UNIVERSITÄT BONN

U-Nets for medical Image-Segmentation

Lokesh Veeramacheneni, Moritz Wolter

September 25, 2024

High-Performance Computing and Analytics Lab

Overview

Why do we need segmentation?

Semantic Segmentation

Semantic Segmentation Architectures

- Encoder-Decoder
- U-Net

Focal Loss

Evaluation Metric - Intersection-over-Union

Why do we need segmentation?

Classification

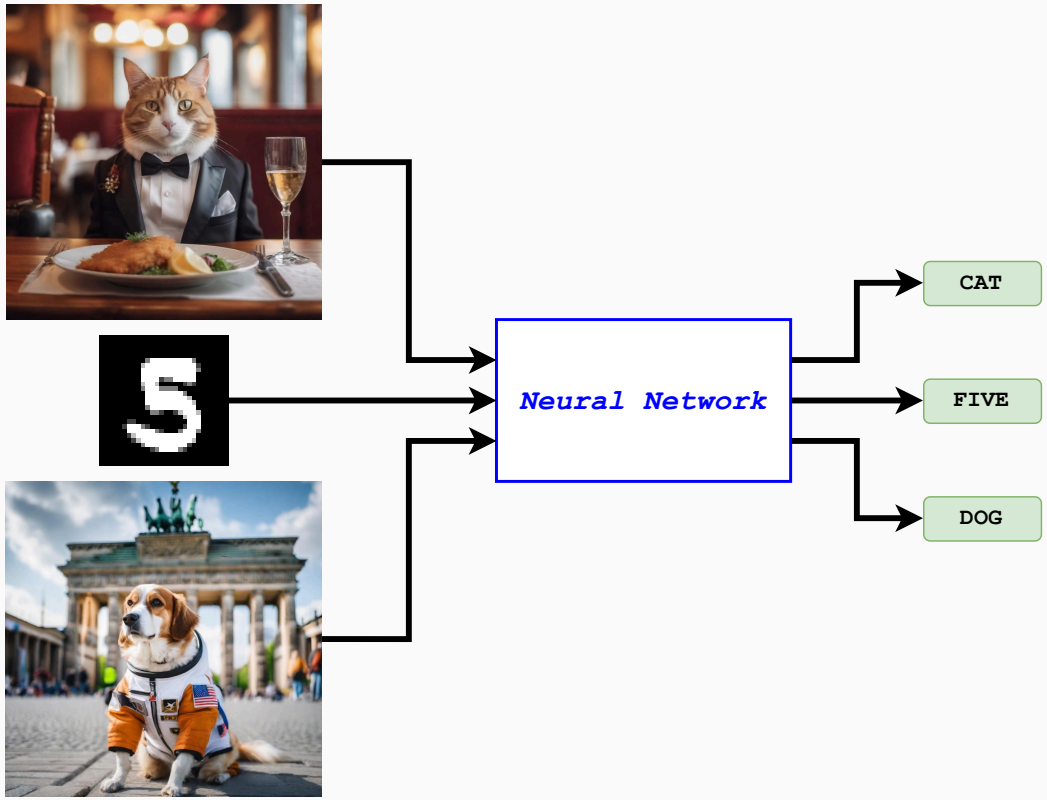
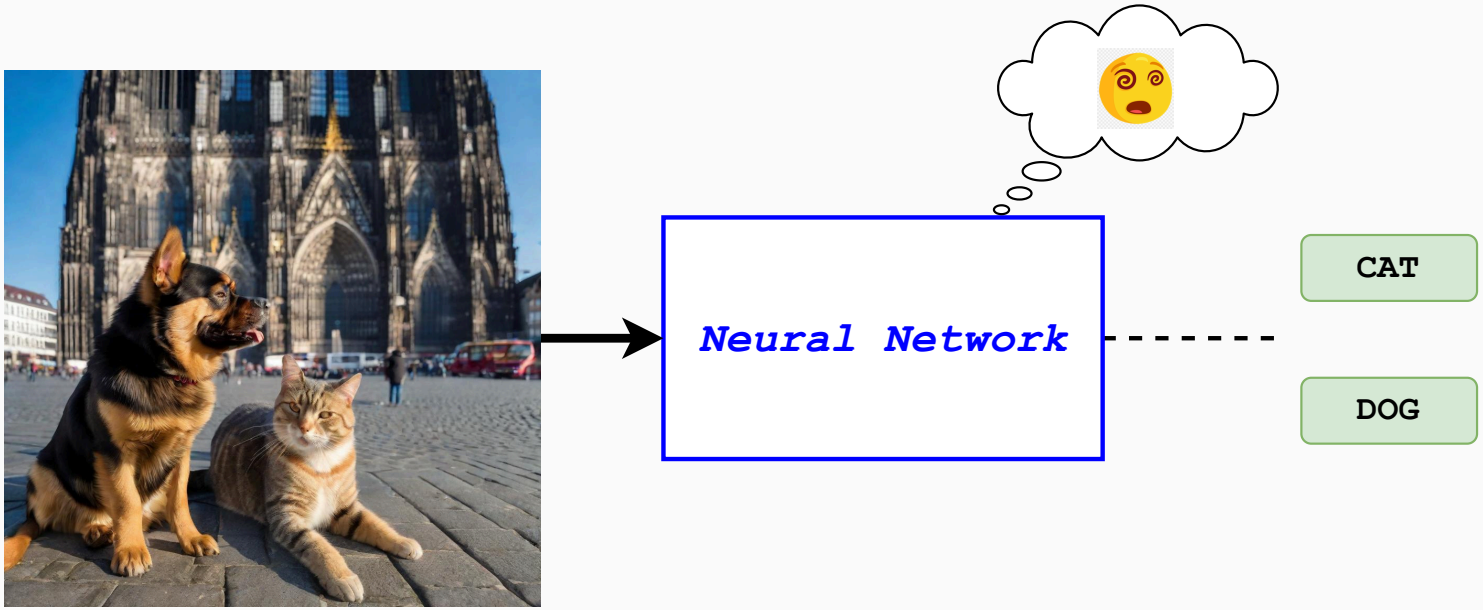


Image classification problem, the network is trained to assign single class label to an entire image. Like in above image as Cat, Five and Dog.

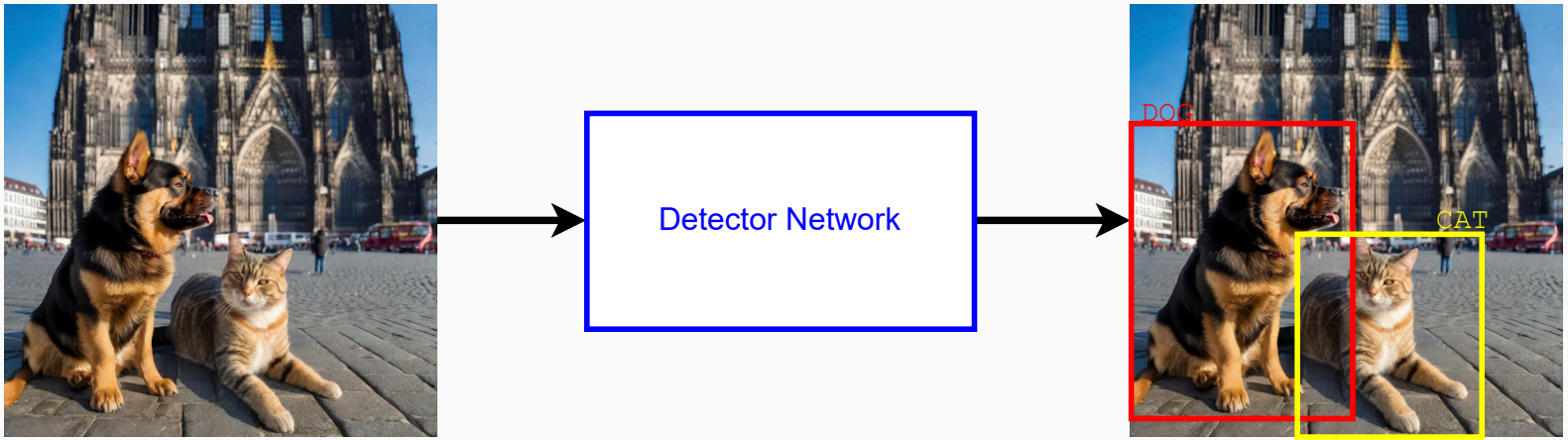
Classification

What about me?



Sol'n 1: Object detection

Train the network to regress a bounding box over the objects of interest.



Detector network

Famous detection network is R-CNN proposed in [Gir+14].

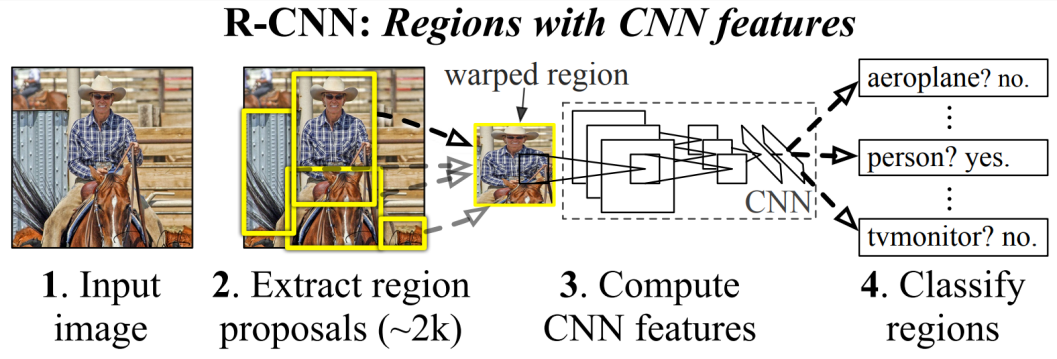


Figure: Given an image, first we extract around 2k Regions of Interest (RoI) using various search algorithms. Then we train a classifier to identify objects in these regions.

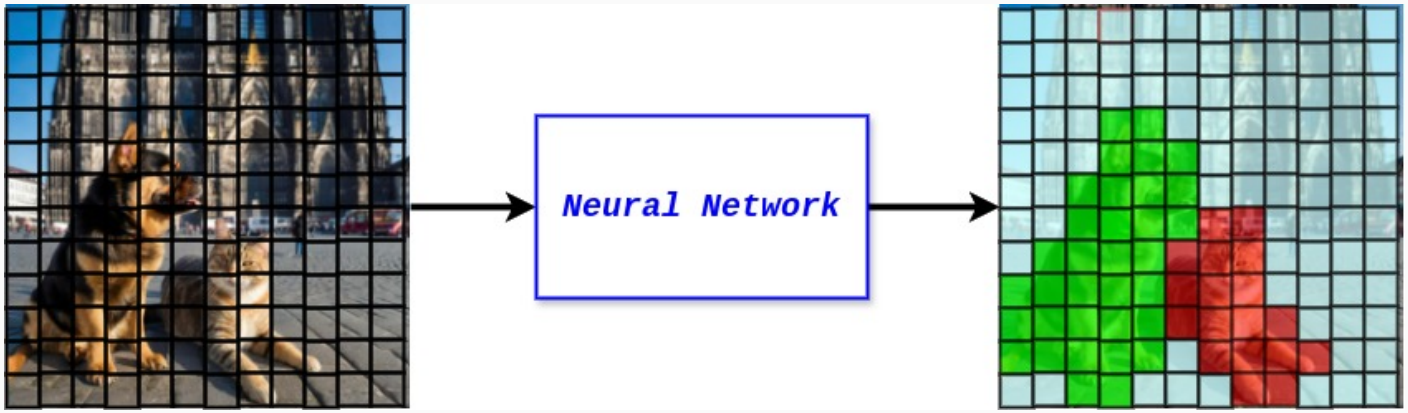
Later developments, automated the region proposal extraction by replacing search algorithms by neural networks.

A disadvantage with this approach is that if we need precise localization of objects, we need to extract large number of region proposals.

Semantic Segmentation

Sol'n 2: Segmentation

Segmentation can be defined as classification at pixel level $\rightarrow$ Network is forced to learn fine details



- **Classification**: What object in the image?
- **Segmentation**: What object in each pixel in the image?

Segmentation isn't a magic bullet. It is computationally expensive, and evolved as orthogonal task to object detection.

Autonomous Driving - Laser segmentation

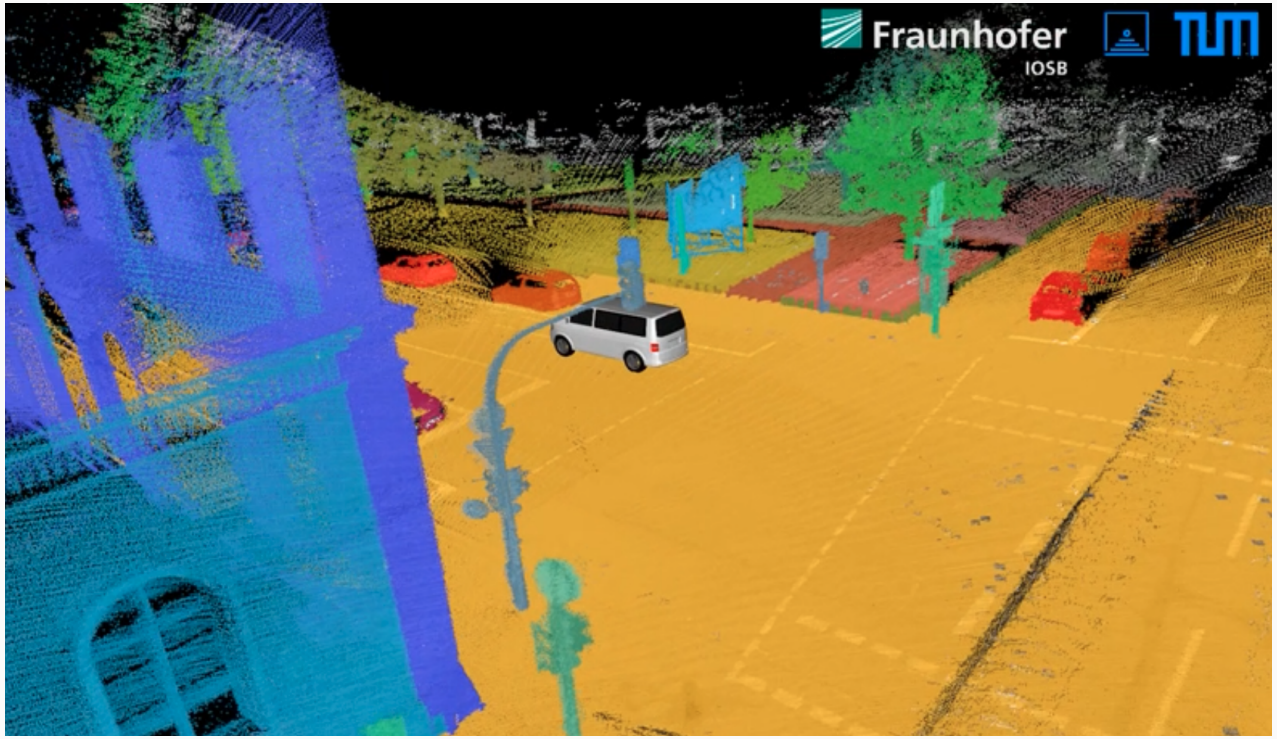


Figure: A part of segmented point-cloud from TUM City Campus dataset [Geh+17]. Full video can be found here.

Robotics

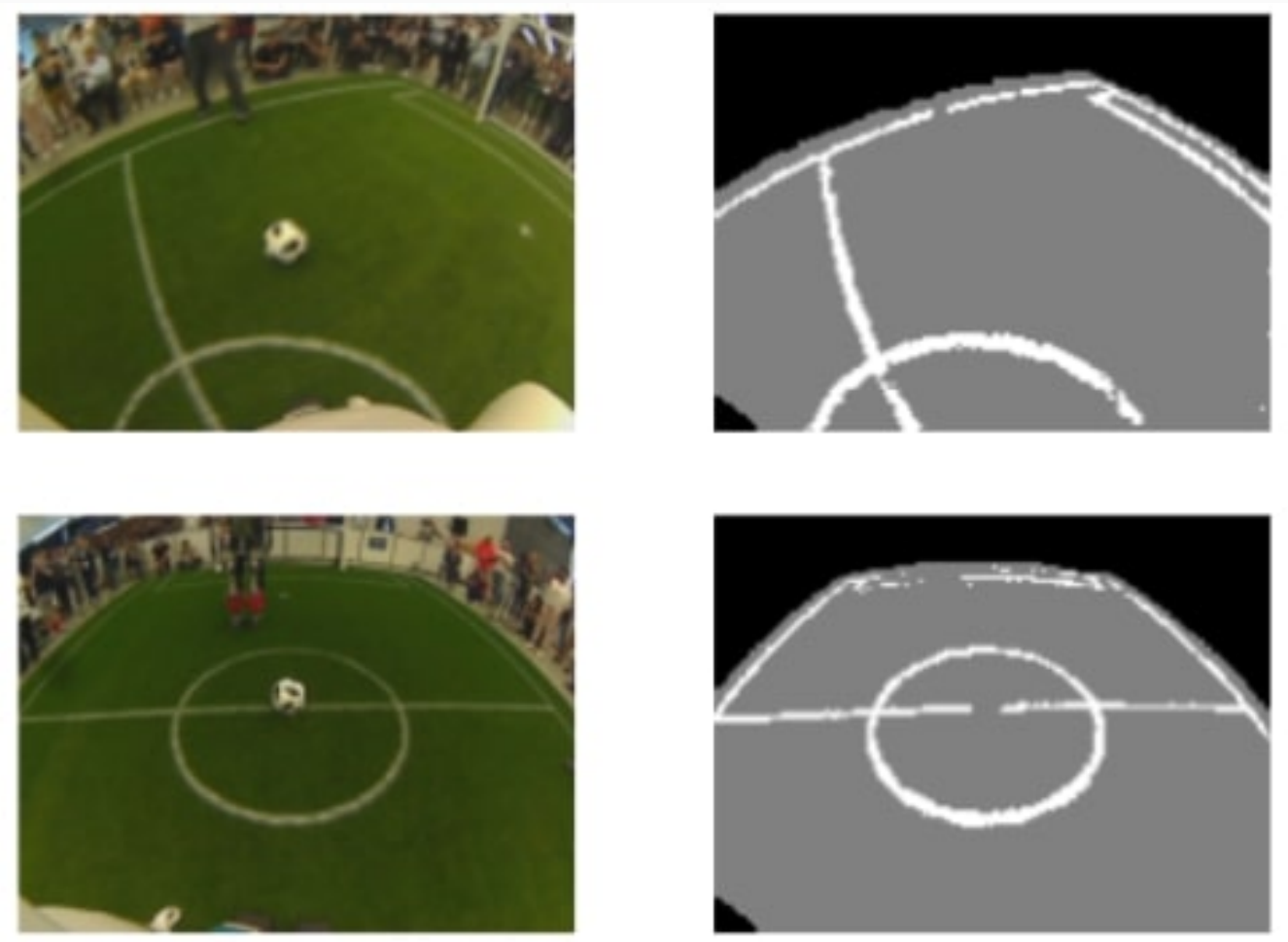


Figure: Segmentation in soccer robots to understand the boundaries of the playing field.

Medical Image segmentation i

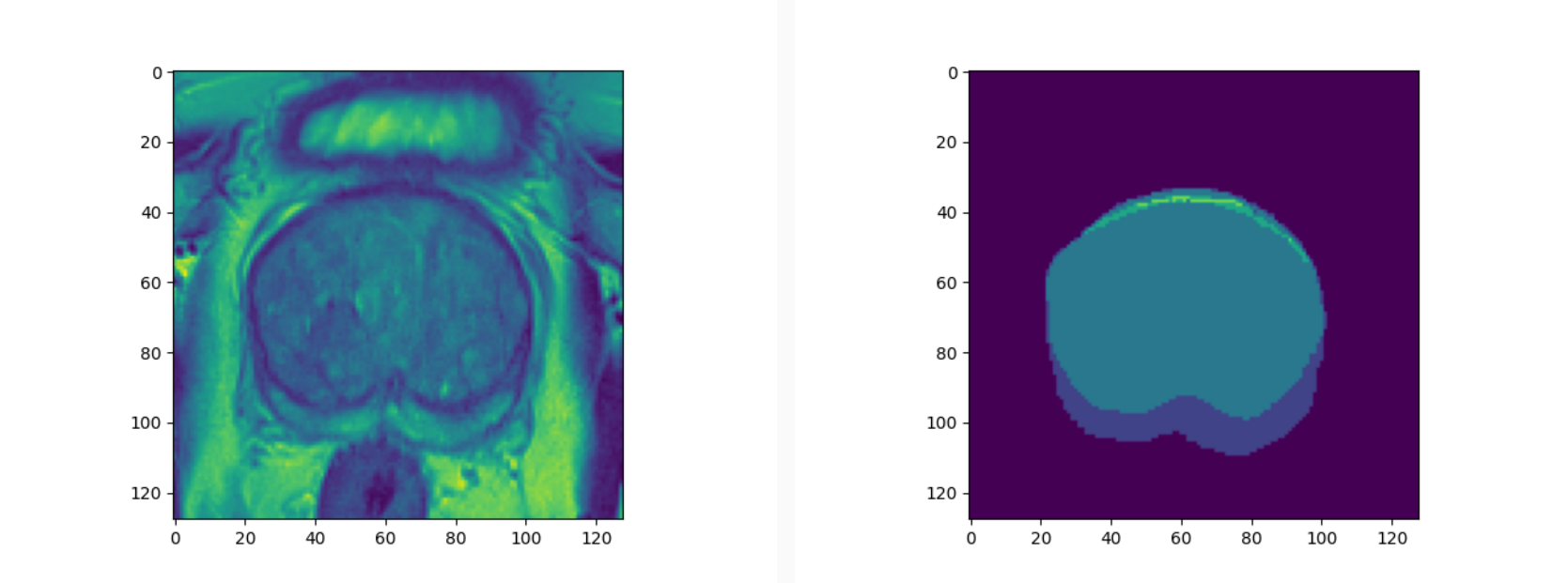


Figure: A transversal prostate image and it's expert segmentation.

Medical Image Segmentation ii

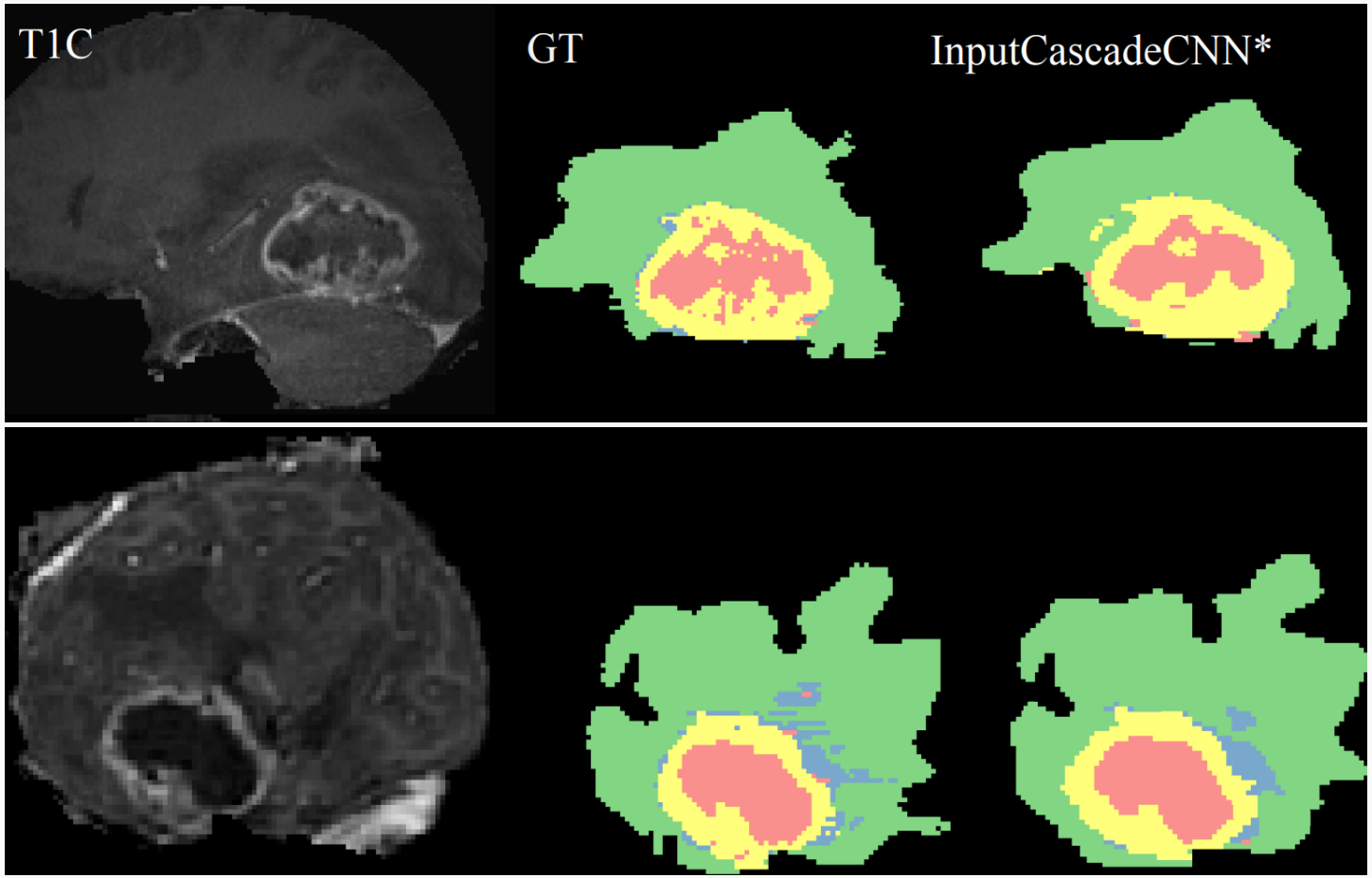


Figure: Brain tumor segmentation using MRI images with T1 weighted image (T1C) on left, ground truth (GT) in the center and DNN segmentation (InputCascadeCNN) on right [Hav+17].

Newspaper Understanding

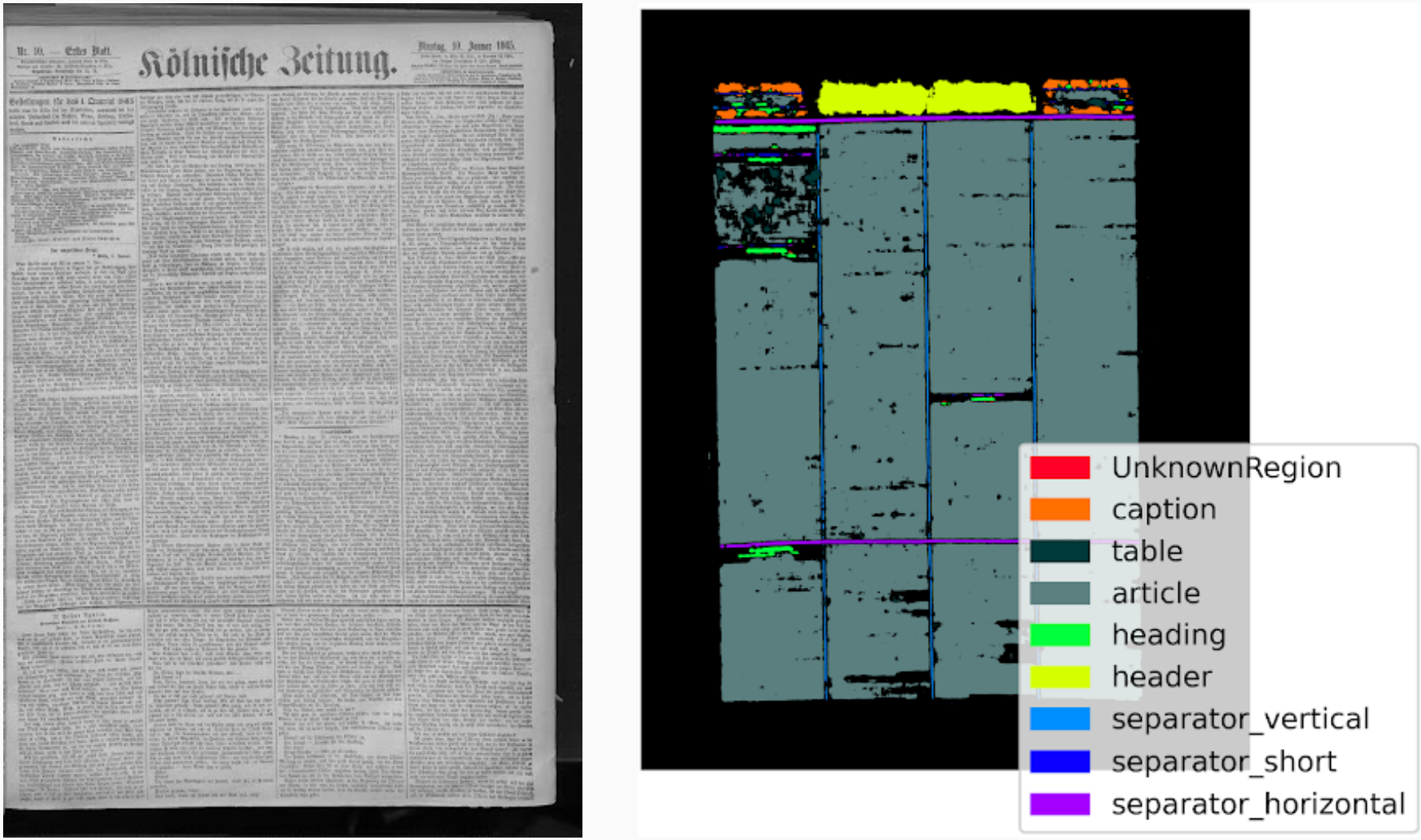


Figure: Understanding historical newspaper formats using semantic segmentation.

Segmentation Types

- Semantic Segmentation: Classify each pixel in the image
- Instance Segmentation: Classify pixel based on the instances of the object
- Panoptic Segmentation: Semantic + Instance Segmentation

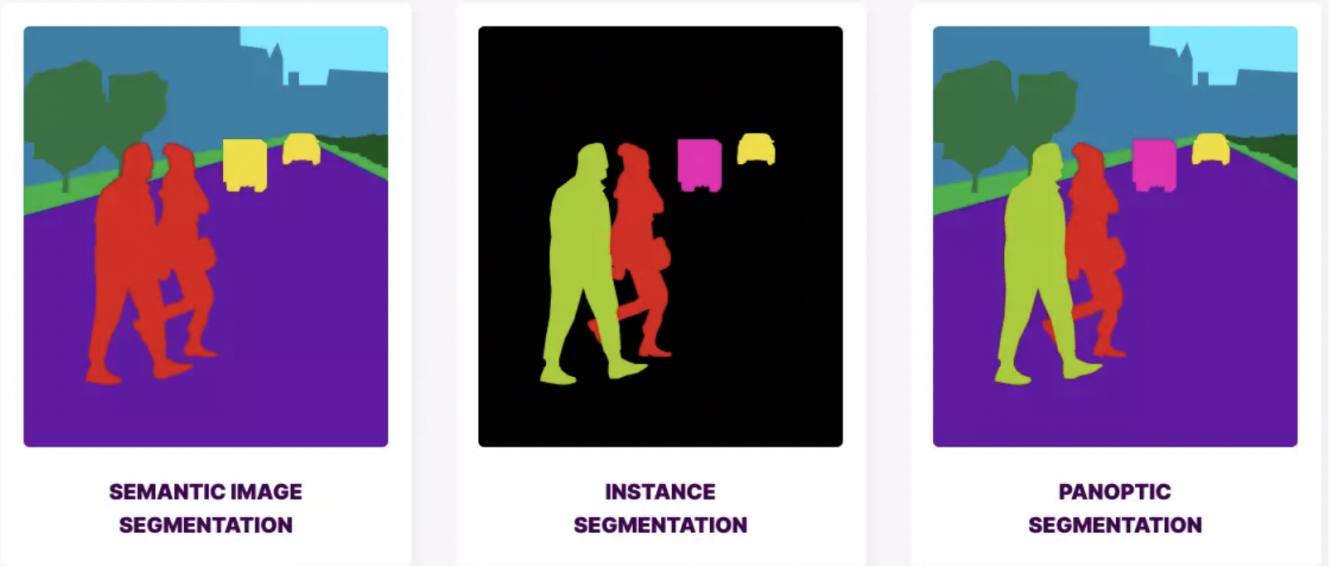


Figure: Types of segmentation [Blo22].

Semantic Segmentation Architectures

Recap: Pooling

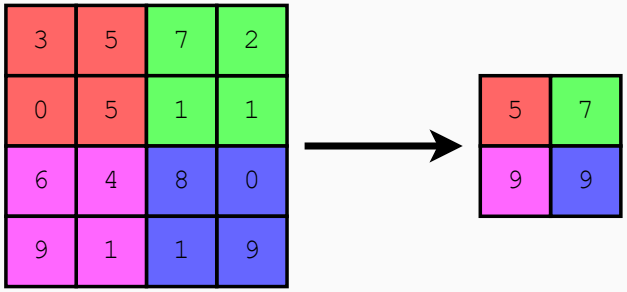


Figure: Max-Pooling applied on image of shape 4×4 with kernel size 2×2

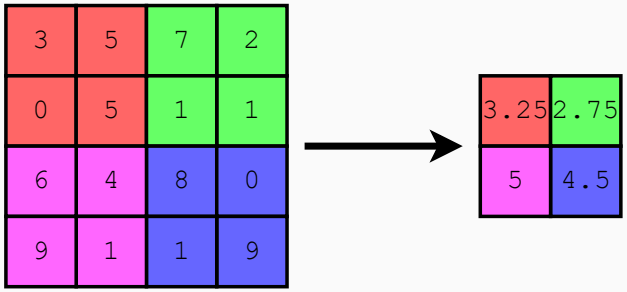


Figure: Average-Pooling applied on image of shape 4×4 with kernel size 2×2

Recap: Convolution

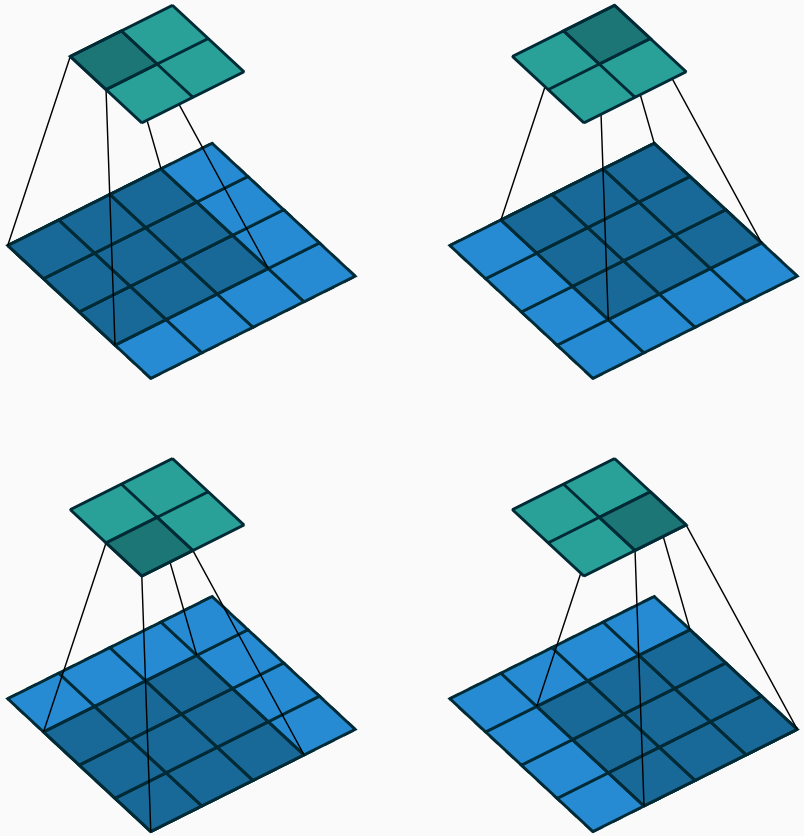
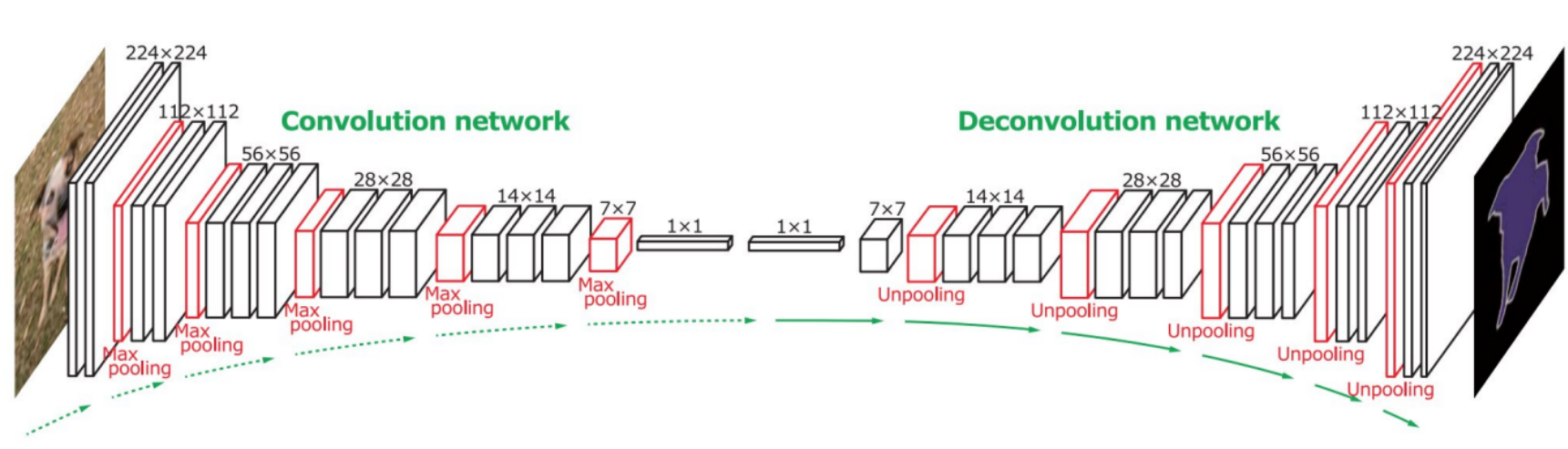


Figure: Visualization of fully padded convolutions with unit strides [DV16].

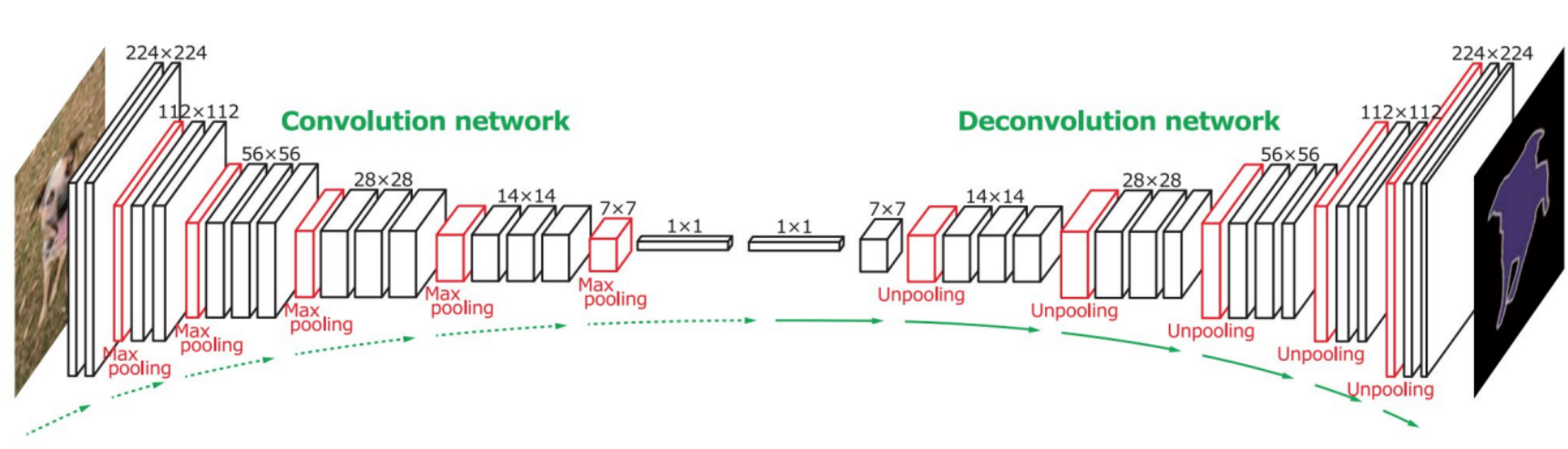
Convolutional Encoder-Decoder

As the name suggests, it is a combination of encoder model (convolution block) and decoder model (deconvolution block).



An example encoder-decoder model proposed in [NHH15]. Encoder block consist of convolutional layers for feature extraction. Intermediate max-pool layers provide aggressive downsampling of spatial features. Decoder block consititute of unpooling layers followed by convolutional layers.

Building blocks of decoder



There are majorly two special layers used in decoder block.

- Unpooling (Average and Max)
- Transposed Convolution

These layers are used to increase the spatial resolution of the data.

Average Unpooling

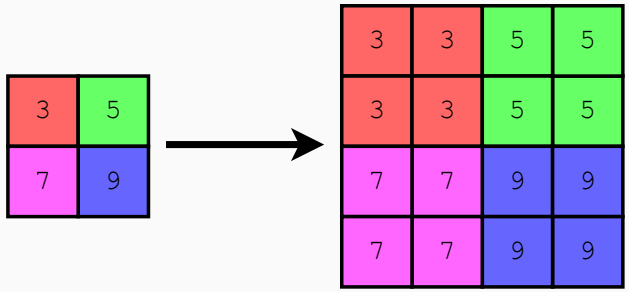


Figure: Average-Unpooling applied on image of shape 2×2 with kernel size 2×2

```
>>> input_ = th.tensor([[3., 5.], [7., 9.]]).unsqueeze(0).unsqueeze(0)
>>> input_
tensor([[[[3., 5.],
          [7., 9.]]]])
>>> avg_unpool2d = th.nn.Upsample(size=(4, 4), mode="nearest")
>>> avg_unpool2d(input_)
tensor([[[[3., 3., 5., 5.],
          [3., 3., 5., 5.],
          [7., 7., 9., 9.],
          [7., 7., 9., 9.]]]])
```

Unpooling

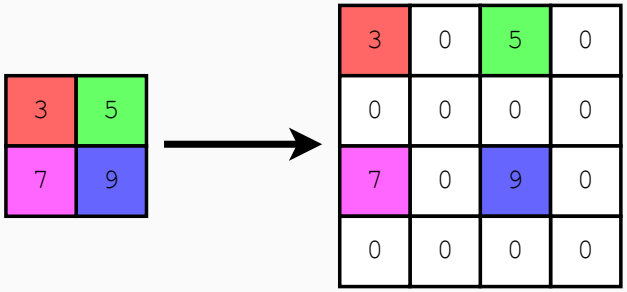


Figure: Max-Unpooling applied on image of shape 2×2 with kernel size 2×2

```
>>> input_ = th.tensor([[3., 5.], [7., 9.]]).unsqueeze(0).unsqueeze(0)
>>> input_
tensor([[[[3., 5.],
          [7., 9.]]]])
>>> indices_ = th.tensor([[0, 2], [8, 10]]).unsqueeze(0).unsqueeze(0)
>>> indices_
tensor([[[[ 0,  2],
          [ 8, 10]]]])
>>> unpool_layer = th.nn.MaxUnpool2d(kernel_size=2)
>>> unpool_layer(input_, indices_)
tensor([[[[3., 0., 5., 0.],
          [0., 0., 0., 0.],
          [7., 0., 9., 0.],
          [0., 0., 0., 0.]]]])
```

Conv vs ConvTranspose

Given Input image (H_{in}, W_{in}) convolved with Kernel (K_h, K_w).

Convolution output size with padding (P_h, P_w) and stride (S_h, S_w) is defined by

$$H_{out} = (\frac{H_{in} - K_h + 2 * P_h}{S_h}) + 1;$$
$$W_{out} = (\frac{W_{in} - K_w + 2 * P_w}{S_w}) + 1$$

Transposed Convolution output size with padding (P_h, P_w) and stride (S_h, S_w) is defined by

$$H_{out} = (H_{in} - 1) * S_h + (K_h - 1) - 2 * P_h + 1;$$
$$W_{out} = (W_{in} - 1) * S_w + (K_w - 1) - 2 * P_w + 1$$

Transposed Convolution

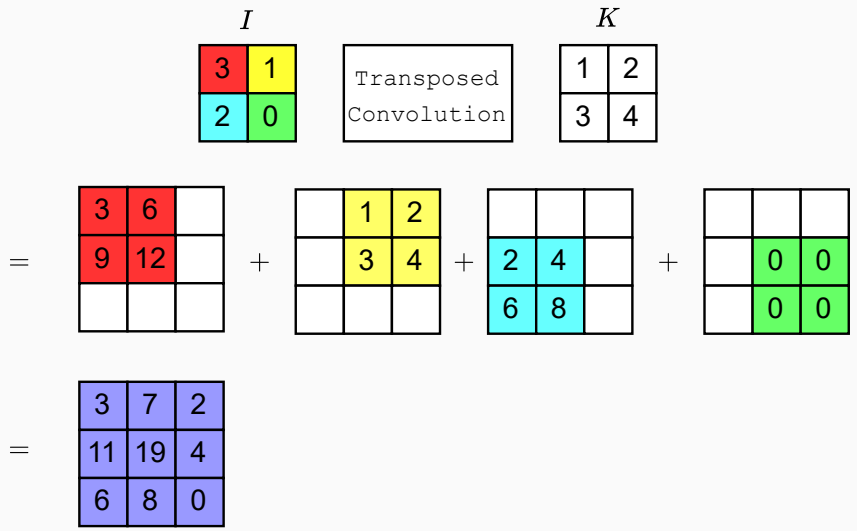


Figure: Transposed convolution applied to image (2×2) with kernel size of (2×2) and stride 1.

```
>>> input_ = th.tensor([[3., 1.], [2., 0.]]).unsqueeze(0).unsqueeze(0)
>>> kernel_ = th.tensor([[1., 2.], [3., 4.]]).unsqueeze(0).unsqueeze(0)
>>> trans_conv = th.nn.ConvTranspose2d(1, 1, kernel_size=2, bias=False)
>>> trans_conv.weight.data = kernel_
>>> trans_conv(input_)
tensor([[[[ 3.,  7.,  2.],
          [11., 19.,  4.],
          [ 6.,  8.,  0.]]]], grad_fn=<ConvolutionBackward0>)
```

Transposed Convolution Strided

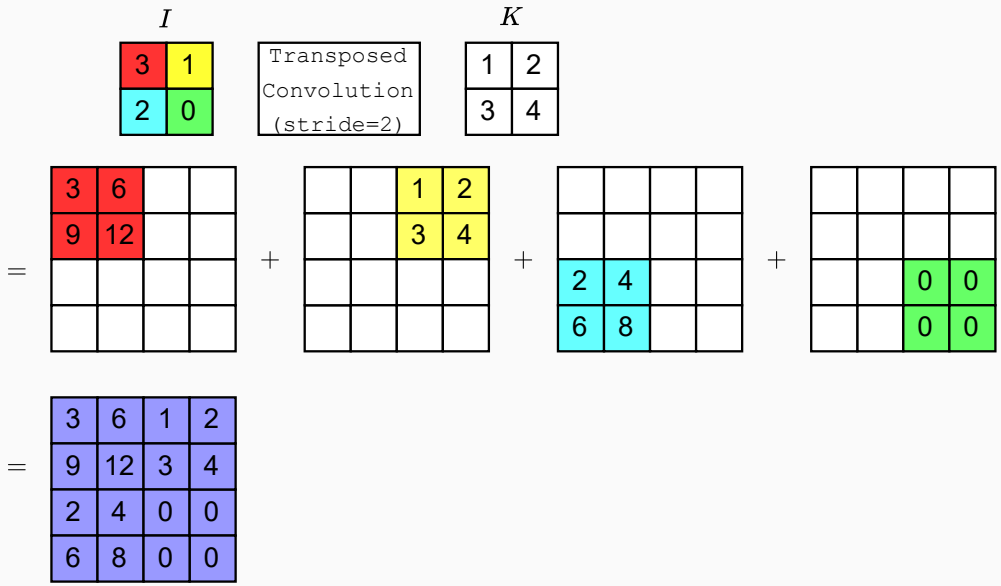


Figure: Strided transposed convolution applied to same image and kernel with stride 2.

```
>>> trans_conv = th.nn.ConvTranspose2d(1, 1, kernel_size=2, stride=2, bias=False)
>>> trans_conv.weight.data = kernel_
>>> trans_conv(input_)
tensor([[[[ 3.,  6.,  1.,  2.],
          [ 9., 12.,  3.,  4.],
          [ 2.,  4.,  0.,  0.],
          [ 6.,  8.,  0.,  0.]]]], grad_fn=<ConvolutionBackward0>)
```

Transposed Convolution Padded

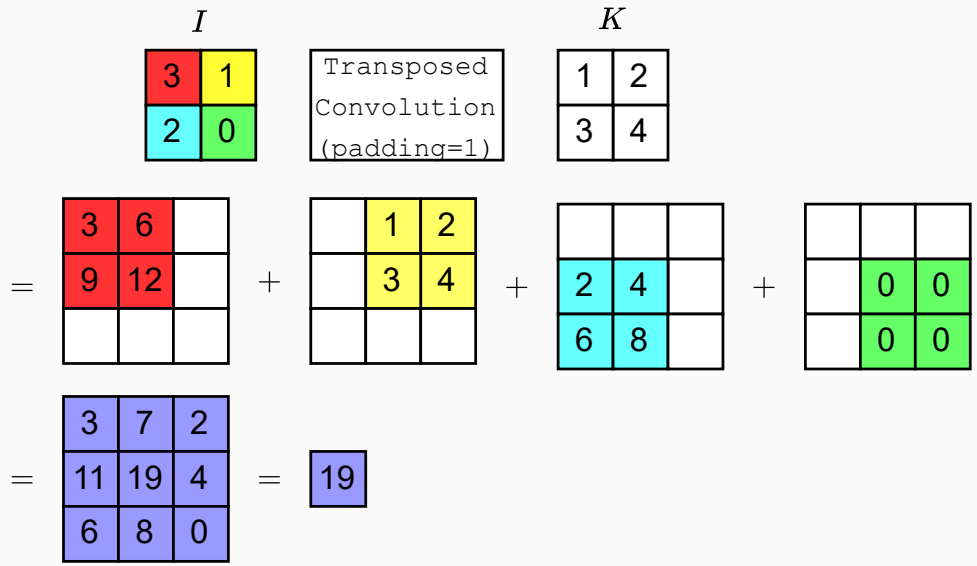


Figure: Transposed convolution with padding being 1.

```
...
>>> trans_conv = th.nn.ConvTranspose2d(1, 1, kernel_size=2, padding=1, bias=False)
>>> trans_conv.weight.data = kernel_
>>> trans_conv(input_)
tensor([[[[19.]]]], grad_fn=<ConvolutionBackward0>)
```

The U-Net structure [RFB15]

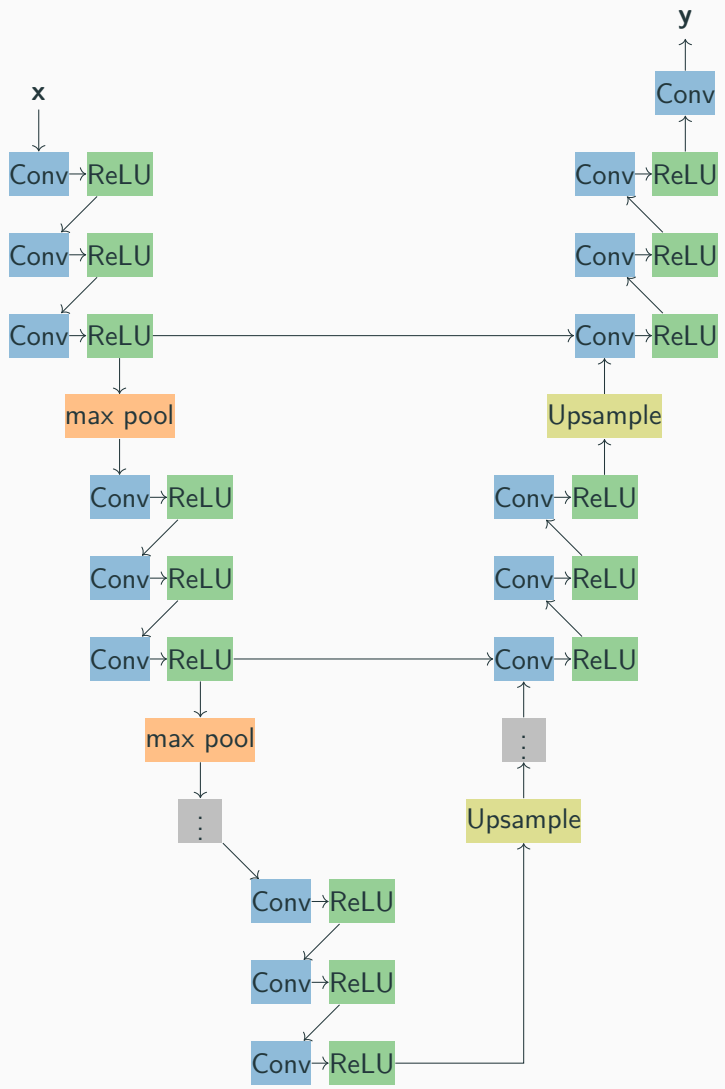


Figure: The U-Net architecture [RFB15].

U-Net Applications

- Semantic Segmentation
- Diffusion [HJA20]

- Protein structure prediction and site localization [Sen+20]
- Neural Style Transfer [Rao+21]

UNet - Prostate segmentation

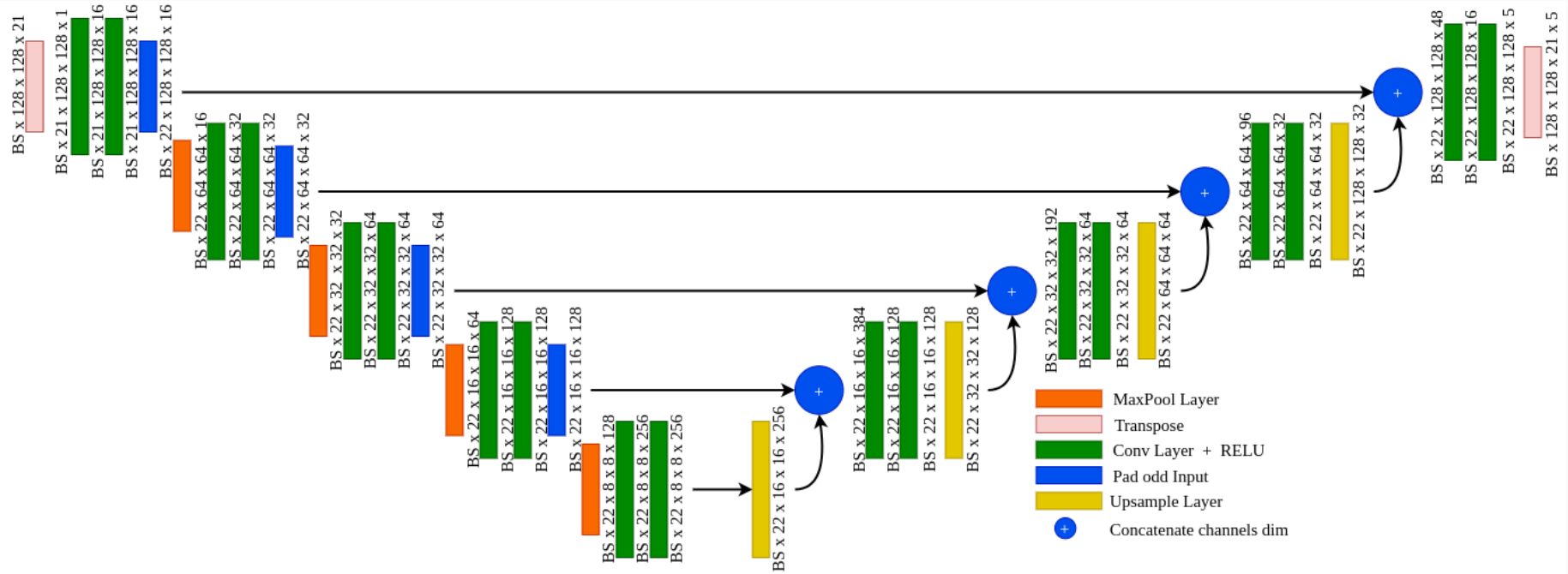


Figure: UNet to be implemented for Prostate segmentation exercise.

Focal Loss

The focal loss for segmentation problems

Focal loss functions are a standard approach in image segmentation, it originally appeared in [Lin+17]. The general idea is to increase the weight of rare classes. If classes are mutually exclusive use,

$$\mathcal{L}(\mathbf{o}, \mathbf{l}) = -\mathbf{l} \cdot (1 - \sigma_{\boldsymbol{s}}(\mathbf{o}))^{\gamma} \cdot \alpha \cdot \ln(\sigma_{\boldsymbol{s}}(\mathbf{o})) \tag{1}$$

to train your U-Net.

Focal Loss Vs Cross-Entropy Loss

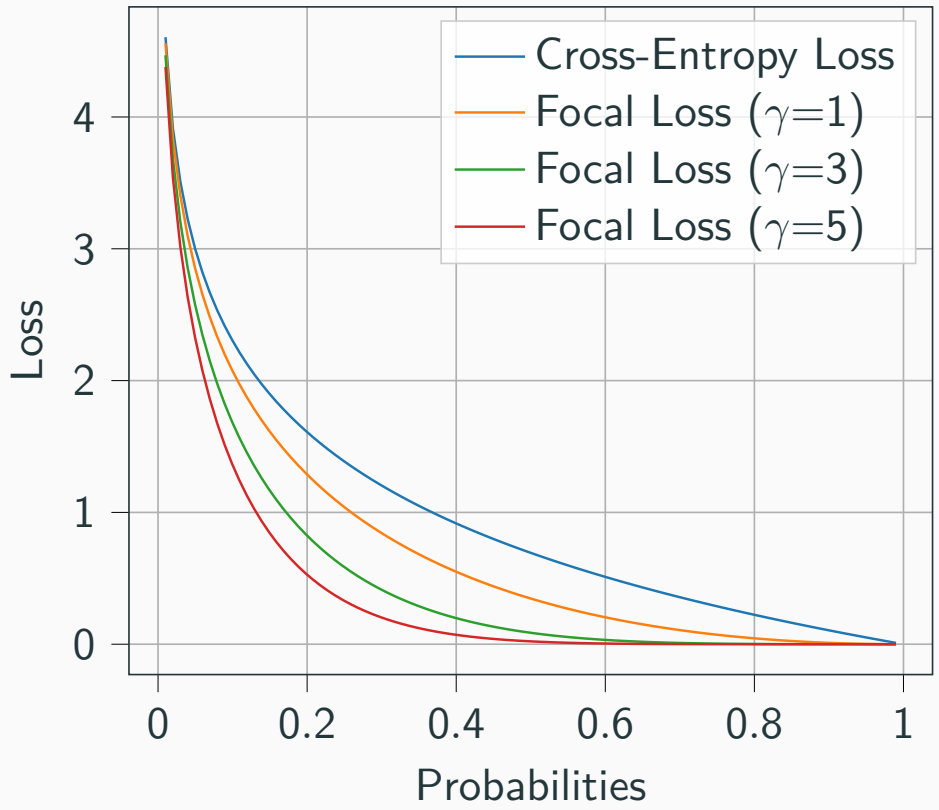
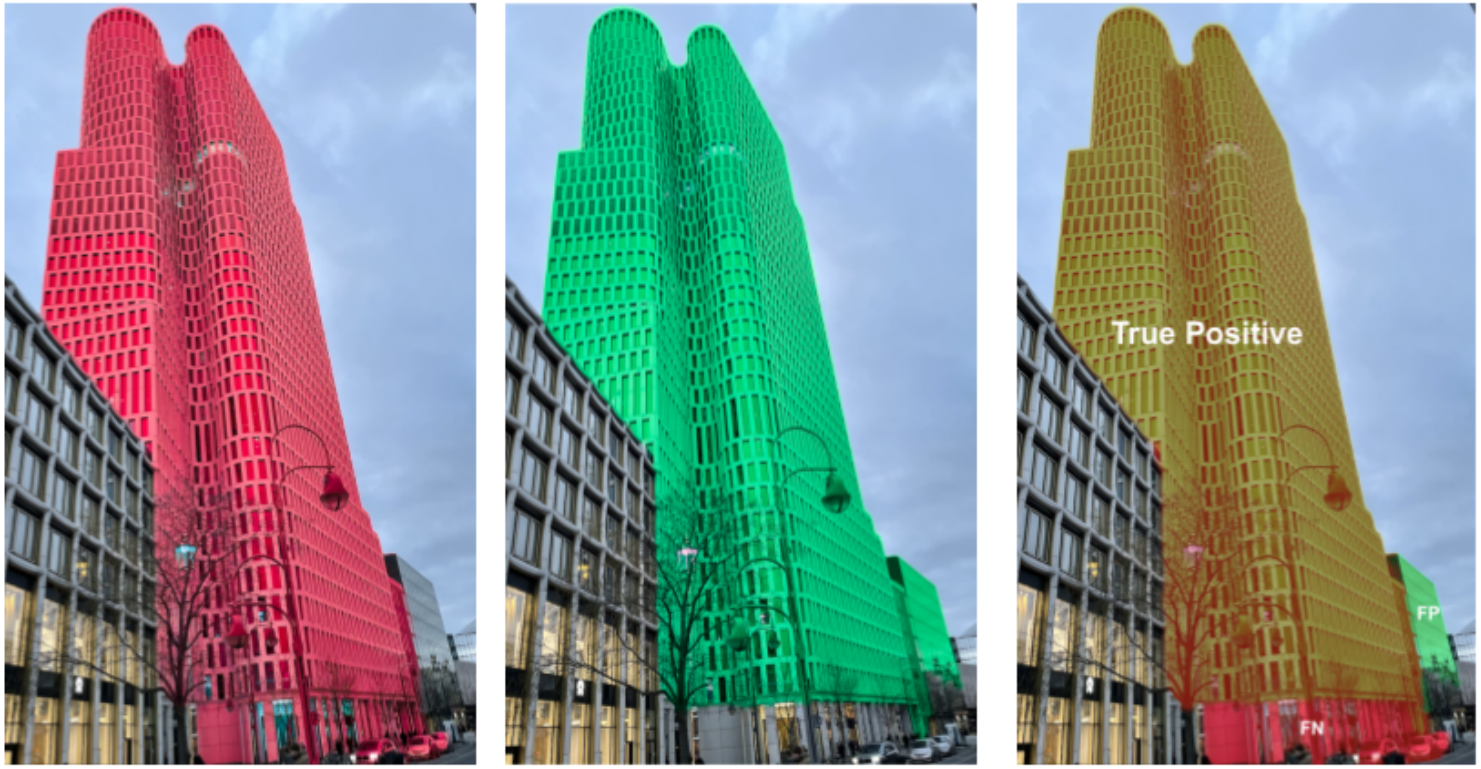


Figure: Evolution of Cross Entropy loss compared with the various γ of Focal loss.

Evaluation Metric - Intersection-over-Union

Intersection over union



Ground Truth Mask Predicted Mask IoU

Intersection over union (IoU)

IoU is calculated as a fraction of area of intersection and area of union between GT and predicted masks.

From above example, IoU can be calculated using confusion matrix by the below formula

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$

where TP, FP and FN are True Positives, False Positives and False Negatives respectively.

IoU ranges between 0 and 1.

References

[Blo22] Mindy news Blog. ***What is Image Segmentation: The Basics and Key Techniques.*** https://mindy-support.com/news-post/what-is-image-segmentation-the-basics-and-key-techniques/. Accessed: 2023-09-26. 2022.

[DV16] Vincent Dumoulin and Francesco Visin. **"A guide to convolution arithmetic for deep learning."** In: *arXiv preprint arXiv:1603.07285* (2016).

Literature ii

[Geh+17] Joachim Gehrung, Marcus Hebel, Michael Arens, and Uwe Stilla. **"An approach to extract moving objects from MLS data using a volumetric background representation."** In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), pp. 107–114.

[Gir+14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. **"Rich feature hierarchies for accurate object detection and semantic segmentation."** In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

Literature iii

[Hav+17] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron C. Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. **"Brain tumor segmentation with Deep Neural Networks."** In: *Medical Image Anal.* 35 (2017), pp. 18–31.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. **"Denoising diffusion probabilistic models."** In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

Literature iv

[Lin+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. **"Focal loss for dense object detection."** In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[NHH15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. **"Learning deconvolution network for semantic segmentation."** In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

Literature v

[Rao+21] Dongyu Rao, Xiao-Jun Wu, Hui Li, Josef Kittler, and Tianyang Xu. **"UMFA: a photorealistic style transfer method based on U-Net and multi-layer feature aggregation."** In: *Journal of Electronic Imaging* 30.5 (2021), pp. 053013–053013.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. **"U-net: Convolutional networks for biomedical image segmentation."** In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

Literature vi

[Sen+20] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. **"Improved protein structure prediction using potentials from deep learning."** In: *Nature* 577.7792 (2020), pp. 706–710.