أكاديمية كاوست
KAUST ACADEMY

Day 3 (part 1): Computer Vision Summary

Note: Do Not depend entirely on this and study from the official slides.

Contributed by: Hassan Mohammed Nasr

Image Segmentation

Task	Image classification	Image segmentation
• Goal	Identify the main/dominant object	Boundaries of all objects and background
• Output	Single label for the while image	Pixel level map, every pixel gets a class
• Structure	Input image --> output ingle label (downsamples image into vector)	Input image --> output image (reserves dimension)

- Types

1 - Semantic Segmentation

- Label class categories only, doesn't distinguish between objects
- Example: cats are colored blue, background red

2 - Instance Segmentation

- Distinguish separate instances of the same class
- Example: Cat 1 is blue cat 2 is red

3- Panoptic Segmentation

- Semantic + instance
- example: color different instances plus including the background also

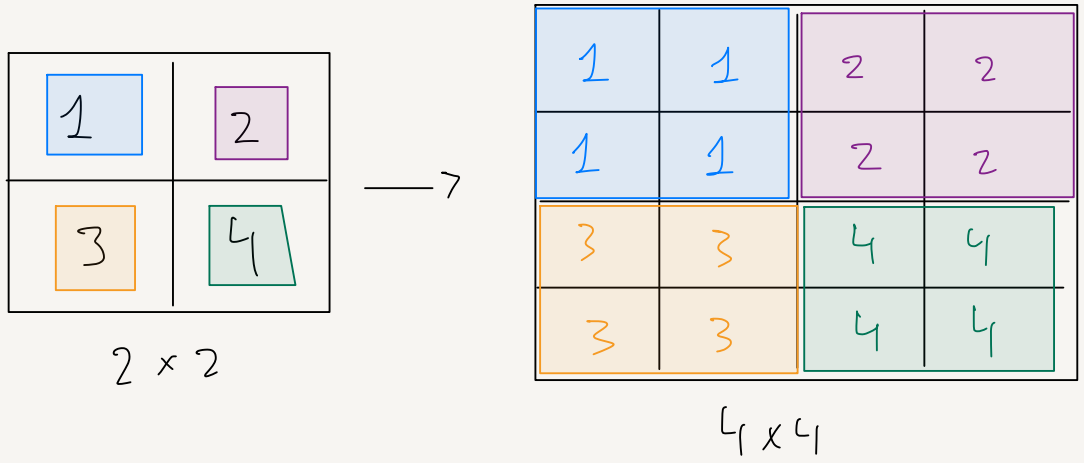
The Architecture Challenge

- Normal CNNs shrink images , while Segmentation needs the same HxW
  - **Problem**: pooling layers reduce spatial resolution. We loose the fine details and size needed for segmentation and doing standard CNNs with original image resolution will be very expensive
  - **Solution** : Use Encoder - Decoder Architecture
    - Encoder (Downsampling): Extract features (normal cnn architecture)
    - Decoder (Upsampling) : Recovers spatial resolution to match input size (opposite of conv)

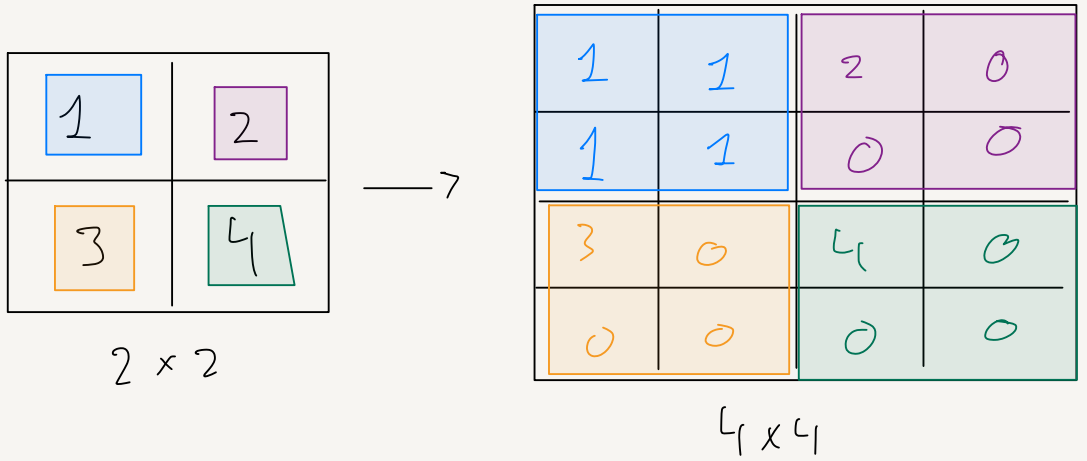
Upsampling techniques

Method	Description	Learnable?
Nearest Neighbor	Duplicate values to fill the larger grid	No
Bed of nails	Place each value in specific place and fills the rest with zeros	No
Max Unpooling	Smart memory: Remembers the index of the max value from the pooling step and puts the value back exactly to preserve spatial details	No
Transpose Convolution	Gold Standard: A reverse convolution with learnable weights that learns how to optimally upsample	Yes 👏

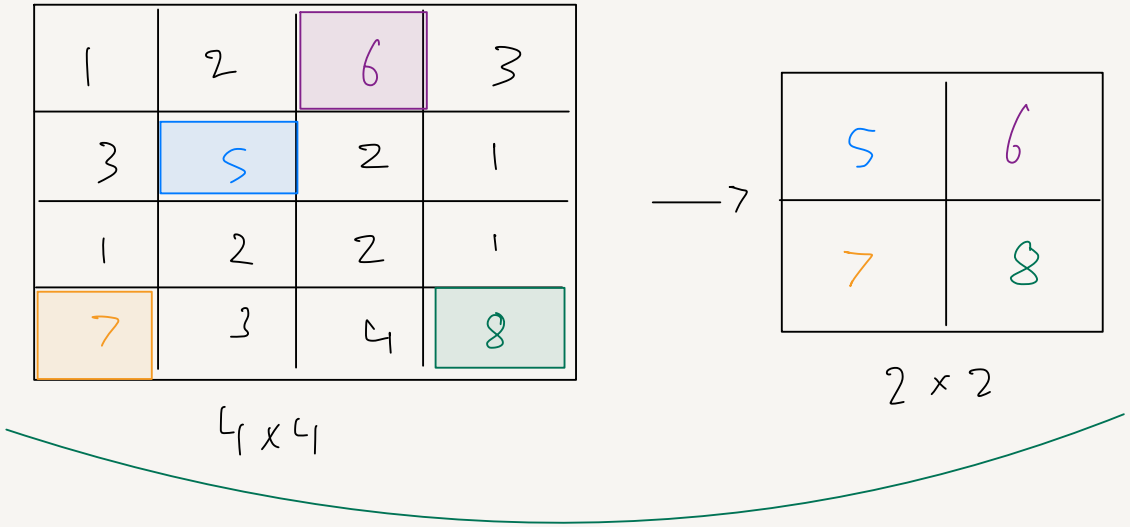
- Nearest Neighbor



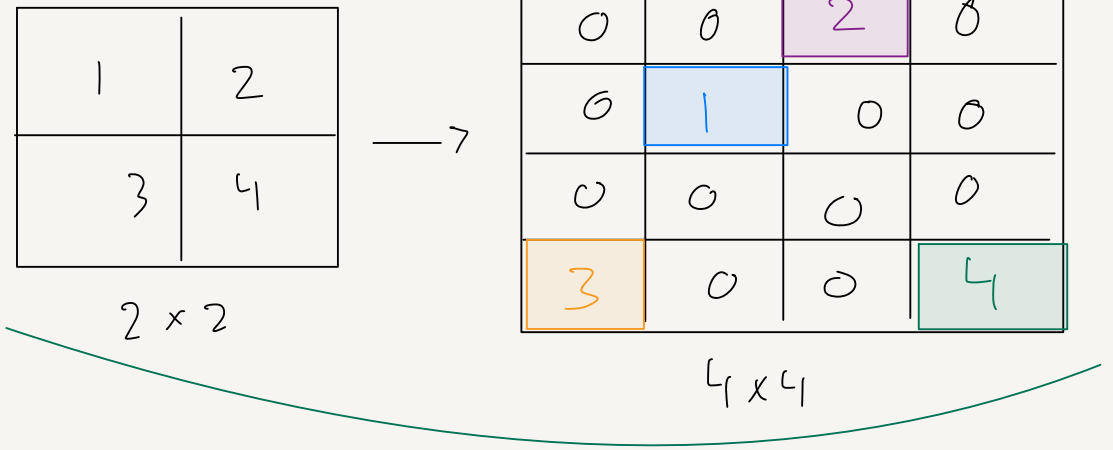
- Bed of nails



- Max Unpooling



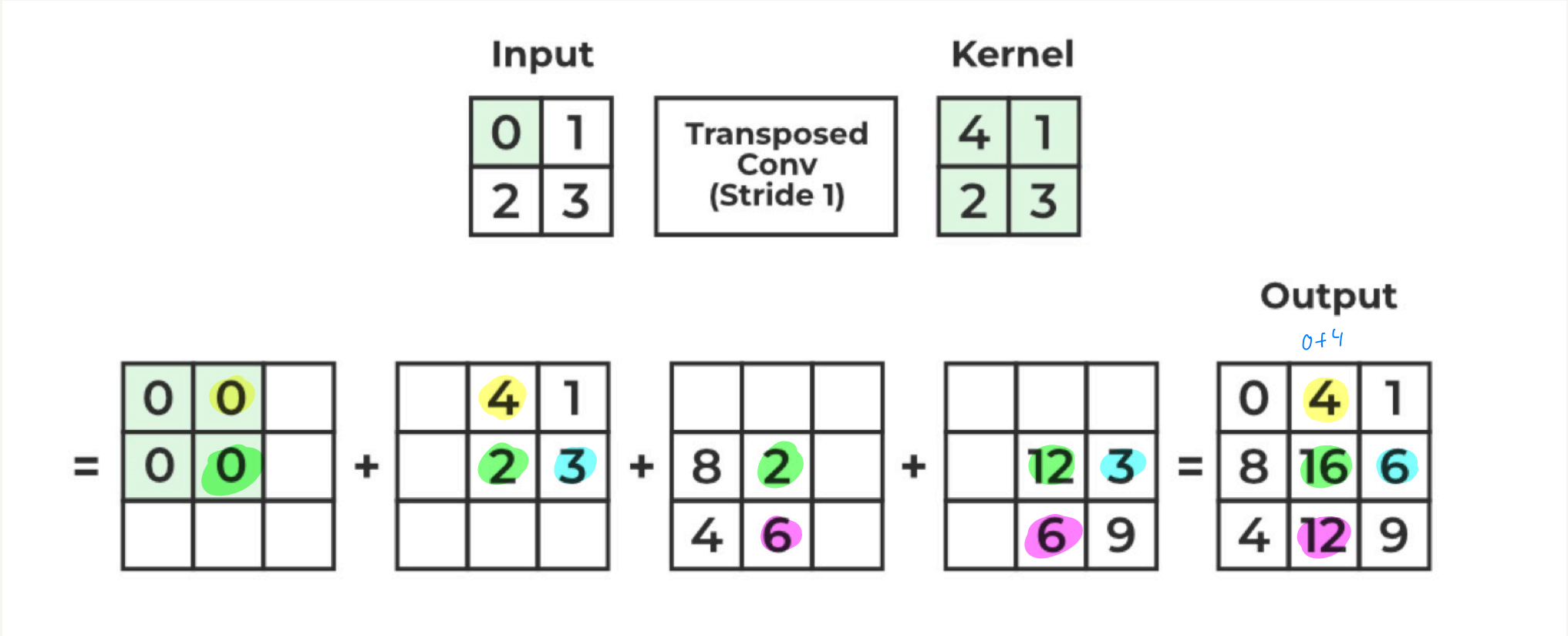
- max pooling



- Max Unpooling

Transposed Convolution (Learnable upsampling)

- The 'opposite' of Normal Convolution
  - Concept : Instead of shrinking the image (downsample), it grows the image (upsampling) using learnable weights.
    - Normal Conv. : put kernel on INPUT - - - > Do dot product - - - > get 1 output (many to one)
    - Transposed Conv. : put kernel on OUTPUT - - - > use input as scaler - - - > copy the scaled kernel to output (one to many)
  - The Mechanism: Each pixel in the input act as a multiplier. It scales the entire kernel, and we stamp that scaled kernel onto the output feature map. Overlapping stamps are summed up



U-Net Architecture

- Problem: Important details and spatial information may be lost during Downsampling

  Solution (U-Net): Introduce Residual connections (skip connections) to preserve spatial information

  Directly connect features from Downsampling layers to upsampling layers as it helps to recover from lost spatial details and improve segmentation accuracy

- U Net structure: Encoder decoder shaped like U
  - Left Side. : Standard CNN plus max pooling to capture context
  - Right Side. : Transposed Conv to recover precise location
- Secret Recipe: Residuals (skip connections)

  Skip connections from the Downsampling to the up sampling: high resolution feature maps from encoder to the corresponding layer in the decoder. The effect is that the decoder gets 'gps coordinates' spatial info from the encoder