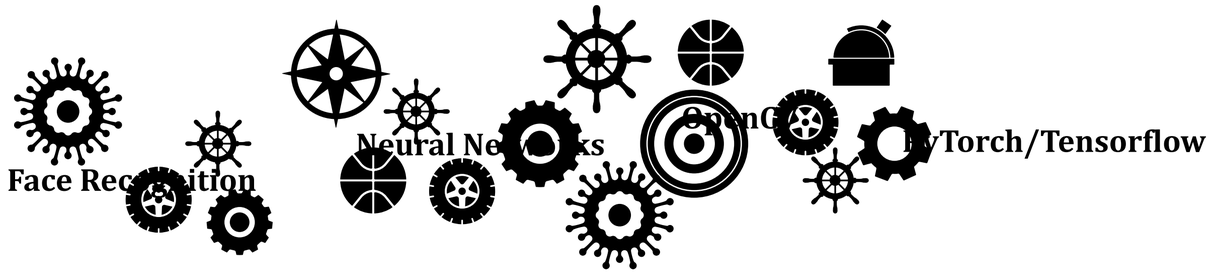
MPSSDEGB
कौशल से रोजगार, समृद्धि अपार

FOUNDATION FOR INNOVATION AND TECHNOLOGY TRANSFER
भारतीय प्रौद्योगिकी संस्थान दिल्ली
Indian Institute of Technology Delhi

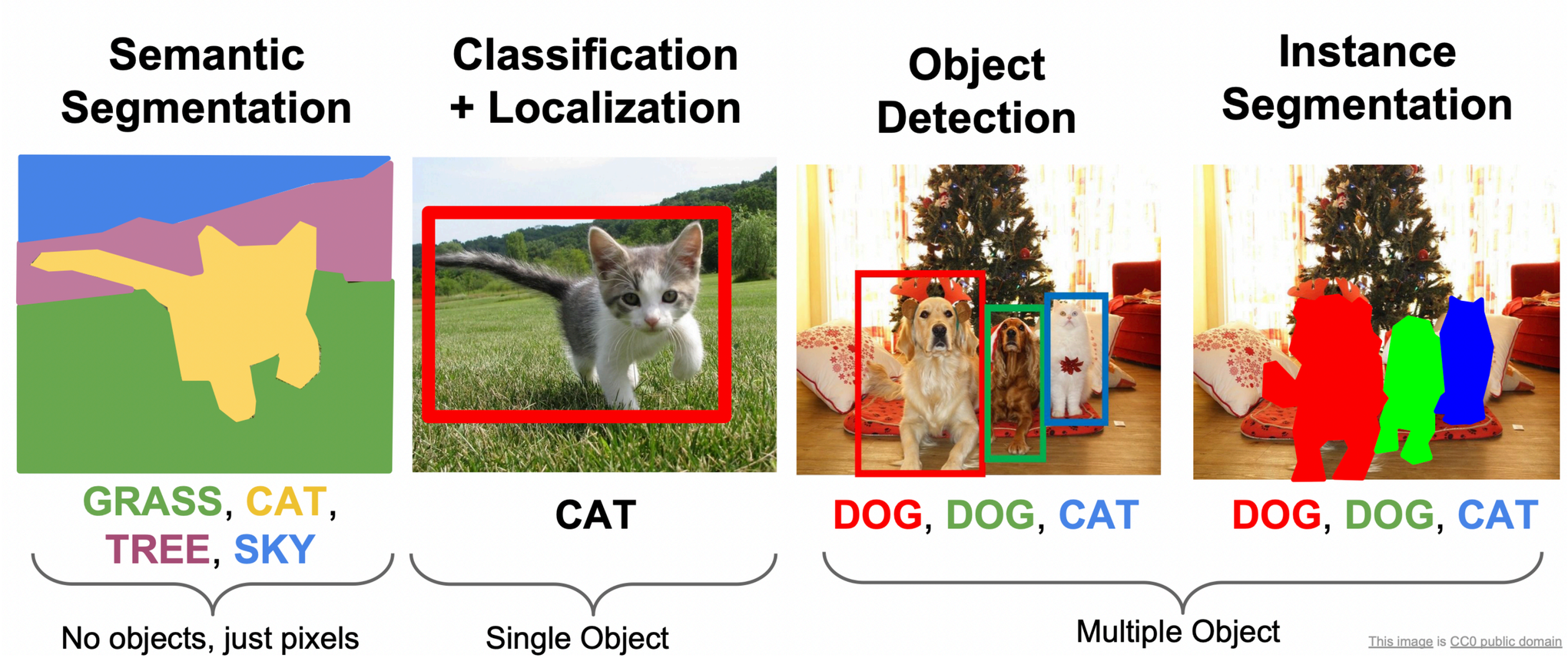
Computer Vision



Deep Learning
Segmentation
Object Detection
Image Classification

The content is added from different sources

Semantic Segmentation



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

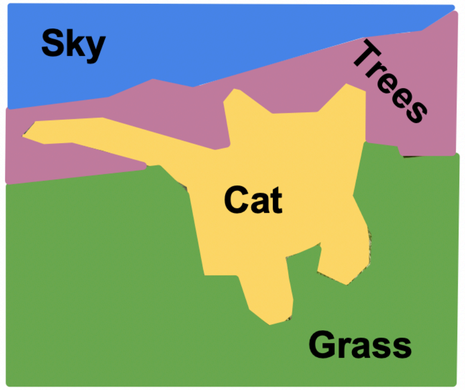
Semantic Segmentation

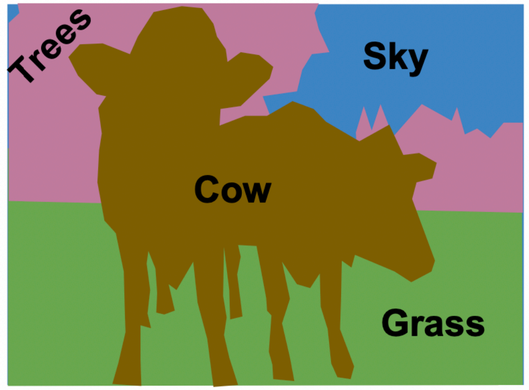
Semantic Segmentation

Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels

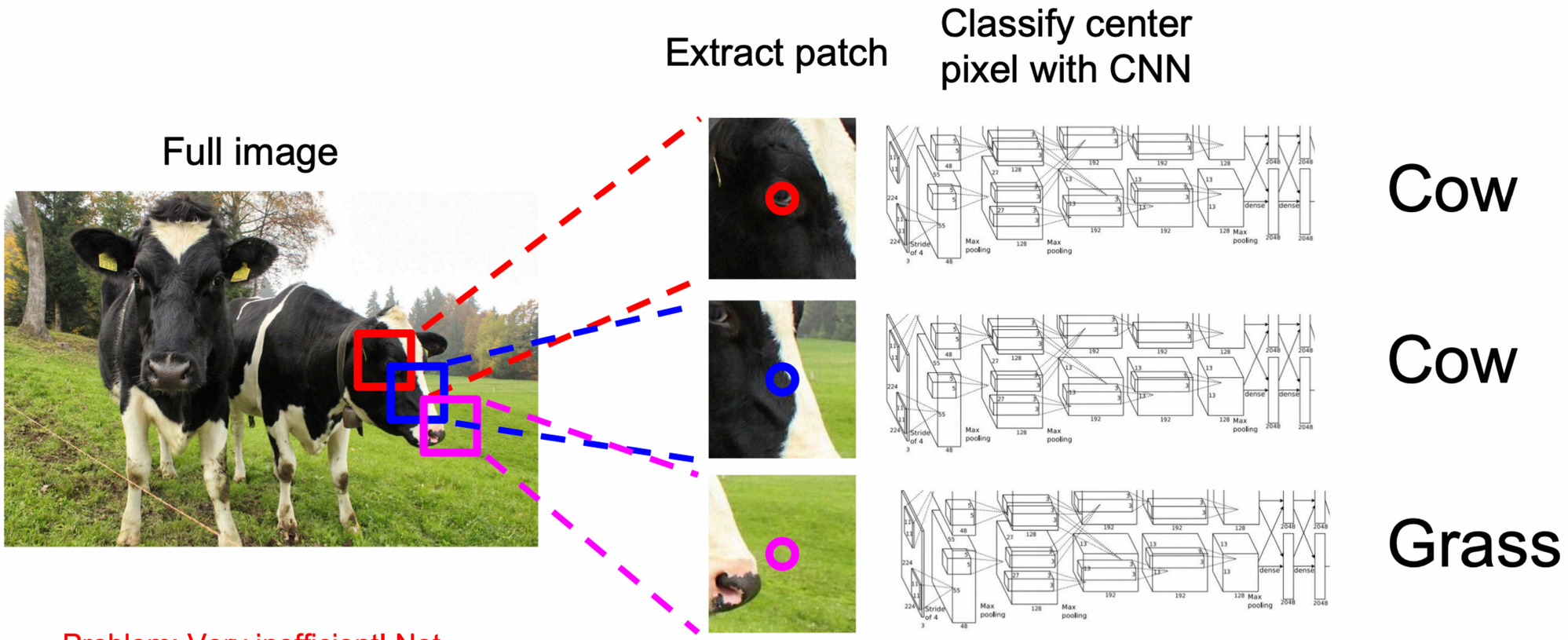
This image is CC0 public domain





http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation Idea: Sliding Window



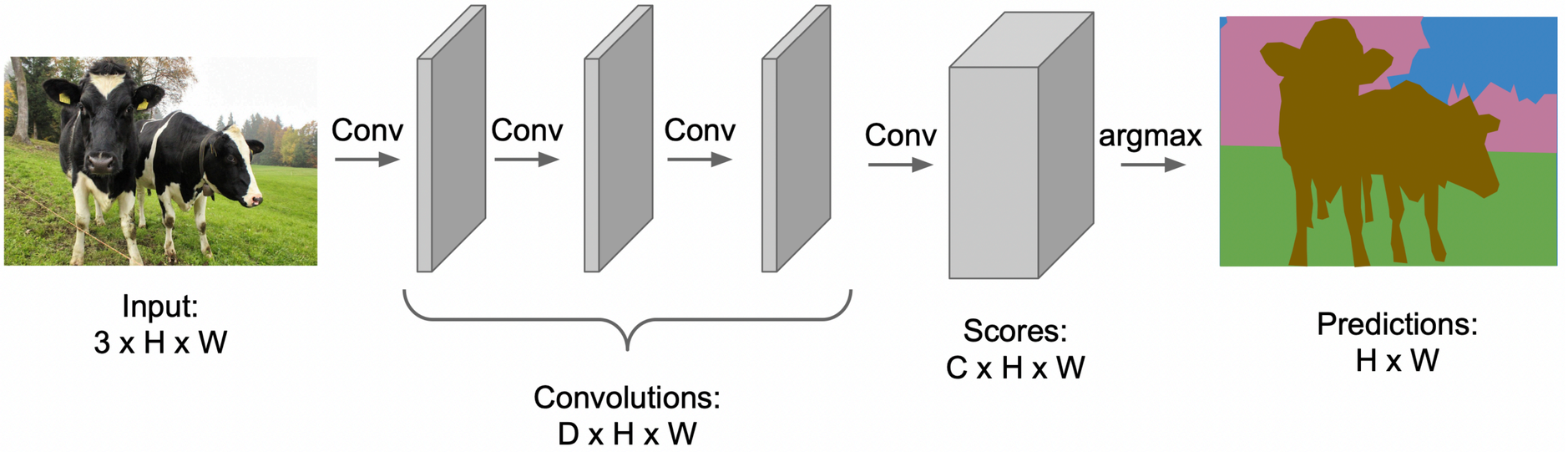
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



Problem: convolutions at original image resolution will be very expensive ...

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation

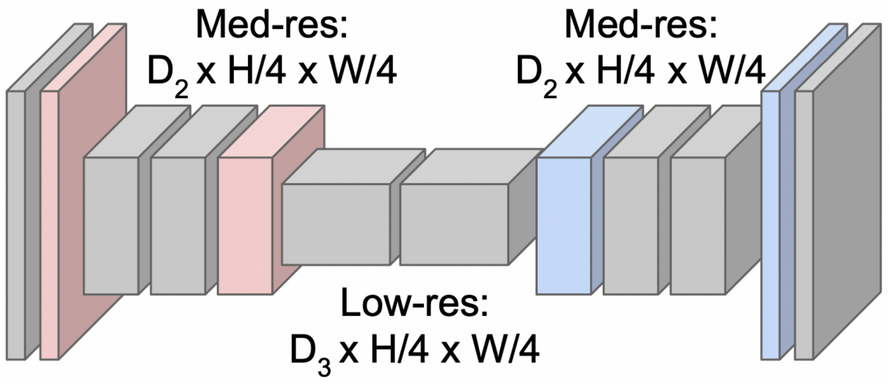
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
???

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

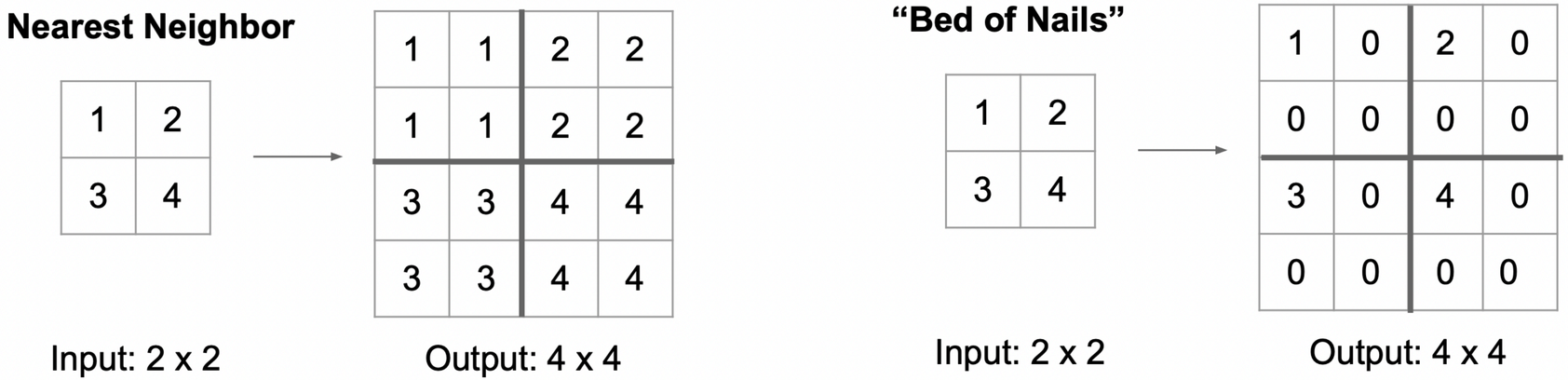
Predictions:
H x W

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation

In-Network upsampling: “Unpooling”



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation

In-Network upsampling: “Max Unpooling”

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

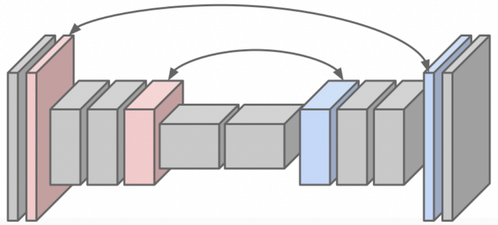
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

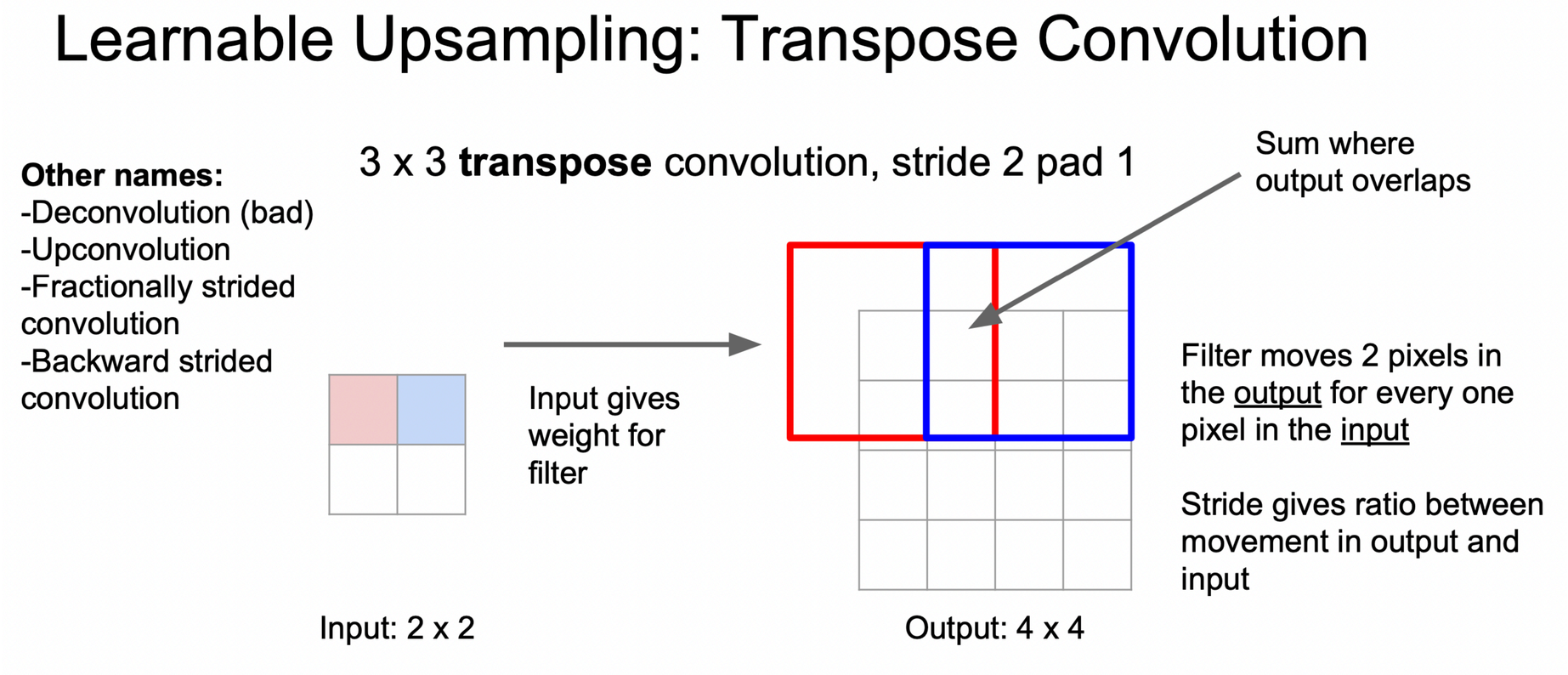
Corresponding pairs of downsampling and upsampling layers



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation



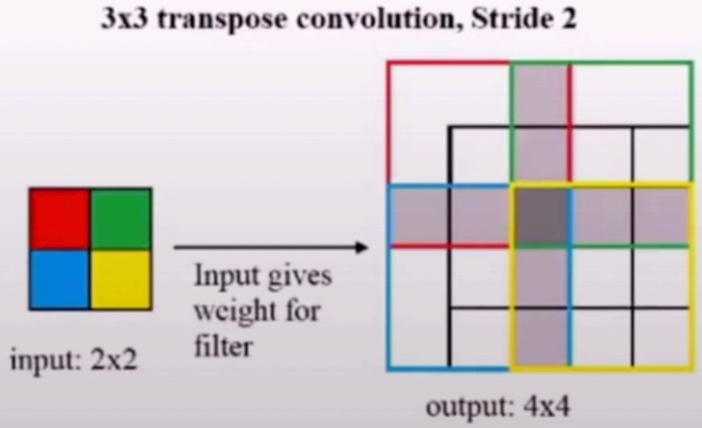
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The content is added from different sources

Semantic Segmentation

Upsampling Techniques: Checkerboard Artifacts

- Transpose convolution has uneven overlap when the kernel size is not divisible by the stride
- The uneven overlaps on the two axes multiply, creating a characteristic checkerboard artifact
- In principle, network could learn weights to avoid this but ...
- ... in practice, networks struggle to avoid it completely



The content is added from different sources

Semantic Segmentation

Checkerboard Artifacts

How to avoid

- Choose appropriate kernel size: use a kernel size that is divisible by the stride, avoiding the overlap issue
- Separate upsampling from convolution to compute features
- For example:
  - Resize the image (e.g., using NN or bilinear interpolation)
  - Add a convolutional layer

The content is added from different sources

Semantic Segmentation

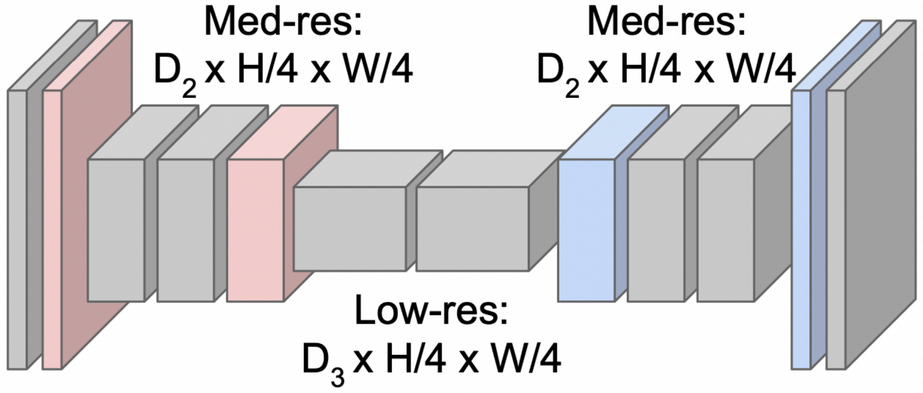
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



High-res:
D1 x H/2 x W/2

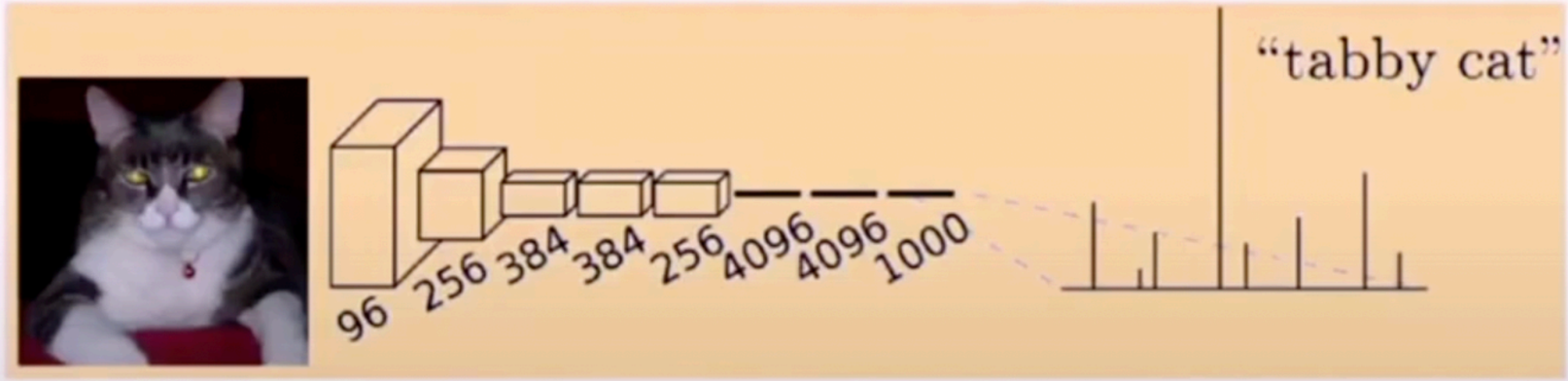
High-res:
D1 x H/2 x W/2

Predictions:
H x W

The content is added from different sources

Semantic Segmentation

- Convolutional neural networks with **fully connected layers** used for classification:
  - CNN layers for feature extraction
  - Output dimensions reduced because of subsampling
  - **Fully connected layers** have fixed inputs and remove spatial information
  - Output: vector encoding class probabilities



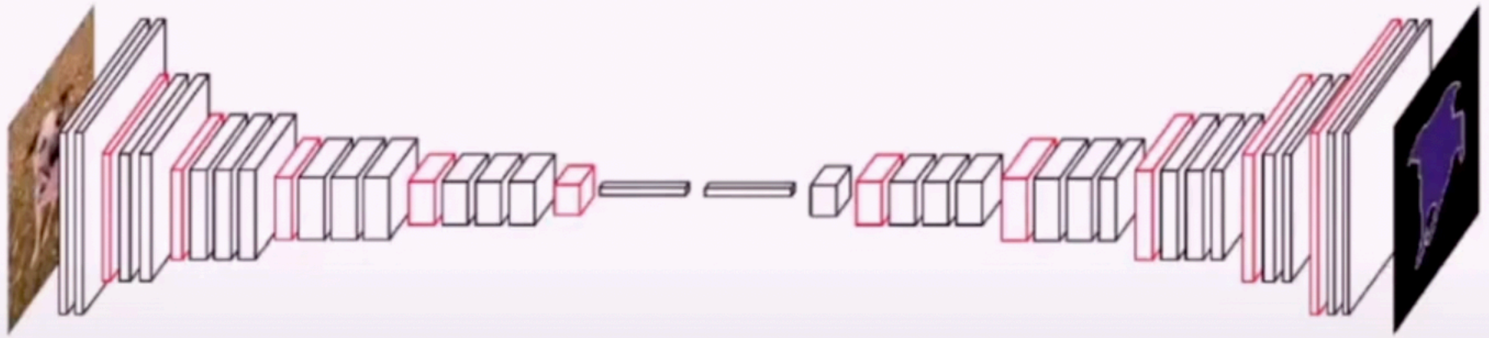
The content is added from different sources

Semantic Segmentation

Encoder and Decoder

- Most methods use a classification network as basis

➔ If it can pickup the class, it should be able to learn the correct features

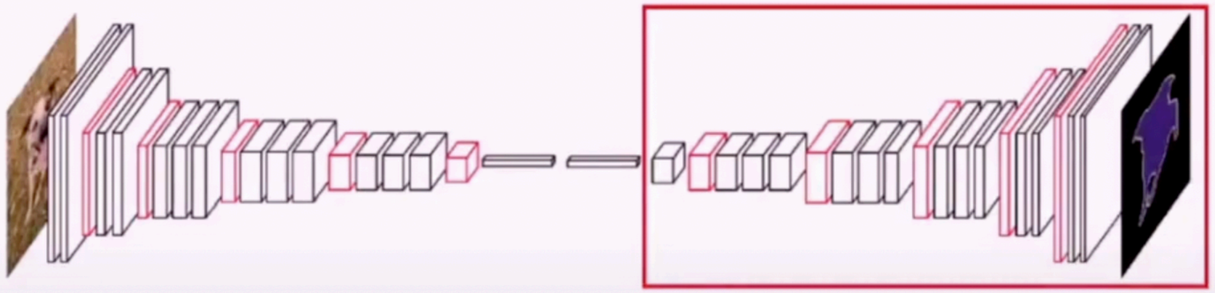


The content is added from different sources

Semantic Segmentation

Encoder and Decoder (cont.)

- Network has to learn to **decode** (map) the output of the encoder to pixel-wise predictions
- Decoder part of the network



Example networks with different decoders:

- Long et al.'s Fully Convolutional Networks [13]
- SegNet [1]
- U-net [21]

Source: modified from: Long et al. 2015

The content is added from different sources

Semantic Segmentation

Integrating Context Knowledge: Combining Where and What

- Semantic segmentation requires integration of information from various spatial scales

→ Need to balance local and global information

- Local information crucial to achieve good pixel-level accuracy
- Global context of the image enables to resolve local ambiguities
- CNNs struggle with this balance

→ Different approaches to integrate local & global information

The content is added from different sources

Semantic Segmentation

Long et al.'s Fully Convolutional Networks [13]

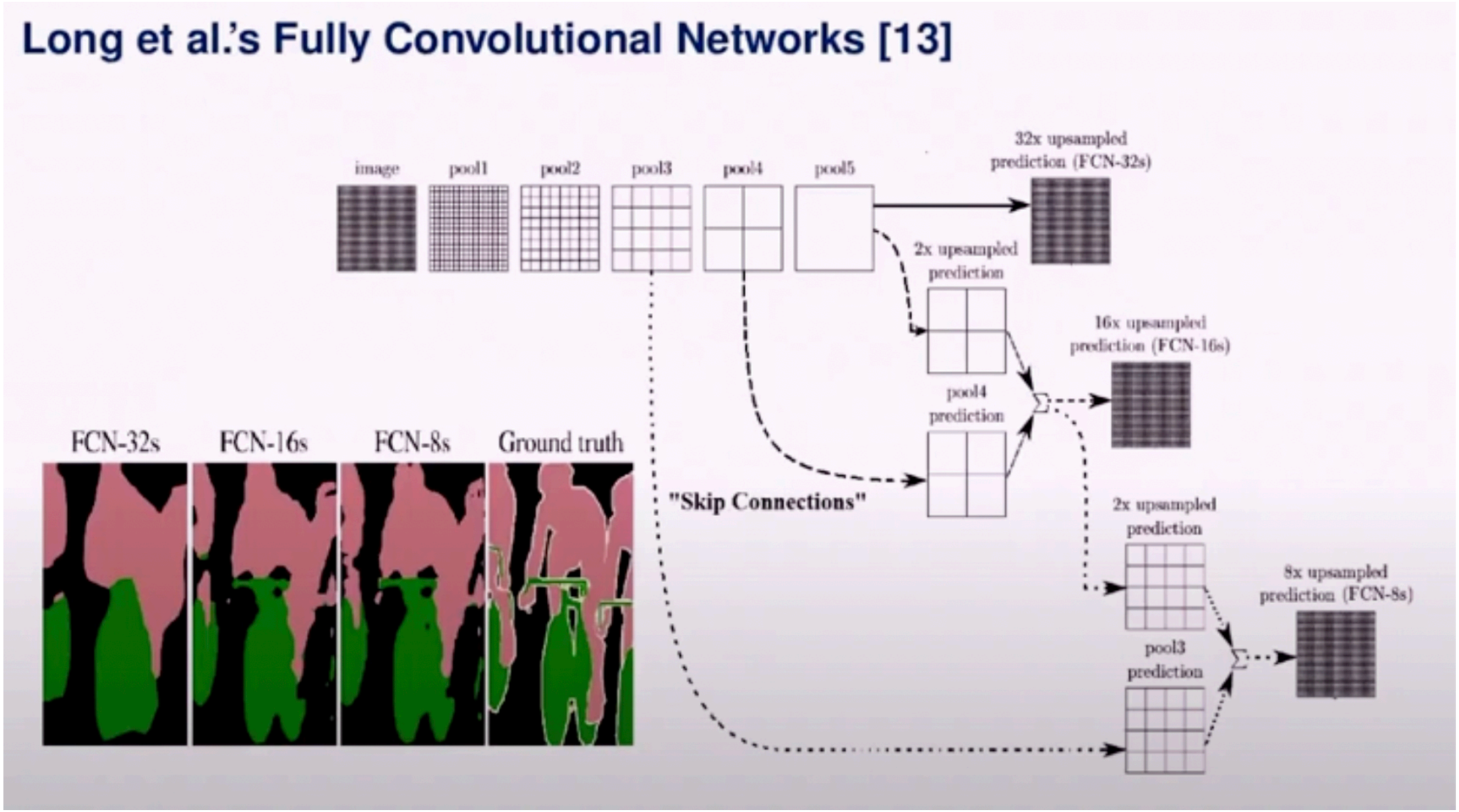
- Upsampling using learnable transposed convolutions
- Idea: Add links combining the final prediction and previous (lower) layers with finer strides
- Additional 1×1 convolution after pooling layer, predictions are added up

→ Make local predictions with global structure

- Network topology is a directed acyclic graph (DAG), with **"skip connections"** from lower to higher layers
- Refinement of coarse segmentation

The content is added from different sources

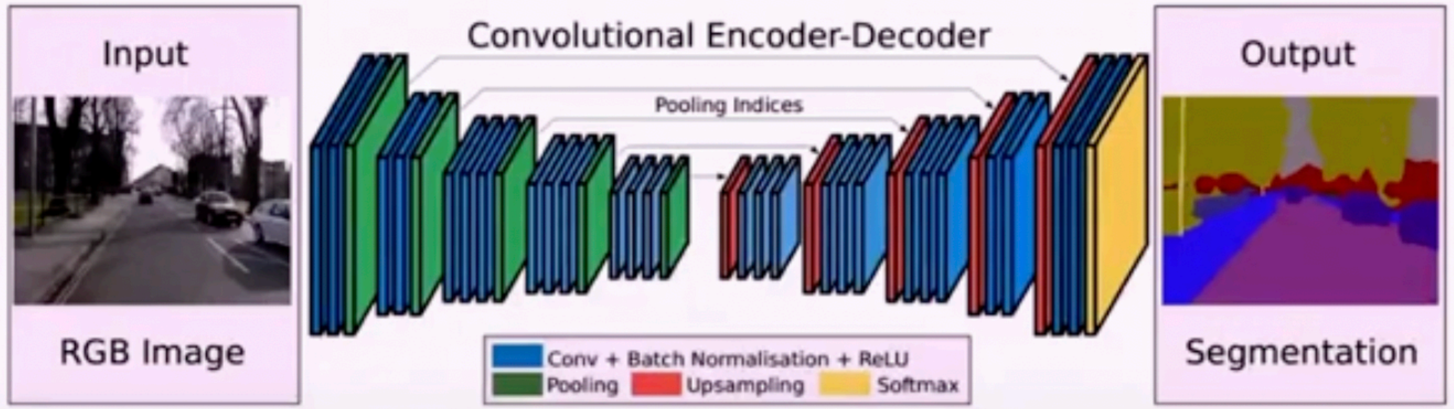
Semantic Segmentation



The content is added from different sources

Semantic Segmentation

SegNet [1]:



- Decoder: Upsampling & convolution layers followed by softmax
- Each upsampling layer corresponds to a **max-pooling layer** in encoder
- Upsampling reuses max-pooling indices from feature maps in encoder

→ provides context information

The content is added from different sources

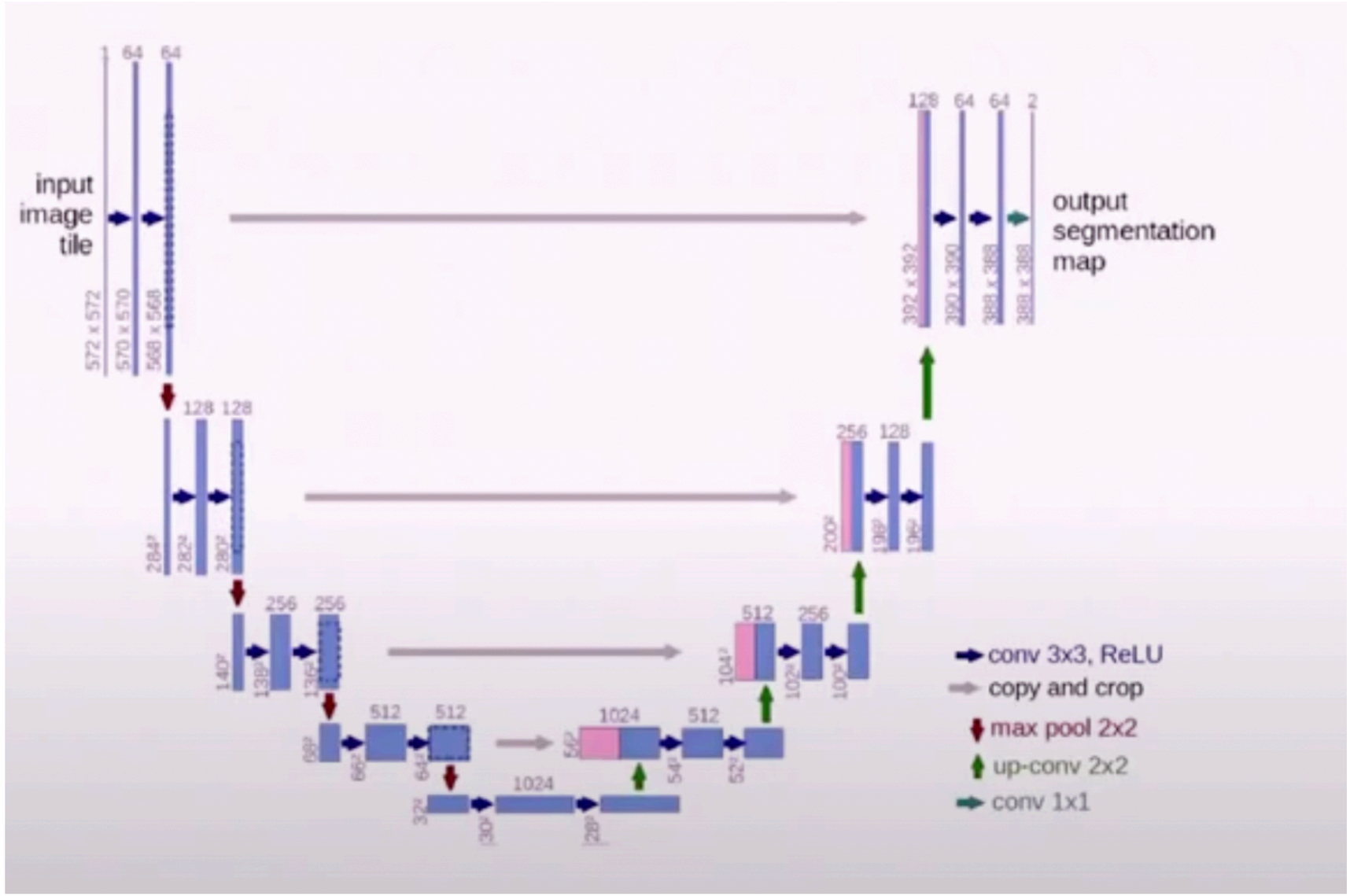
Semantic Segmentation

U-Net [21]

- The network consists of:
  - Encoder: A contracting path to capture context
  - Decoder: A **symmetric** expansion path for localization (decoder)
- Encoder follows typical architecture of a CNN
- Decoder (expansion path) consists of
  - **Upsampling of feature maps** followed by a 2×2 convolution that halves the number of feature channels.
  - **Concatenation** with the corresponding cropped feature map from the contracting path
- The training strategy relies on use of data augmentation
  - Non-rigid deformation
  - Rotation & translation

The content is added from different sources

Semantic Segmentation



Semantic Segmentation

So far: Combination of feature maps from different levels + upsampling

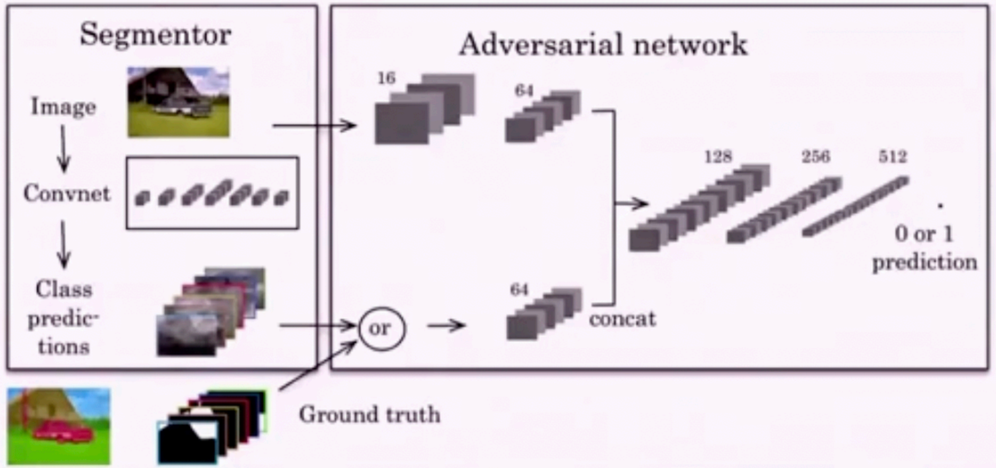
Alternatives & extensions:

- Dilated convolutions
- Network stacks
- Multi-scale networks
- Defer context modeling to another network, e.g. an RNN
- Refinement as a post-processing step with Conditional Random Fields (CRFs)

The content is added from different sources

Semantic Segmentation

Adversarial Networks for Segmentation [14]



Source: Luc et al., 2016

Optimize Segmentor with hybrid loss function with two terms:

1.) Usual pixel-wise multi-class cross-entropy for semantic segmentation

2.) Loss based on min-max game with Discriminator

Semantic Segmentation

- Given a data set of N training images $\mathbf{x}_n$ and a corresponding label maps $\mathbf{y}_n$ the loss function defined as:

$$l(\theta_s, \theta_a) = \sum_{n=1}^{N} \underbrace{l_{mce}(s(\mathbf{x}_n), \mathbf{y}_n)}_{\text{multi-class cross-entropy}} - \lambda \underbrace{\left[l_{bce}(a(\mathbf{x}_n, \mathbf{y}_n), 1) + l_{bce}(a(\mathbf{x}_n, s(\mathbf{x}_n)), 0)\right]}_{\text{adverserial loss} \rightarrow \text{binary classification}}$$

- $(\mathbf{x}_n, \mathbf{y}_n), n \in \{1, \ldots, N\}$: Data set with ground truth labels
- θ_s: Parameters of the Segmentor $s(\mathbf{x}, \mathbf{y})$
- θ_a: Parameters of the Discriminator $a(\mathbf{x}, \mathbf{y})$

→ Segmentor is trained both on the segmentation and on fooling the discriminator

→ **Multi-task learning** with adversarial task