Computational Biology Project

Day 2

Welcome to day 2!

Overview:

- Feedback on day 1
- Go over the day 1 classification workflow
- Introduction to object detection and segmentation
- Supervised vs unsupervised segmentation
- Overview of the UNet model
- Start working through day 2 together

Feel free to ask questions whenever they come up!

Feedback on day 1

We'll take a moment to do a poll on what everyone thought of day 1

Leave comments on:

- Did you enjoy the material and the coding exercises?
- Did you find the difficulty level appropriate?
- Were there any parts of the workflow you wish you learned more about?

Introduction to object detection and segmentation

Object detection:

Determining whether an object of interest is in an image and identifying its location

What is segmentation?

Introduction to segmentation

Object detection:

Determining whether an object of interest is in an image and identifying its location

What is segmentation?

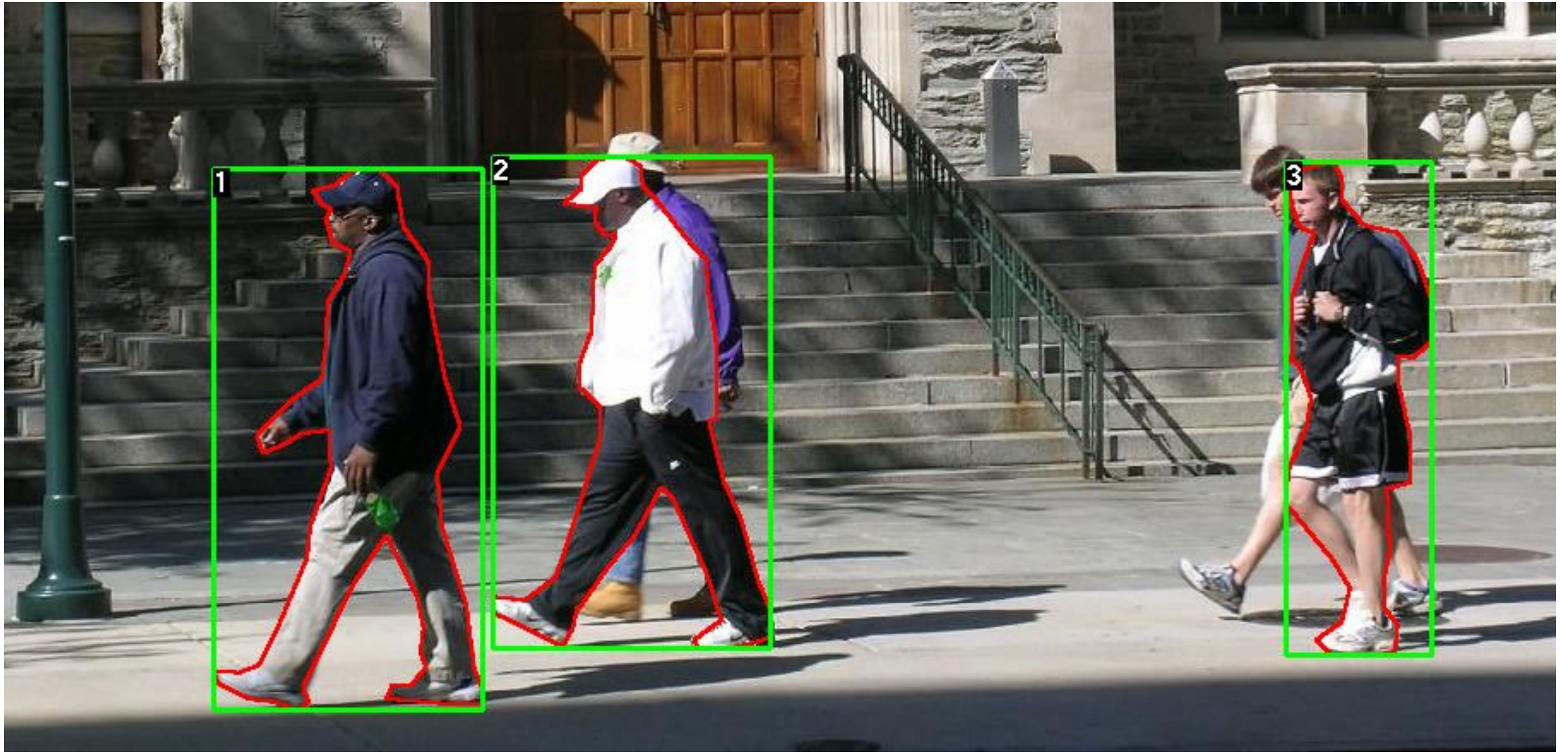
Segmentation is the pixel-by-pixel labelling of an object of interest

Segmentation creates a new image, a segmentation mask, that identifies object of interest with the label 1, where everything else is labeled 0

From Day 1 slides:

Green = bounding boxes, red = bounding contours

Introduction to computer vision: object detection

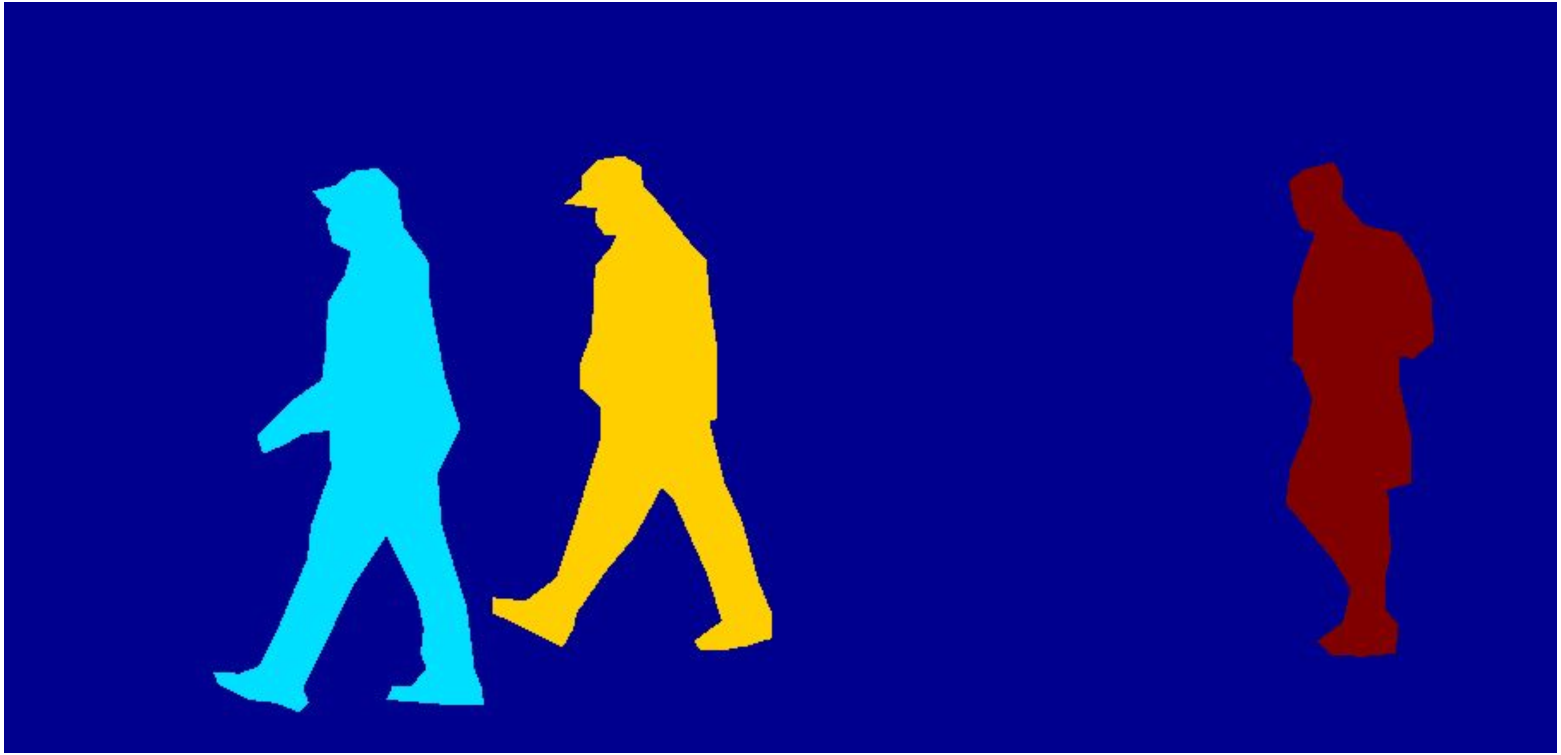


https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

From Day 1 slides:

Blue = label 0, light blue, yellow, red = labels 1, 2, 3

Introduction to computer vision: object segmentation



https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

Introduction to object detection and segmentation

What are some applications of object detection and/or segmentation?

Introduction to object detection and segmentation

What are some applications of object detection and/or segmentation?

- Facial recognition
- Self-driving cars
- Autonomous robots
- Automatic image annotation
- Automatic cancer detection

Segmentation approaches

We can perform segmentation using neural networks in either a supervised or unsupervised setting

What is the difference between supervised and unsupervised learning?

Segmentation approaches

We can perform segmentation using neural networks in either a supervised or unsupervised setting

What is the difference between supervised and unsupervised learning?

Supervised learning: Learning to complete a task by training on input and target pairs of data

Unsupervised learning: Learning to complete a task using just training inputs with no target data

Supervised segmentation

We can train a neural network to identify pixel-by-pixel what is background or foreground by training on pairs of images and their segmentation maps

The most popular segmentation neural network in medical image analysis in the UNet architecture (https://arxiv.org/pdf/1505.04597.pdf)

Supervised segmentation: UNet

Like the classification model, the first part of the UNet is to “encode” the image by using a series of layers to extract the key features

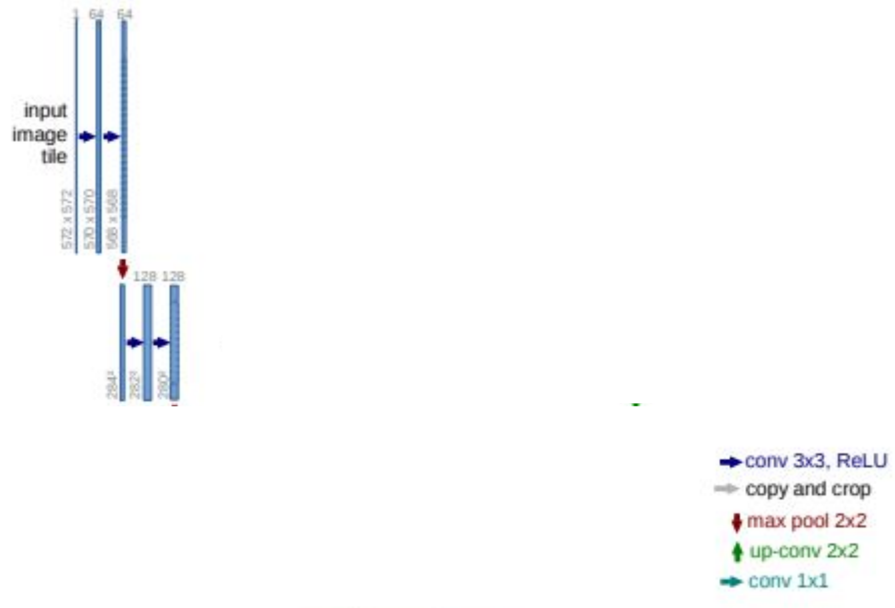


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Like the classification model, the first part of the UNet is to “encode” the image by using a series of layers to extract the key features

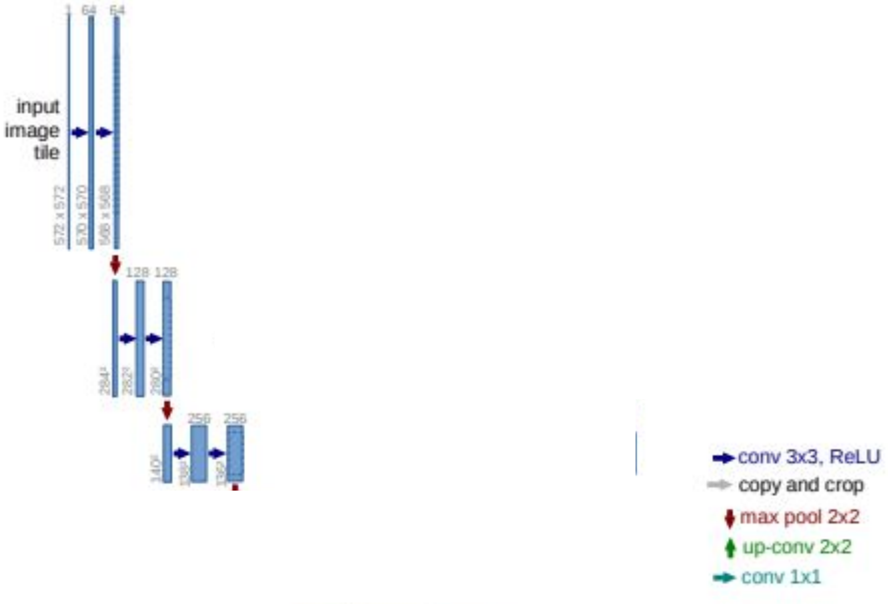


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Like the classification model, the first part of the UNet is to “encode” the image by using a series of layers to extract the key features

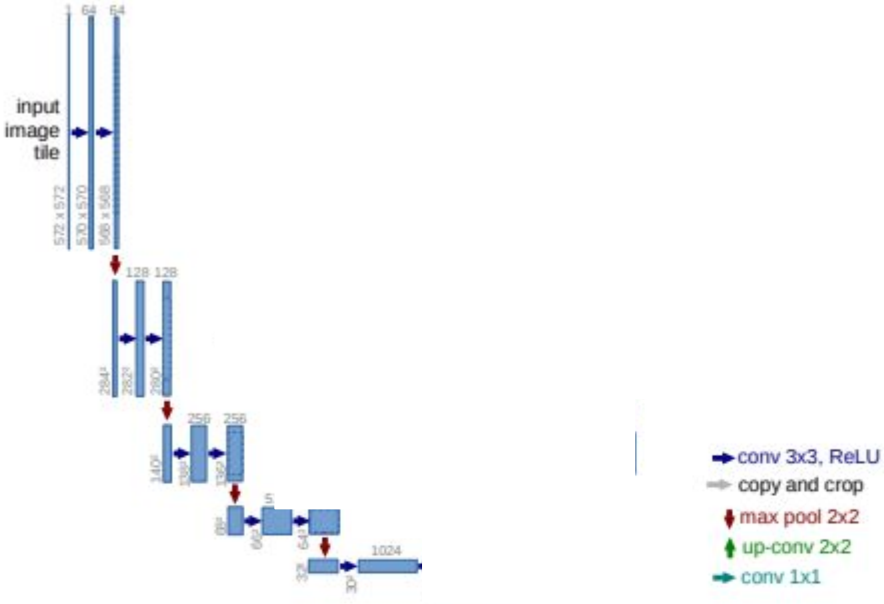


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Then, the network has a series of operations to “decode” the encoded feature vector, using upsampling operations and information from the encoding path

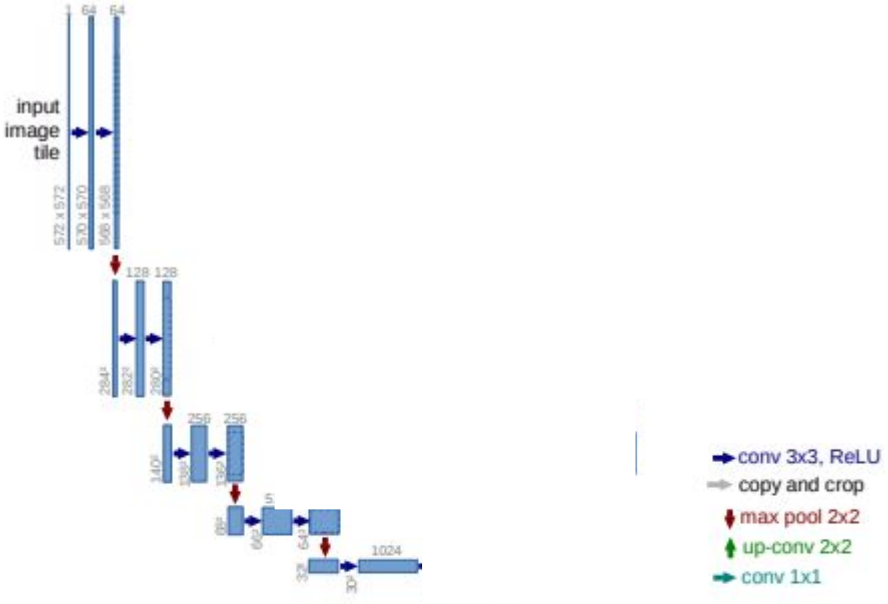


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Then, the network has a series of operations to "decode" the encoded feature vector, using upsampling operations and information from the encoding path

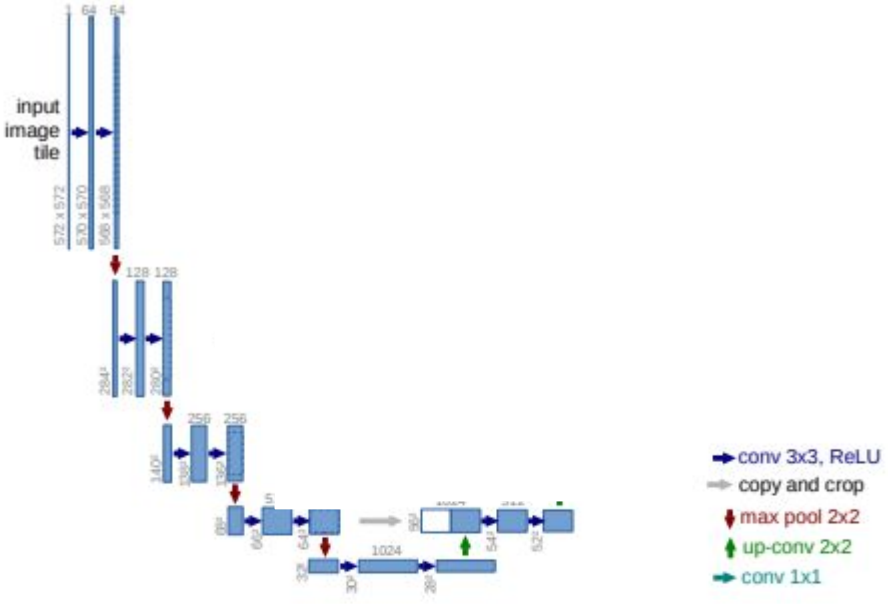


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Then, the network has a series of operations to “decode” the encoded feature vector, using upsampling operations and information from the encoding path

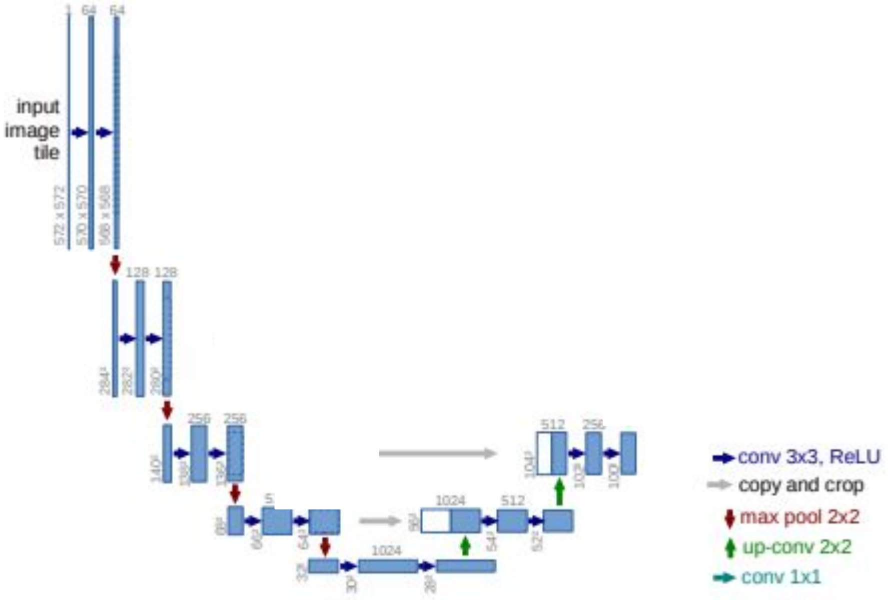


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

Then, the network has a series of operations to “decode” the encoded feature vector, using upsampling operations and information from the encoding path

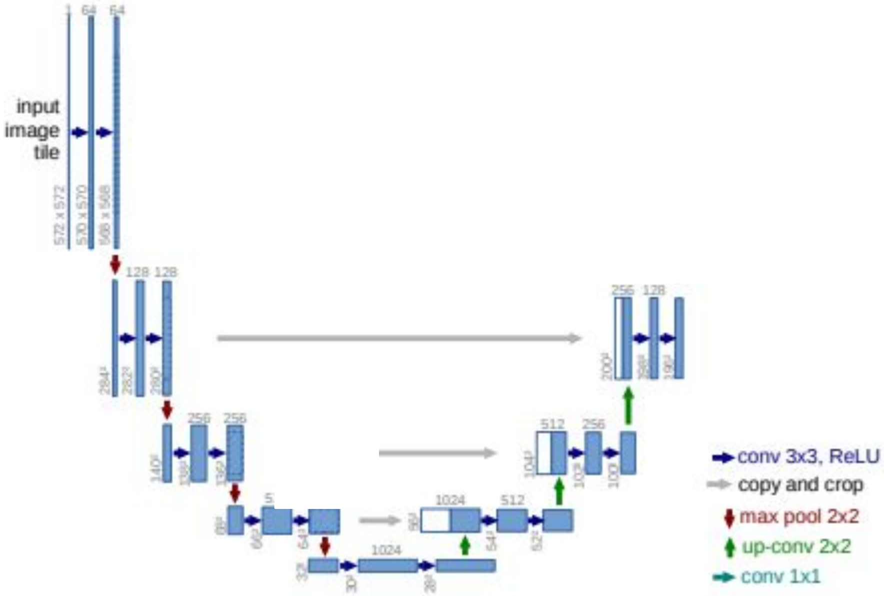


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Supervised segmentation: UNet

The final result is a segmentation map the same size as the image, that identities pixel by pixel the foreground and background of the image

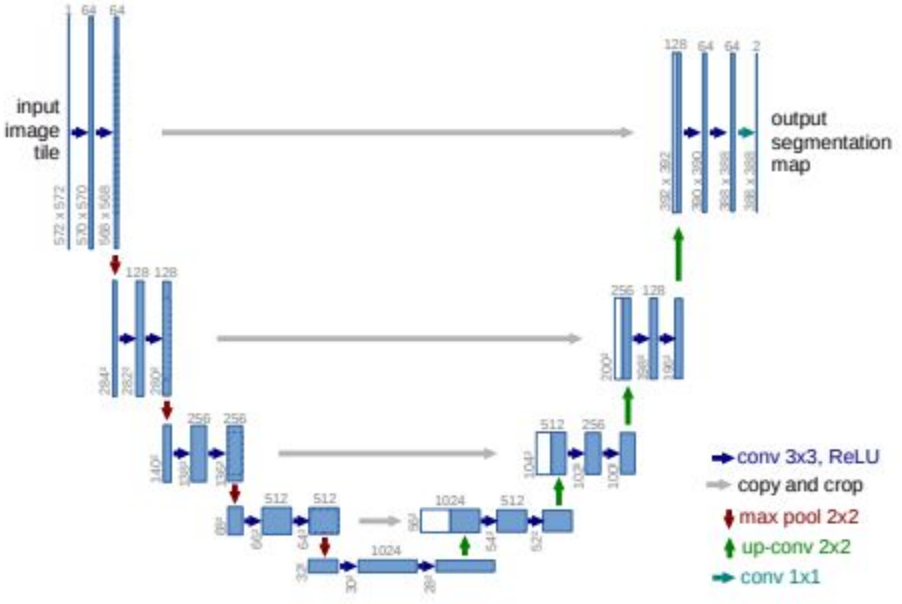


Figure 1 from https://arxiv.org/pdf/1505.04597.pdf

Unsupervised segmentation

How can we learn to detect objects of interest, say tumours in a cancer dataset, without learning from any examples of images and segmentations?

Unsupervised segmentation

How can we learn to detect objects of interest, say tumours in a cancer dataset, without learning from any examples of images and segmentations?

By reframing the problem as anomaly detection

If objects of interest aren't always present, we can study all the other examples where no objects are present and model what this looks like. Using this idea, objects of interest will be anomalies to our model

Unsupervised segmentation example:

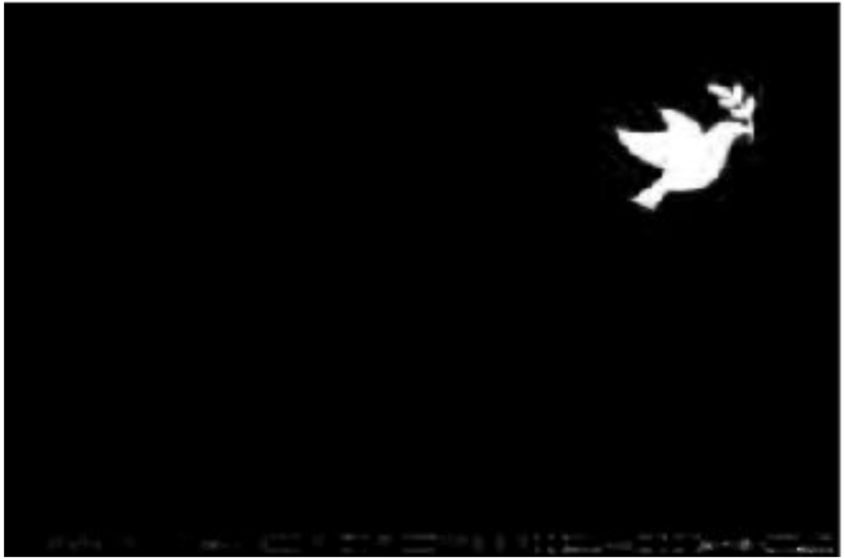
Example with no anomalies

Example with an anomaly

Unsupervised segmentation example:

By modelling only examples without anomalies, we can incorporate ways of detecting the anomalies

Example with anomaly



Detected anomaly

Evaluating a segmentation network's performance

Dice metric: popular evaluation metric, standard way to measure the accuracy of a segmentation

The dice metric measures the overlap of area between a predicted segmentation and ground truth segmentation, then normalizes this overlap by the size of both predicted and ground truth segmentations

$$Dice\ metric = \frac{2 * (\#\ overlapping\ pixels)}{(\#\ pixels\ in\ prediction) + (\#\ pixels\ in\ ground\ truth)}$$

Evaluating a segmentation network's performance

The dice metric measures the overlap of area between a predicted segmentation and ground truth segmentation, then normalizes this overlap by the size of both predicted and ground truth segmentations

$$\textit{Dice metric} = \frac{2 * (\#\ \textit{overlapping pixels})}{(\#\ \textit{pixels in prediction}) + (\#\ \textit{pixels in ground truth})}$$

Why should we normalize by the size rather than just use the overlapping area to measure accuracy?

Day 2 - Segmentation workflow

- **Visualization of the liver cancer dataset**
  - Warm-up/Activity 1: Display examples of tumors images and their segmentations from the dataset
- **Experiment with the Monai segmentation workflow:**
  - Activity 2: fill in the blanks of the training loop
  - Activity 3: add a validation step to the training loop to save the best model
- Evaluate the segmentation model:
  - Activity 4: Display the test results ie. test images, ground truth segmentation, model output)
  - Activity 5: Calculate dice score

Day 2 - Segmentation workflow

- Visualization of the liver cancer dataset
    - Activity 1: Display examples of tumors images and their segmentations from the dataset and then add a validation set in the training loop to save the best model
- **Experiment with the Monai segmentation workflow:**
    - Activity 2: fill in the blanks of the training loop
    - Activity 3: add a validation step to the training loop to save the best model
- Evaluate the segmentation model:
    - Activity 4: Display the test results ie. test images, ground truth segmentation, model output)
    - Activity 5: Calculate dice score

Day 2 - Segmentation workflow

- Visualization of the liver cancer dataset
  - Activity 1: Display examples of tumors images and their segmentations from the dataset and then add a validation set in the training loop to save the best model
- Experiment with the Monai segmentation workflow:
  - Activity 2: fill in the blanks of the training loop
  - Activity 3: add a validation step to the training loop to save the best model
- **Evaluate the segmentation model:**
  - Activity 4: Display the test results ie. test images, ground truth segmentation, model output)
  - Activity 5: Calculate dice score
  - If time, try to experiment with improving the model!