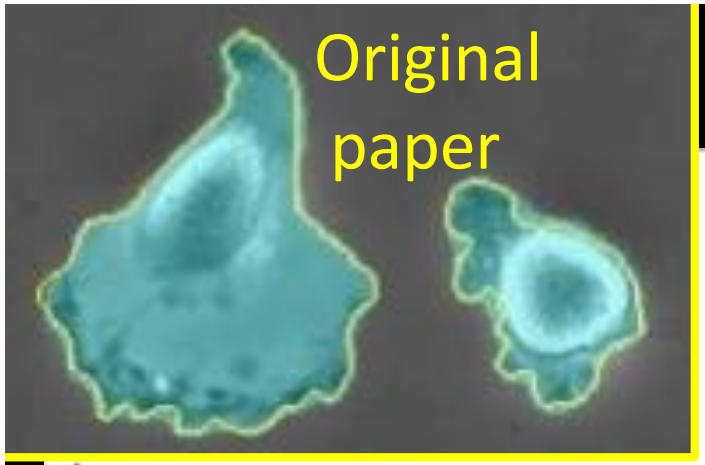

Decoding branch for processing while upsampling
X1
X1/2
X1/4
X1/8
X1/16
X1
X1/2
X1/4
X1/8


X1
X1/2
X1/4
X1/8
X1/16
Features with smaller (fine) spatial dimension
Upsampling
Upsampling
Features with larger (coarse) spatial dimension
Turn out we need features from all the scale level

U-Net (encoder-decoder w/ skip connection)

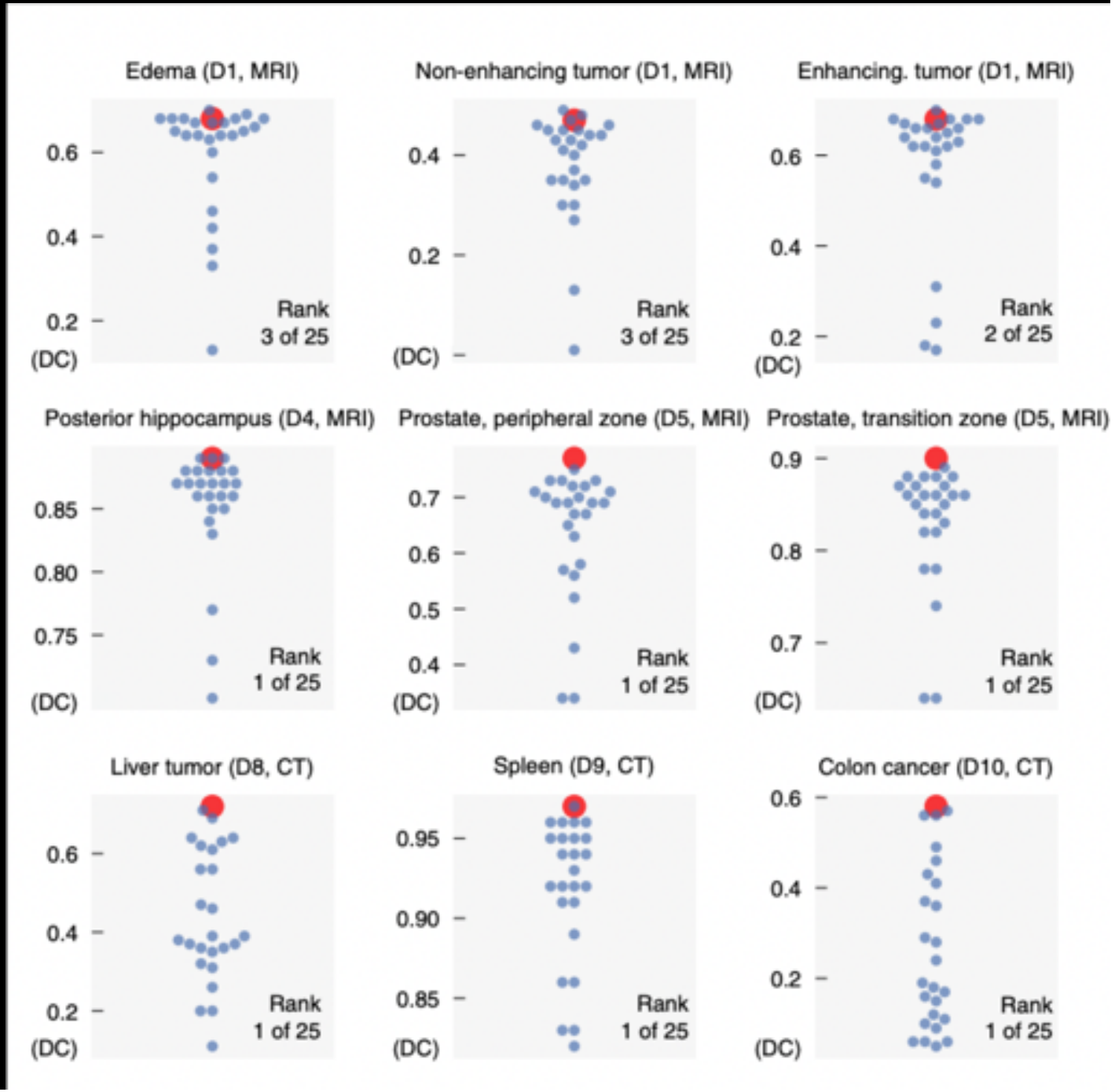


Skip connections: translate full size details (find and coarse) from left to right



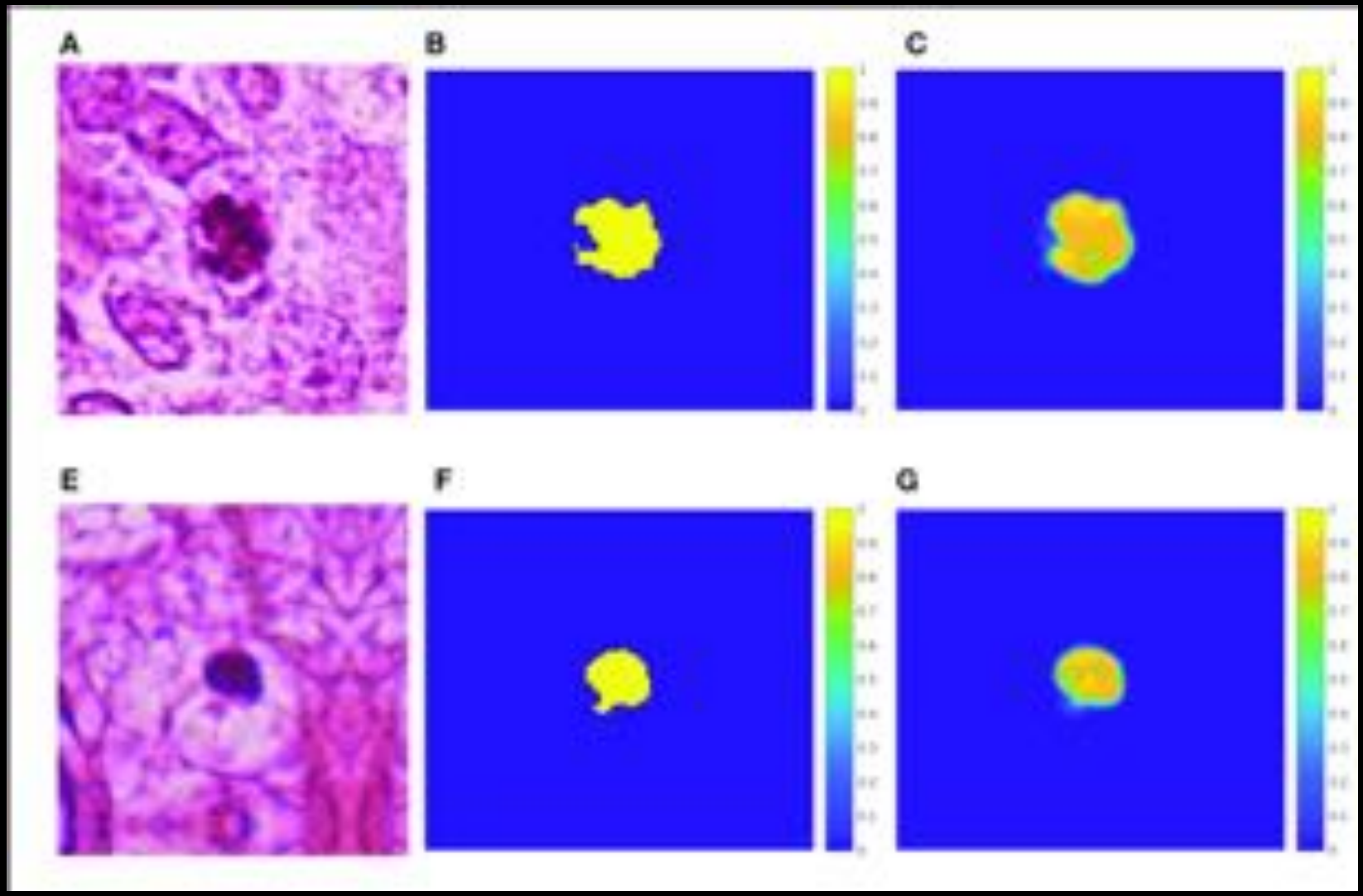
Applications

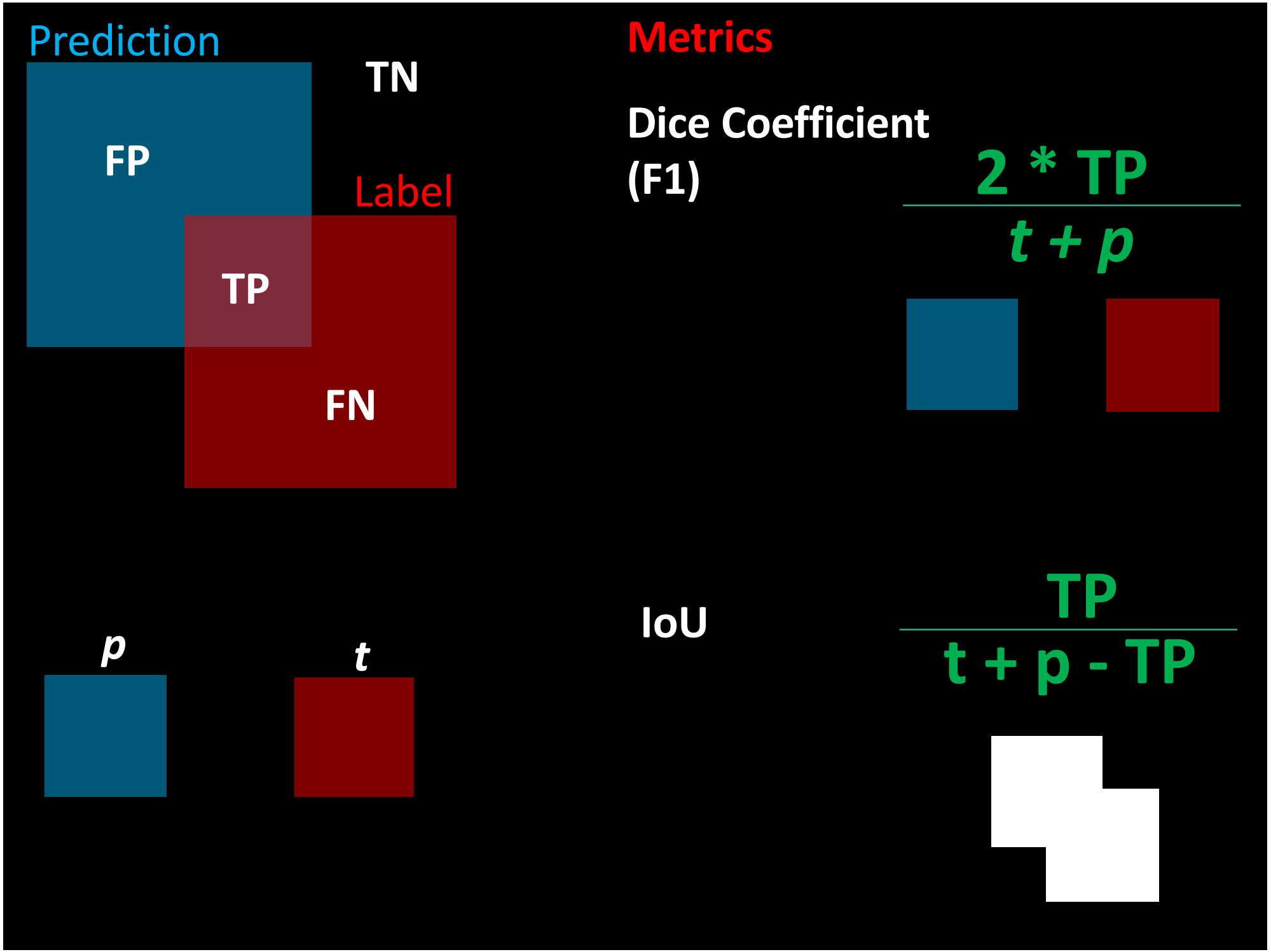
nn (not new) unet



Loss and metrics of segmentation

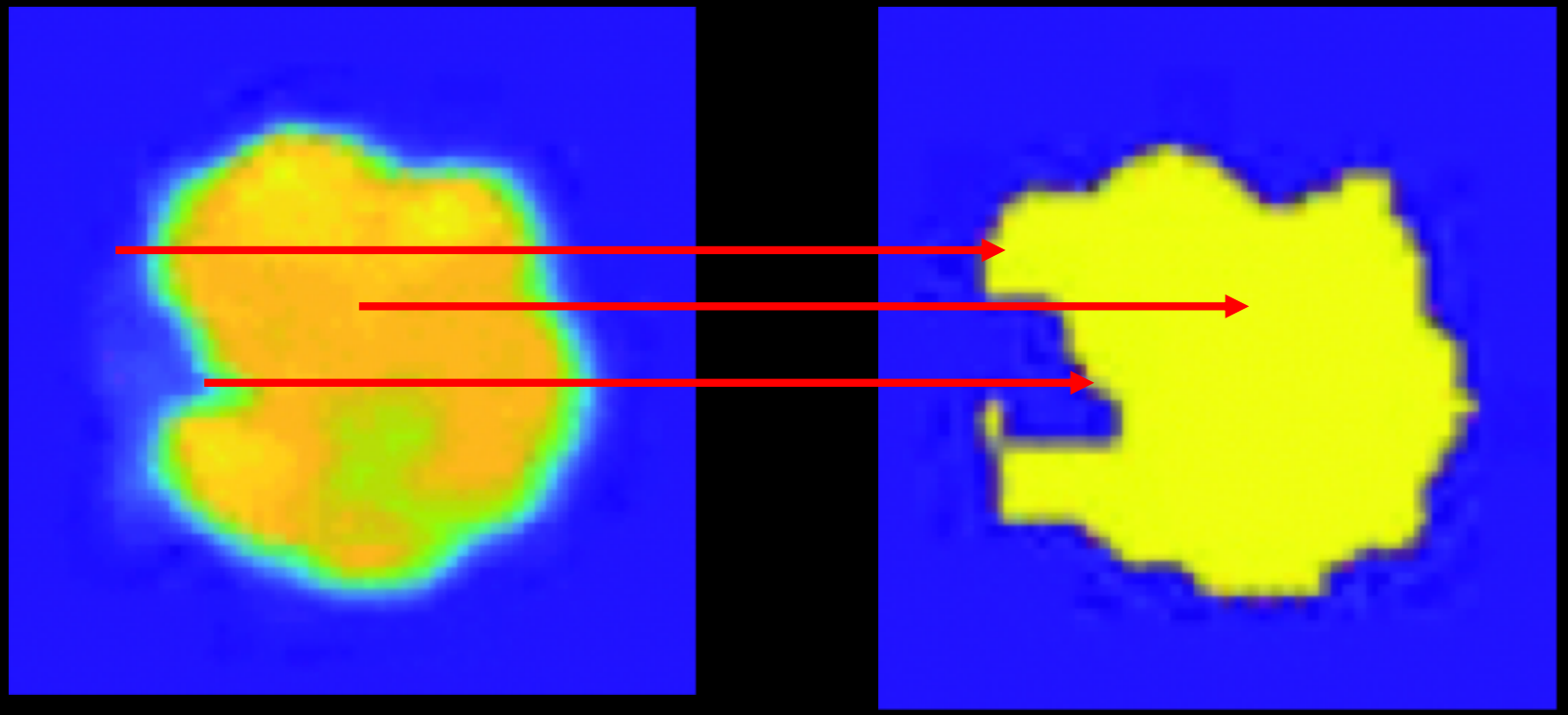
Original Ground-truth Prediction (probability)




Prediction
TN
FP
Label
TP
FN
p
t
Metrics
Dice Coefficient
(F1)
2 * TP
t + p
IoU
TP
t + p - TP

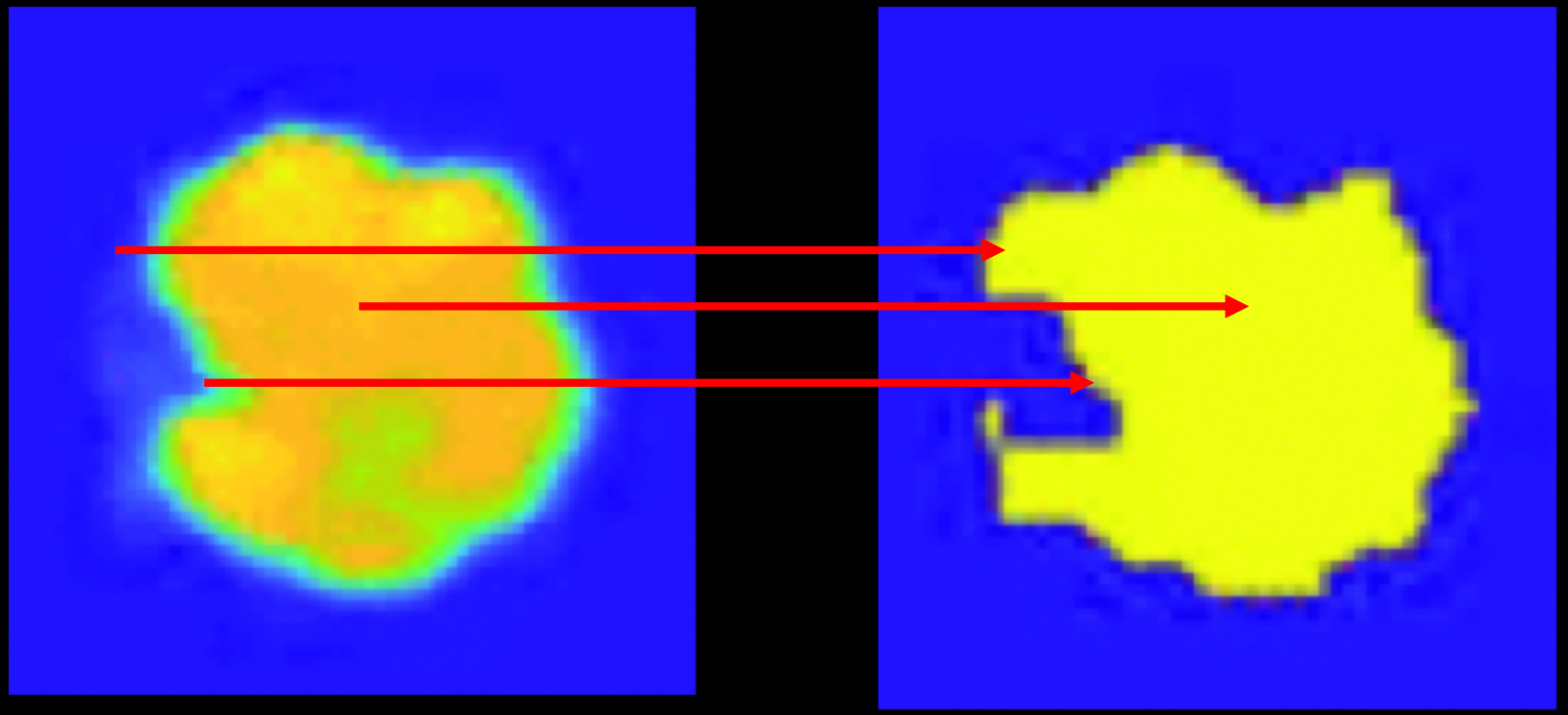
Prediction (probability)

Ground-truth



Its basically pixel-by-pixel classification !!

Prediction (probability) Ground-truth



Prob.	Label	Cross entropy = -log(q)*p	
0.01	1	-log(0.01) * 1 = 2	False Negative. (FN)
0.95	1	-log(0.95) * 1 = 0.02	True Positive (TP)
0.6	0	-log(1-0.60) * (1-0) = 0.53	False Positive. (FP)


U-Net Taxonomy
Transformers
Transformer Complement to CNN-based U-Net
1. TransU-Net
2. TransBTS
3. GT U-Net
4. CoTr
5. UNETR
6. MTU-Net
7. TMU
8. UCTransNet
9. ScaleFormer
Standalone Transformer Backbone for U-Net Design
10. MedT
11. Swin-UNet
12. MISSFormer
13. D-Former
Probabilistic Design
Graphical Model Algorithms
14. MRF-UNet
15. Bayesian Skip Net
Variational Auto Encoder (VAE) Regularization
16. VAE U-Net
17. Probabilistic U-Net
Rich Representation Enhancement
Multi-scale Fusion
18. Focal Tversky Attention U-Net
19. Polar Transformation M-Net
20. MFP-UNet
Multi-modality Fusion
21. Dense Multi-path U-Net
22. Cascaded U-Net
Leveraging Depth Information
23. 3D Attention U-Net
Backbone Design Enhancement
Residual
24. Residual U-Net
25. V-Net
26. H-DenseUnet
Multi-Resolution
27. MultiResU-Net
28. DR-VNet
29. DoubleU-Net
Re-considering Convolution
30. DU-Net
31. SSD-UNet
Recurrent U-Net
32. DRU-Net
33. R2U-Net
Skip Connection Enhancement
Increasing the Number of Skip Connections
34. U-Net++
35. U-Net 3+
36. BiO-Net
Processing Feature Maps within the Skip Connections
37. Attention U-Net
38. RA-UNet
39. Attention UNet++
40. Projective Skip-Connections
Combination of Encoder and Decoder Feature Maps
41. BCDU-Net
42. CR-Unet
Bottleneck Enhancement
Attention Modules
43. MA-Net
44. SA-UNet
45. RIC U-Net
46. FRCU-Net
47. SMU-Net
Multi-scale Representation
48. ASPP-FC DenseNet
49. COPLE-Net
50. ICS

Nested UNet

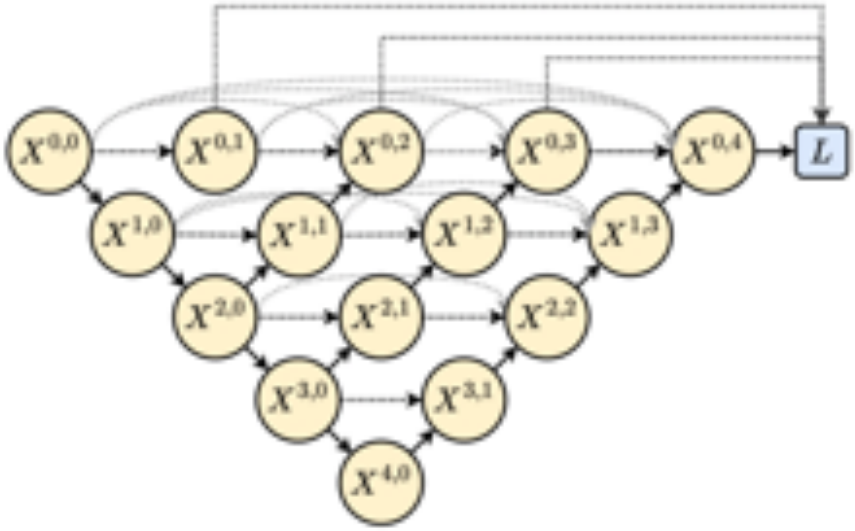


Fig. 6: Architecture of the U-Net++ [67]. The U-Net++ uses a nested structure of convolutional layers $X^{i,j}$ (indicate j-th convolution layer in the i-th scale) connected through the skip connection paths to model multi-scale representation.

Fully-connected Skip Connection

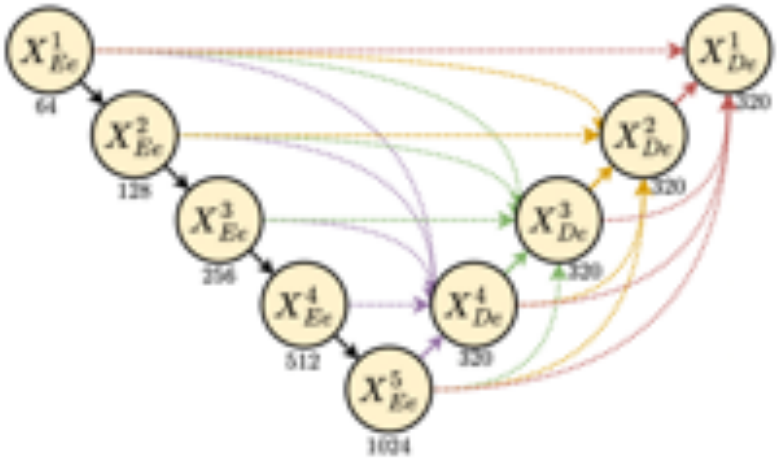
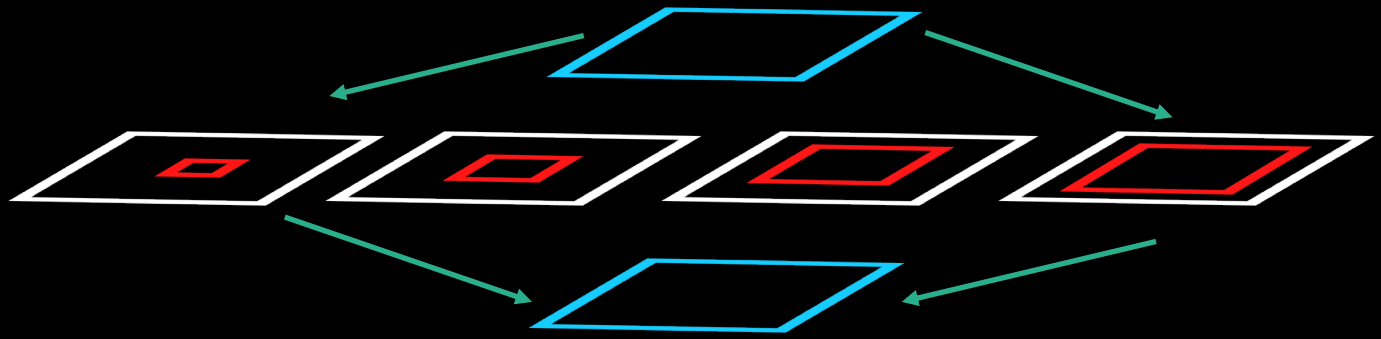
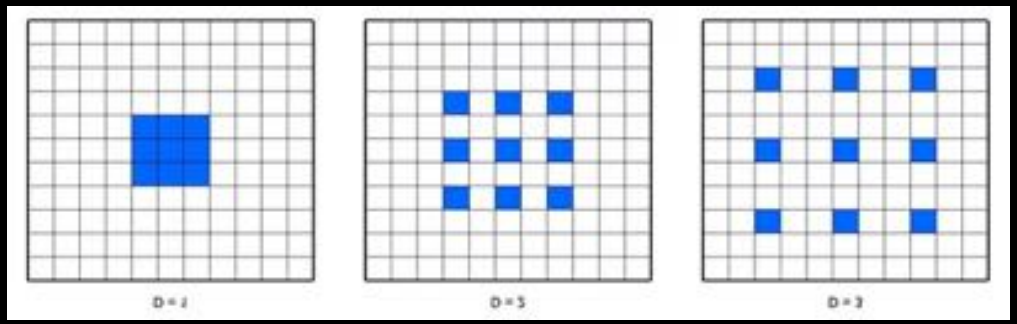


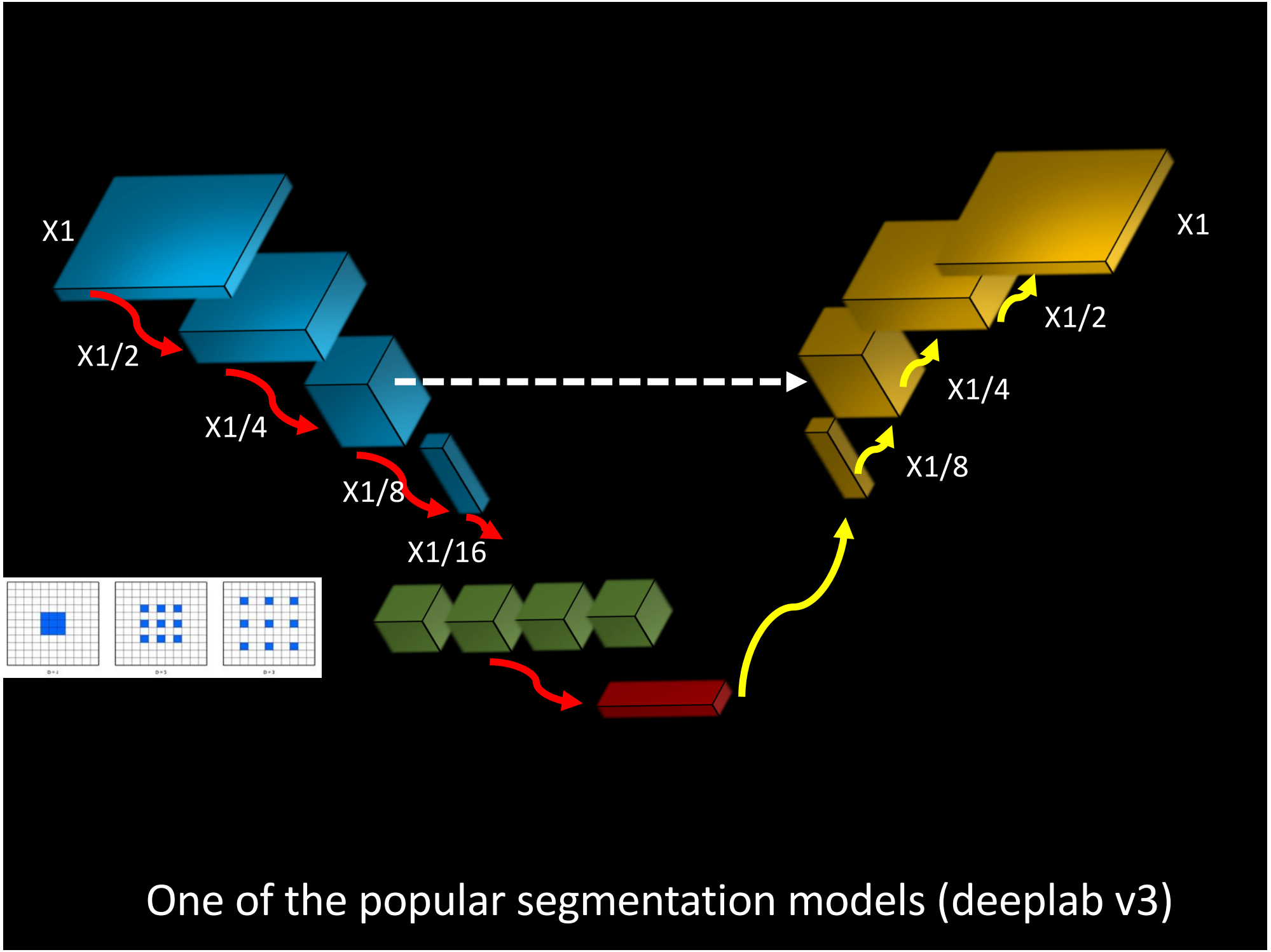
Fig. 7: Architecture showing the full-scale skip connections of the UNet3+ [68]. X^i_{En} and X^i_{De} denote the encoder and decoder path feature maps at i-th scale, respectively. In addition, each subscript under each feature map symbol represents the number of feature maps to the corresponding block.

Another popular segmentation model: Deeplab Families

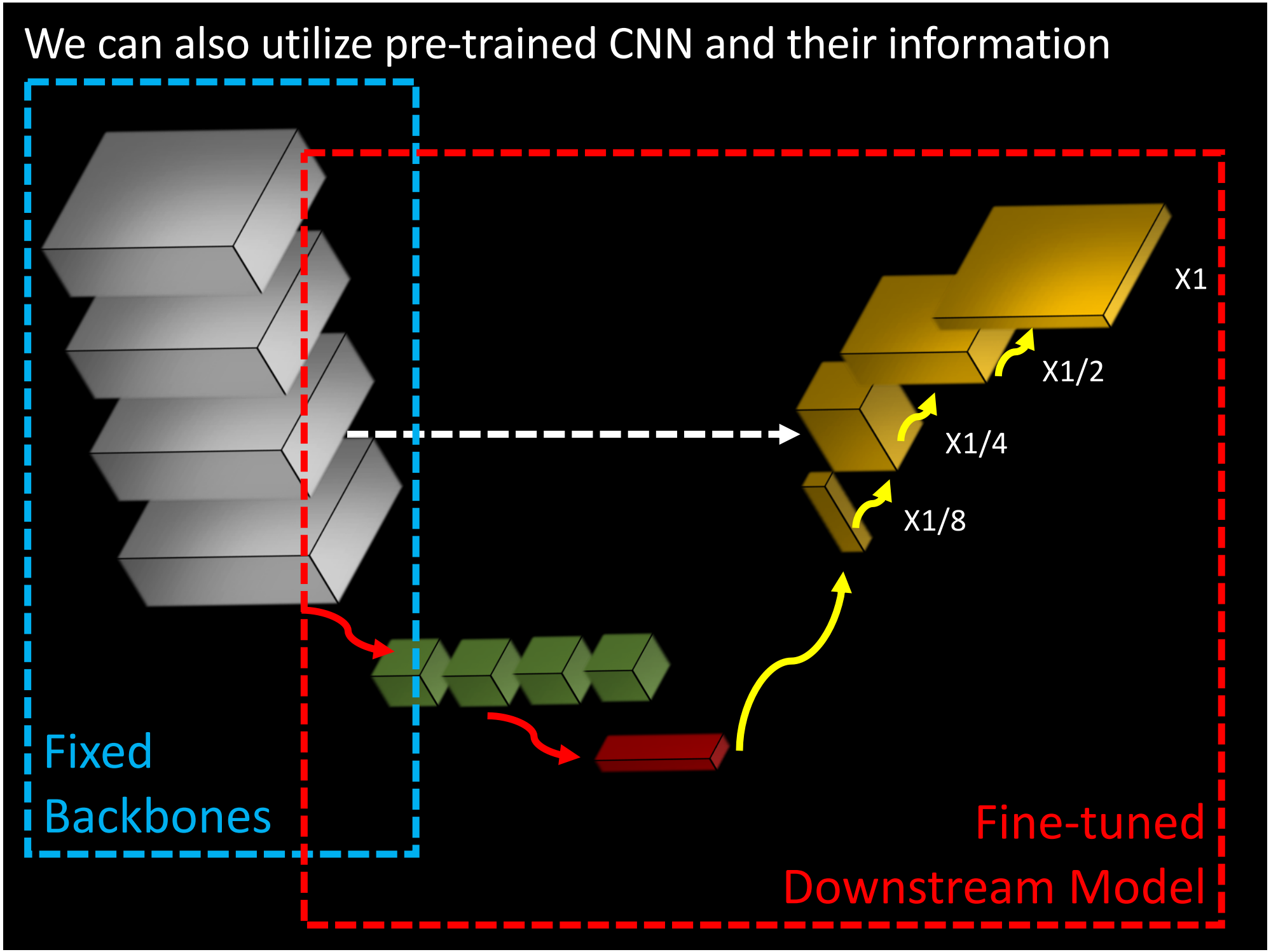


Spatial Pyramid Pooling





One of the popular segmentation models (deeplab v3)


We can also utilize pre-trained CNN and their information
X1
X1/2
X1/4
X1/8
Fixed
Backbones
Fine-tuned
Downstream Model

Backbones:
features extractors pretrained with a massive dataset for classification, and
useful for "downstream" tasks

Q1: In stomach in this picture?

Q2: Can you point it out? → Downstream tasks

Q3: Can you draw the outline? → Downstream tasks

Deep Learning researchers (Mildly exaggerated)

How to use an existing model with bare-minimum understanding:

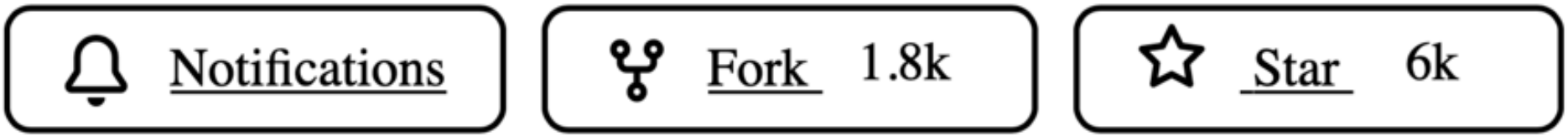
Welcome to the new nnU-Net!

Click here if you were looking for the old one instead.

Coming from V1? Check out the TLDR Migration Guide. Reading the rest of the documentation is still strongly recommended ;-)

https://github.com/MIC-DKFZ/nnUNet

Make sure the code is trustworthy &
can be reproduced



Usually, there is a pre-defined data format

```
nnUNet_results/
├── Dataset002_Heart
│   ├── nnUNetTrainer__nnUNetPlans__2d
│   │   ├── fold_0
│   │   ├── fold_1
│   │   ├── fold_2
│   │   ├── fold_3
│   │   ├── fold_4
│   │   ├── dataset.json
│   │   ├── dataset_fingerprint.json
│   │   └── plans.json
│   └── nnUNetTrainer__nnUNetPlans__3d_fullres
│       ├── fold_0
│       ├── fold_1
│       ├── fold_2
│       ├── fold_3
│       ├── fold_4
│       ├── dataset.json
│       ├── dataset_fingerprint.json
│       └── plans.json
```

First, understand the pre-defined format

uuUNet_raw
Dataset001_heart
imagesTr
heart_0001_0000_nii_gz
imagesTs
heart_0001_0000_nii_gz

Load the XXX.nii.gz file for me and
Print out the datatype and dimension

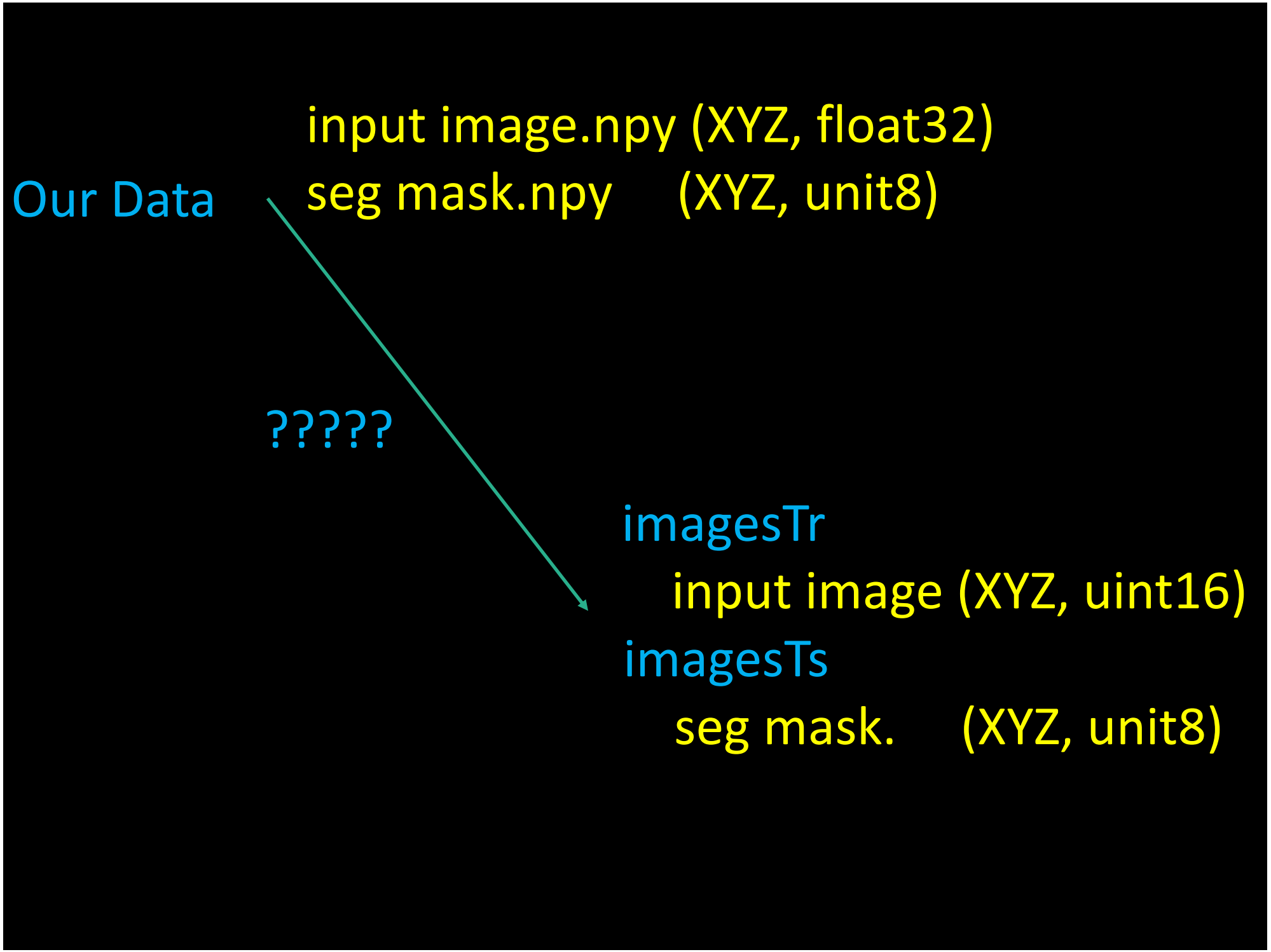
imagesTr

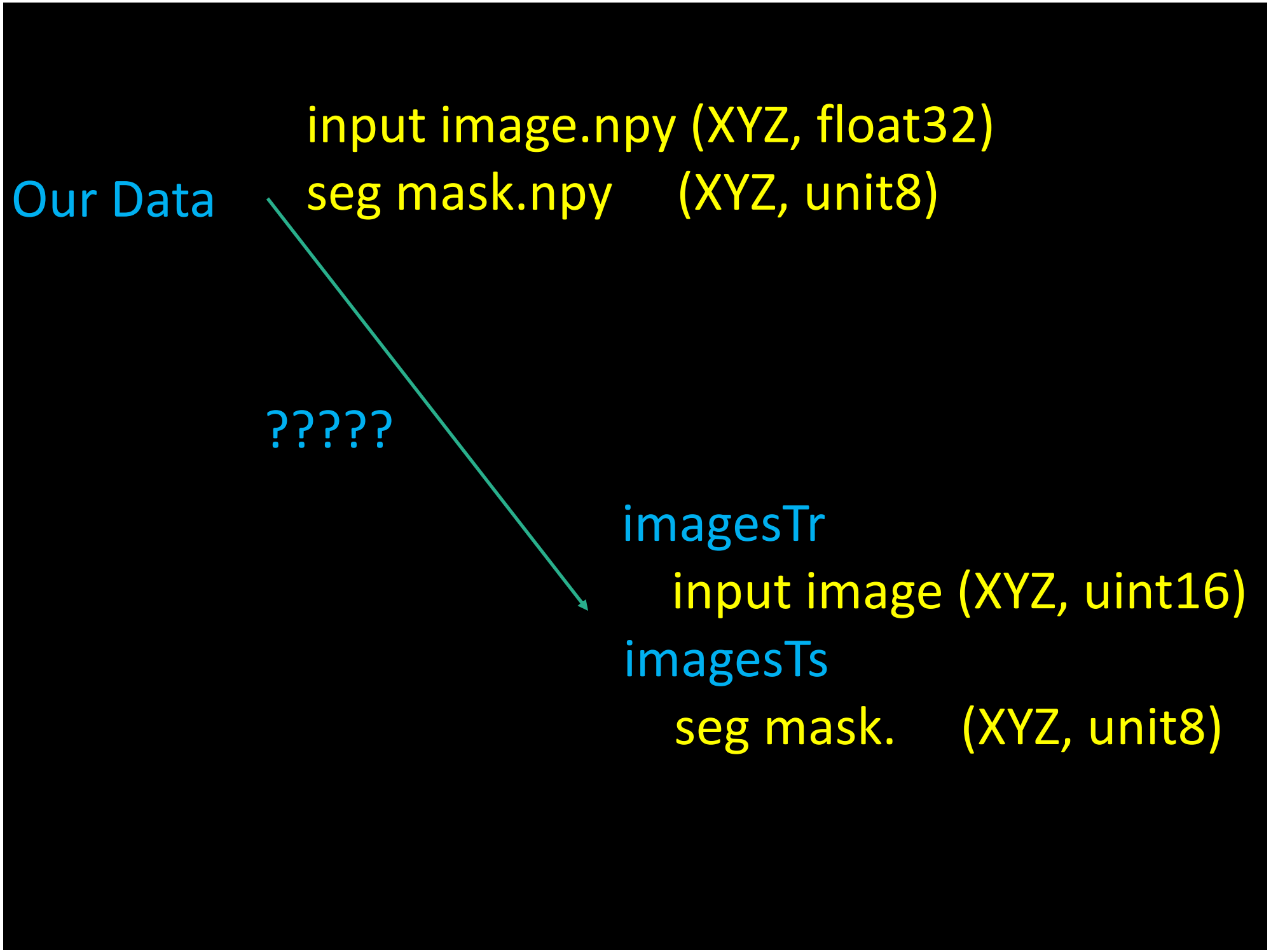
heart_0001_0000_nii_gz

imagesTs

heart_0001_0000_nii_gz

```
>>> # Load the NIfTI file
>>> img = nib.load('heart_0001_0000.nii.gz')
>>> data = img.get_fdata()
>>> # Print shape and data type
>>> data.shape
(256, 256, 120)  # Example dimensions - height, width, and number of slices
>>> data.dtype
dtype('unit16')
```

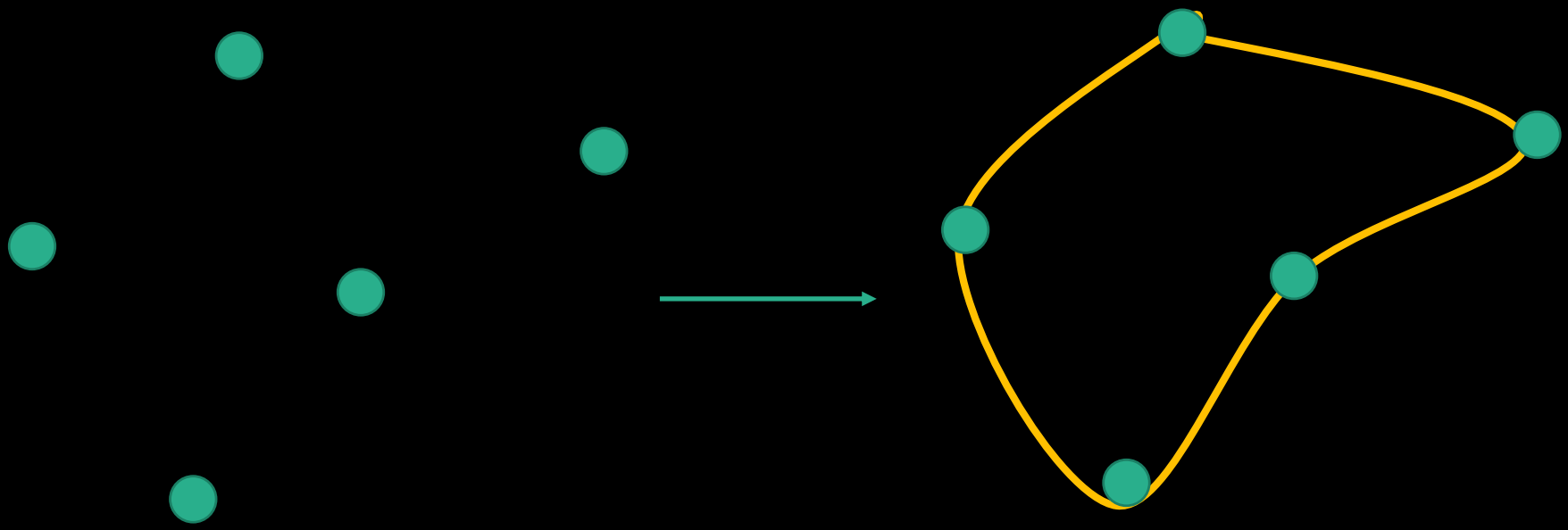

input image.npy (XYZ, float32)
Our Data
seg mask.npy (XYZ, unit8)
?????
imagesTr
input image (XYZ, uint16)
imagesTs
seg mask. (XYZ, unit8)


input image.npy (XYZ, float32)
Our Data
seg mask.npy (XYZ, unit8)
?????
imagesTr
input image (XYZ, uint16)
imagesTs
seg mask. (XYZ, unit8)

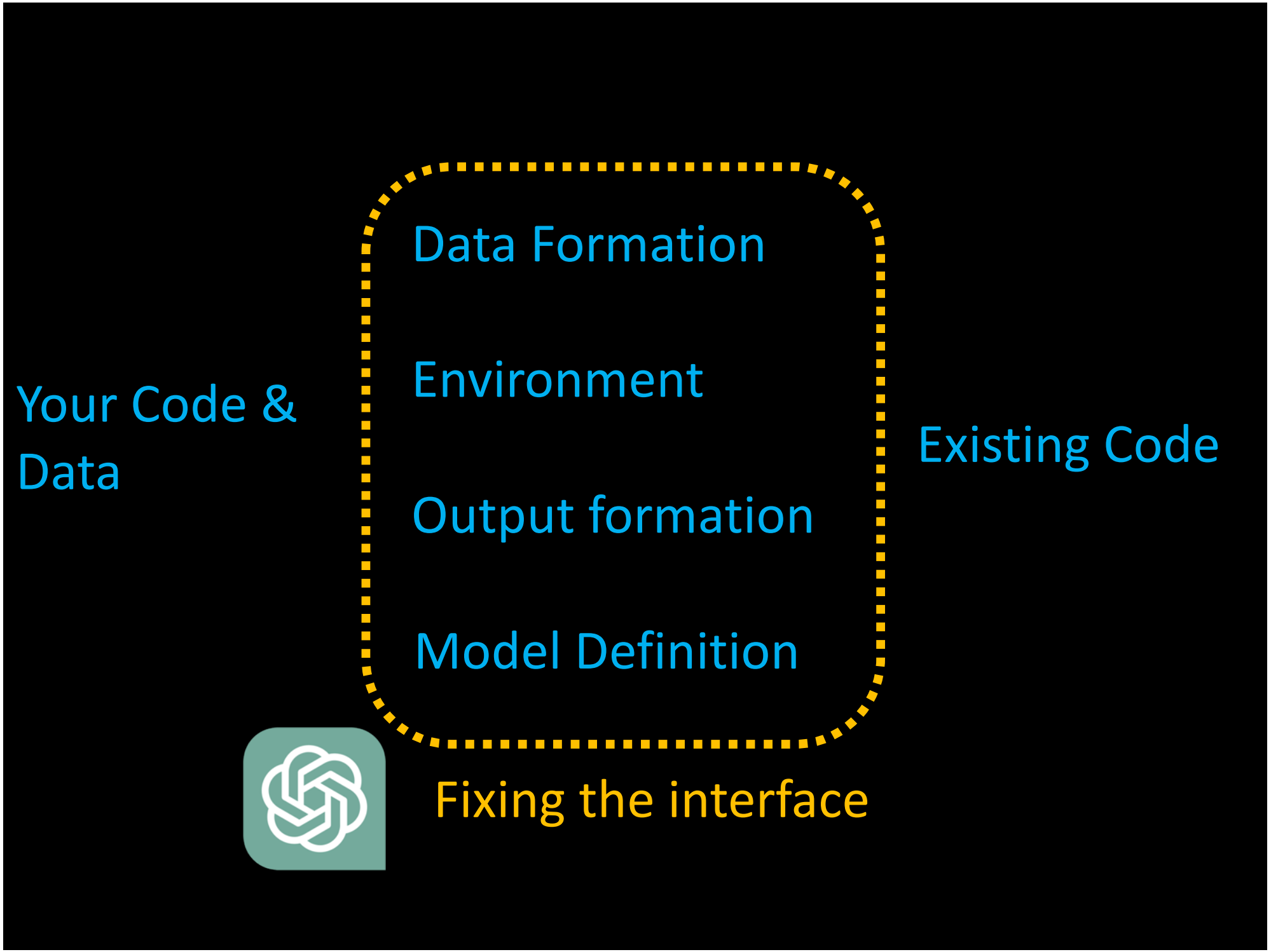
I have a 3D (MRI, mask) pair of (float32 and uint8), help me to convert them to .nii.gz of uint 16 and uint 8

```
>>> import numpy as np, nibabel as nib

>>> mri = np.load('mri.npy').astype(np.uint16)  # float32 -> uint16

>>> mask = np.load('mask.npy').astype(np.uint8)  # keep uint8

>>> nib.save(nib.Nifti1Image(mri, affine=np.eye(4)), 'mri_uint16.nii.gz')

>>> nib.save(nib.Nifti1Image(mask, affine=np.eye(4)), 'mask_uint8.nii.gz')

>>> print(f"Saved shapes: {mri.shape}, {mask.shape}")

Saved shapes: (256, 256, 120), (256, 256, 120)
```

Change annotation format:



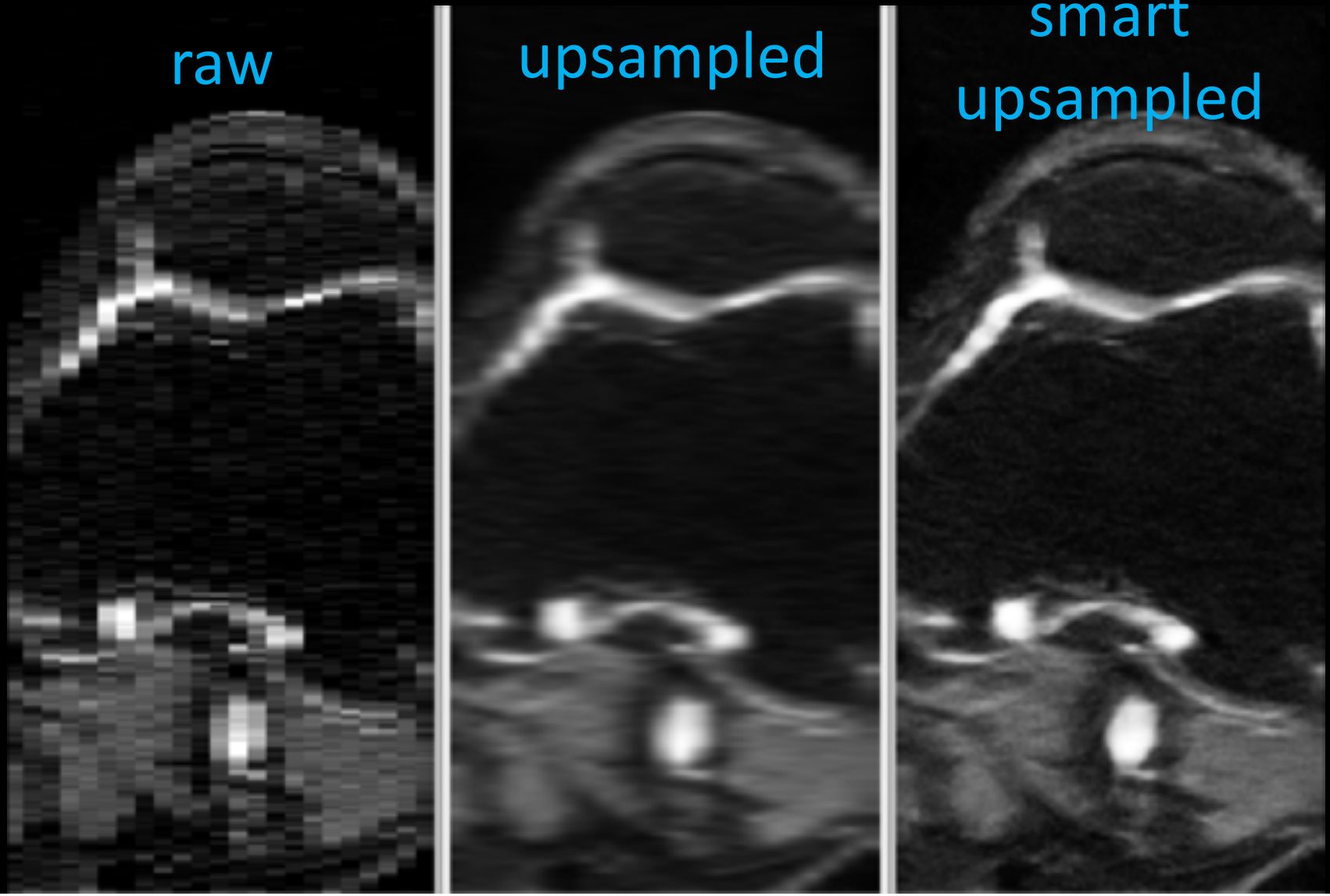
I have a key point files of (x,y,z),
Help me to convert them to closed area…


Your Code &
Data
Data Formation
Environment
Output formation
Model Definition
Existing Code
Fixing the interface

Think about what is routine

What required intelligence!

We don’t usually operate in a 3D high-resolution space (i.e. anisotropic resolution):

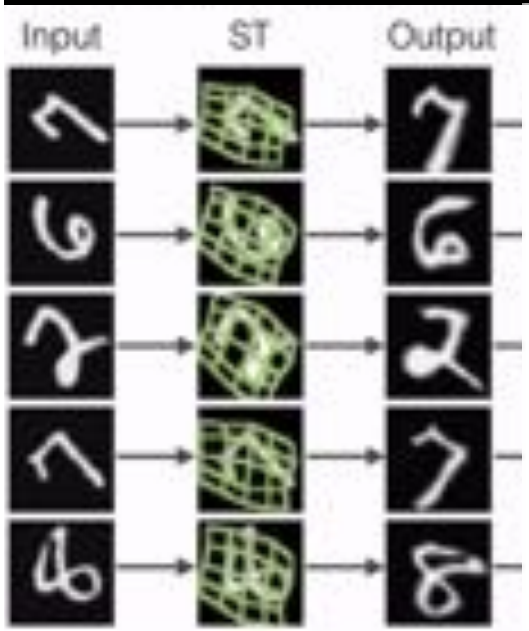


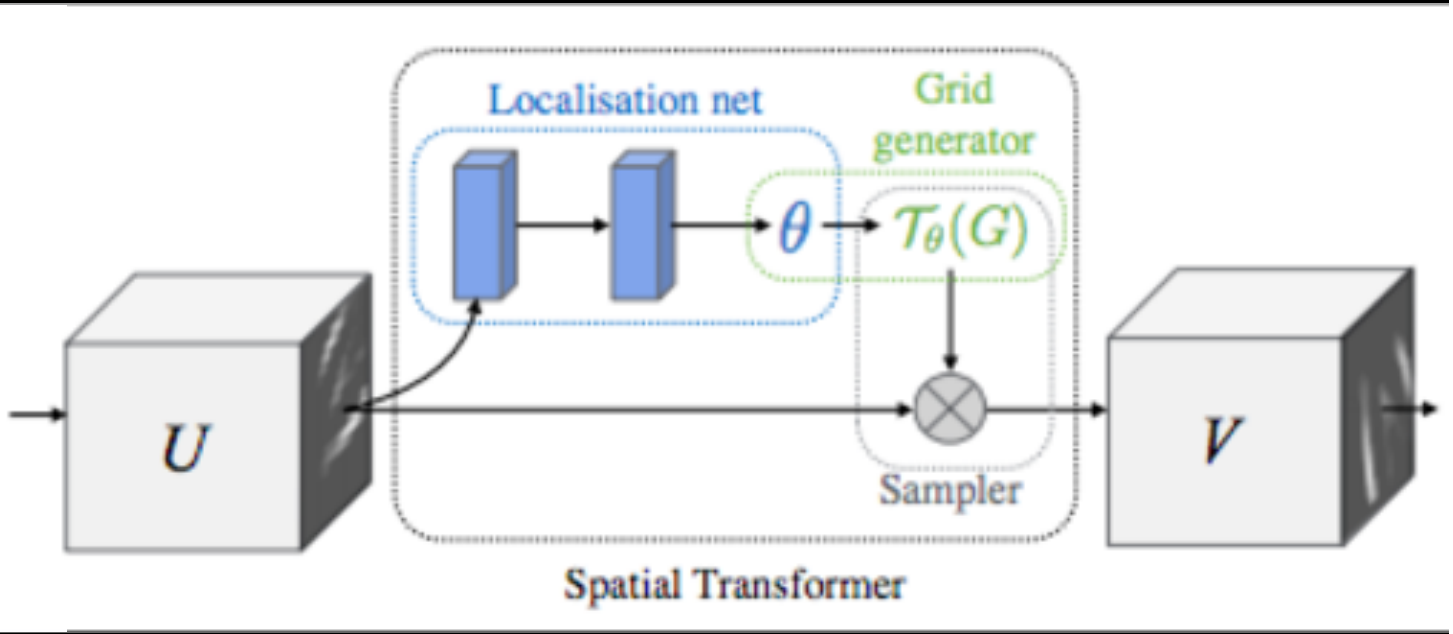
Front
Annotation

Side
Annotation

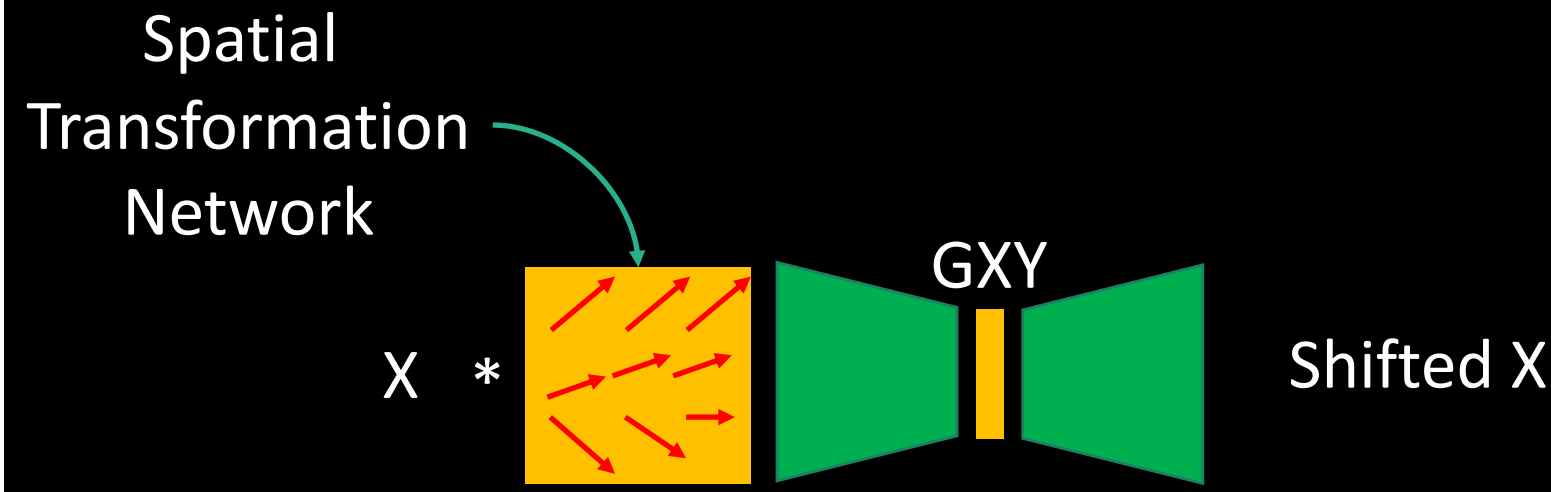
Resample?
Smoothing?
Multi-view
annotation?

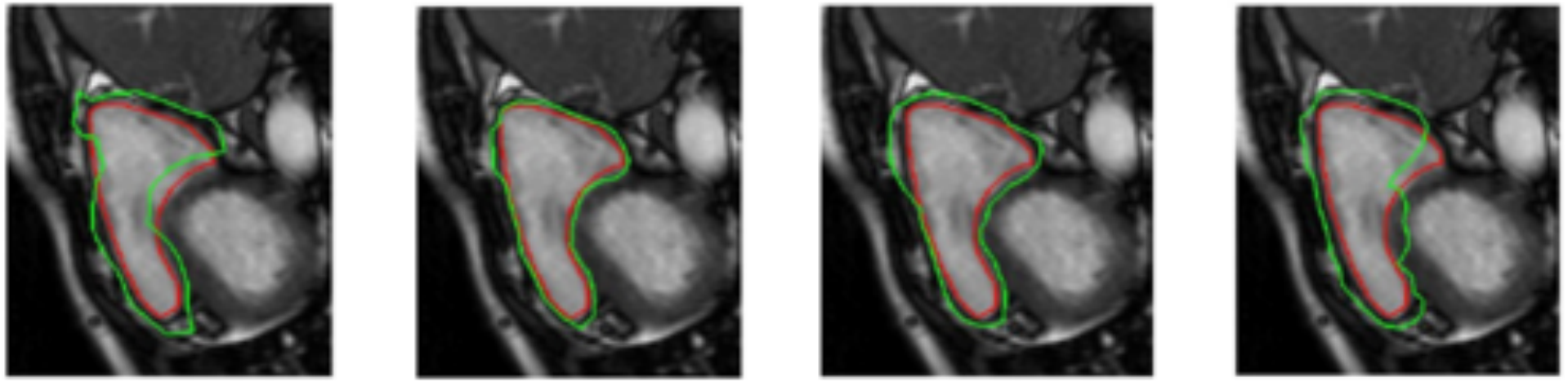
Imaging Registration by Spatial Transformation





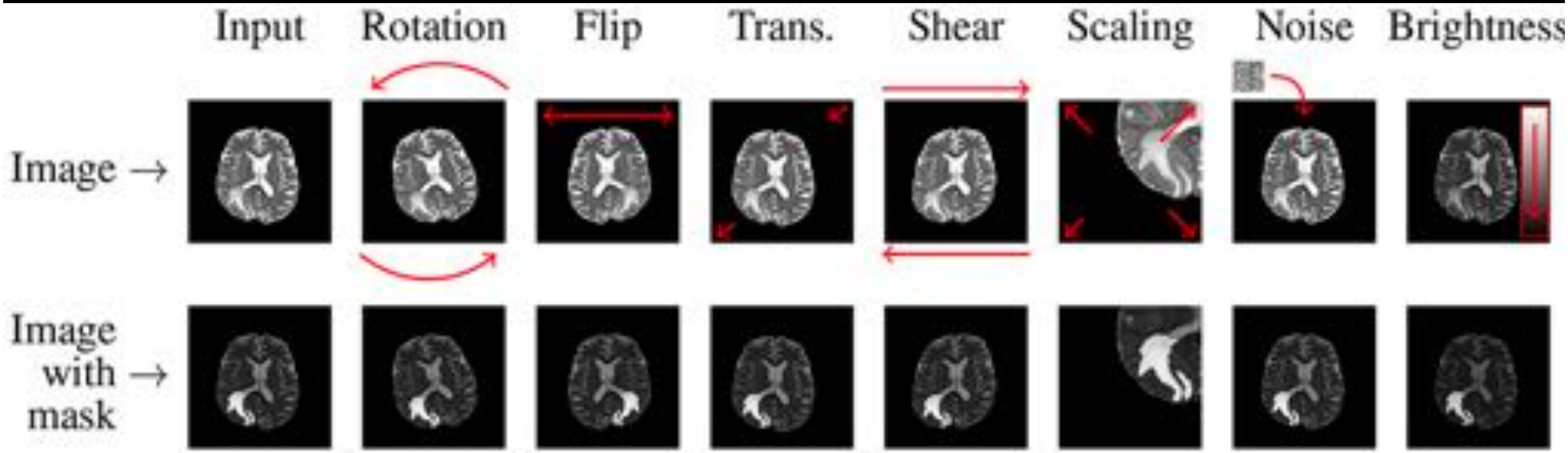
Imaging Registration by Spatial Transformation





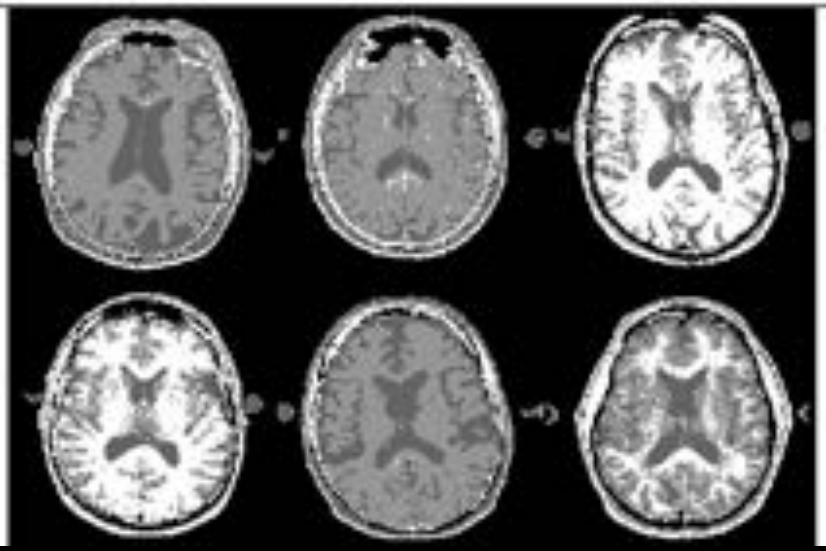
ELASTIC REGISTRATION OF MEDICAL IMAGES WITH GANS

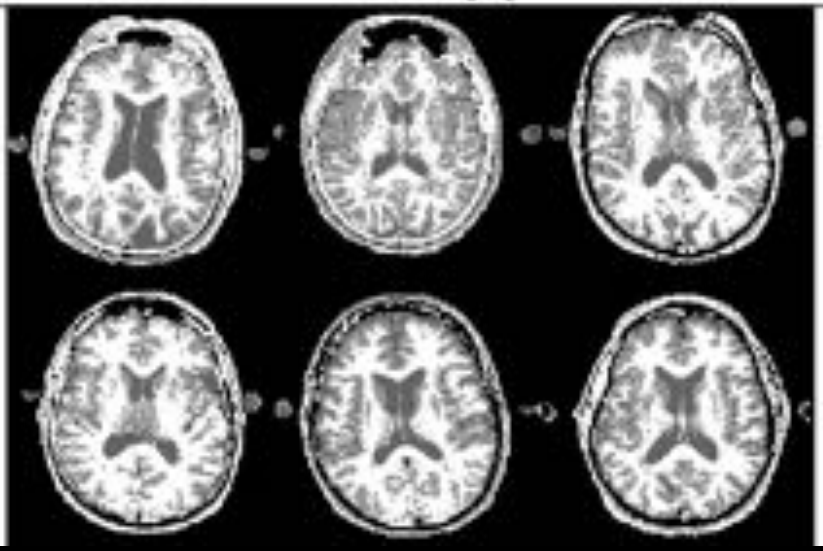
Data Augmentation:



Intensity Normalization / augmentation

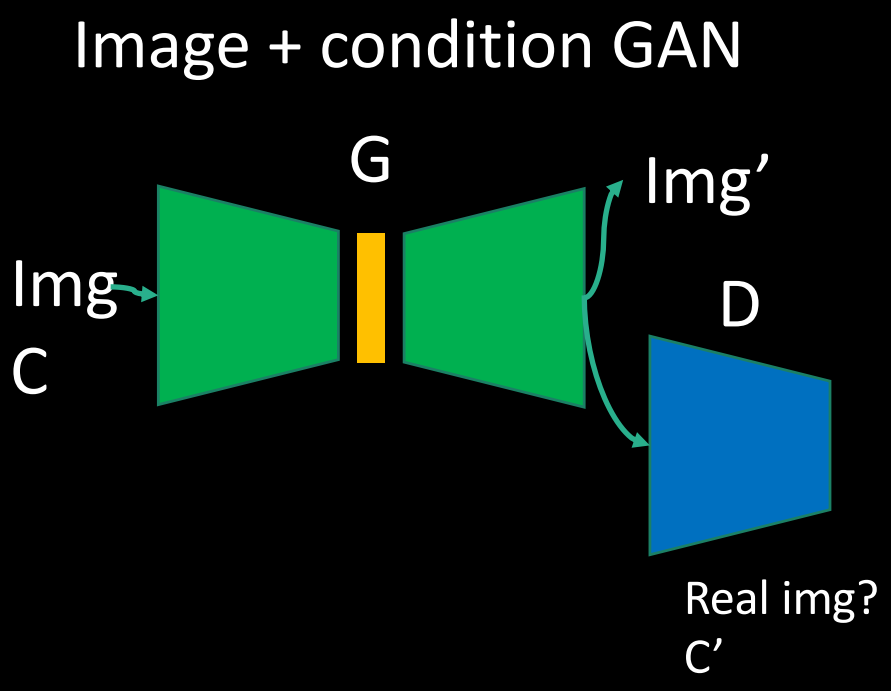
paper read at IEEE ISBI 2004

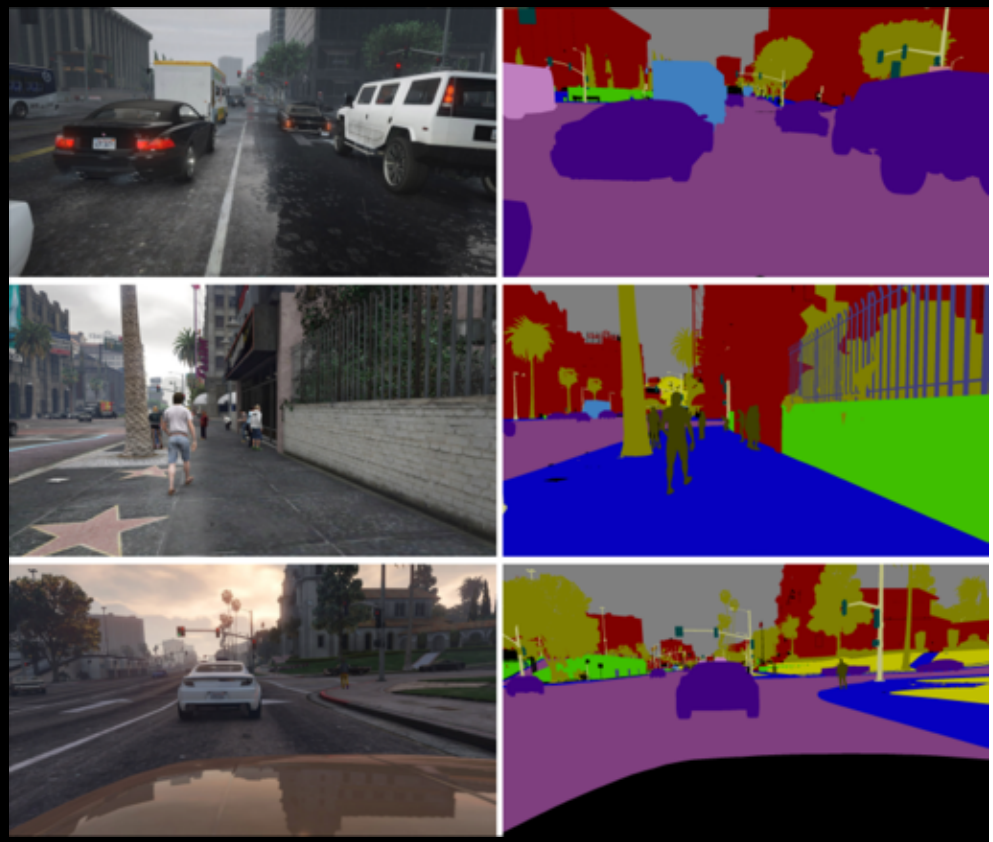




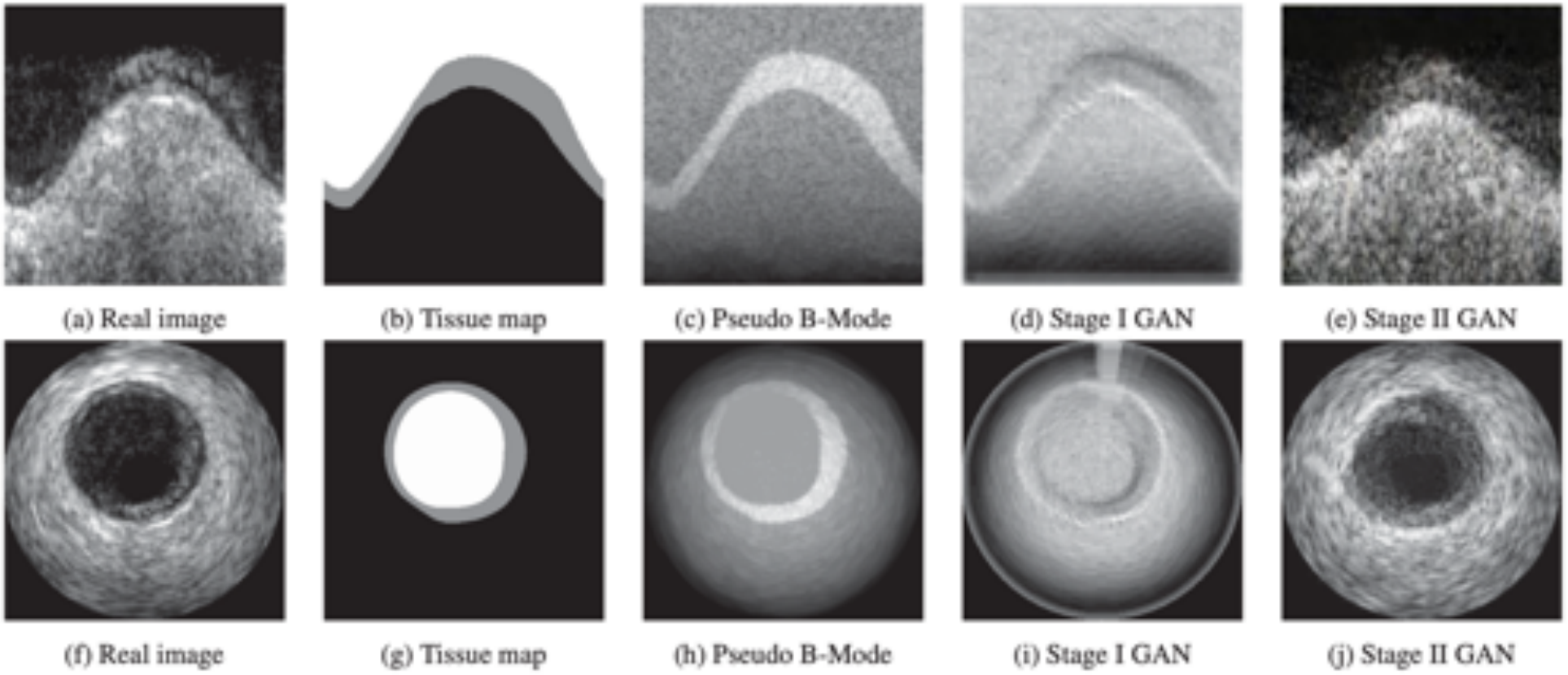
Patch size

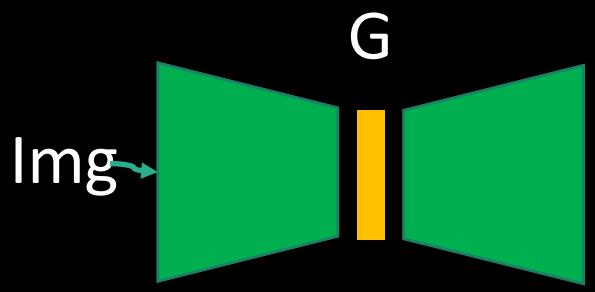
“Smart” Data Augmentation:





Mask Conditioned Image Generation



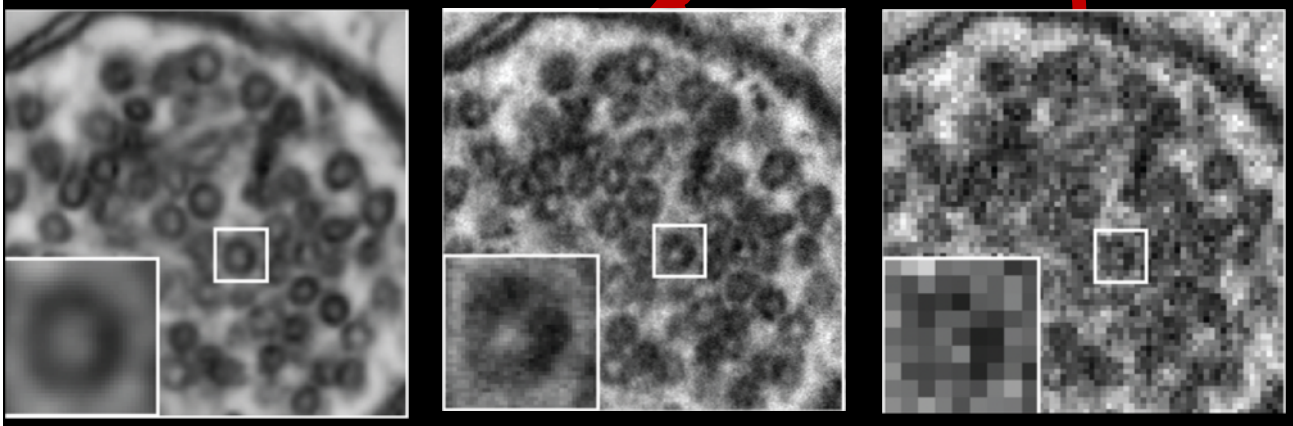


You got free images with free mask!

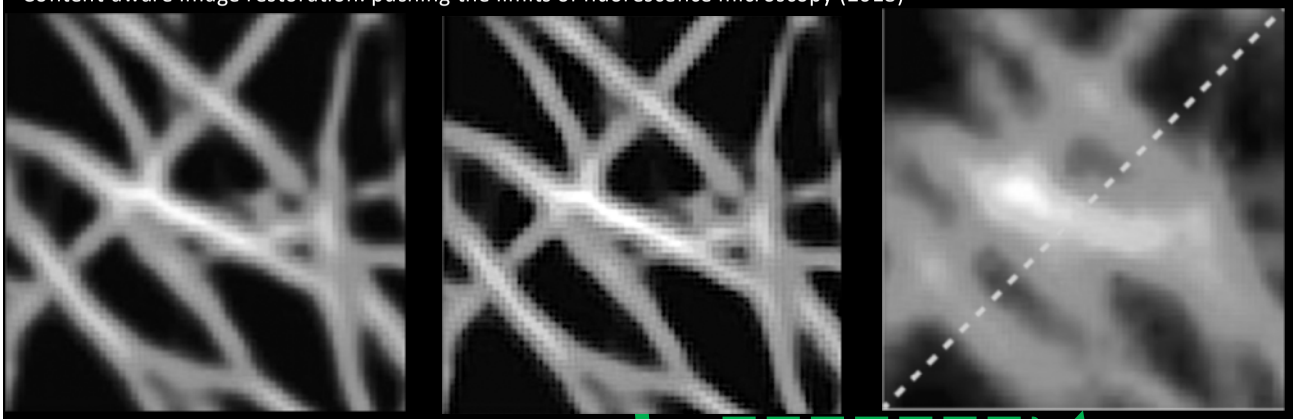
Mask Conditioned Image Generation (2)

Learn

Deep Learning-Based Point-Scanning Super-Resolution Imaging (2019)



Content-aware image restoration: pushing the limits of fluorescence microscopy (2018)

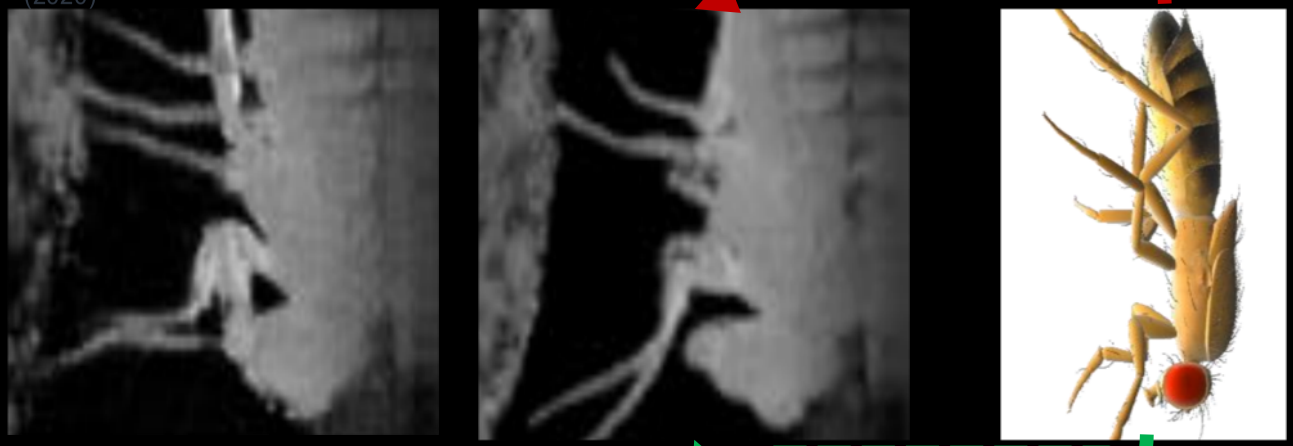


Syntheti

c

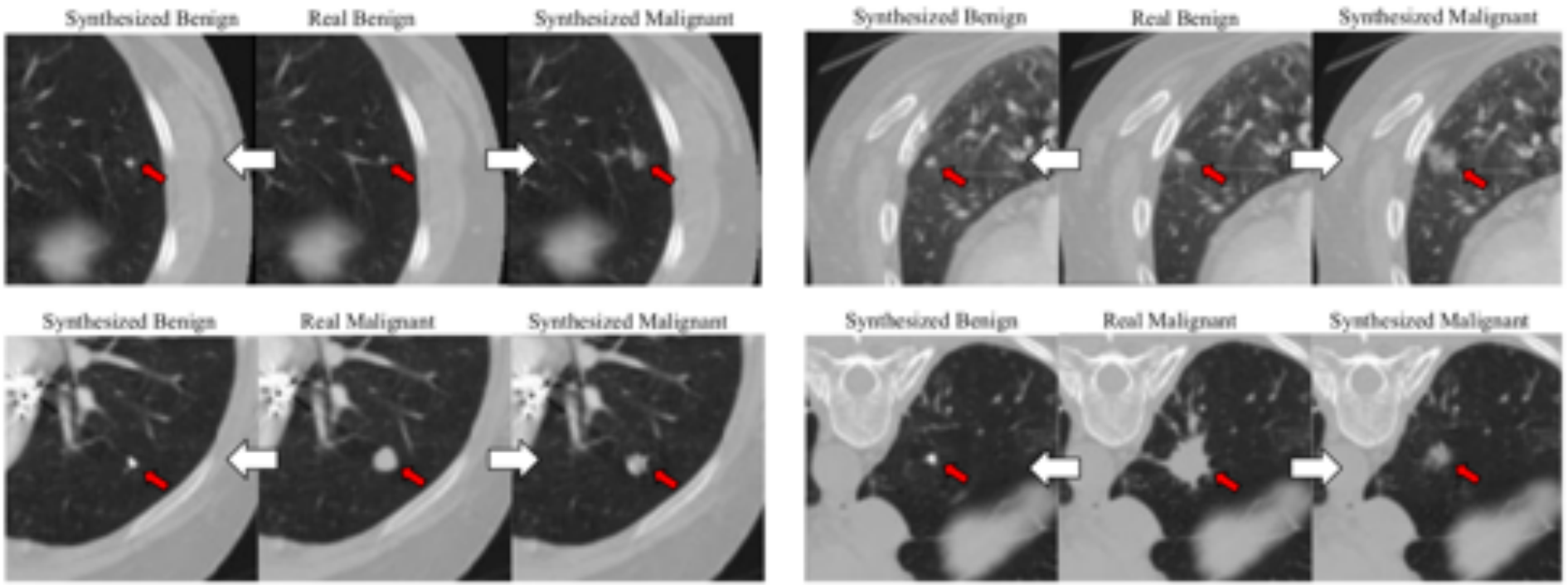
Learn

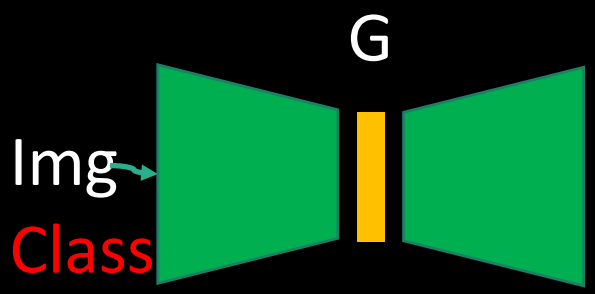
Deformation-Aware Unpaired Image Translation for Pose Estimation on Laboratory Animals (2020)



Synthetic

Class + Mask Conditioned Image Generation





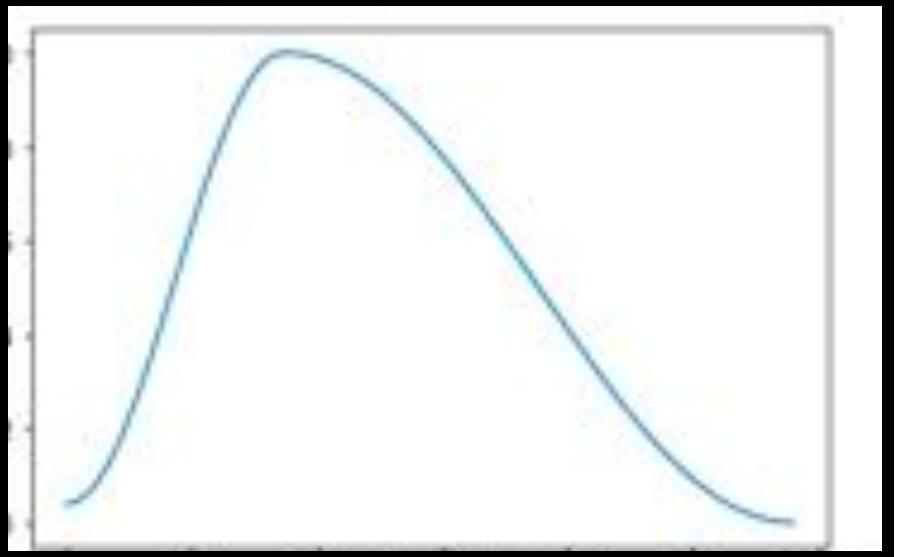
CLASS-AWARE ADVERSARIAL LUNG NODULE SYNTHESIS IN CT IMAGES (MICCAI 2020)

Finally, Hyper-parameters Tuning

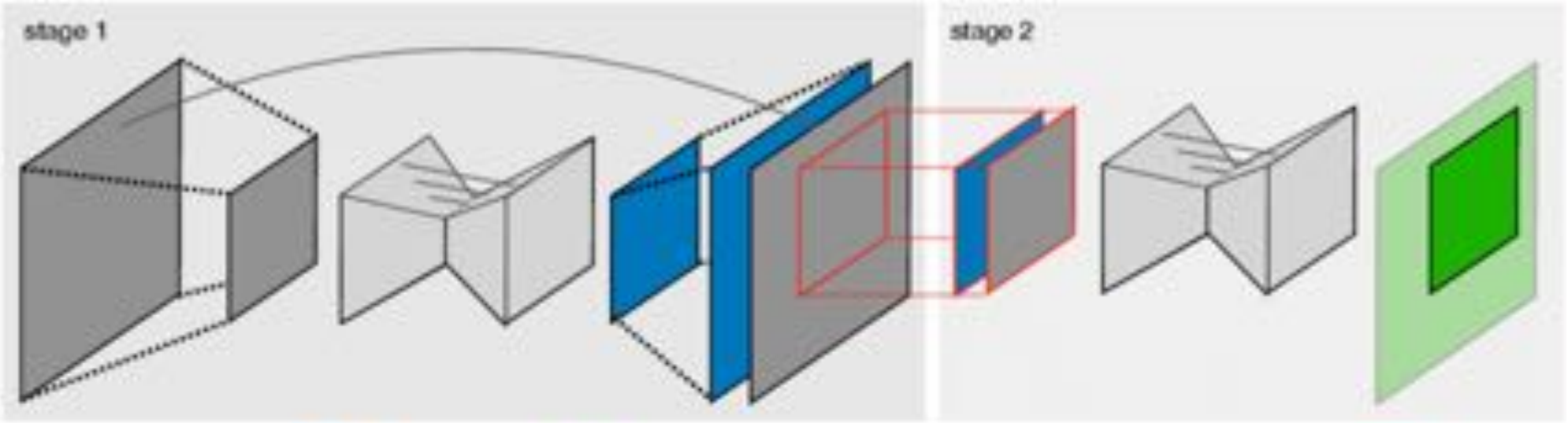
Optimizer: SGD with momentum

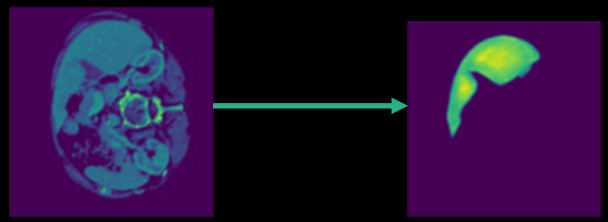
Loss: Dice & Cross Entropy

Learning Rate: Polynomial Learning Rate

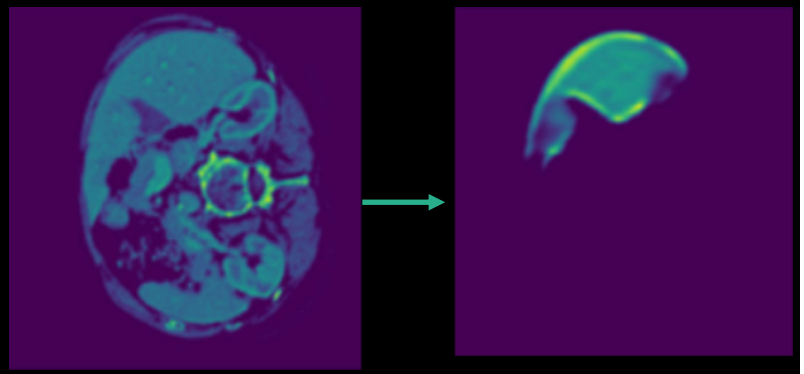


“Cascade” Segmentation Model



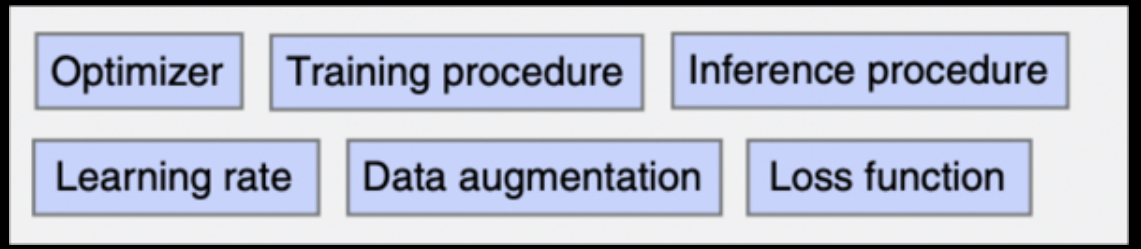


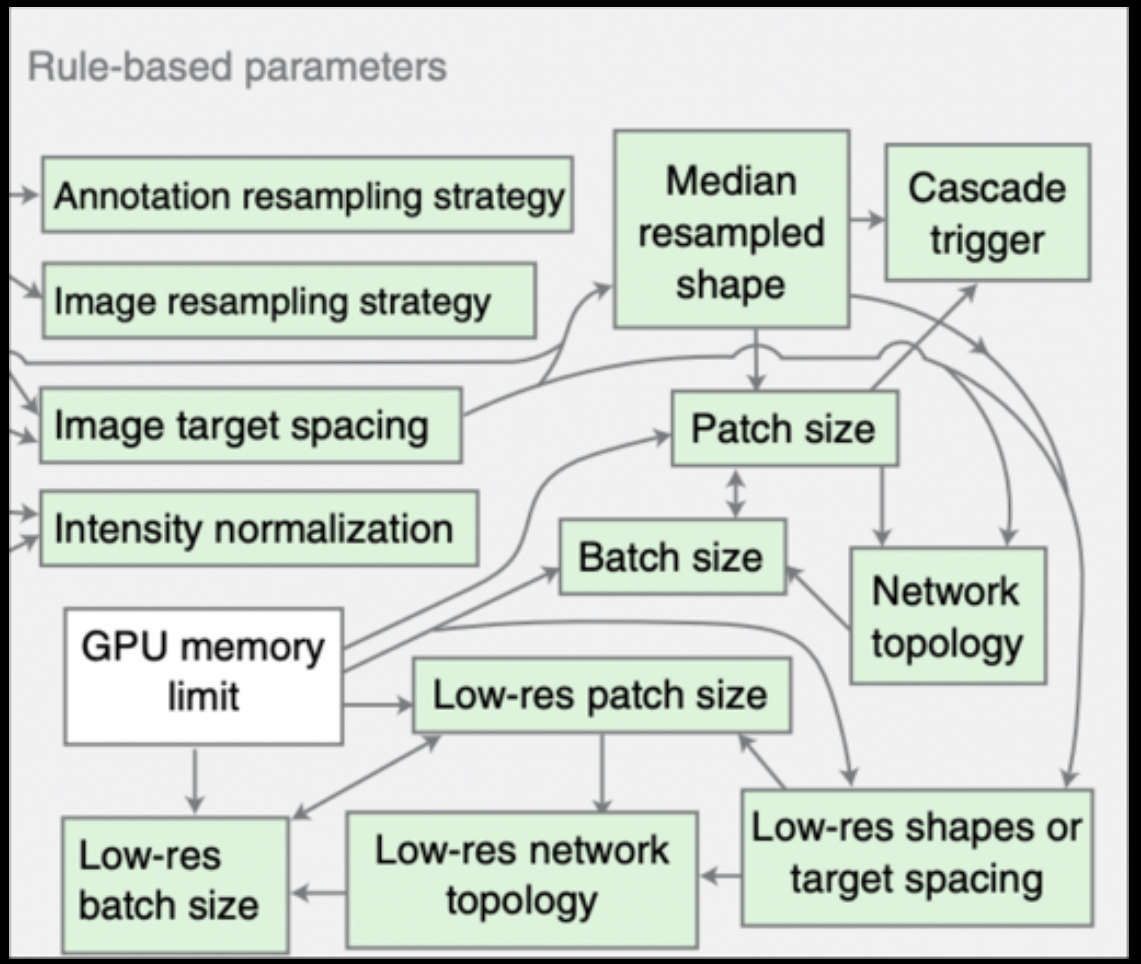
One stage for overall shape



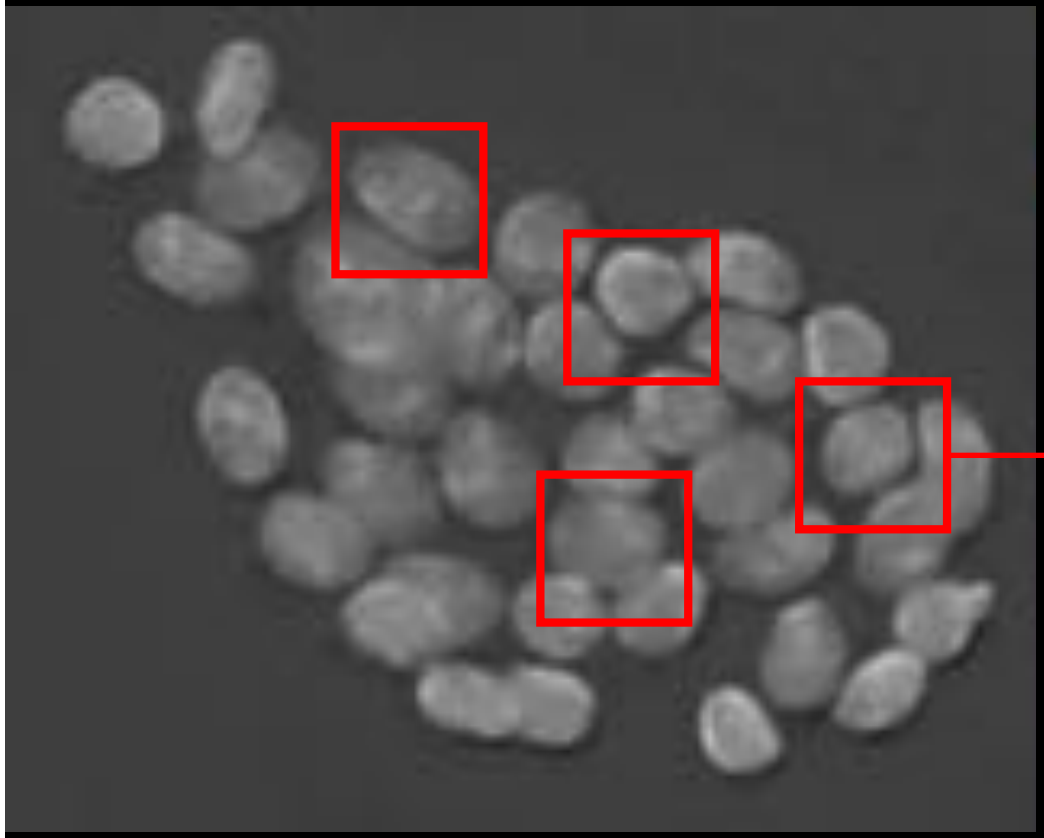
One stage for details

nn (not new) unet





Detection
/ Instance Segmentation

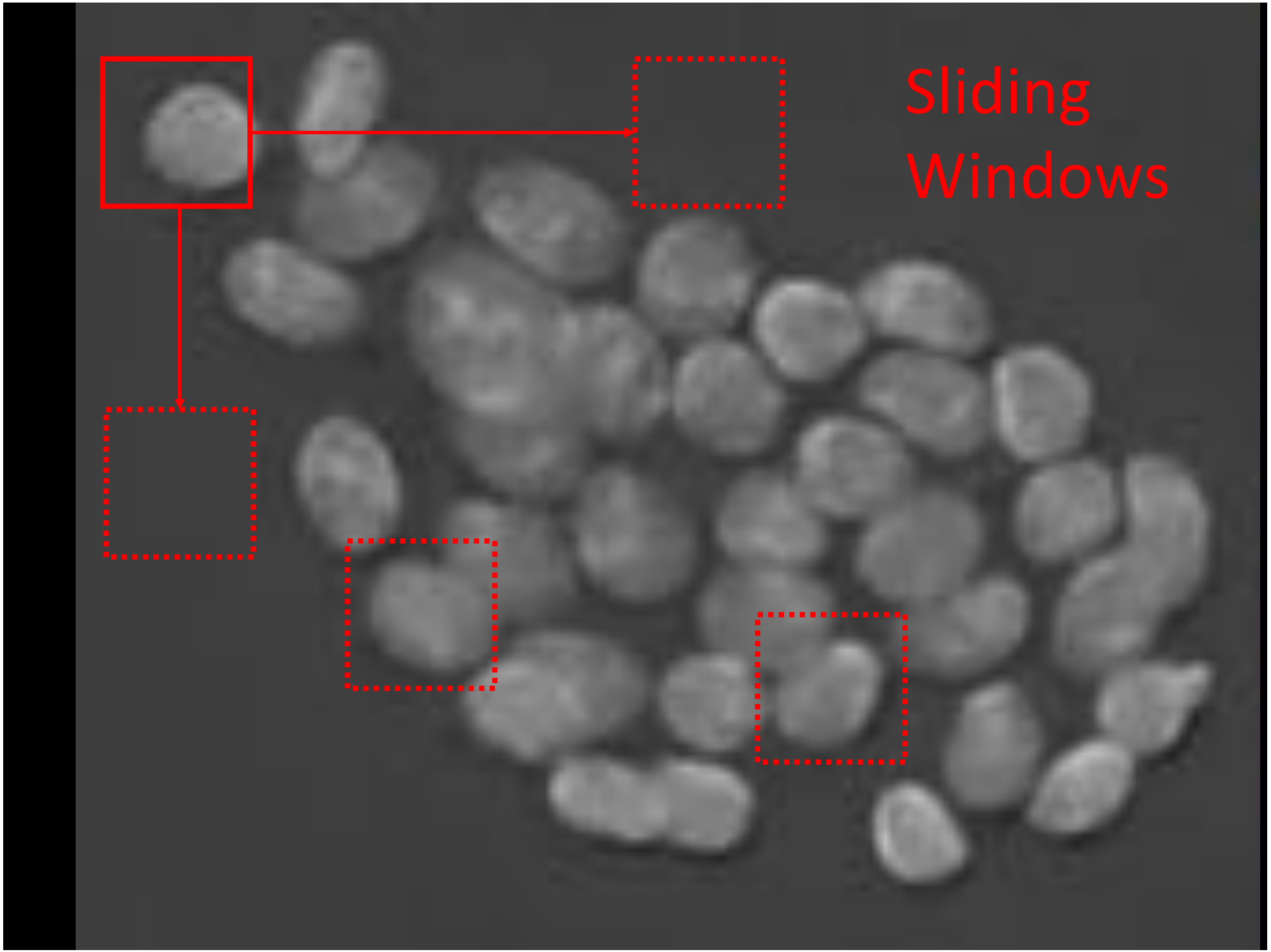


Class: cell

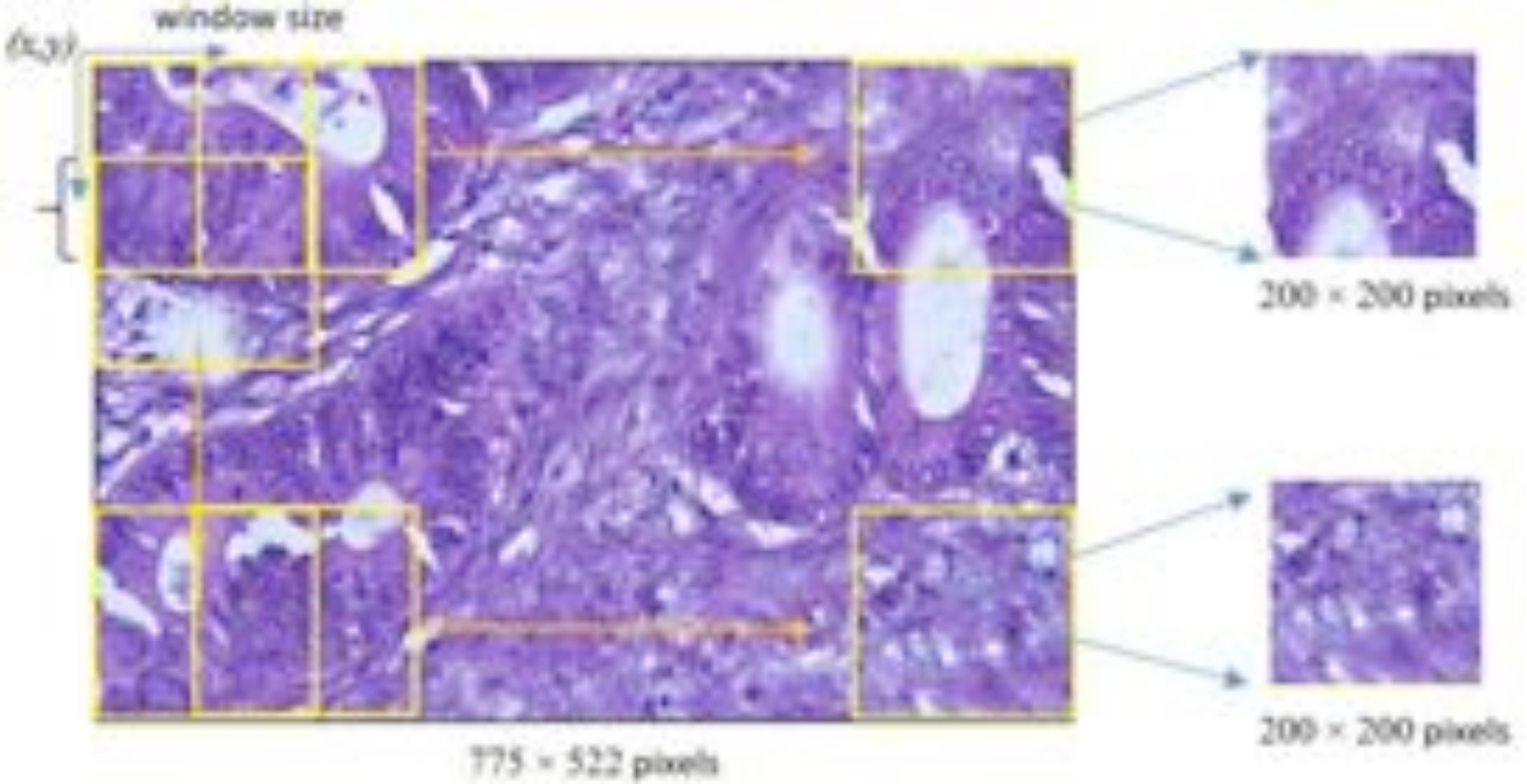
X = 60 pts
Y = 40 pts
W = 5.2 pts
H = 4.3 Pts

Multi Class
Grade A 30%
Grade B 30%
Grade C 40%

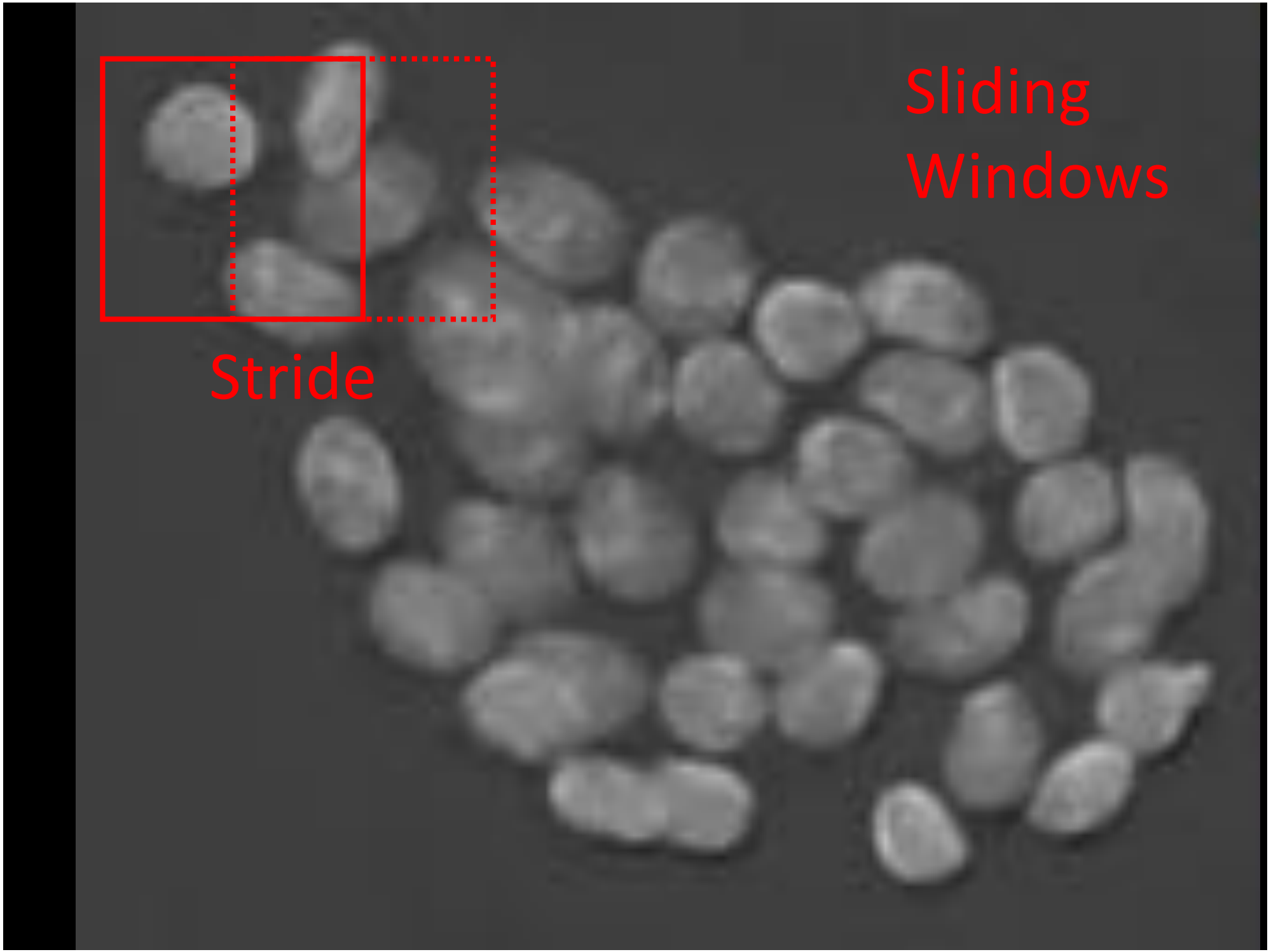
X = 60 pts
Y = 40 pts
W = 5.2 pts
H = 4.3 Pts


Sliding
Windows

Sliding Windows

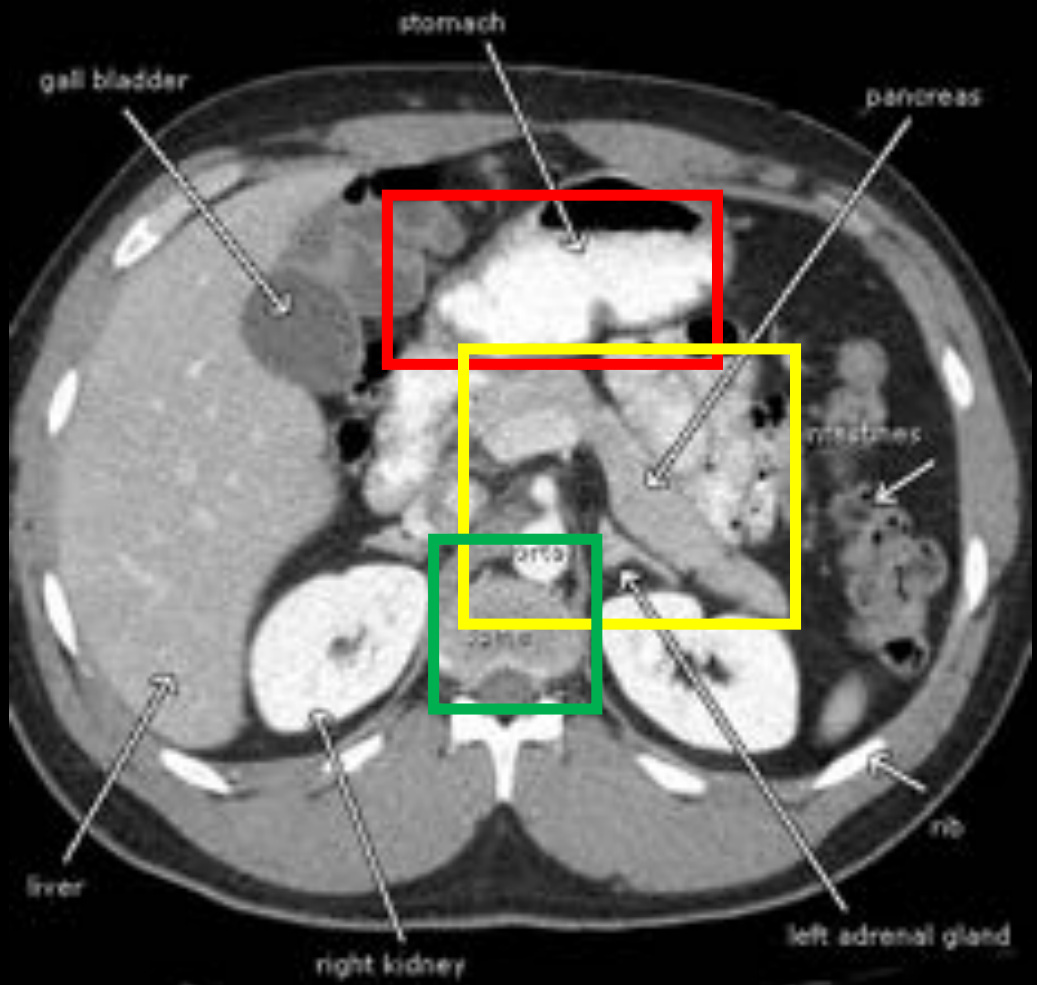


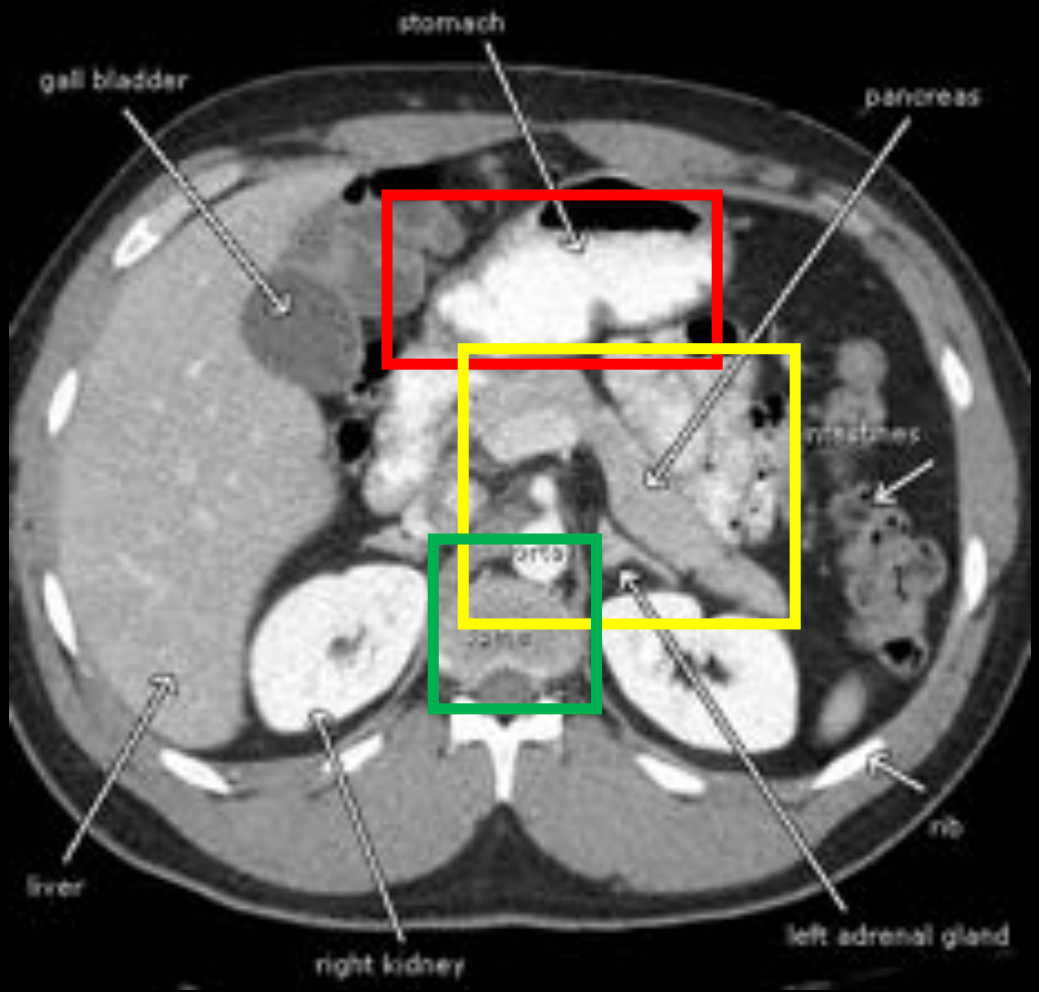
Conditional sliding windows: An approach for handling data limitation in colorectal histopathology image classification


Sliding
Windows
Stride

Sliding Windows Shortcomings:

- slow
- no global context





Different size

Different aspect ratio

May be overlapping

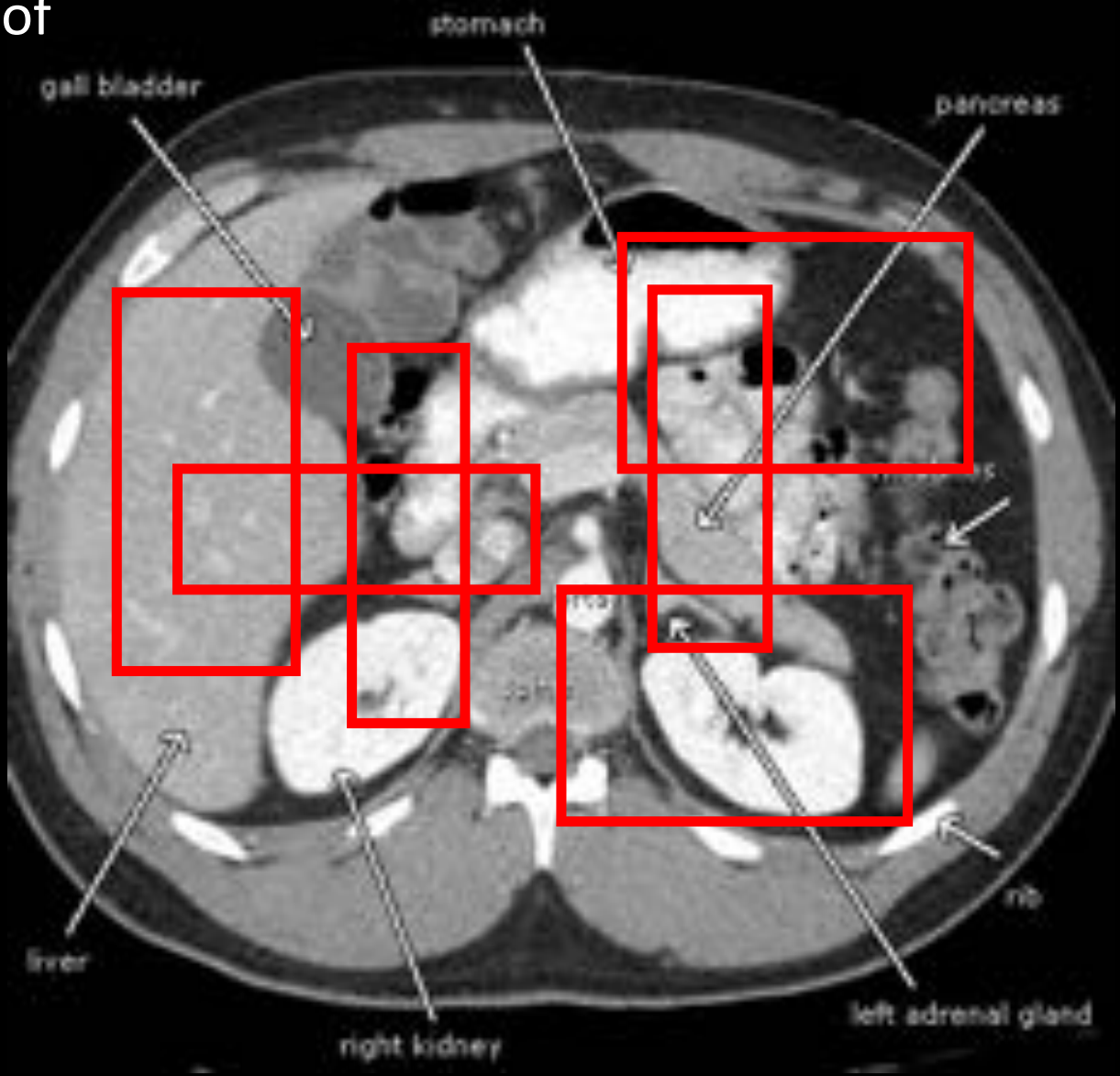
(Sometimes) need global context

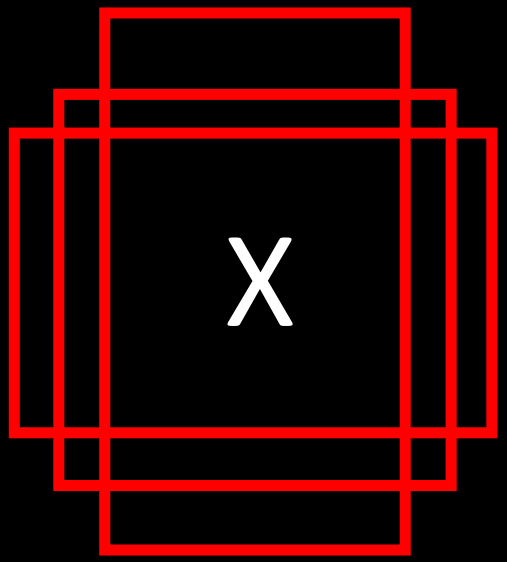
We can propose a bunch of Regions with different

Location,

Size,

And aspect ratio

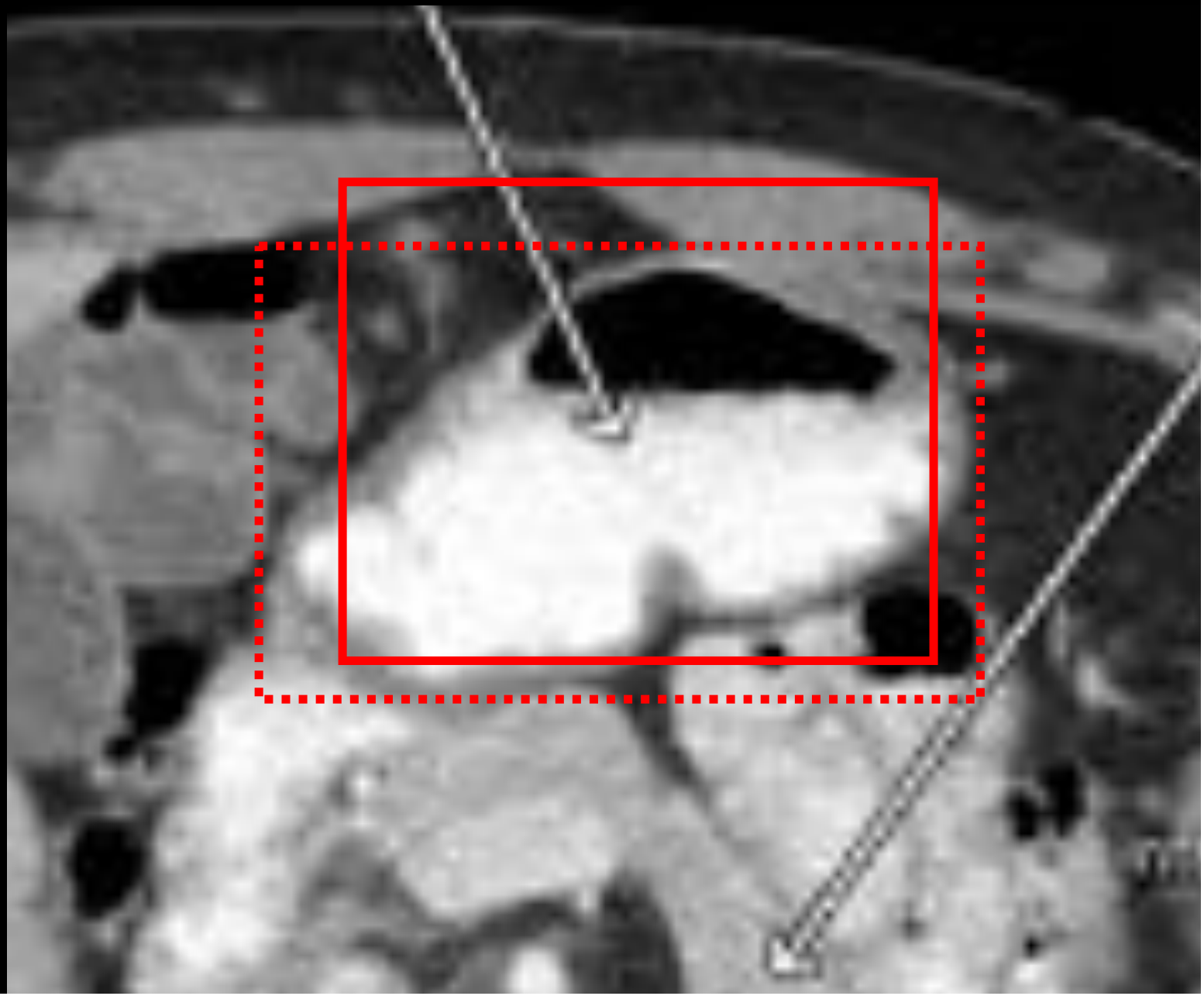




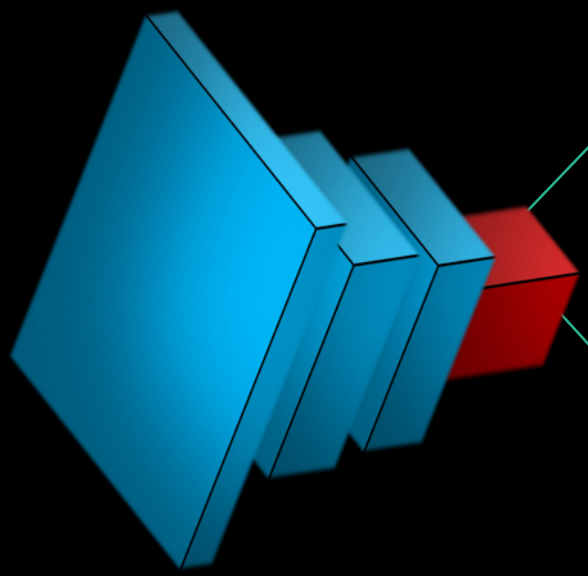
All the bounding boxes around the same regions should share some similiar information

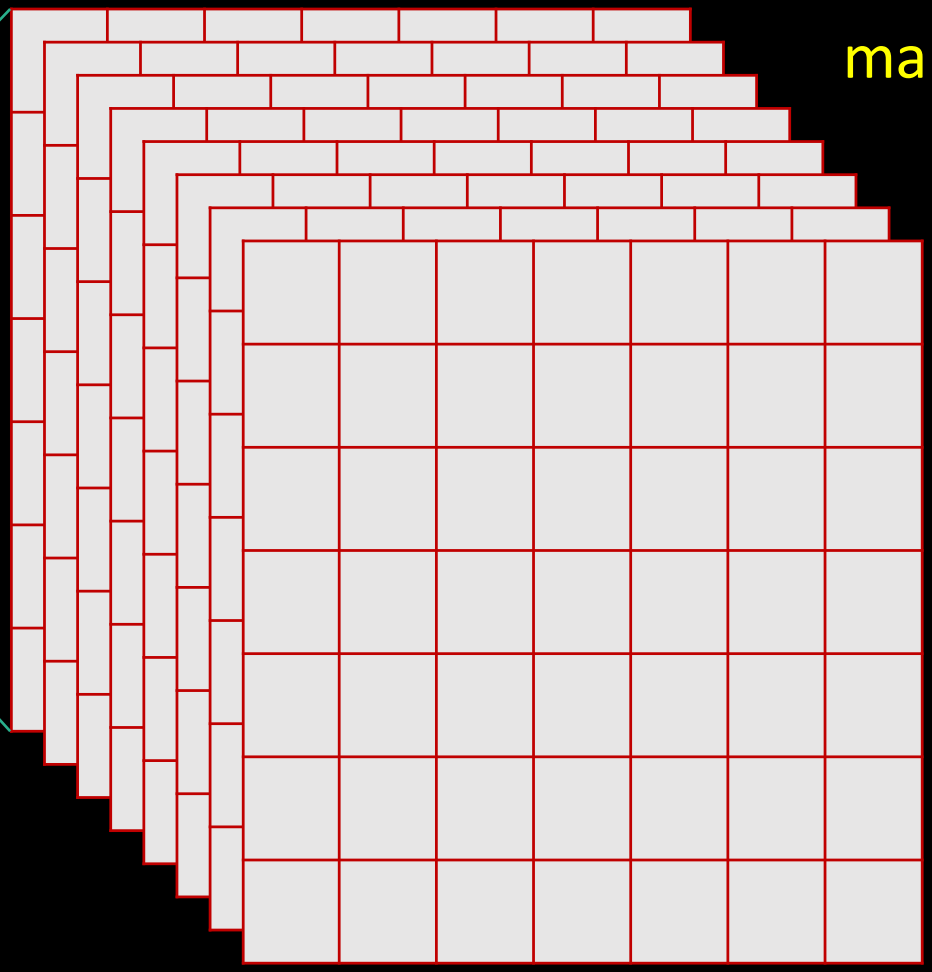
We call the center an “Anchor”

We can propose “rough” bounding boxes then refine it



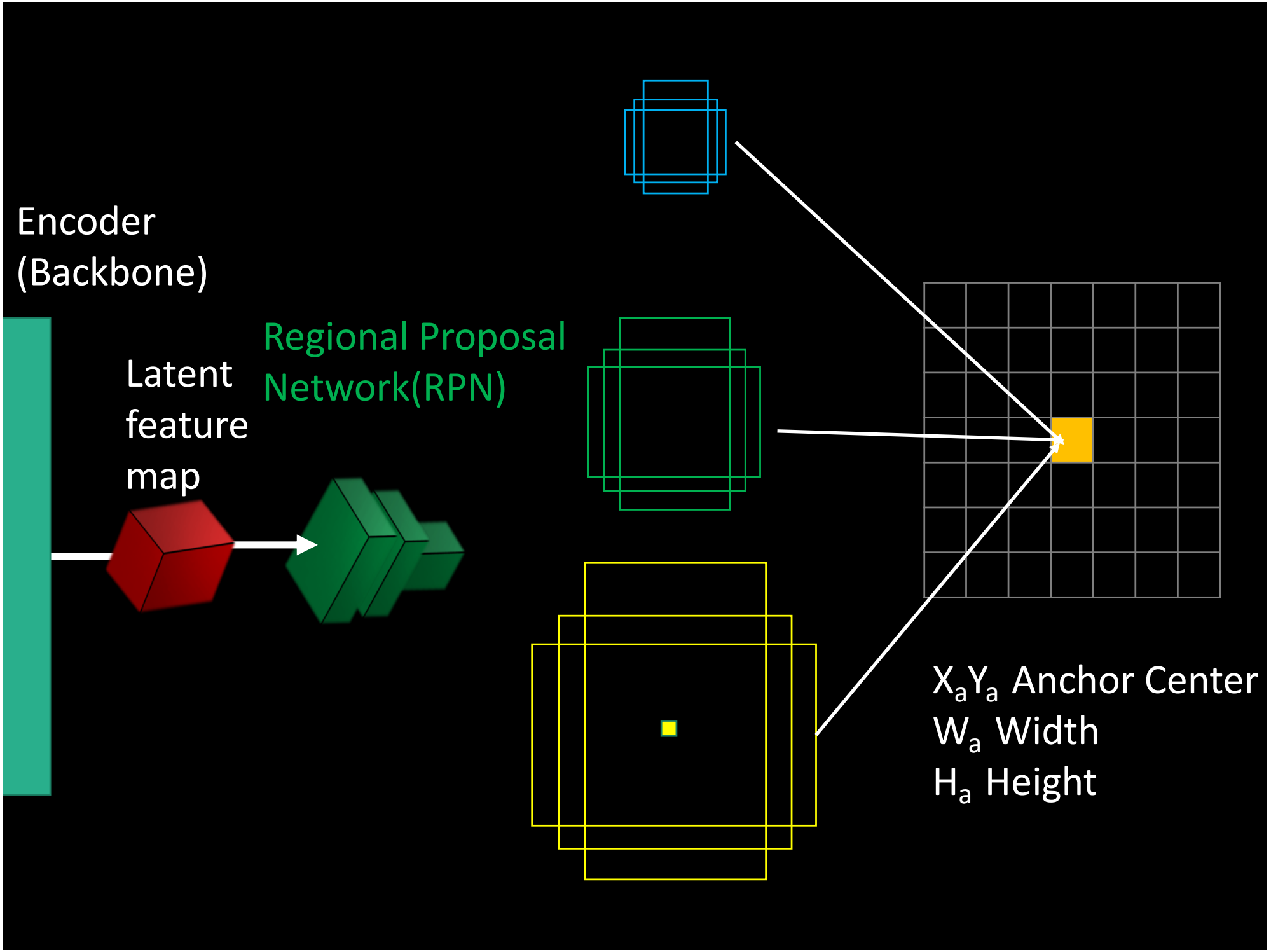
Latent Space often contained a rough spatial information (for example, in a 7x7 grid)

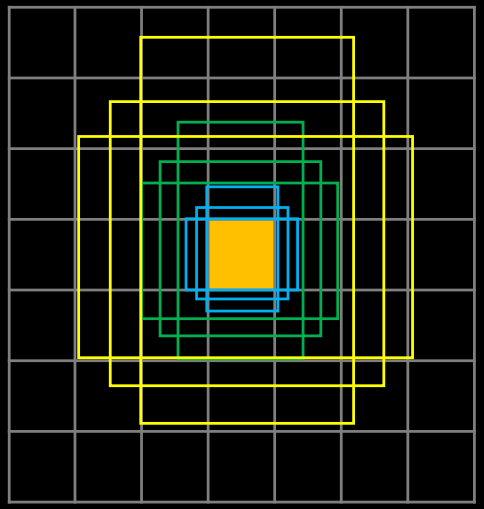




Latent feature map

From a backbone model


Encoder
(Backbone)
Latent
feature
map
Regional Proposal
Network(RPN)
X_aY_a Anchor Center
W_a Width
H_a Height



For every anchor point:

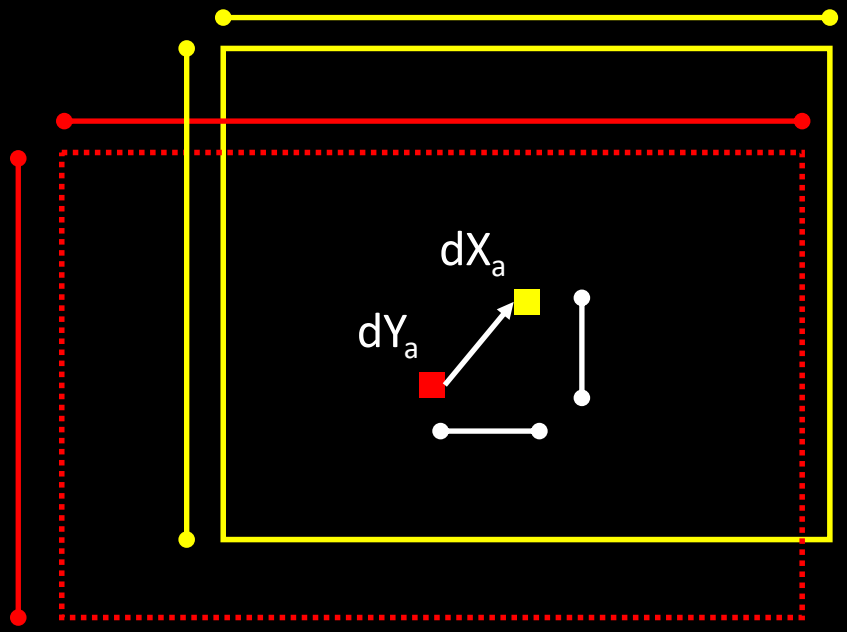
<u>Q1: Is it an object?:</u>

9 (boxes) * c (classes) output probability

Bounding box regression

<u>Q2: How far should the box be adjusted?</u>

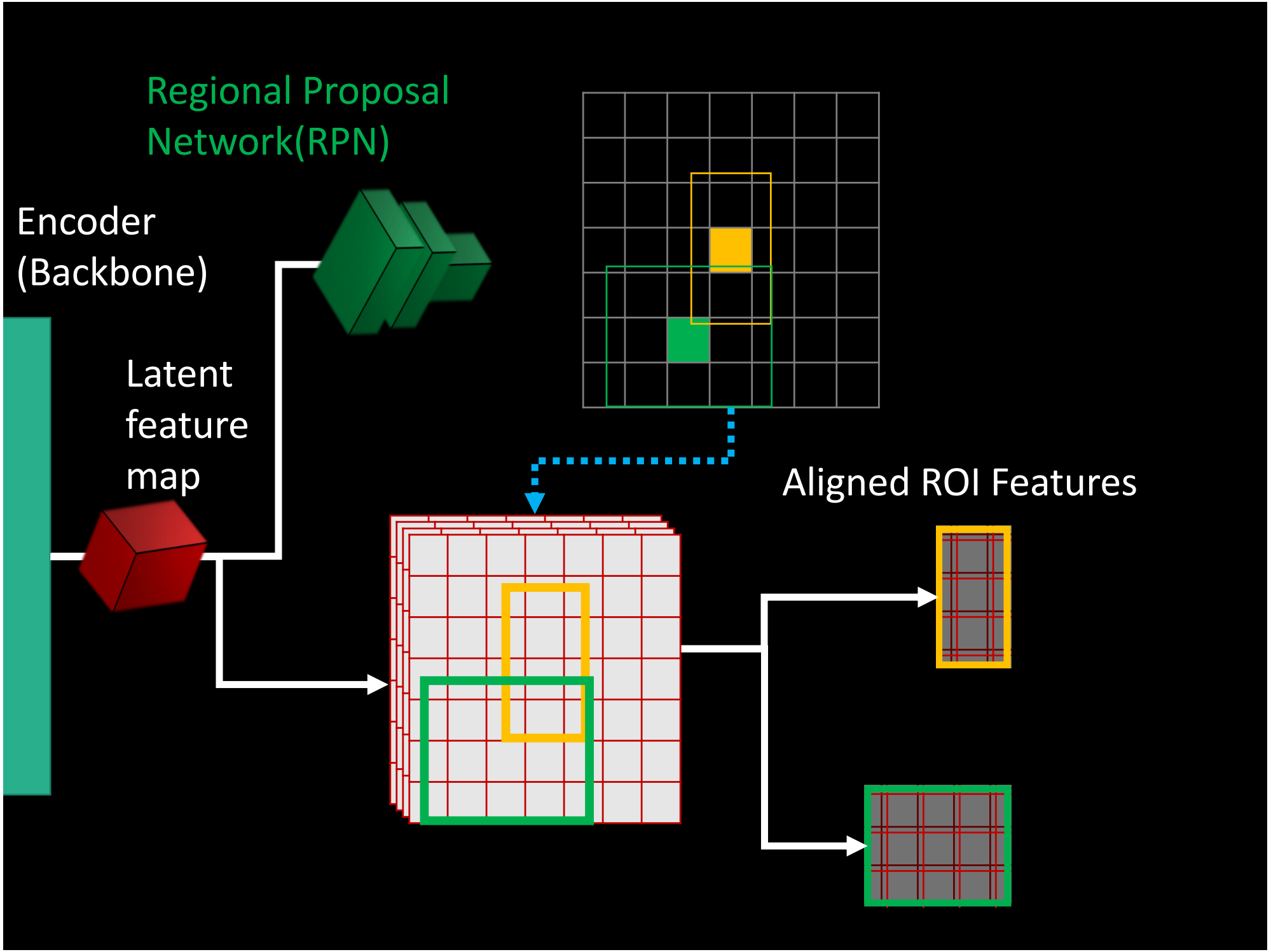
9 * 4 (x, y, w, h) regression values

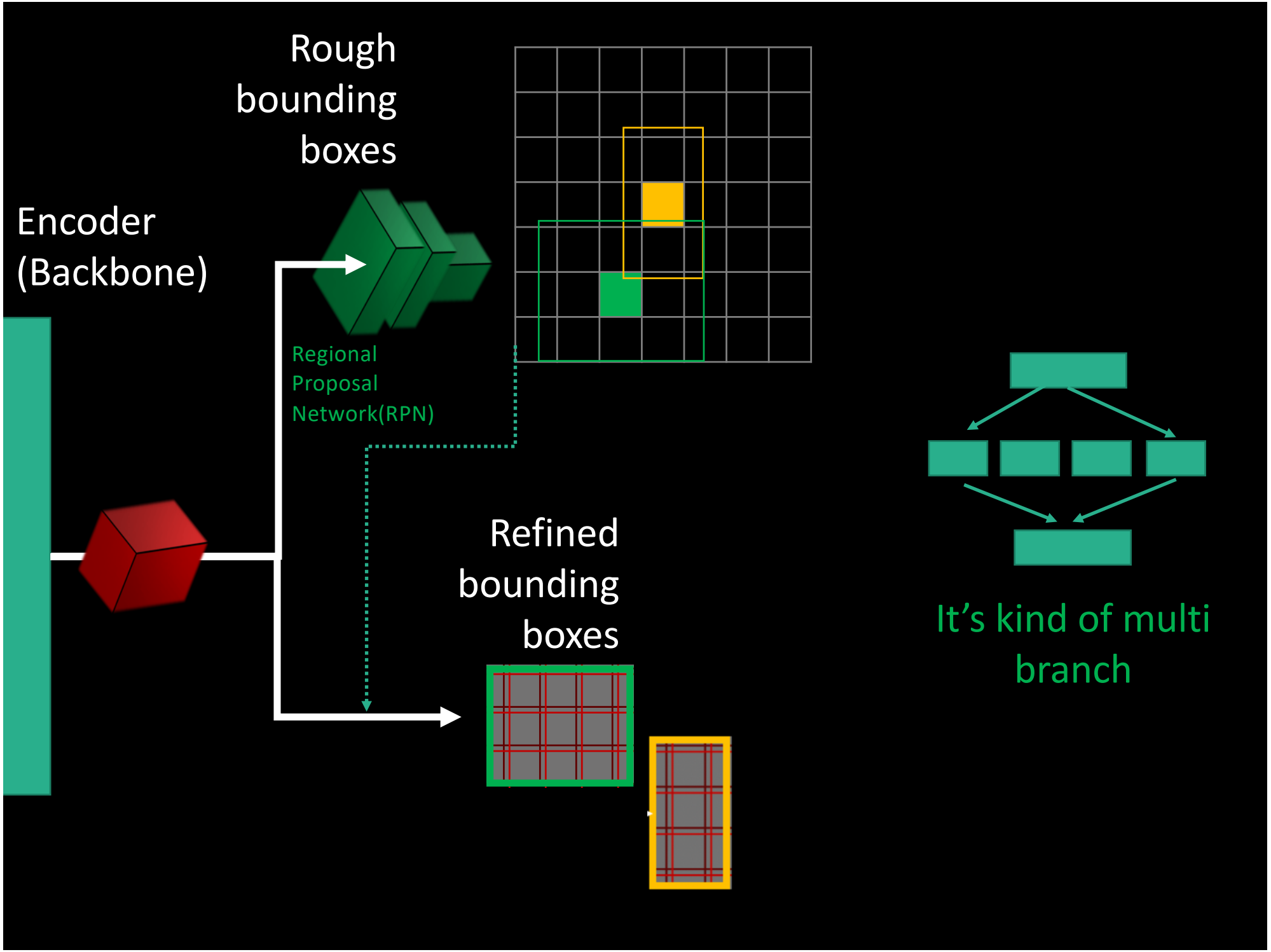


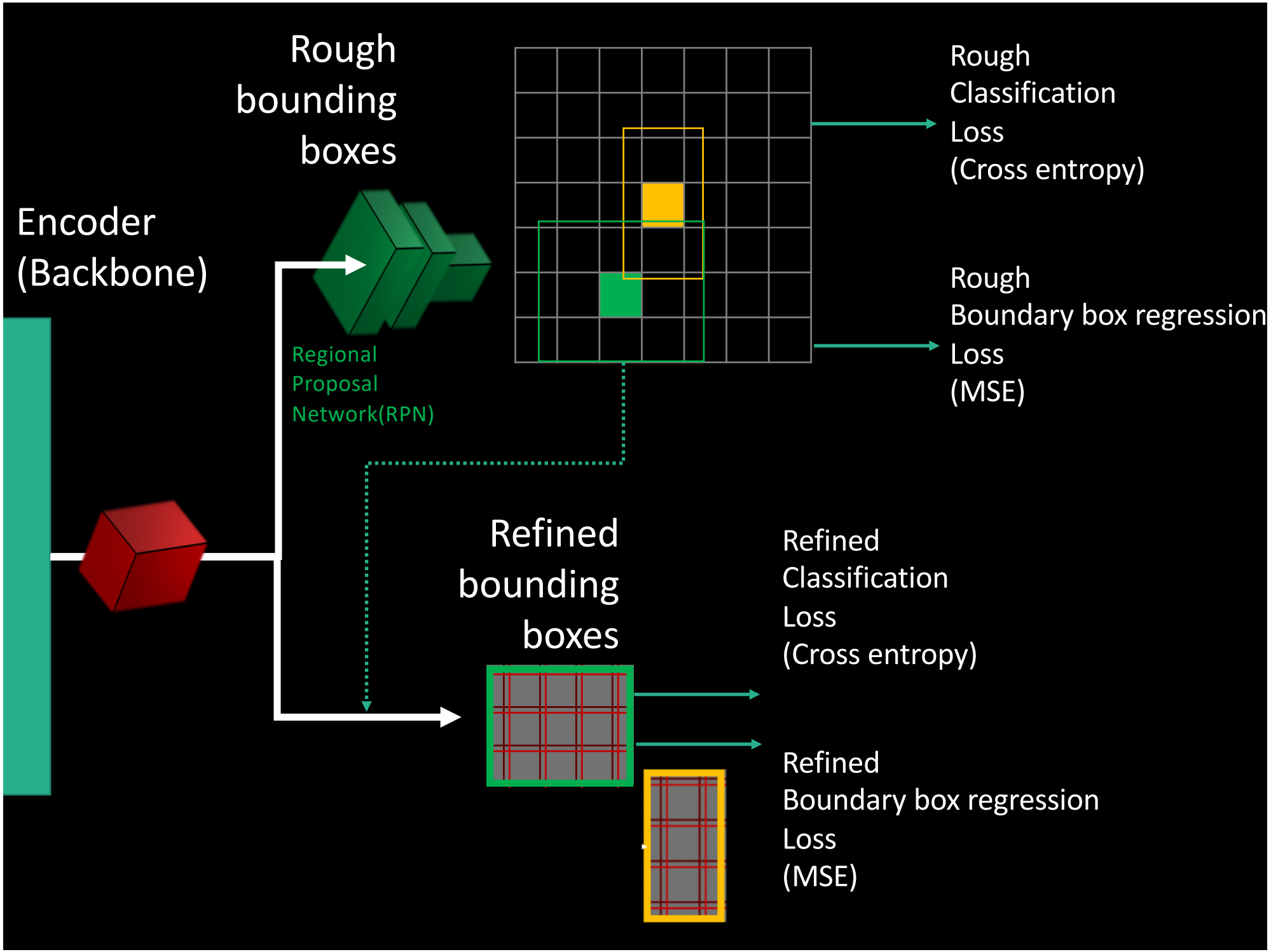
$dX_a dY_a$ Anchor Center

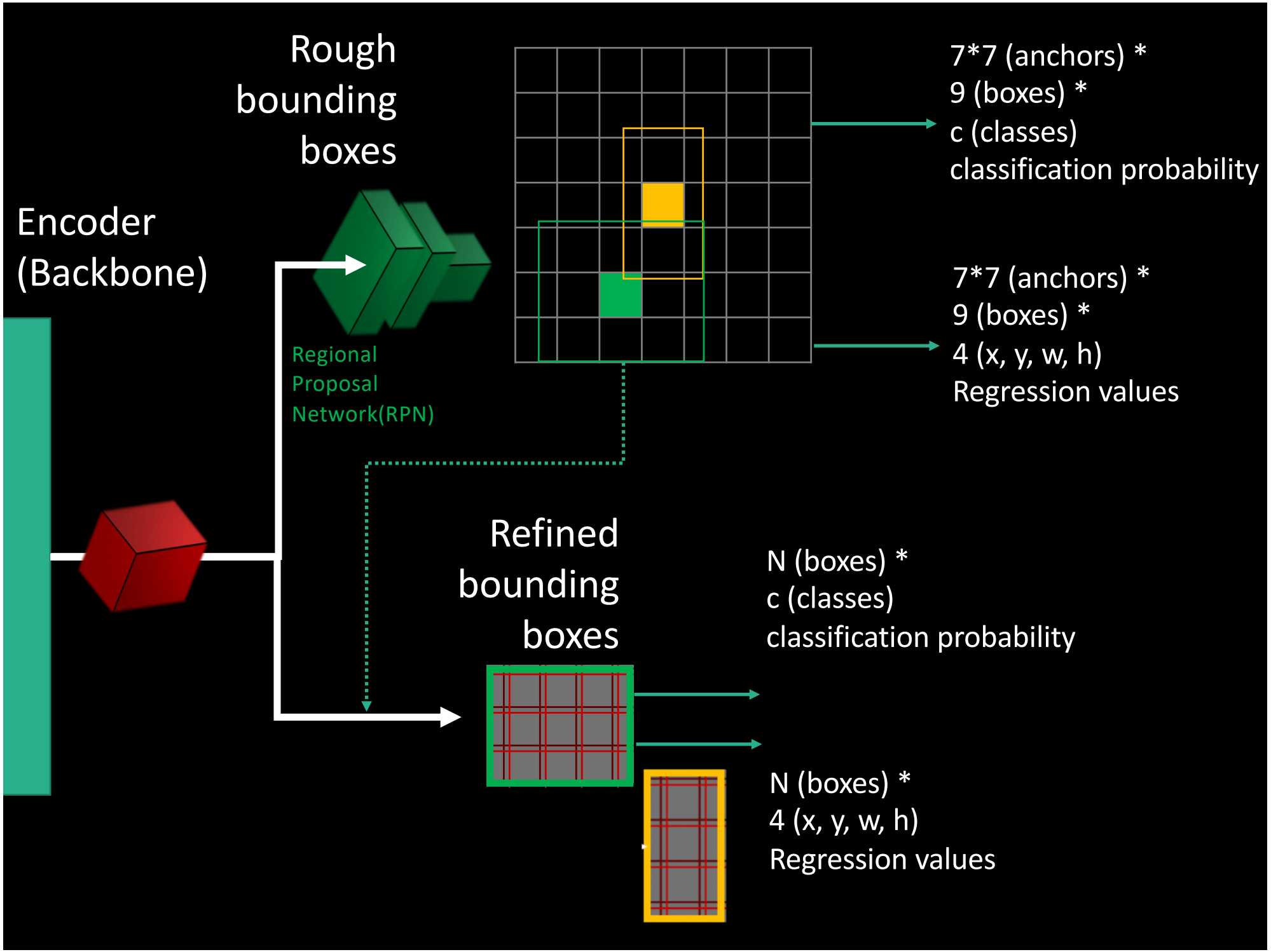
dW_a Width

dH_a Height

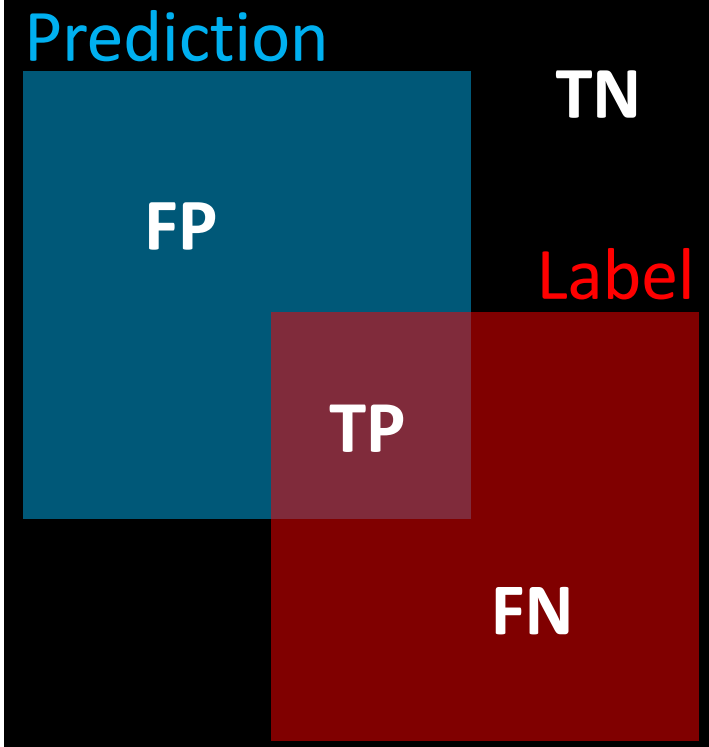

Regional Proposal Network(RPN)
Encoder (Backbone)
Latent feature map
Aligned ROI Features


Rough
bounding
boxes
Encoder
(Backbone)
Regional
Proposal
Network(RPN)
Refined
bounding
boxes
It’s kind of multi
branch


Rough
bounding
boxes
Encoder
(Backbone)
Regional
Proposal
Network(RPN)
Rough
Classification
Loss
(Cross entropy)
Rough
Boundary box regression
Loss
(MSE)
Refined
bounding
boxes
Refined
Classification
Loss
(Cross entropy)
Refined
Boundary box regression
Loss
(MSE)

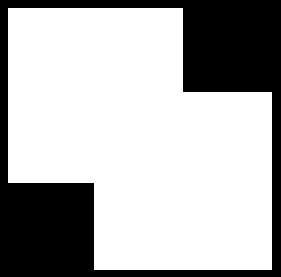

Rough
bounding
boxes
Encoder
(Backbone)
Regional
Proposal
Network(RPN)
7*7 (anchors) *
9 (boxes) *
c (classes)
classification probability
7*7 (anchors) *
9 (boxes) *
4 (x, y, w, h)
Regression values
Refined
bounding
boxes
N (boxes) *
c (classes)
classification probability
N (boxes) *
4 (x, y, w, h)
Regression values

Performance Metrics of Segmentation

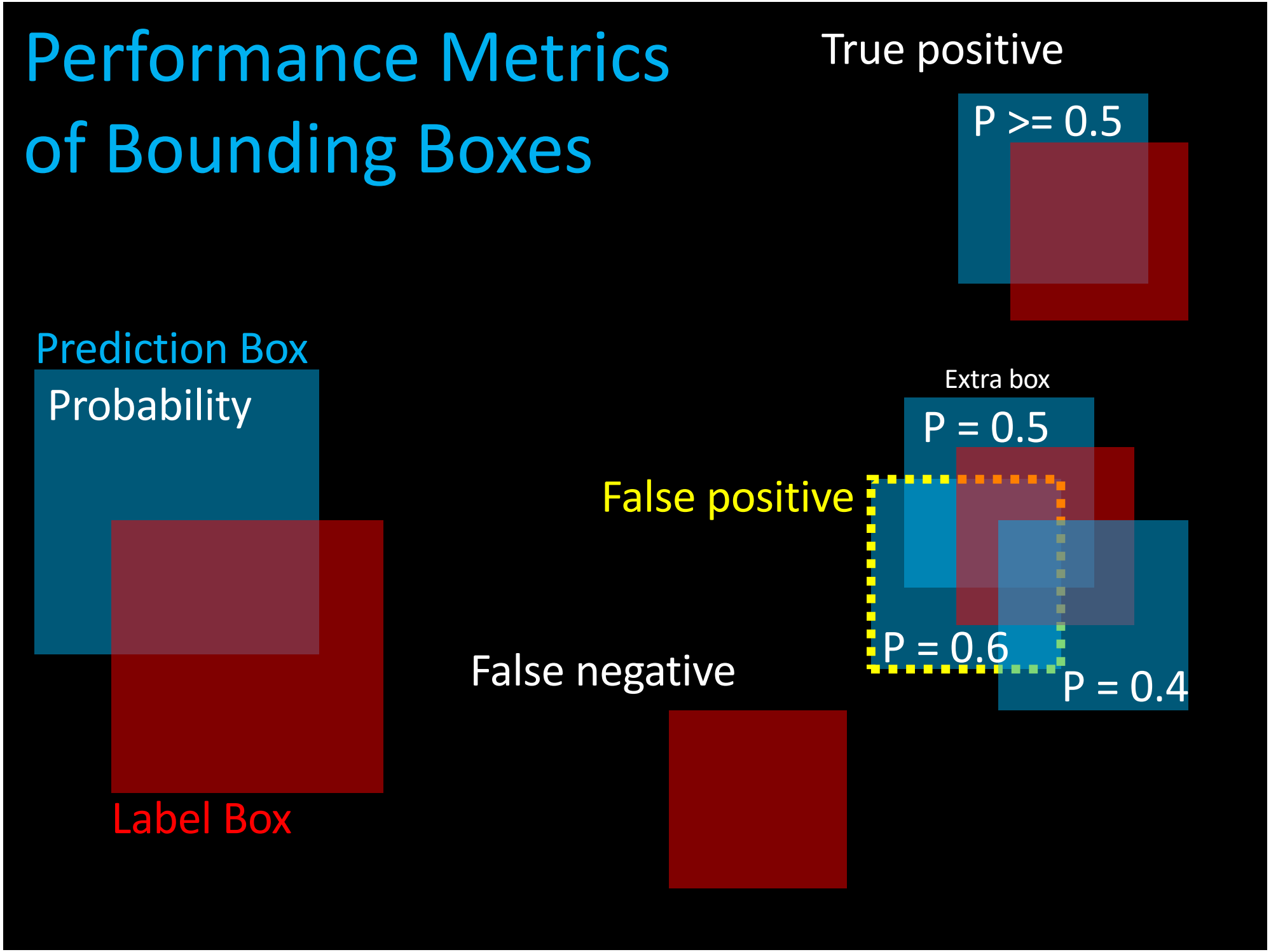


IoU

$$\frac{TP}{t + p - TP}$$



Performance Metrics of Bounding Boxes



Typical Components of a Instance segmentation Project

Task:

Instance Segmentation

Data

Image
+
Bounding boxes/ Segmentation mask

+
PREPROCESSING

Models

EX:
Resnet101
(backbone)
+
MaskRCNN
(architecture)

Metrics

IoU
F1
Dice coeff.

Loss

Segmentation Loss

Cross-entropy

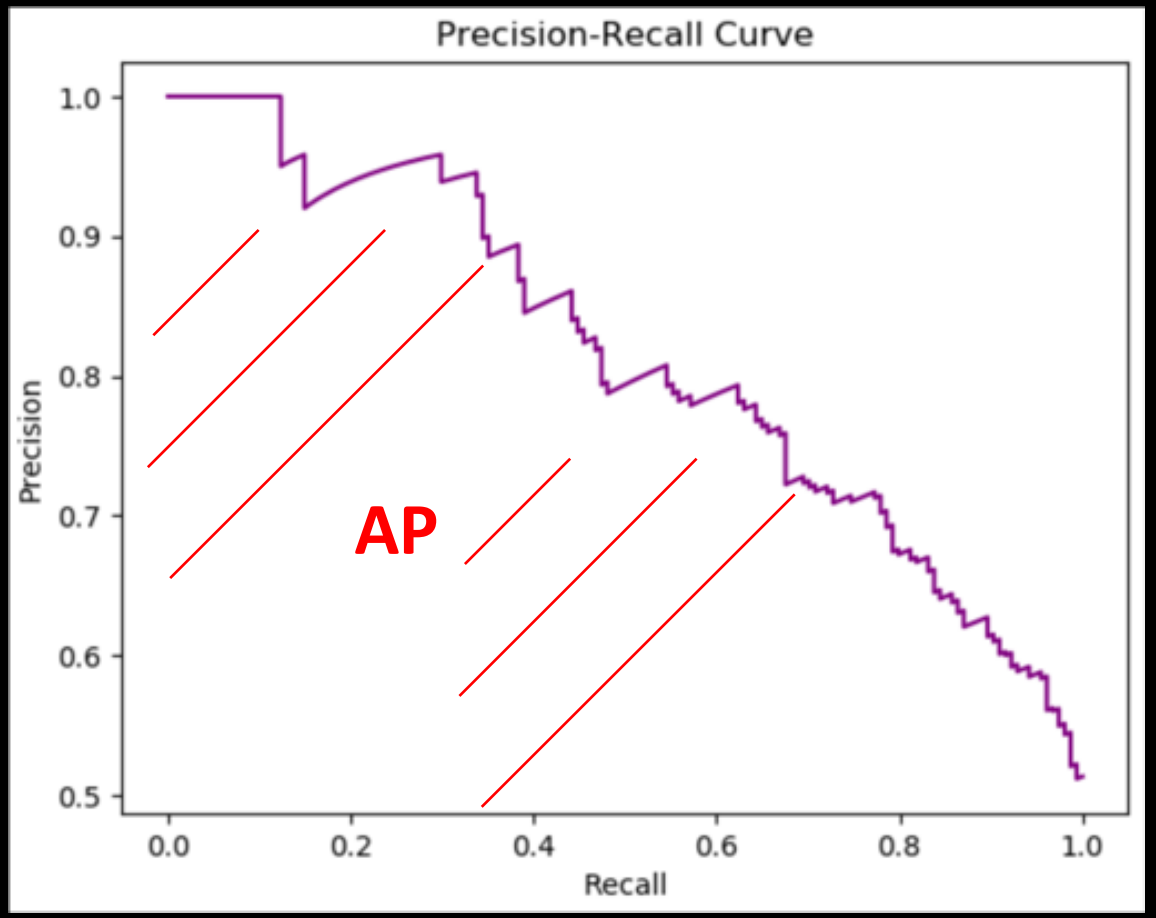
Bounding-box Regression loss

Precision

$$\frac{TP}{TP + FP}$$

Recall

$$\frac{TP}{TP + FN}$$



We often calculated mAP (mean over all the classes)