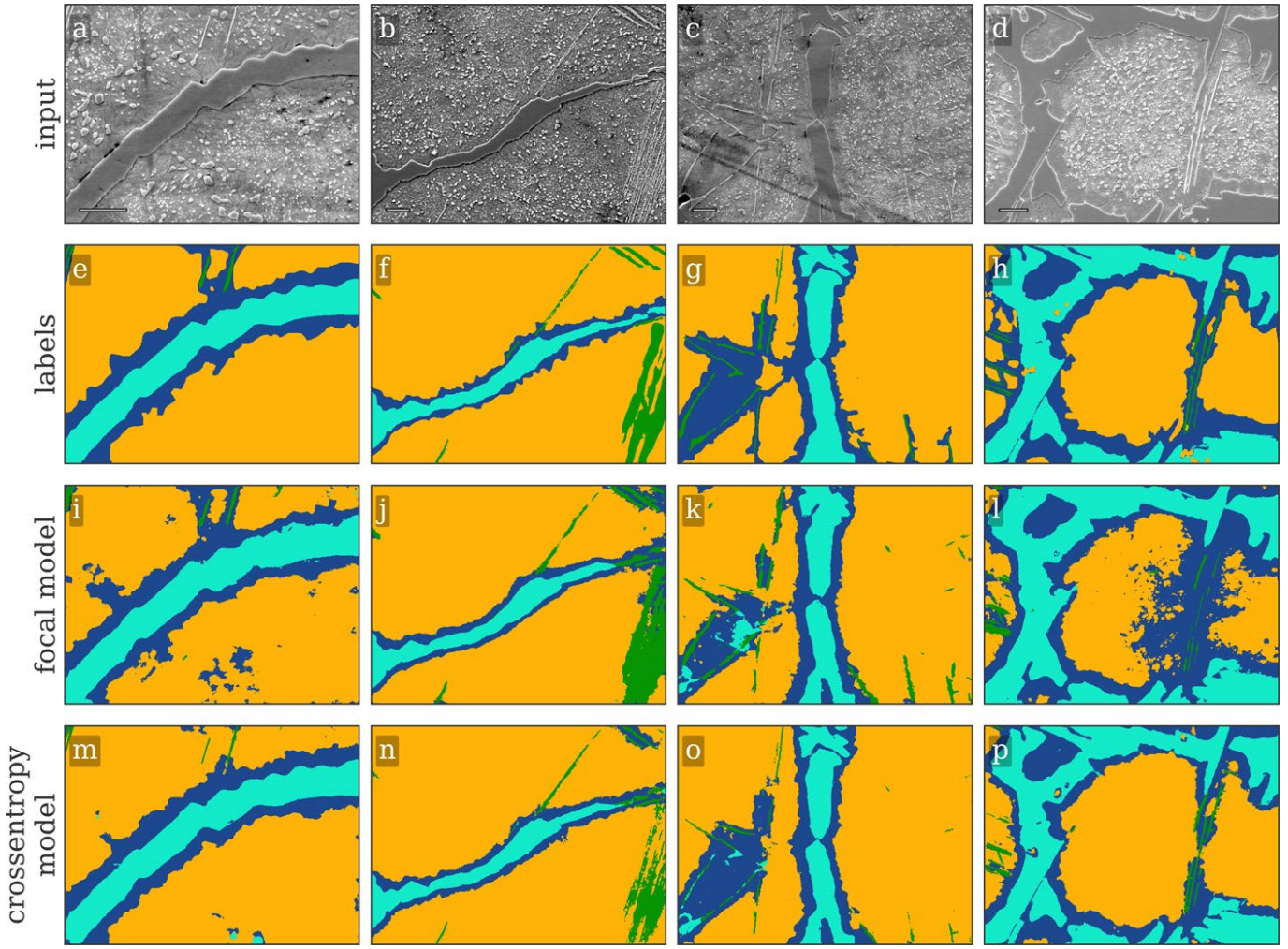
Image segmentation

Image segmentation is becoming an increasingly important part of materials informatics

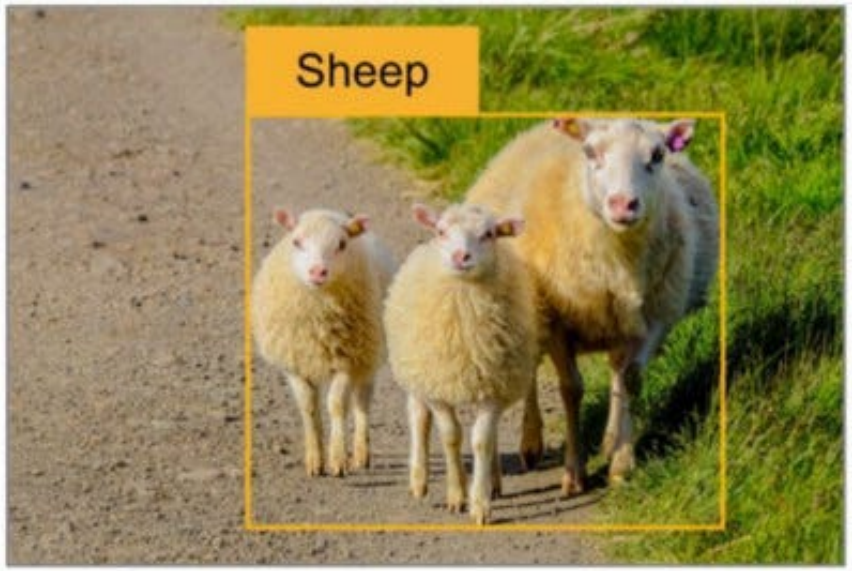
Definition: Image segmentation is the process of partitioning an image into multiple segments or pixels with similar attributes to simplify its representation. It's a critical step in image analysis, helping in the identification of regions of interest, such as grain boundaries in materials science.

Goal: The primary goal is to make an image more meaningful and easier to analyze by isolating features of interest from the background or differentiating between different material phases.

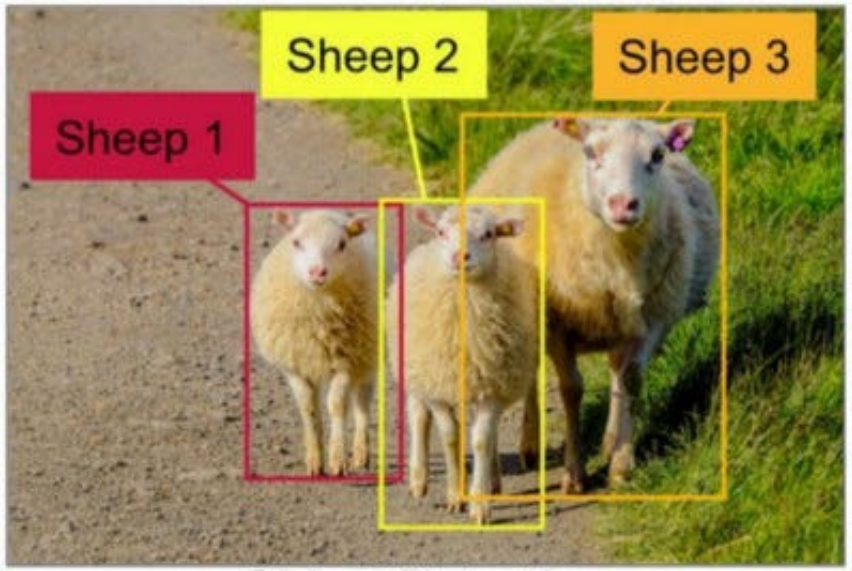


DeCost, Brian L., et al. Microscopy and Microanalysis 25.1 (2019): 21-29.

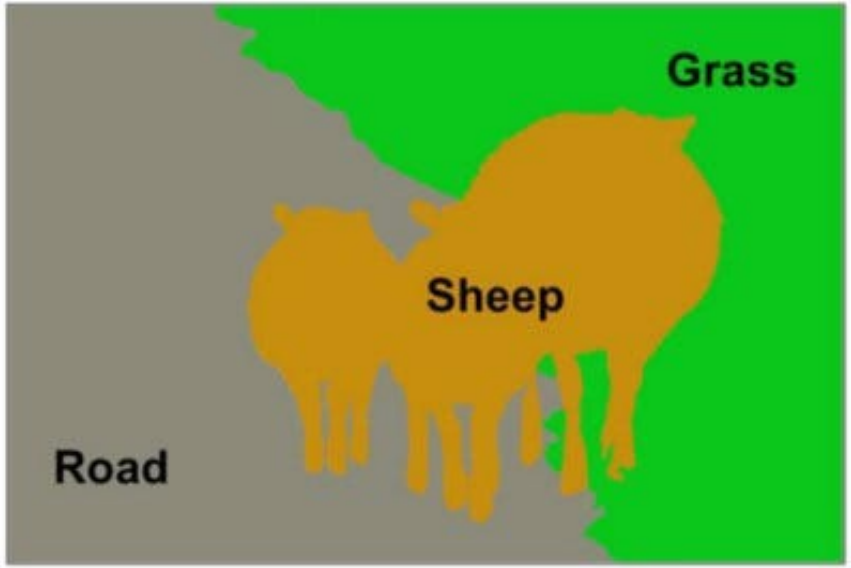
Recognition vs semantic segmentation vs object detection vs instance segmentation



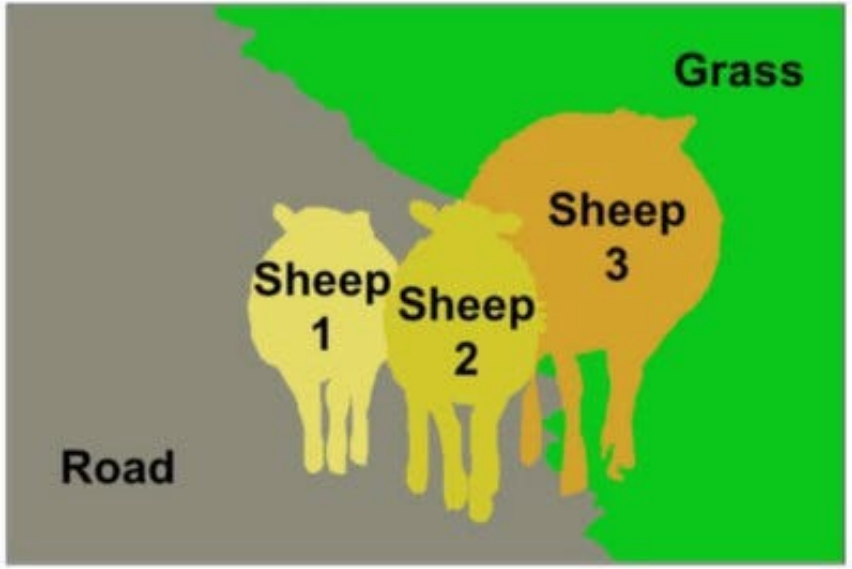
Classification + Localization



Object Detection



Semantic Segmentation



Instance Segmentation

Image segmentation involves several techniques

Techniques:

1. Thresholding
2. Edge detection
3. Region growing

Deep learning techniques:

1. Convolutional neural nets
2. U-Net
3. Segment Anything Model (SAM)

Thresholding is something you've probably come across before

Thresholding: The simplest form of segmentation, based on pixel intensity. The basic idea is to select a threshold value, and then all pixels above this value are considered part of one segment, while those below are considered another.

Math is simple for thresholding

$$I(x,y) \rightarrow S(x,y)$$

$$S(x,y) = \begin{cases} 1 \; if \; I(x,y) > T \\ 0 \quad otherwise \end{cases}$$

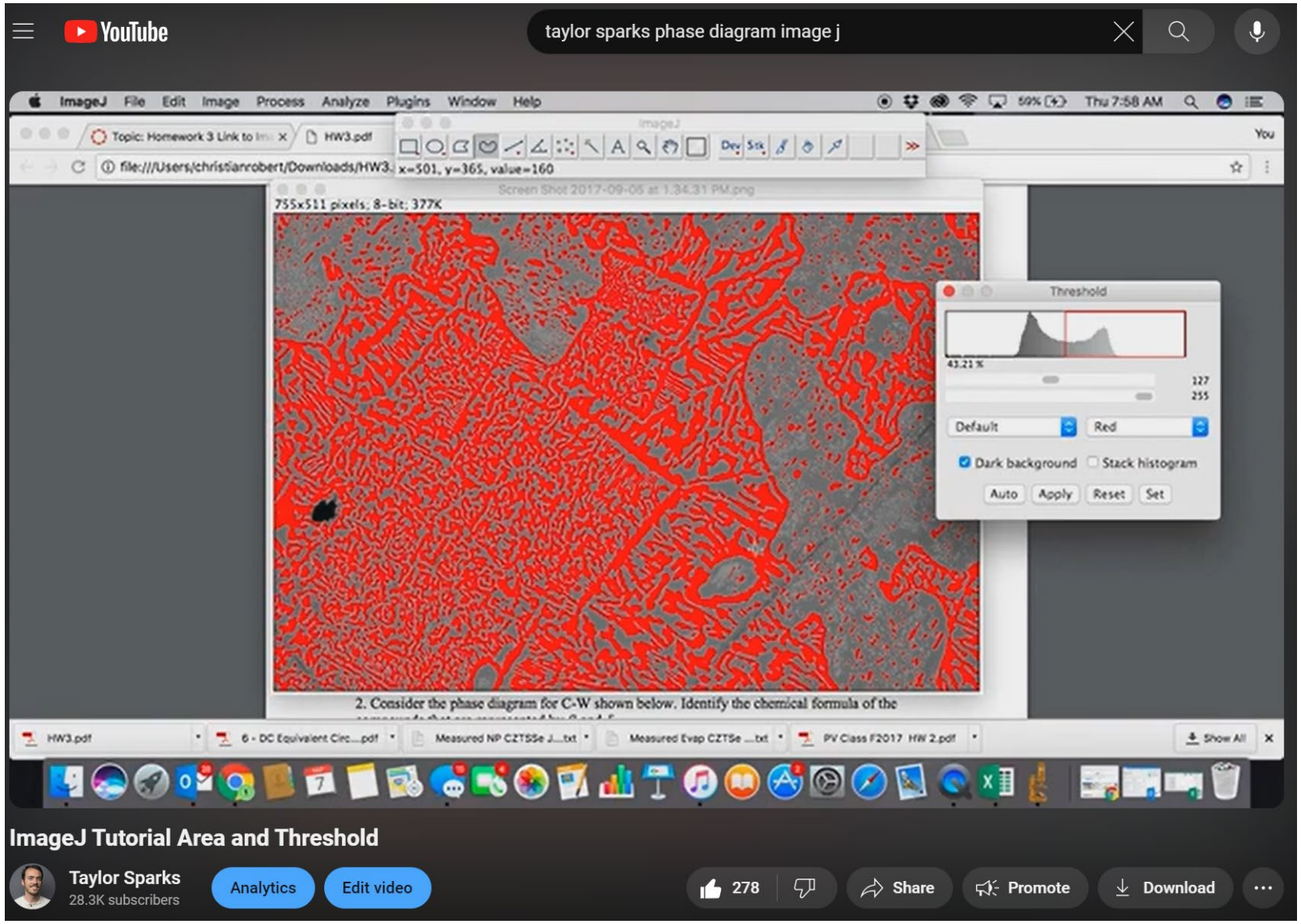
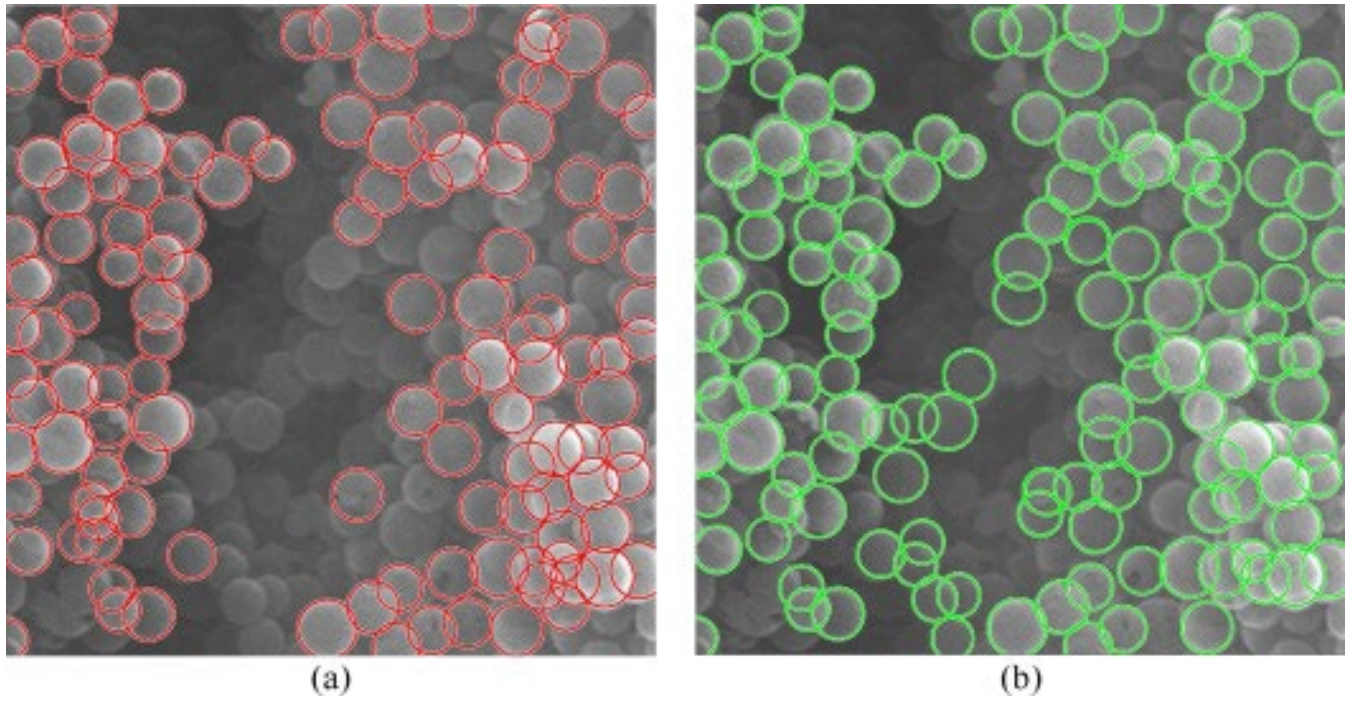


Image segmentation techniques include thresholding, edge detection, and region growing



Edge Detection: Identifies boundaries between different regions based on discontinuities in pixel intensity. Common methods include Sobel, Canny, and Laplacian of Gaussian (LoG).

Math depends on the operator used

$$G_x = S_x * I \; and \; G_y = S_y * I$$

Overall gradient G becomes

$$G = \sqrt{G_x^2 + G_y^2}$$

Meng, Yingchao et al. Micron, 106, 34-41 (2018)

Sobel filter example

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

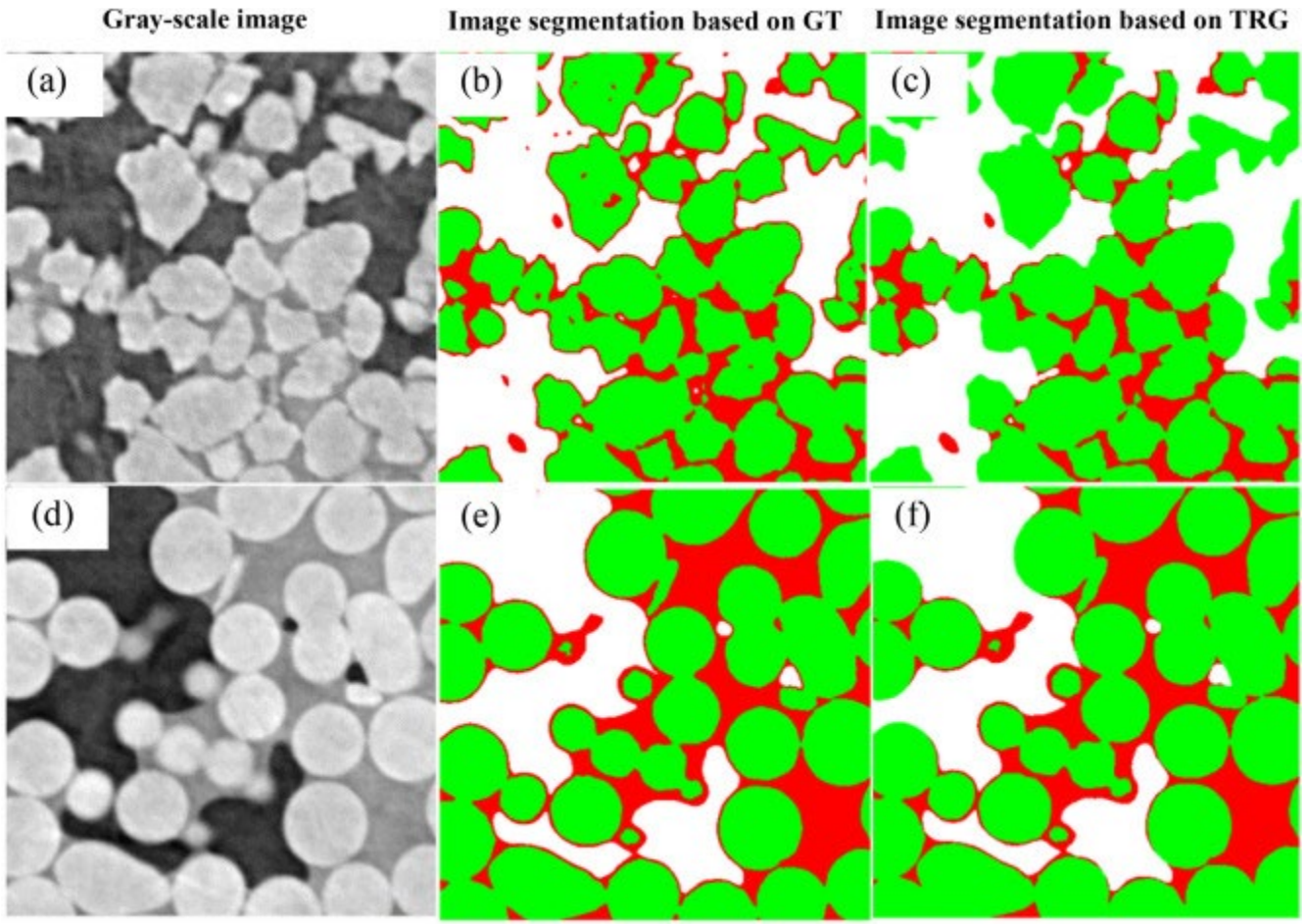
Image segmentation techniques include thresholding, edge detection, and region growing

Region Growing: Starts with a set of seed points and grows regions by appending to each seed those neighboring pixels that have similar properties (e.g., intensity).

Pixel p_i in region R with some pixel p_j in neighborhood that gets added to R if it satisfies some similarity criterion

$$C(p_j) = \begin{cases} True & if \ |I(p_j) - \mu_R| < T \\ False & otherwise \end{cases}$$

μ_R is avg pixel intensity in region R



Cheng and Wang, Powder Technology, 368, 80-89 (2020)

U-Net has become a very popular tool for image segmentation

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

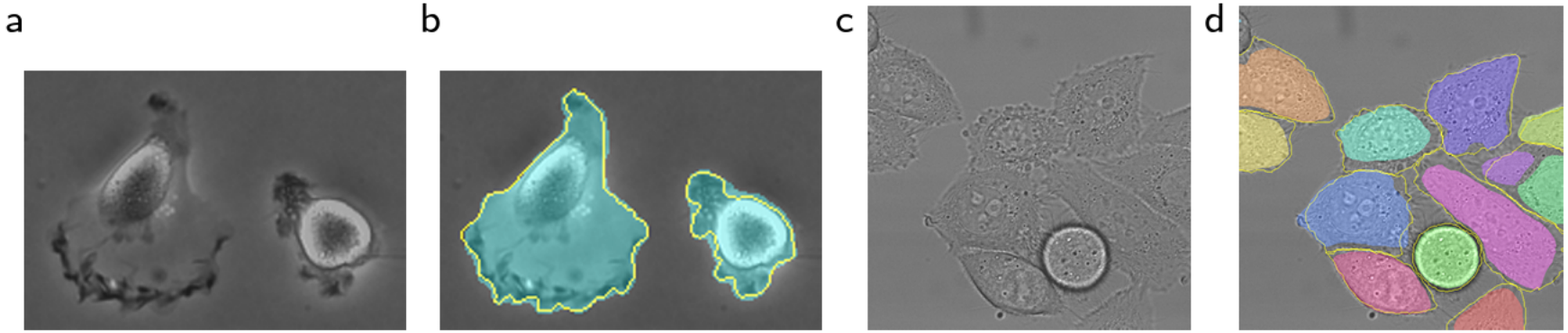
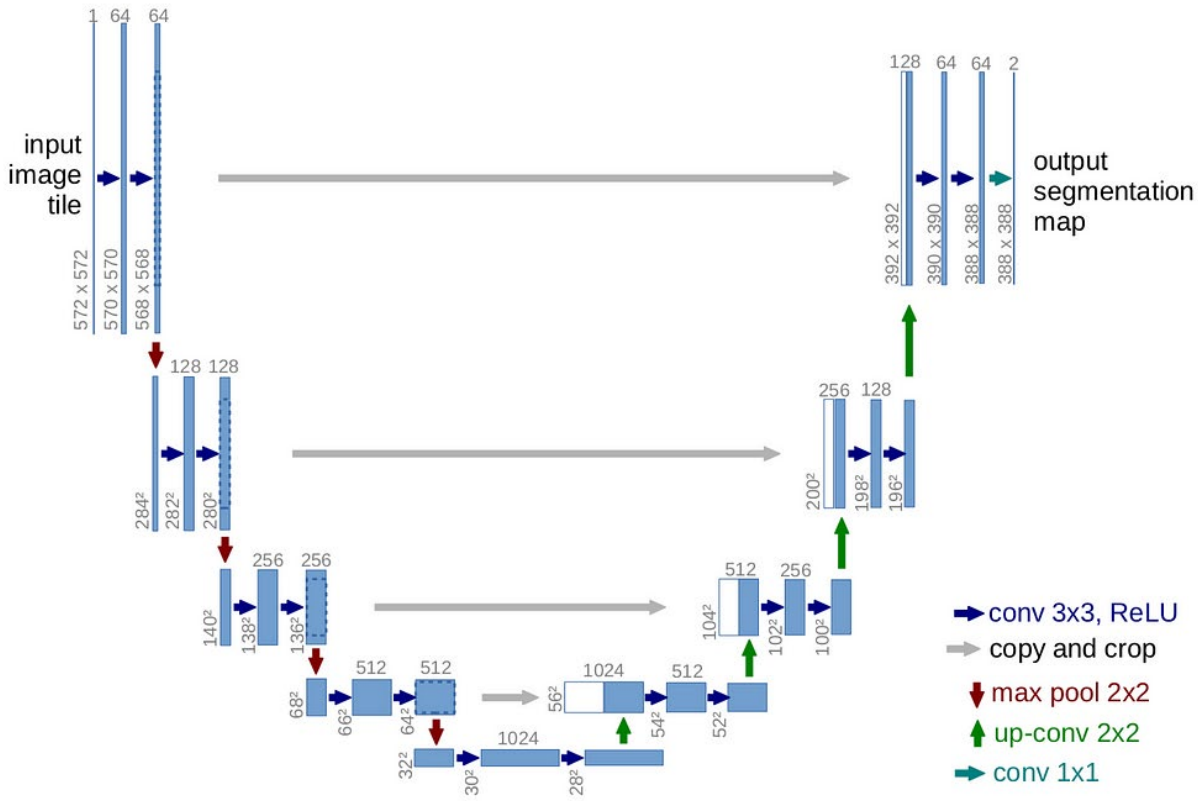


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

U-Net has become a very popular tool for image segmentation

Contracting Path (encoder): Composed of repeated application of two 3x3 convolutions (each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for down-sampling). At each down-sampling step, the number of feature channels is doubled.

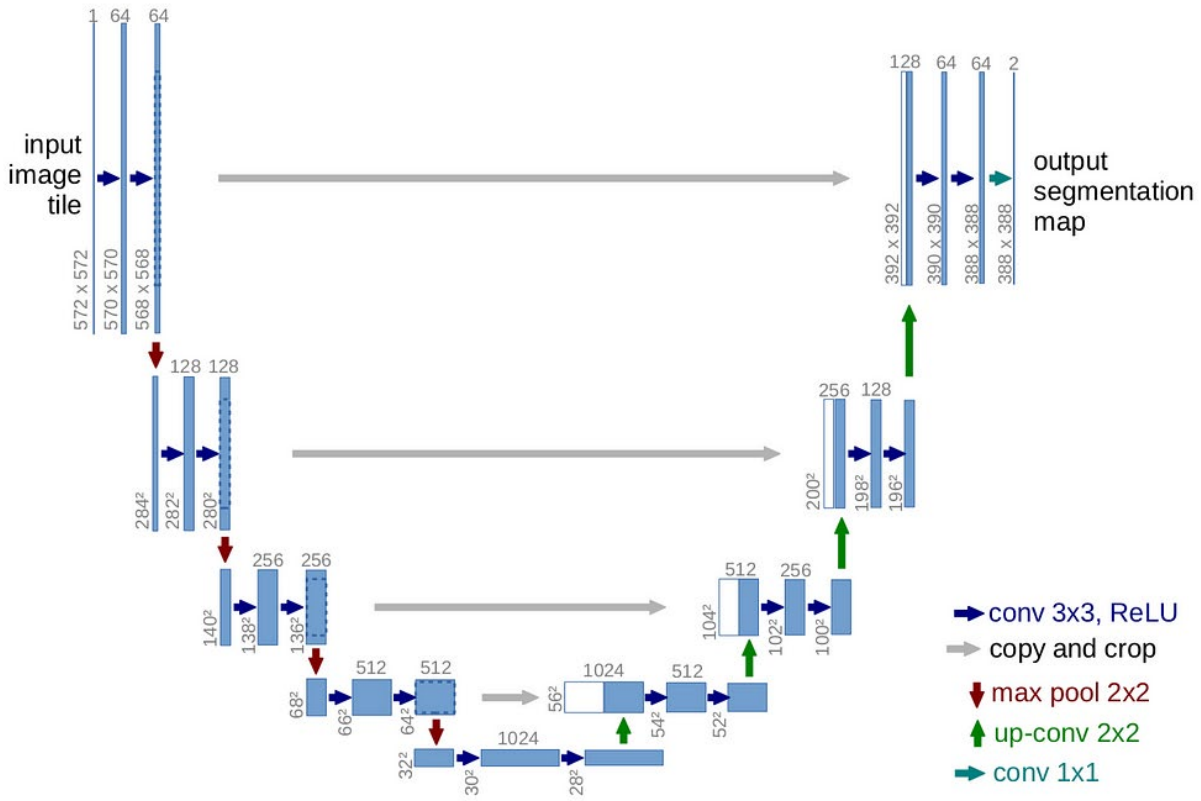
Expanding Path (decoder): Consists of repeated upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU.



The concatenation step is really what differentiates U-Net from other CNNs

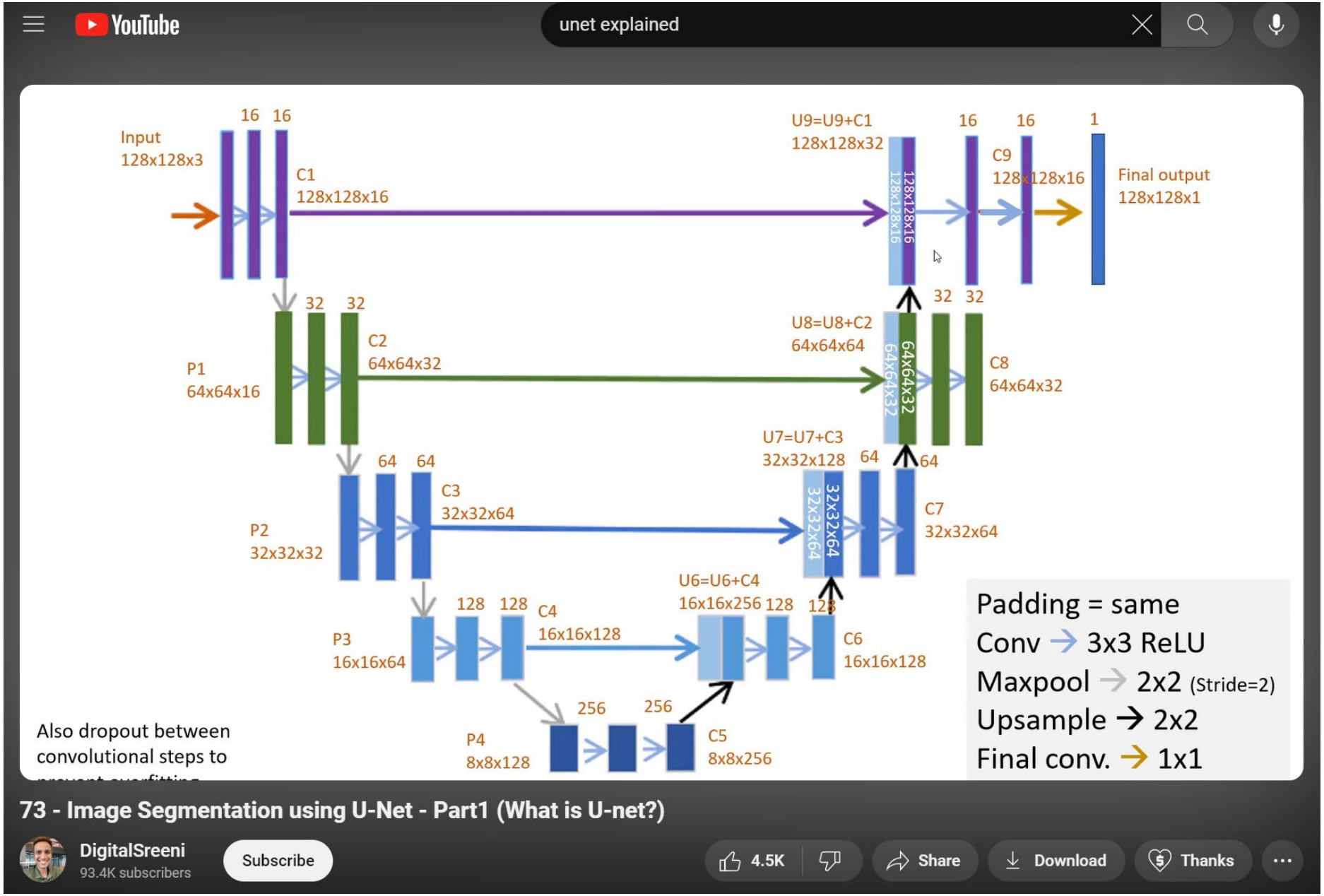
“Copy and Crop” mechanism:
Concatenation of feature maps helps provide localized information!

“The main contribution of U-Net in this sense is that while upsampling in the network we are also concatenating the higher resolution feature maps from the encoder network with the upsampled features in order to better learn representations with following convolutions. Since upsampling is a sparse operation we need a good prior from earlier stages to better represent the localization.”

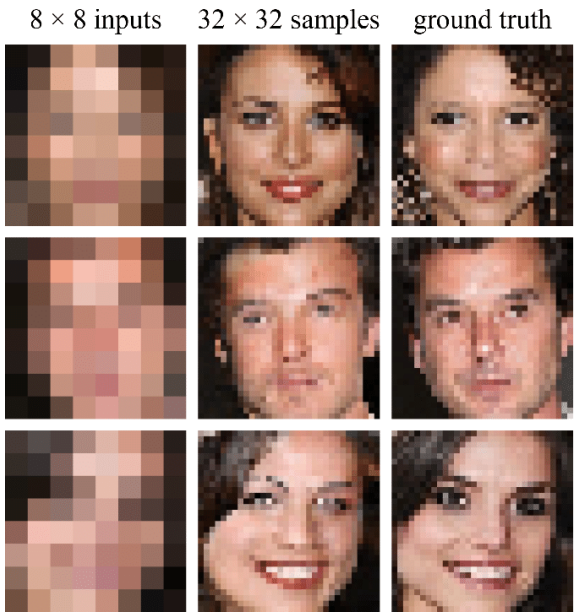


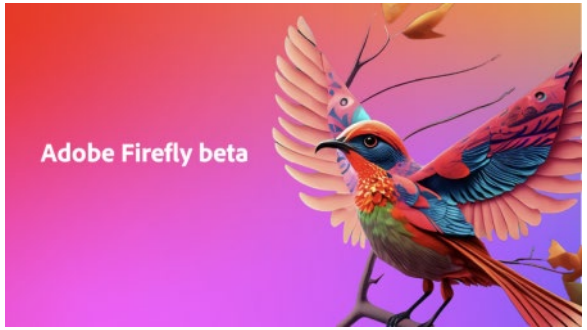
https://developers.arcgis.com/python/guide/how-unet-works/

Exact U-Net architecture is tunable



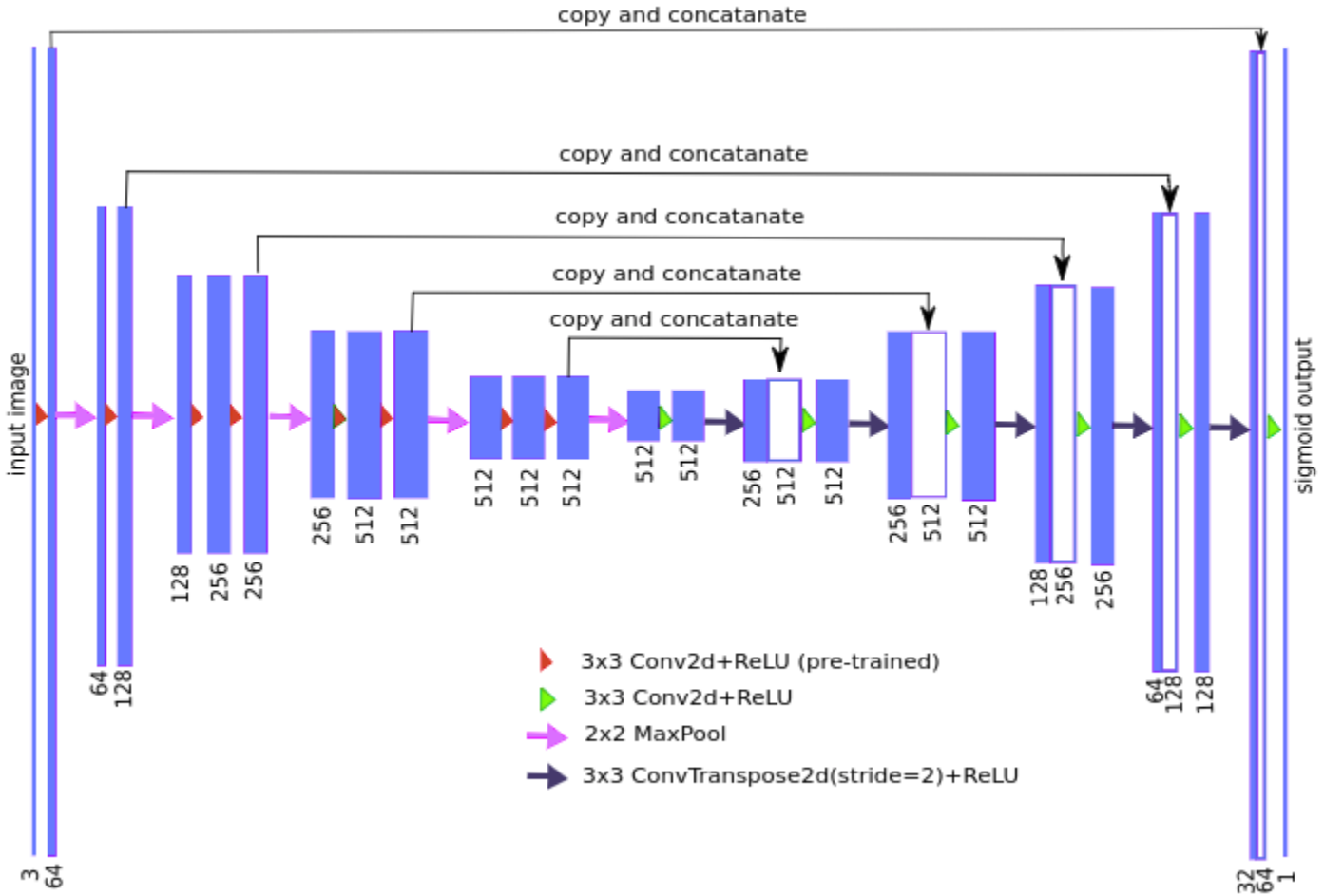
U-Net is useful for segmentation, but also super-resolution, and generative models





Transfer learning is letting us go beyond "vanilla" U-Net architectures

The contraction path is really just an encoder…. So why not leverage beefy ImageNet encoders like VGG with pretrained weights? Then we just focus on the decoder.



https://medium.com/@keremturgutlu/semantic-segmentation-u-net-part-1-d8d6f6005066

Meta's Segment Anything Model (SAM) is now the current SOTA

SAM is "the world's first massive scaled, promptable, interactive, foundation image segmentation model"

Kirilov et al. https://arxiv.org/abs/2304.02643 2023

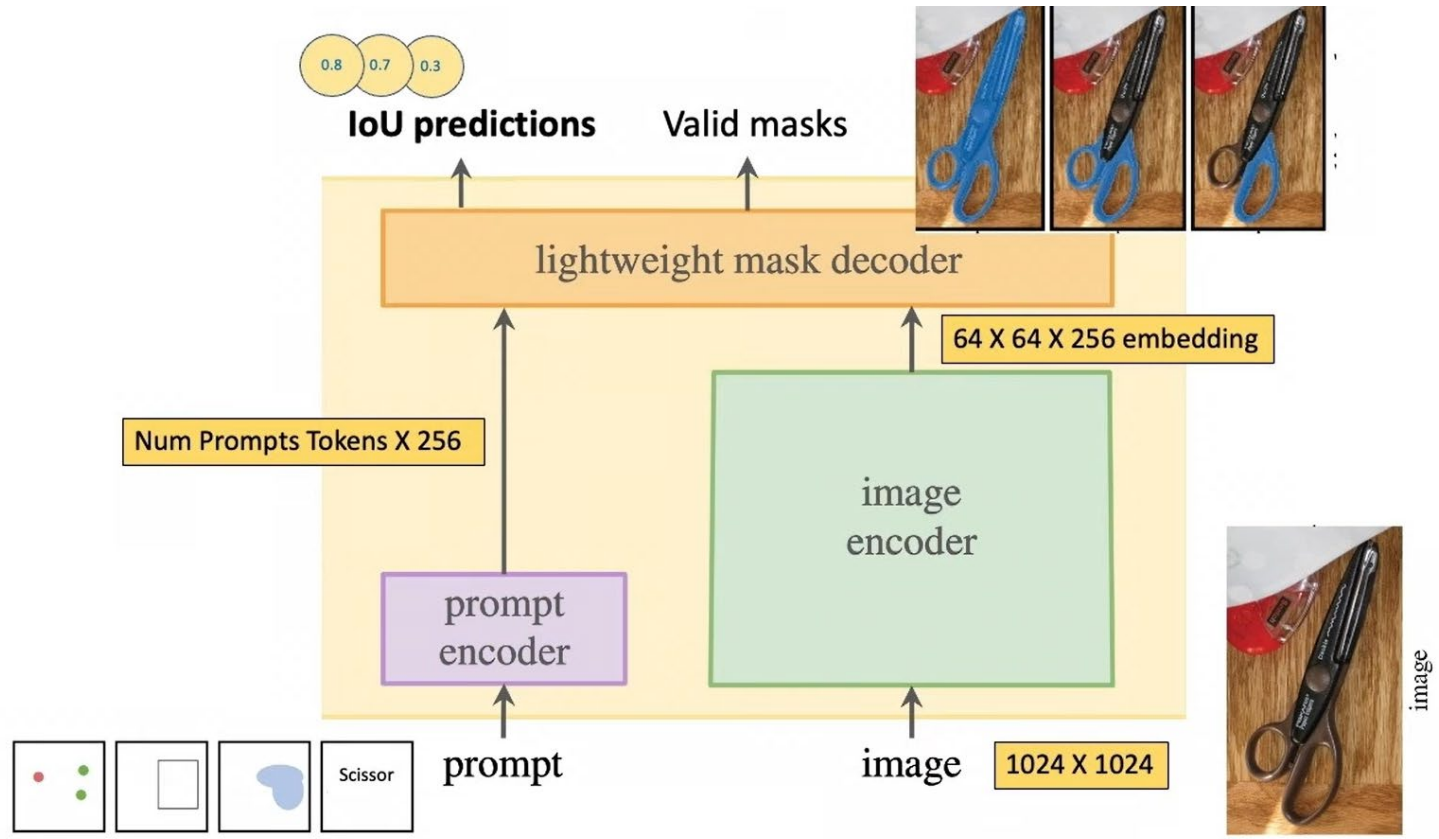
How SAM works

SAM takes in an input….

- image and breaks it down into embeddings
- Prompt (multi modal, clicks, box, shading, text)

Merges these and outputs

- predicts 3 masks
- whole, part, subpart



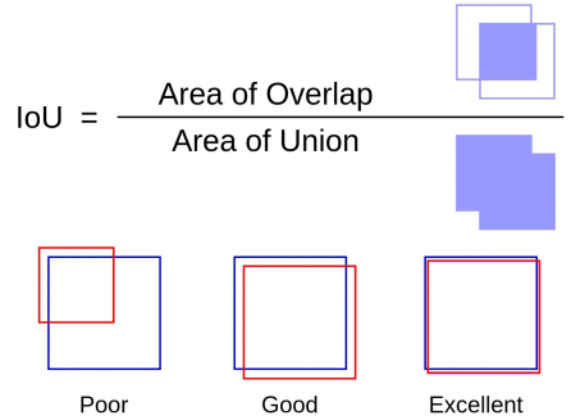
Kirilov et al. https://arxiv.org/abs/2304.02643 2023

The masks are learned via a custom loss-function emphasizing new objects

Total loss depends on focal loss and dice loss in 20:1 ratio

Dice loss is similar to IoU (intersection over union)

- The mask with highest IoU is used to calculate loss



Focal Loss is modified cross-entropy loss

- with higher weights on hard mis-classified examples, and lower weight on easy examples

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

Kirilov et al. https://arxiv.org/abs/2304.02643 2023

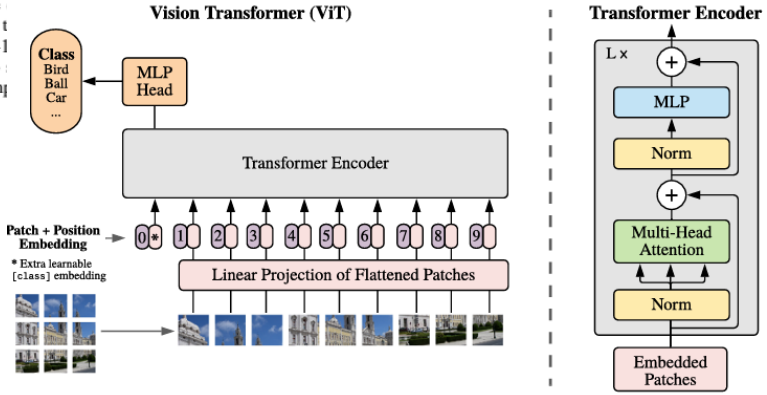
Separating the image encoder and prompt encoder allows for versatile use

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*,
Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transforme
very well on image
data and transferred t
(ImageNet, CIFAR-1
results compared to
stantially fewer comp

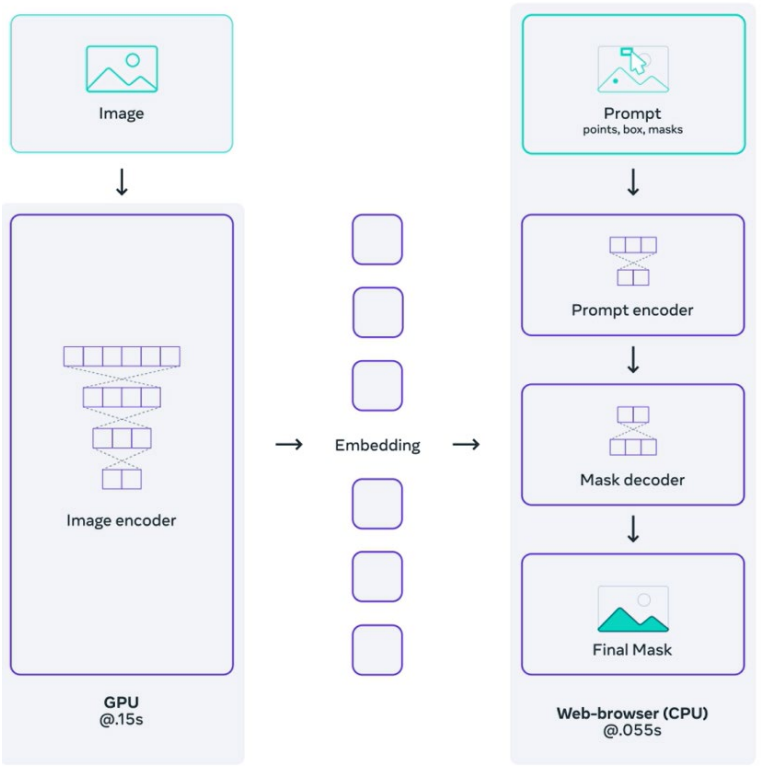


100s of millions of parameters, runs on cloud-based GPU server with slower latency

SAM was trained on 11M images with 1B+ masks.

- You can download the full SA-1B dataset!

Interactive prompt is faster latency



Kirilov et al. https://arxiv.org/abs/2304.02643 2023, Dosovitskiy et al https://arxiv.org/abs/2010.11929

The image encoder is a “masked encoder”

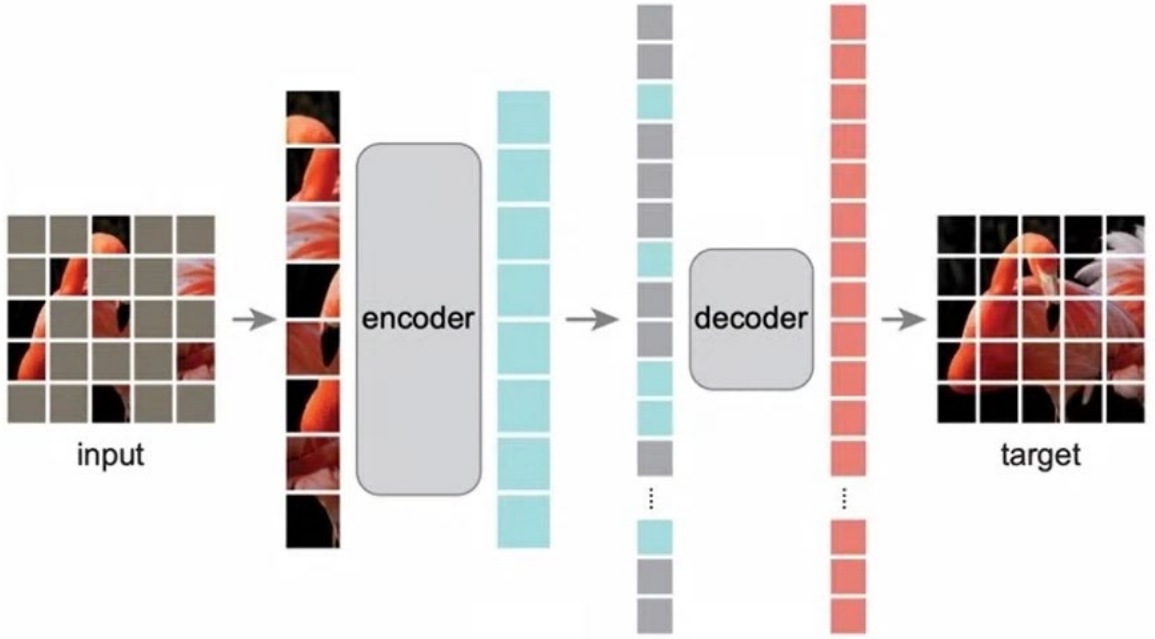


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

He et al. https://arxiv.org/abs/2111.06377

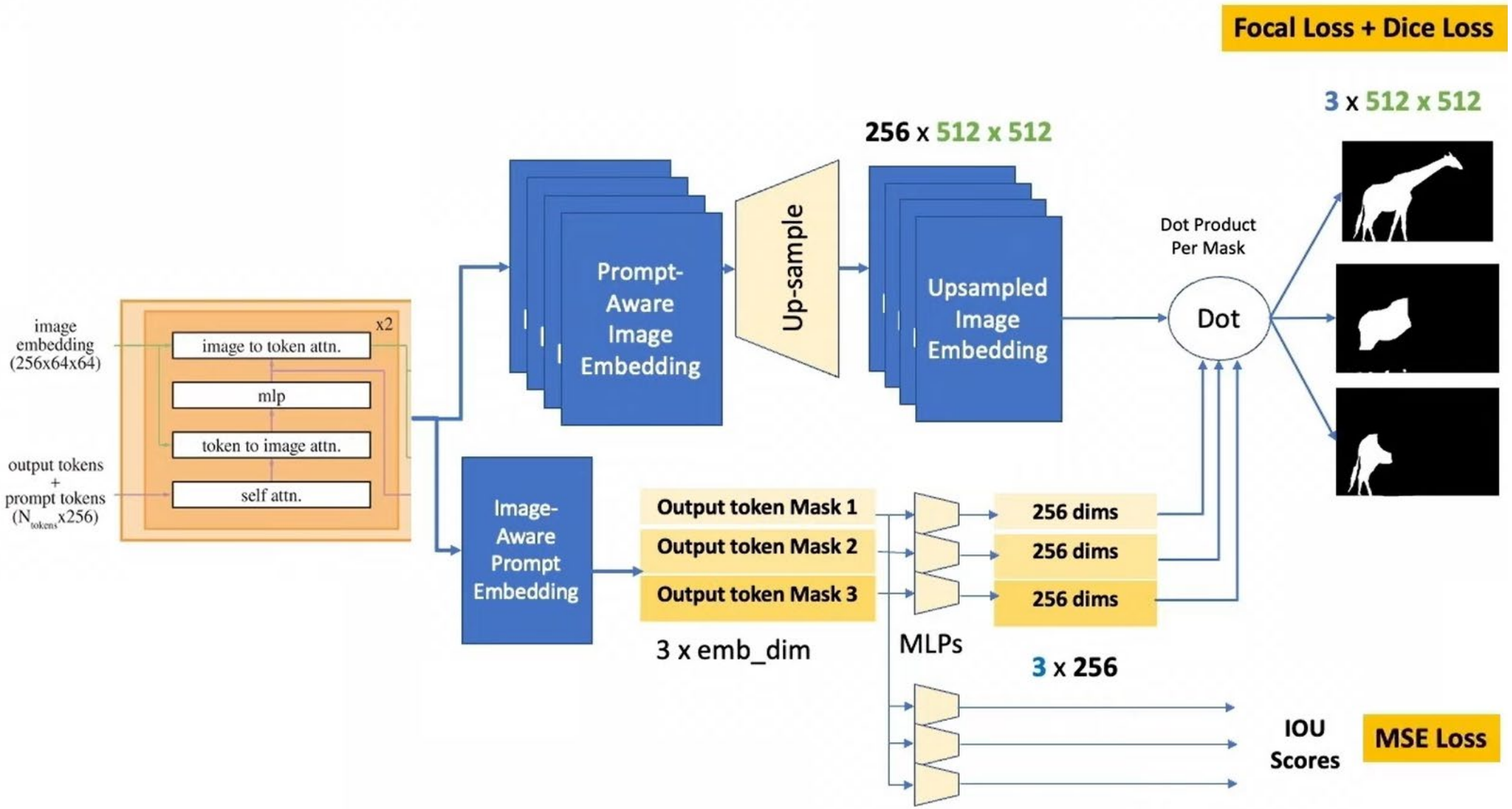
The prompt encoder had various inputs

Encoding points
- Positional encoding of point (x,y)
- Embeddings indicating foreground vs background

Encoding bounding boxes
- Positional encodings for upper left and lower right points

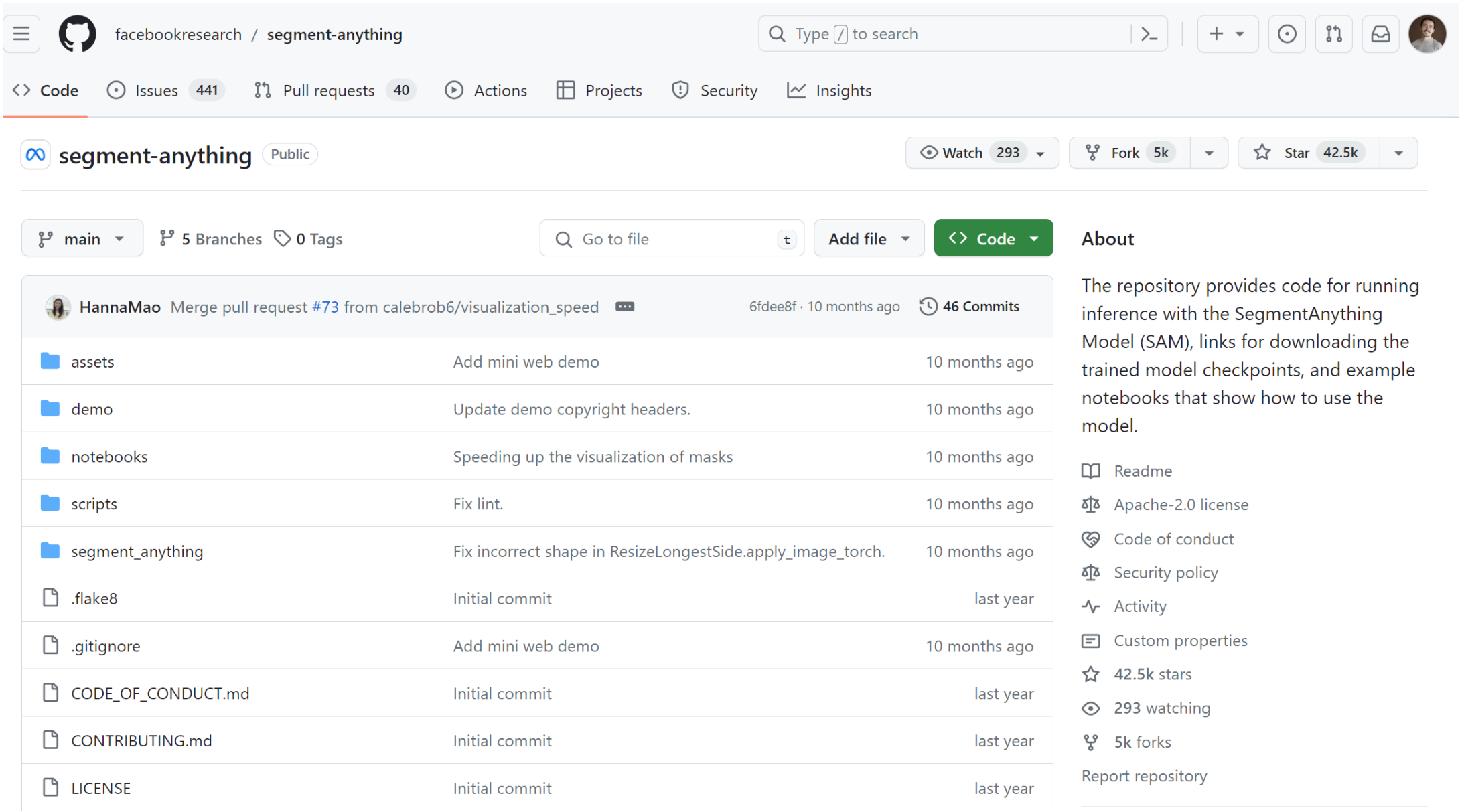
Ecoding text prompts
- Pretrained CLIP embeddings

The decoder architecture



Explaining the Segment Anything Model - Network architecture, Dataset, Training. Neural Breakdown with AVB, YouTube

Best of all, SAM is open source!



Explaining the Segment Anything Model - Network architecture, Dataset, Training. Neural Breakdown with AVB, YouTube

Diffusion models