Application of object segmentation techniques

Viktória Kabai
Postgraduate specialist training mathematics expert in data analytics and machine learning
Supervisor: Szabolcs Szalánczi

ELTE Faculty of Science
2022

Contents of the thesis:

1. INTRODUCTION
2. BASIC CONCEPTS OF IMAGE SEGMENTATION
    - 2.1. Definitions of Computer Vision and its Applications
    - 2.2. Overview of Early Image Segmentation Approaches
3. DEEP LEARNING IMAGE SEGMENTATION
    - 3.1. Milestone algorithms and their development
    - 3.2. Building blocks of Image Segmentation Deep Learning algorithms
4. APPLICATION OF SOME DEEP LEARNING ALGORITHMS
    - 4.1. Data Description
    - 4.2. VGGNet
    - 4.3. ResNet
    - 4.4. U-Net
    - 4.5. Transfer Learning
    - 4.6. VGG16 - U-Net
    - 4.7. Summary of results
5. REFERENCES

Thesis objectives

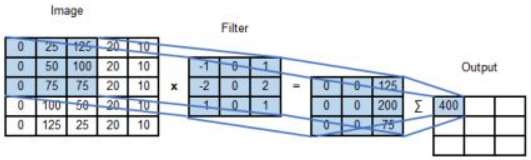
The goal was to deepen the understanding of deep learning image segmentation architectures and the ways of adapting them to specific use cases:

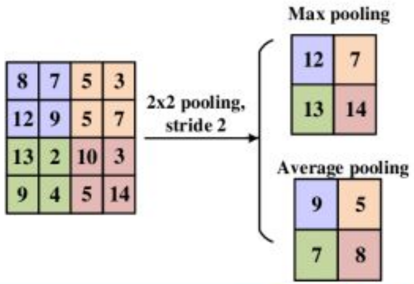
- Discuss the history of the image segmentation field and the stages it went through before landing on deep learning algorithms;
- Cover in detail some of the milestone algorithms that shaped the research around image segmentation and the circumstances made these advances possible;
- Make experiments on implementing the discussed algorithms from scratch and importing them via transfer learning.

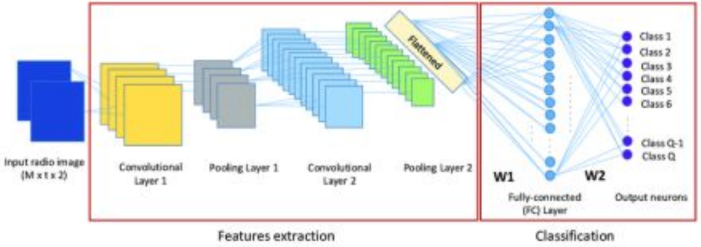
Basics of Deep Learning Image Segmentation

Convolution Neural Networks

- **Layers:** convolutional, fully connected, sub-sampling (pooling)
- **Activation functions:** ReLU, Sigmoid, Softmax
- **Regularization:** batch normalization, dropout
- **Loss functions:** cross-entropy
- **Metrics:** pixel accuracy







Data Description

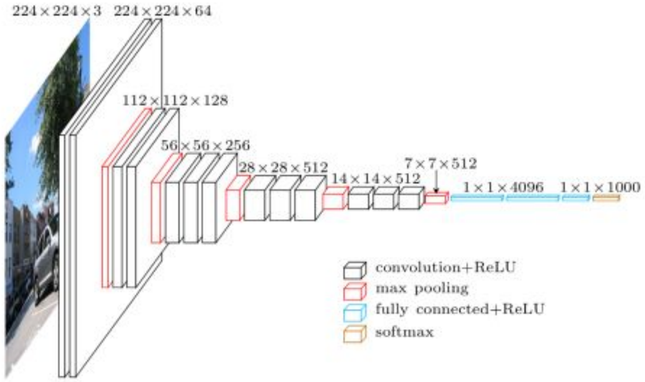
COCO (Common Objects in Context) dataset that is a large scale object detection, segmentation and captioning dataset with 330K images (>200K labelled), 1.5 million object instances and 80 categories.

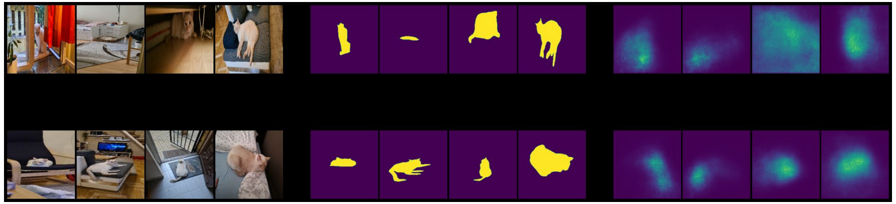
Application of Deep Learning Models I. – VGG

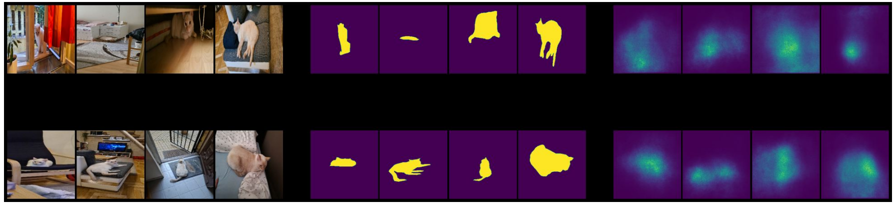
256x256x3 sized images as our inputs and our goal is to do binary semantic segmentation on cat images from the COCO dataset, changes to the structure:

- adjust the input sizes to our needs
- use batch sizes of 5 and learning rate of 1e-5
- contract the fully connected layers to size 1000
- last layer 256*256 size fully connected layer
- reshape the output of the last layer to return to 2D matrix shape tensor instead of a vector of 65536 length
- comparable with the original masks of the images
- train the networks for 25 epochs

VGG16 results: validation accuracy of around 0.813 and test accuracy of 0.914 and image results shown below (left)
VGG19 results: validation accuracy of 0.814 and test accuracy of 0.914 and image results shown below (right)



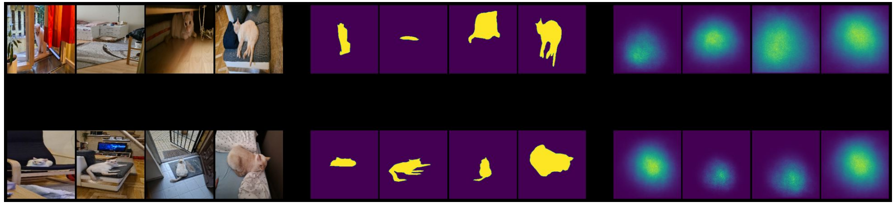


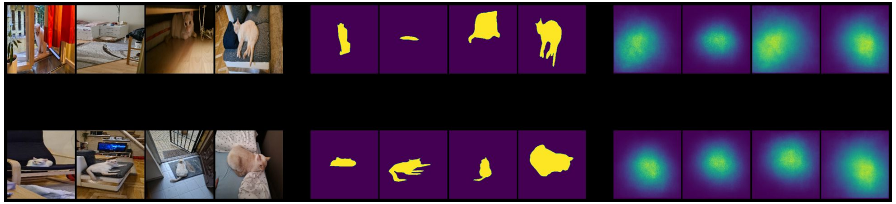


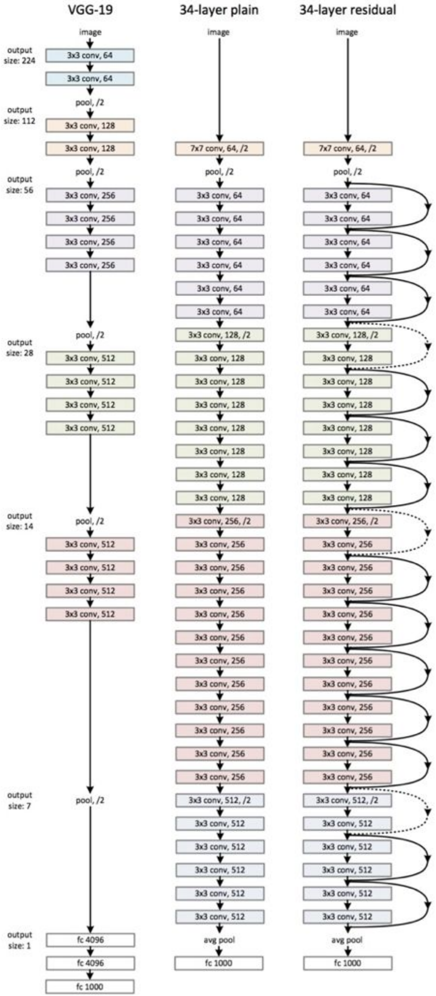
Application of Deep Learning Models II. - ResNet

- adjust the input sizes to 256x256x3
- use batch sizes of 5 and learning rate of 1e-5
- contract the fully connected layers to size 125
- last layer 256*256 size fully connected layer
- reshape the output of the last layer to return to 2D matrix shape tensor instead of a vector of 65536 length
- comparable with the original masks of the images
- train the networks for 25 epochs

ResNet50 results: validation accuracy of 0.812 and test accuracy of 0.914 and image results shown below (top)
ResNet101 results: validation accuracy of 0.815 and test accuracy of 0.914 and image results shown below (bottom)



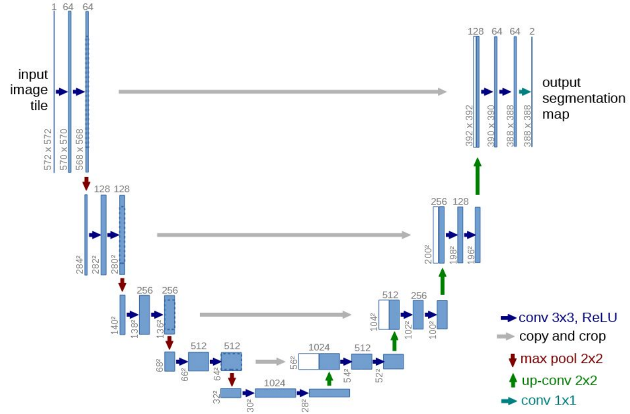


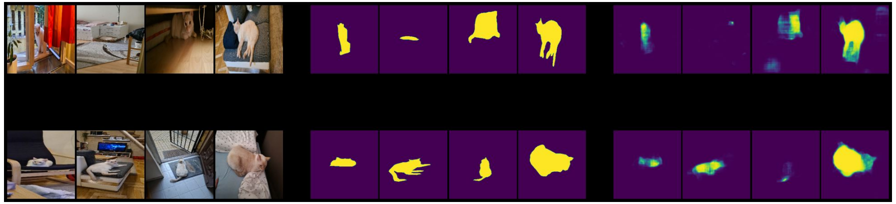


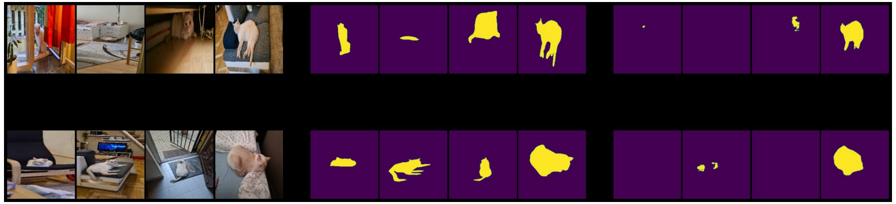
Application of Deep Learning Models III. - U-Net

- adjust the input sizes to 256x256x3
- use batch sizes of 10 and learning rate of 1e-3
- start with 16 features and go up
- train the networks for 25 epochs

U-Net results: validation accuracy of around 0.934 and test accuracy of 0.965 and image results shown below (left)
Image results with threshold set to 0.9 instead of 0.5 (right)



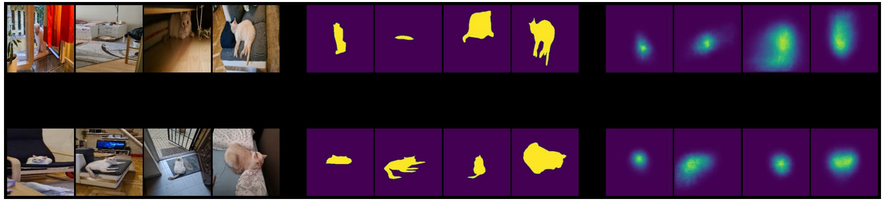


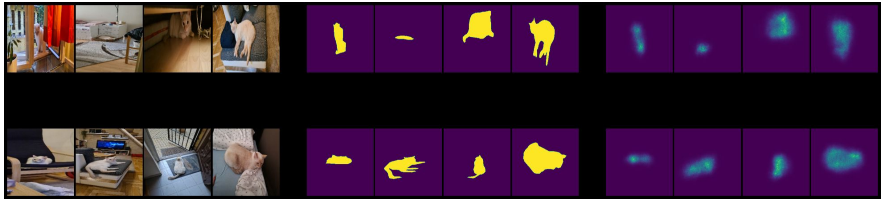


Application of Deep Learning Models IV. - Transfer Learning with VGG and ResNet

- use every convolutional layer from the pre-trained models
- build our own dense (fully connected) layers (4096 and 1000 neurons in VGG and 500 neurons in ResNet)
- last ones size 256*256 and softmax activation function
- reshape them as in the original VGG and ResNet models
- same batch size, epoch number, learning rate, optimizer, loss and metric as in the VGG and ResNet models before

VGG16 results: validation accuracy of 0.814 and test accuracy of 0.915
ResNet50 results: validation accuracy of 0.816 and test accuracy of 0.915

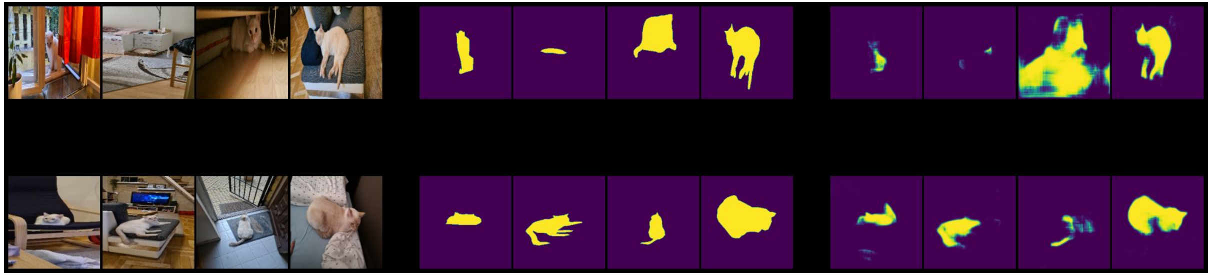




Application of Deep Learning Models V. - VGG - U-Net

- encoder part the architecture - pre-trained VGG16 model
- at the bottom we have 3 convolutional layers with the same kernel size and 512 features without max pooling
- in the decoder we have 4 blocks of 3 convolutional layers with upsampling and 3x3 kernels, with the feature number going up from 512 to 64 block-by-block
- use dropout, batch normalisation
- output is created by a convolutional layer of size 1 and sigmoid activation

Accuracy of this model tops at 0.942 on validation set and 0.929 on test data



Conclusions

The goal of this thesis was to explore and understand more deeply the available state-of-the-art deep learning image segmentation algorithms. For that we went through several milestone models that shaped the field of research behind it.

We discussed the structures among others VGG and ResNet models, which all share the fact that they combine convolutional and fully connected layers. Experiment were conducted by building the algorithms from scratch to see on hand how they behave and what are their specific characteristics.

We showed, in line with our expectations, that models with fully connected layers need a lot more training time and data size (ResNet and VGG models) to perform better, while U-Net, that was specifically designed for small amounts of data and is fully convolutional model, is working very well with limited amounts of data and can achieve great accuracy.