UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Image Segmentation

Lecture 14

Course of:

Signal and imaging acquisition and modelling in environment

24/04/2024

Federico De Guio - Matteo Fossati

What is Semantic Segmentation?

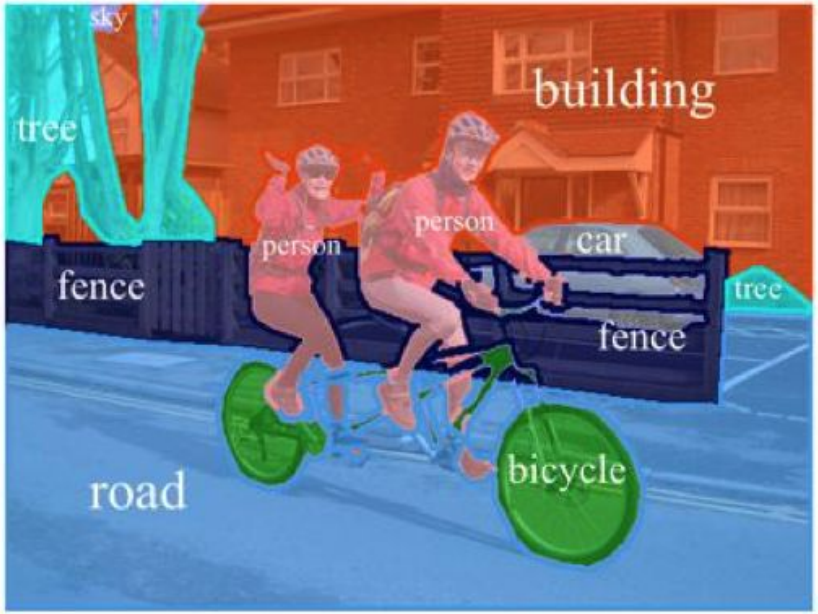
- The task of semantic segmentation is to **classify each pixel in a given image**
  - Input: images
  - Output: regions, structures → line segments, curve segments, circles, etc

- We need to process the image
  - Filters, gradient information, color information, etc.

What is the goal?

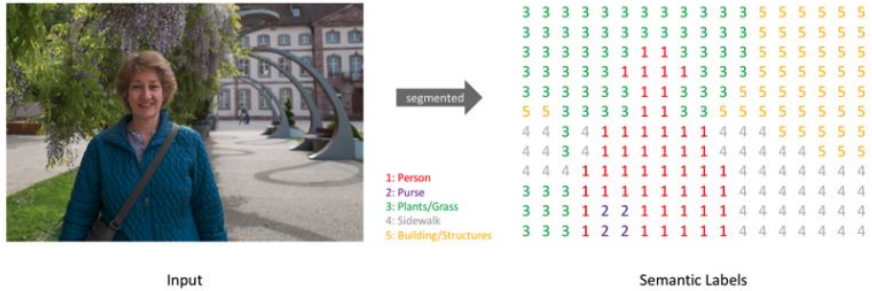
- We want to **recognize and understand what is in the image at pixel level.**

"Two persons riding on a bike in front of a building on the road. And there is a car."

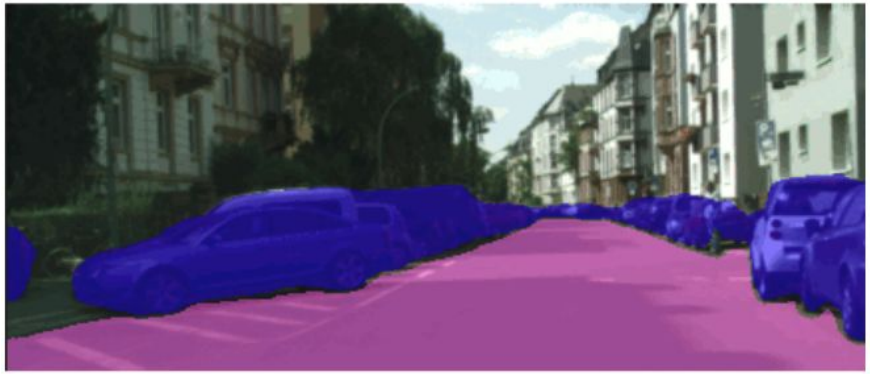


In practice..

- Take an image of size W x H x 3
- Generate a W x H matrix containing the predicted class ID's corresponding to all the pixels
  - we want to know **which pixel belongs to which entity**

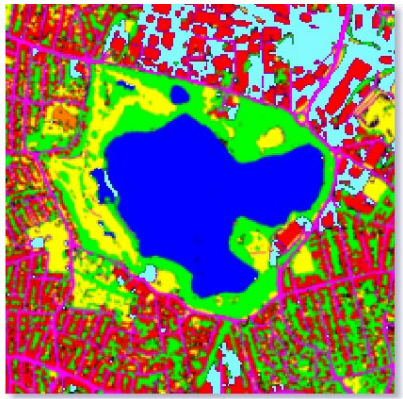


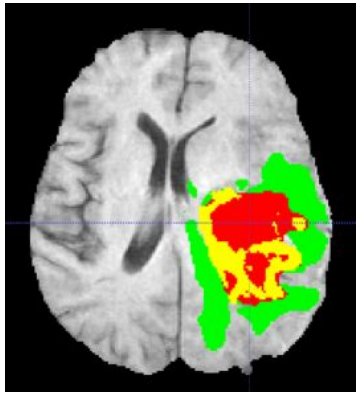
- Semantic segmentation is **different from object detection**
  - Does not predict any bounding boxes around the objects
  - It doesn't distinguish between different instances of the same object



Why is it interesting?

- Robot vision **and understanding**
  - Autonomous driving
- **Medical** purposes
  - Identify tumors, etc
- **Satellite** image analysis
  - Automated land mapping



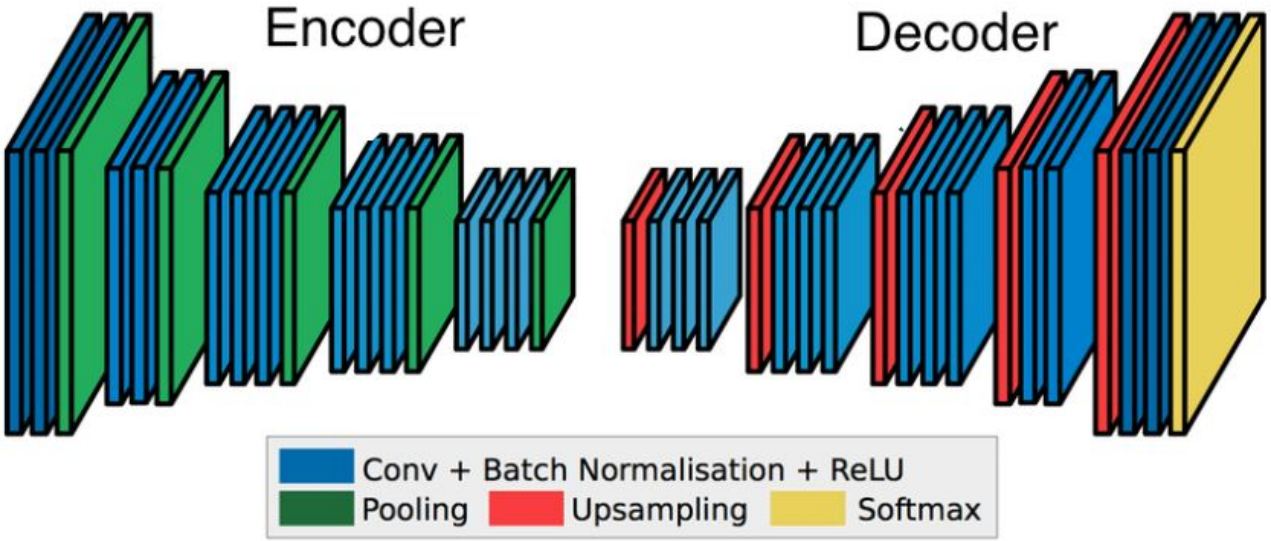


Question for you:

how would you build a NN

to perform such task?

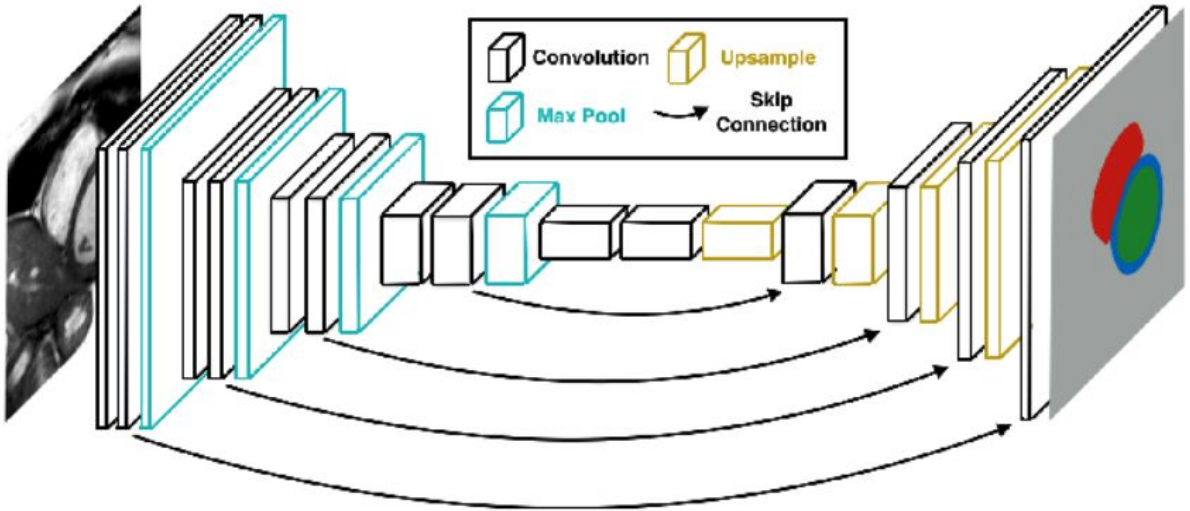
ML and image segmentation



- **Encoder - Decoder structure**
- Left half of the net: same structure used to perform classification

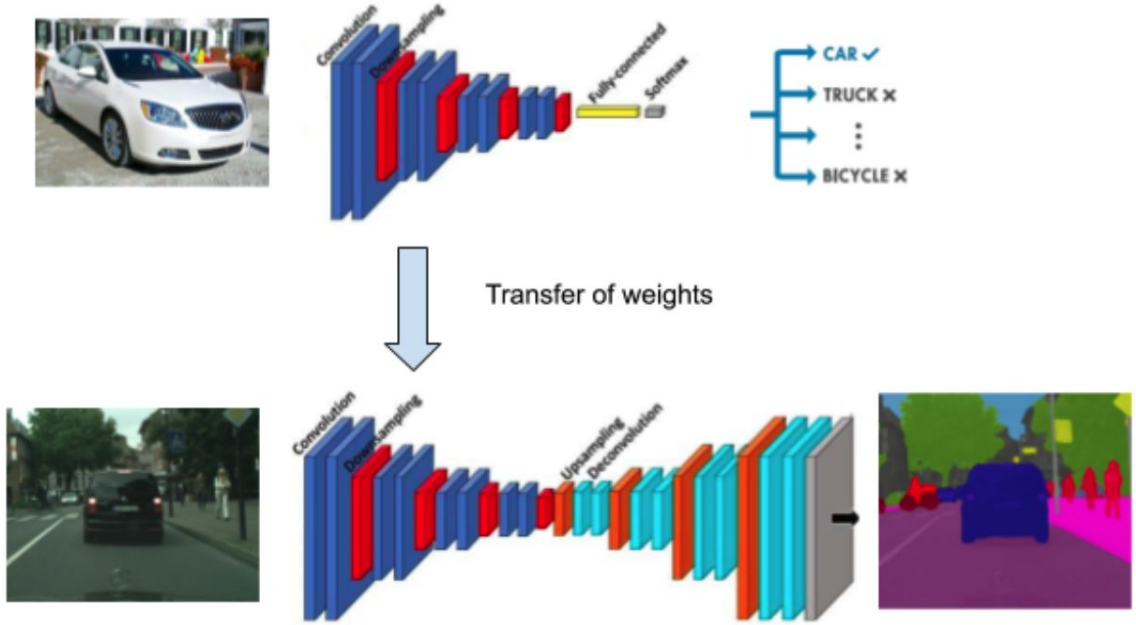
Skip connections

- The encode-decode operation may result in a **loss of low-level information**
  - Results in not accurate boundaries in the output image
- As a solution, **let the decoder access the low-level features** produced by the encoder layers
  - Intermediate outputs of the encoder are added/concatenated with the inputs to the intermediate layers of the decoder at appropriate positions.



Transfer learning

- Models trained for image classification contain meaningful information that can be re-used in segmentation
  - Using Resnet or VGG pre-trained on ImageNet dataset is a popular choice
  - A note on transfer learning here



Some popular segmentation models: FCN

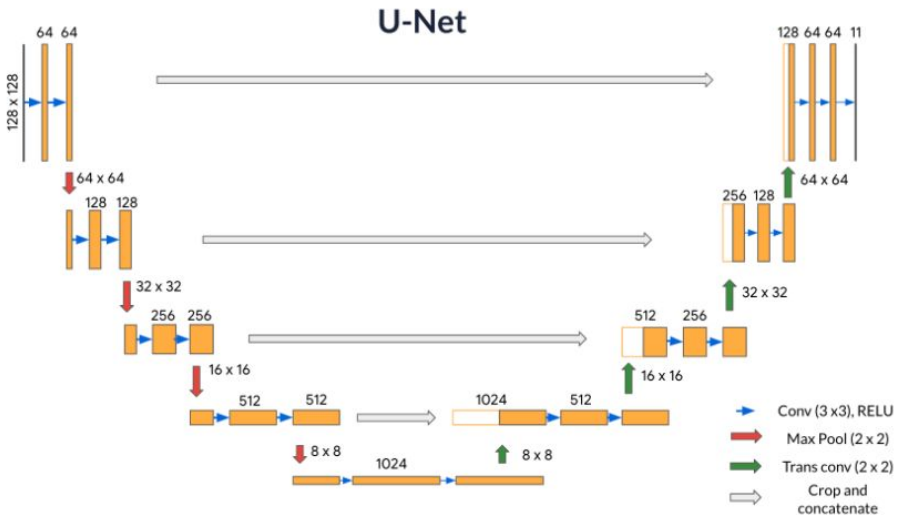
Some popular segmentation models: Xception

Some popular segmentation models: SegNet

Some popular segmentation models: ResNet

A popular segmentation models: UNet

- The UNet was designed in 2015 to process biomedical images
  - Classify whether there is a disease
  - Localize the area of abnormality → input and output have the same size



UNet: downsampling block

3.2 - Encoder (Downsampling Block)

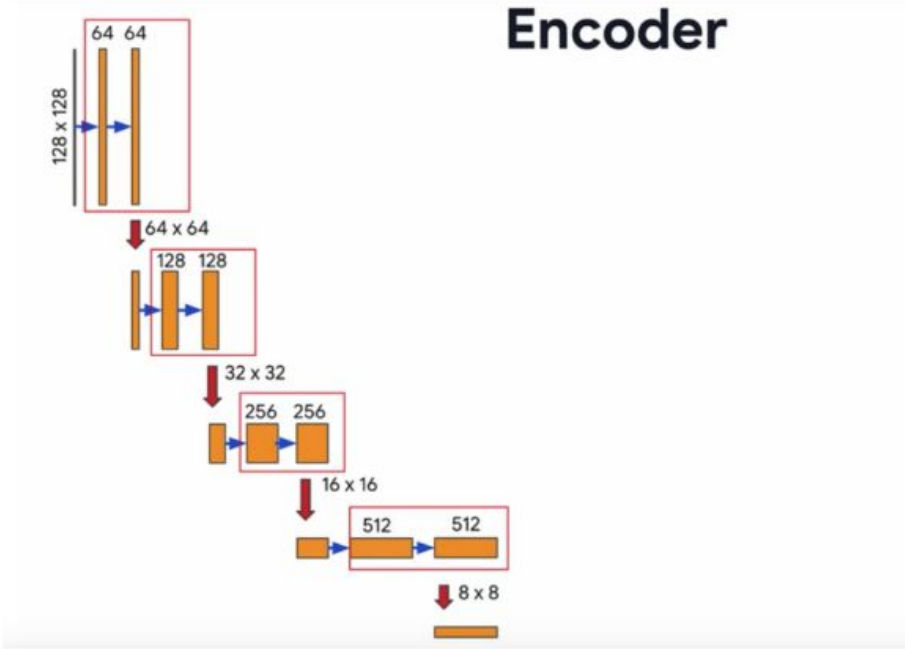


Figure 3: The U-Net Encoder up close

UNet: upsampling block

3.3 - Decoder (Upsampling Block)

The decoder, or upsampling block, upsamples the features back to the original image size. At each upsampling level, you'll take the output of the corresponding encoder block and concatenate it before feeding to the next decoder block.

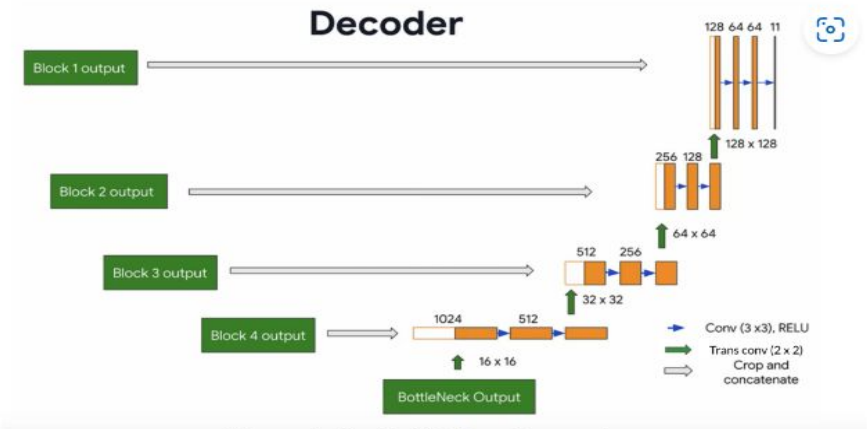


Figure 4: The U-Net Decoder up close

An example:

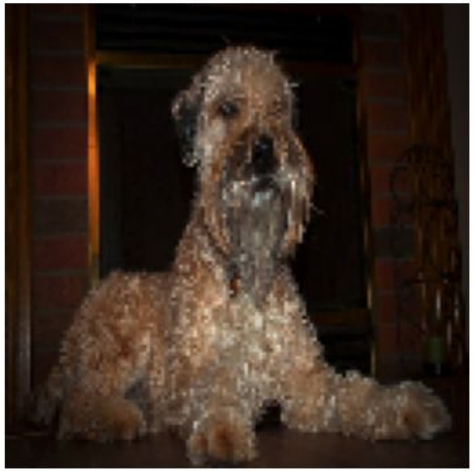
Segmentation on pets images

Perform segmentation on a dataset available from TF

- The UNet architecture is used in this case. Find the full example here.

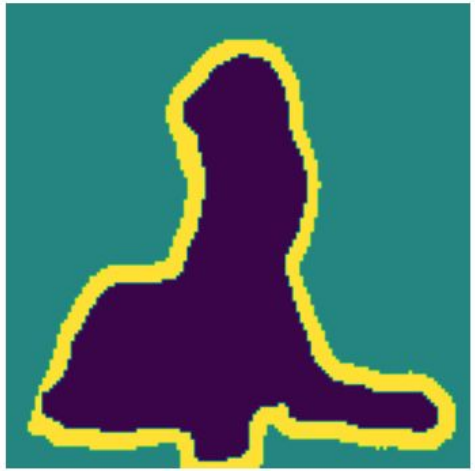
- Basically an RGB classification is performed pixel by pixel
  - 34M of parameters overall → they correspond to the pixels of the input image
  - Very quick convergence: 90% accuracy after 20 epochs

Input Image



True Mask

Predicted Mask



Your turn

Decide where to land your drone

- 400 total images are available here
  - Some picture has been discarded, but there is correspondence in the folders

- The idea is to build a network to perform segmentation on the images
  - UNet is a good choice, but different architectures can be tested and compared (FCN, SegNet, ResNet, …)
  - The number of layers and parameters is something you have to tune
  - 90% accuracy can be achieved with 150 epochs, training is reasonably fast

- Once the segmentation is performed, use an algorithm or ML to spot where to land your drone
  - Far away from trees, houses, etc

- Suggestion:
  - The GPU memory on CoLab is 15GB. Minimizing the I/O helps during the training phase
  - Use the tensorflow dataset following the pets example to load the data in batches on the GPU
  - Reduce the image size and normalize the masks

Data preparation