DEPARTMENT OF COMPUTER SCIENCE
AND MATHEMATICS

UNIVERSITY OF APPLIED SCIENCES MUNICH

Master's Thesis in Computer Science

On the generalization capabilities of interactive segmentation methods

Alexander Fertig

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS

UNIVERSITY OF APPLIED SCIENCES MUNICH

Master's Thesis in Computer Science

On the generalization capabilities of interactive segmentation methods

Author:	Alexander Fertig
Supervisor:	Prof. Dr. David Spieler
Advisor:	Advisor
Submission Date:	Submission date

I confirm that this master's thesis in computer science is my own work and I have documented all sources and material used.

Munich, Submission date Alexander Fertig

Acknowledgments

Abstract

1. Introductions
   a) DL in Industry
   b) Application of DL and gathering Labels
2. Basics
   a) ML, Dl, CNN
   b) segmentation segmentation (and IoU)
   c) Interactive segmentation segmentation (Methods of comparison)
3. Methods
   a) Extreme Points
   b) IOG
4. Benchmark
   a) Motivation and structure of the Benchmark
   b) Applied Methods
   c) Evaluation (or put Evaluation as own chapter)
5. Conclusion

Contents

1 Introduction

1.1 Section

1.1.1 Subsection

2 Theory

2.1 ML, DL and CNN

The last decade was revolutionary for the sector of information technology. Due to technical advancement, the computational power of processors, especially Graphical Processing Unit (GPU) rise significantly. Further, the wider creation and use of data introduced the domain of big data, that allows operators to gain more insights and benefits. Both of these recent advancements benefited another field of study commonly described as Artificial Intellegence (AI). The term AI describes machines that are show characteristics of human intelligence that allow them to handle various tasks. But there are several gradations, that hide behind the powerful bus word AI and are illustrated in this section.

2.1.1 Machine Learning

Machine Learning (ML) is a subsection of AI and is based on the ability of algorithms to analyze, detect and learn patterns in various kinds of data. Applying these learned pattern ML algorithms may reach human level performance or even better, but they are limited specifically to their scope. For other tasks mostly a new ML algorithms needs to be defined [1]. In contrast AI may be able to solve various tasks and learn independently by showing strong characteristics of human intelligence.

[1]Henning Steiner, Hessischer Rundfunk: *Selbstlernende Maschinen - wie Künstliche Intelligenz entsteht*: `https://www.hr-inforadio.de/podcast/wissen/selbstlernende-maschinen---wie-kuenstliche-intelligenz-entsteht,podcast-episode-53312.html`

2.1.2 Deep Learning

2.1.3 Convolutional Neural Networks

2.2 Image Segmentation

2.2.1 General

Image segmentation is an advanced task of modern computer vision. The term *segmentation* means to obtain regions or structures from an image. In order to partition the image into segments a high level of understanding is required. Modern techniques of Deep Learning (DL) have proven themselves to be most adequate for this task. Deep Convolutional Neural Networks (CNNs) are usually applied nowadays to perform image segmentation. In this context segmentation is treated as a classification task with K classes. A class k is assigned to every pixel of the image. The output is a segmentation map, which has the size of the input image each pixel containing a label of its class k. Pixels with a the same class label form a segment, that may be further processed afterwards. There two main variants to perform image segmentation:

- **Semantic segmentation** classifies each pixel with one class. There is no differentiation made if there are multiple objects of one class, they all belong to the same segment.

- **Instance segmentation** differentiates between different objects, which have the same class, by assigning them a unique label. In the result several segments may have the same class, but are treated as independent instances of this class.

As classification is a problem of supervised learning, this also applies on image segmentation. Therefore, a dataset with labels on pixel-level is required. A label $\hat{y}$ is represented by a map, that has the same size as the corresponding input image x and contains a class label for each pixel $\hat{y}_{rc}$ with r and c referring to the corresponding row and column of the map. In this context the label $\hat{y}$ is also referred to as mask or Ground Truth (GT).

To train a segmentation network a loss function is required, that considers the loss of every pixel in the image and optimizes the prediction for each pixel individually. In [Jad20] several loss functions are examined. Jadon concludes, that there is no universal loss function, instead their performance depends on the characteristics of the dataset. Cross entropy loss works best on a balanced dataset, while for imbalanced datasets the dice coefficient or focal loss is suitable.

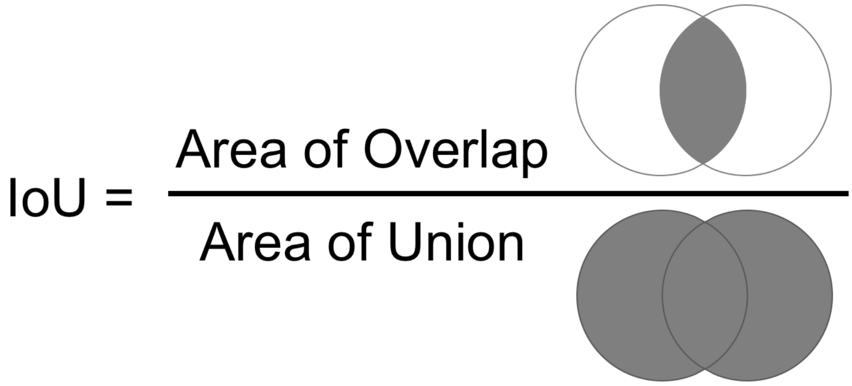

Figure 2.1: Intersection over Union. The *area of overlap* represents the intersection of the GT with the made prediction. The *area of union* represents the total area of GT and the prediction [SM18].

2.2.2 Evaluation Metric

To ensure an objective comparison of several methods a evaluation metric is required, which incorporates the basic idea of segmentation. As this challenge is an classification task on pixel-level, a measure of evaluation is the Overall Pixel (OP) accuracy, which represents the proportion of all correctly labeled pixels in an image. Further, the OP measurement can be refined by calculating the accuracy for each class. This results in the Per-Class (PC) accuracy, which represents the proportion of correctly labeled pixels of one class. The most commonly used evaluation metric is the Intersection over Union (IoU), also known as the Jaccard Index, which is used in the PASCAL VOC challenge [Eve+10] since 2008 [CP13]. The IoU measures the ratio of overlap (true positives) between GT and prediction and of the total area. It is defined as

$$IoU = \frac{\text{true positives}}{\text{true positives} + \text{false negatives} + \text{ false positves}} \tag{2.1}$$

and is calculated for each instance or segmentation class. To evaluate all instances or classes of an image or a dataset the IoU is averaged, which results in the mean Intersection over Union (mIoU) [2].

An advantage of this metric is the inclusion of *false positives* and *false negatives* into the calculation. A limitation of the IoU metric is that the correctness of the segments boundaries is not taken into account. In order to compensate this issue, Csurka

[2]TensorFlow, *tf.keras.metrics.MeanIoU*: `https://www.tensorflow.org/api_docs/python/tf/keras/metrics/MeanIoU`

suggests in [CP13] to combine the IoU with another complementary metric, evaluating the boundary of a segment. Regardless, the IoU is a suitable and informative metric, which is also the most common to evaluate semantic segmentation models.

2.2.3 Architecture

For image classification established CNN architectures follow a common scheme: A multi dimensional input image is processed and continuously downsized to a one dimensional tensor in order to make one prediction. In contrast, for image segmentation a prediction is made for each pixel of the image. Therefore, an adaption in architecture required, that enables the model to make a prediction for every pixel of the image. In the following characteristics of important architectures and components are examined.

Encoder-Decoder-Architecture

The Encoder-Decoder-Architecture as its name anticipates is based on two main parts: the encoder network and the decoder network, visualized in Figure 2.2. Representatives of the encoder-decoder-architecture are among others the U-Net [RFB15], the DeConvNet [NHH15] and the SegNet [BKC17].

The encoder network is very similar to a CNN. It consists out of convolution and pooling layers, that reduce the size of the feature maps and extract features. The encoder networks of the DeConvNet [NHH15] and the SegNet [BKC17] are even represented by of a popular CNN, the VGG-16 [SZ15]. In this context the process of applying the encoder network is also called *downsampling*, due to the size reduction of the feature maps.

The decoder network is the counterpart of the encoder network. It reconstructs the feature maps to their original size, which is also referred to as *upsampling*. To reach this original size often a reversed architecture of the encoder network is used. The elemental components of this reconstruction are the operations *unpooling* and *transpose convolution*, introduced in the following.

After the encoder network generally a softmax classifier is applied, that predicts the class for each pixel. The output is a probability map with K channels for K number of classes [BKC17].

Unpooling. The unpooling operation is the equivalent of the pooling operation. Instead of reducing the size of feature maps F_c^n, they are enlarged. As for pooling, no features are learned and there exist multiple methods to perform unpooling, two of them are illustrated in Figure 2.3. Nevertheless, unpooling is not capable to fully reconstruct information lost during the process of downsampling. The result for *bed of*

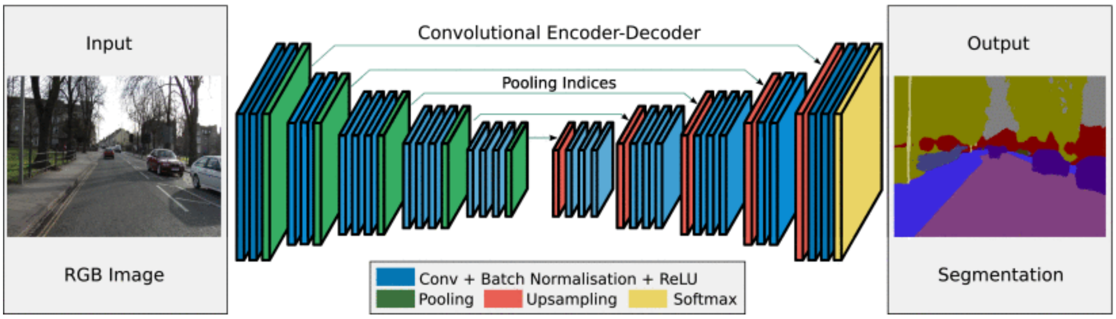

Figure 2.2: Encoder-Decoder-Architecture from SegNet. On the left the encoder network, which reduces the size of the feature maps while processing. On the right is the decoder network, which reconstructs the feature map to the size of the original input. The yellow layer on the very right is the classification layer, here represented as softmax layer to create the output segmentation. Copyright ©2017 Creative Commons License. Reprinted by permission from [BKC17].

nails are sparse feature maps F_c^n, while for *nearest neighbor* the feature maps F_c^n contain redundant information.

Transposed Convolution. As for pooling it is unpooling, the counterpart of convolution is transposed convolution [3] [4]. Therefore, these operations also share common features and characteristics, like learnable filters or hyperparameters as *kernel size*, *padding* and *stride*. Transposes convolution can be used to enlarge feature maps or dense sparse feature maps, as created by the unpooling method "bed of nails". In [NHH15] Noh observes, that the lower layers of the decoder network handle coarse details (e.g., location, shape and region), while the higher layers capture the fine and more complicated details. This leads to a coarse-to-fine approach for the reconstruction through the decoder network. In literature transposed convolution is also referred to as *deconvolution* [NHH15], *inverse convolution* [BKC17] or *backwards convolution* [LSD15].

Skip Connections

Another architectural component frequently used for the task of image segmentation are skip connections, alternatively also named lateral or shortcut connections. A skip

[3] Vincent Dumoulin and Francesco Visin, 2018, "A guide to convolution arithmetic for deep learning": `https://arxiv.org/abs/1603.07285`

[4] F.-F. Li, J. Johnson and S. Yeung, 2018, "Stanford Lecture Detection and Segmentation": `http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf`

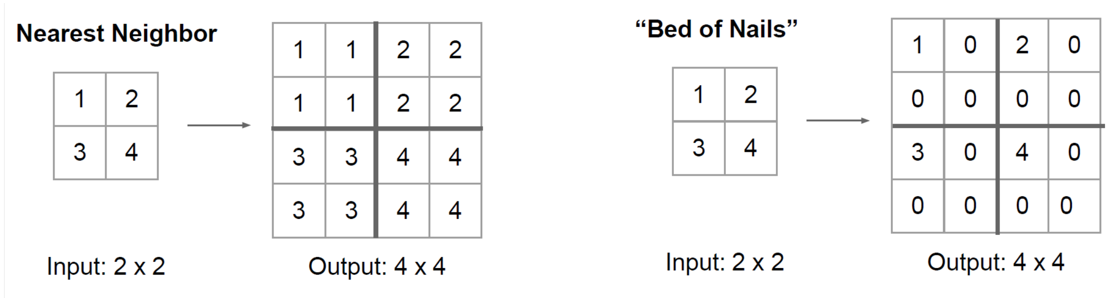

Figure 2.3: Unpooling methods *nearest neighbor* and *bed of nails*. With *nearest neighbor* enlarged feature maps are filled up with the same input value. With *bed of nails* sparse feature maps are created, that contain the input value filled up with zeros. *TODO recreate figure with drawing program*

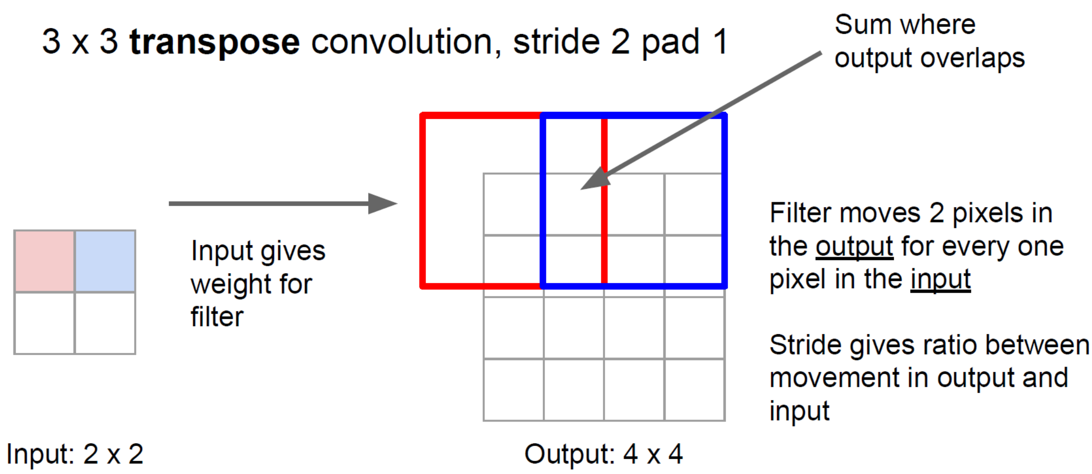

Figure 2.4: Example of the transposed convolution [18] with $kernalsize = 3 \times 3$, $stride = 2$ and $padding = 1$. On the left is the input of size 2×2 px before the application of transposed convolution. On the right is the output of size 4×4 px. The red and blue square visualize the application of the convolution kernel with $stride = 2$. *TODO recreate figure with drawing program*

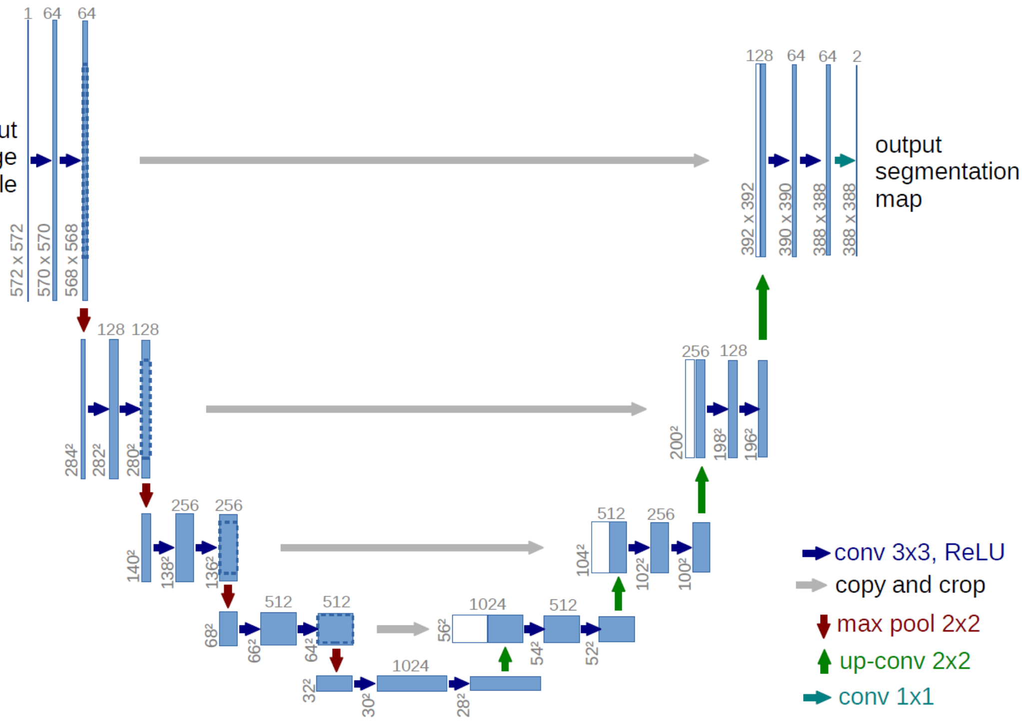

Figure 2.5: U-Net architecture. The left part of the shown network architecture represents the encoder network, while the right part represents the decoder network. In between, the skip connections establish additional lateral links (visualized in gray) between the encoder and decoder network. The skip connections exist on several levels to persistently combine classification and localization information. Copyright ©2015 Springer Nature. Reprinted by permission from [RFB15].

connection is a link between two layers, that are not ordered strictly consecutively. The receiving layer may takes multiple inputs, one from the sequential previous layer and another from the layer connected by the skip connection. These inputs are combined by the concatenation operation [5]. Skip connections can be integrated in other architectures as e.g., the encoder-decoder-architecture from [RFB15] shown in Figure 2.5.

The task of segmentation segmentation aims to answer the questions of classification *What is in the image?* and the question of localization *Where is it in the image?*. While downsampling, the network extracts features and learns to answer the question of

[5] Concatenation is a frequently used operator to fuse multiple layer outputs in the form of tensors into a single tensor, as in PyTorch: `https://pytorch.org/docs/stable/generated/torch.cat.html`

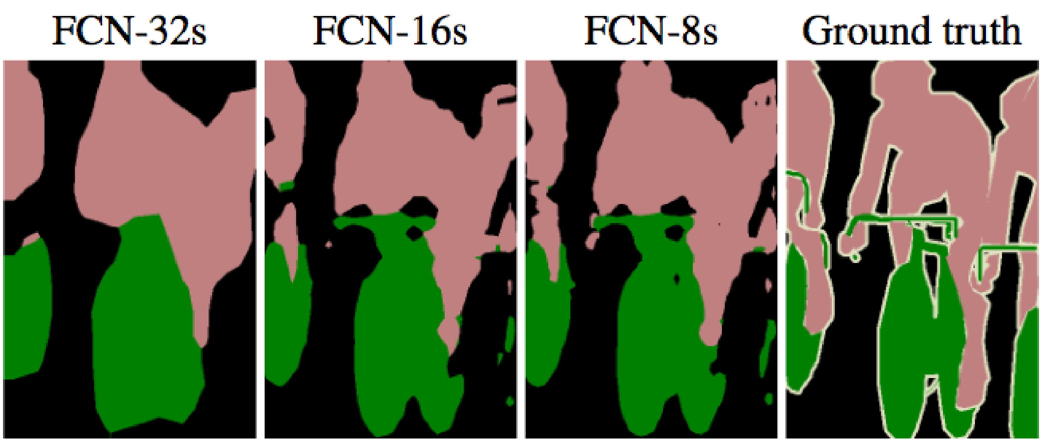

Figure 2.6: Results of several variants of the FCN. It can be observed that the FCN-32s creates a relatively coarse prediction compared to the networks with skip connections. In contrast the FCN-8s achieves the best result with significantly improved level of detail and sharper borders. Copyright ©2015 IEEE. Reprinted by permission from [LSD15].

classification. During this process the size of feature maps decrease and localization information is lost. As a result it gets harder to perform a detailed reconstruction and answers the question of localization. One solution to neutralize this effect, is the adverting from the idea of building a strictly sequential architecture and instead include skip connections. By doing so, layers that still contain localization information can be directly connected to layers that contain the developed classification information.

The Fully Convolutional Network (FCN) introduced by Long, is based on the encoder-decoder-architecture with a relatively small decoder. To compensate the absent of a deep decoder and refine the network, Long applies skip connections, that combine lower layers with the final prediction layer. In order to compare the effect of skip connections three models are created: one without skip-connection (FCN-32s) and two with skip connection (FCN-16s and FCN-8s). The number indicates the upsampling factor required from this point to the final predictions. The FCN-16s and FCN-8s require less upsampling, due to their fusion with a lower layer. The results shown in Figure 2.6 highlight the effect of skip connections in order to solve the question of localization.

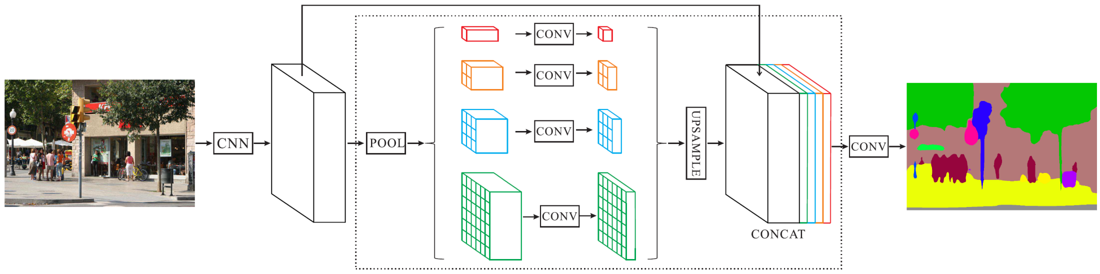

Figure 2.7: PSP Network with the pyramid pooling module in the middle. The pyramid pooling model contains four pyramid levels illustrated in different colors. The respective sizes of the pyramid levels are 1×1, 2×2, 3×3 and 6×6. After the passage of the pyramid levels the results are upsampled and concatenated. The number of pyramid levels and their size can be modified. Copyright ©2015 IEEE. Reprinted by permission from [Zha+17].

Pyramid Scene Parsing Network

In [Zha+17] it is stated that in segmentation architectures the receptive field does not include enough context information. Further, Zhao claims that in order to differentiate between various classes the context information on global and subregion level is useful. To address this problem the Pyramid Scene Parsing (PSP) Network is introduced, that aims to enlarge the receptive field. To improve the context information different subregions can be fused, similar as in [He+15]. In [Zha+17] the pyramid pooling module is proposed, which is a hierarchical structure using multiple processing streams, also referred to as pyramid levels. Each pyramid level applies convolution operations with different filter sizes resulting in feature maps of different pyramid scales. Afterwards the feature maps are upsampled to a mutual size and then concatenated, as illustrated in Figure 2.7.

DeepLab

DeepLab is a DL model for semantic segmentation developed by researchers from Google and first published in [Che+18a]. In order to improve the segmentation result, three main techniques were introduced: Atrous Convolution, Atrous Spatial Pyramid Pooling (ASPP) and Conditional Random Fields (CRF).

- **Atrous convolution** or dilated convolution modifies the kernel used for the convolution operation. The size of the kernel is extended and the upcoming gabs between the parameters are filled up with zeros. The benefit is the coverage of a greater receptive field, without increasing the number of convolution parameters and so the computational load.

- **ASPP** is based on the concept of Spatial Pyramid Pooling (SPP) introduced in [He+15]. SPP aims to combine images of different resolutions in order to obtain multi-scale information without increasing the computation time. ASPP applies atrous convolution to the concept of SPP. A input is applied to several atrous convolution kernels of different sizes and the result is their combined output.

- **CRF** aim to achieve sharper boundaries by considering the surrounding pixels before performing classification. The functionality can be reviewed in detail in [Che+18a] and [KK11]. In contrast to most other segmentation models DeepLab does not use skip connections, but instead relies on CRF in order to recover fine details and the boundaries of objects to answer the question of localization.

2.2.4 Data

As segmentation segmentation is a problem of supervised learning it requires GT. For a dataset to be suitable in the field of DL among others the following criteria should be met: quantity, quality and representation capabilities.

- The quantity of a dataset used for training a DL model is crucial for its success. In general, small datasets, may not cover all vital characteristics to completely map a given objective. It has been shown in [BB01], that the performance of networks can improve significantly using a larger dataset for training. Also, in [HNP09] the effect of larger datasets is examined. It is claimed that, using a larger dataset for training can benefit the networks performance more than modifying the architecture of the network [Gér17]. This highlights the importance of datasets with sufficient quantity to increase the performance of networks.

- The quality of the training data has a high impact on the model performance as well. Data, that is inconsistent, incomplete, erroneous or too noise, can lead to significant decrease in performance [GAD17]. Training with poor quality data makes it more difficult for a model to detect and understand the elemental features and patterns, that are required by a model to perform well [Gér17].

- The capability of dataset to represent a given problem is another elemental characteristic. To enable a model to generalize and perform well, it is essential for the training data to be representative to the problem [Gér17]. The best approach to do so, is to include samples of this specific problem or of samples from the same domain. But instead, often general 'all-use-datasets', like Pascal VOC [Eve+10], COCO [Lin+14] or ImageNet [Den+09], are used as training data on a specific problem, that is not covered within the samples of these datasets. This may results in a decrease of performance, because the capabilities of DL models are strongly connected with the representation of the data [GBC16].

It can be a challenge to obtain a dataset, that meets these criteria. The creation of new image datasets are considered to be very expensive in time and cost. Datasets for image segmentation are even more difficult to create due to the high effort required to label images on pixel-level. Especially, uncommon, restricted or private domains (e.g., medical or industrial domains) are rarely covered in public datasets. For example, the manufacturing process in a closed industrial environment may contain unique objects or uncommon surroundings, that are hardly ever represented in common datasets.

New approaches have been created, in order to facilitate the process of creating new dataset and label images with pixel-level accuracy. An efficient and common way is a program, that simplifies labeling process by providing an user interface and multiple methods to create and save label. These programs are often called *labeltools* or *annotation tools* and due to the high demand on labeled training data there are various labeltools available[6]. To simplify the quite manual process of labeling for a human user there are interactive methods (see Chapter 2.3), that support the applicant in creating a label. Another approach is to create synthetic datasets like the SYNTHIA dataset [Zol+19] and use them to as training data for semantic segmentation [Che+19].

2.2.5 State-of-the-art

An overview about the performance of previously introduced and current state-of-the-art networks is given in Table 2.1. As benchmark dataset the Pascal VOC test set [Eve+10] and as metric the mIoU were selected, due to their widespread usage. Notable is the rapid increase on performance over the last years, which emphasizes the relevance and research interest on this field of study.

2.2.6 Application

Image segmentation finds application in various tasks and is widely used over different domains. Due to its capability to perform classification on pixel-level it is often applied on scene understanding [LSF09] or the evaluation of satellite images [Li+18]. In the field of autonomous driving semantic segmentation is used for street scene analysis

[6]E. Cerna, *Image Annotation Tools: Which One to Pick in 2020?* `https://bohemian.ai/blog/image-annotation-tools-which-one-pick-2020/`

[7]Papers with code, Semantic Segmentation on PASCAL VOC 2012 test: `https://paperswithcode.com/sota/semantic-segmentation-on-pascal-voc-2012`

Model	mIoU (val)	mIoU (test)
FCN [LSD15]	-	62.2
DeconvNet [NHH15]	-	72.5
DeepLab-CRF [Che+18a]	77.7	79.7
PSPNet [Zha+17]	-	85.4
DeepLab3+ [Che+18b]	84.6	89.0
EfficientNet [Zop+20]	90.0	90.5

Table 2.1: Comparison of semantic segmentation models on Pascal VOC 2012. Other overviews show the evolution of the state-of-the-art performance over time [7].

[Cor+16] [MG15] [Neu+17]. In medicine this method can be used to segment cancer cells, tumors [RFB15] or blood cells [Tra+18]. Further, it is applied in order to fulfill abstract tasks like the reconstruction of indoor scenes [Dai+17]. This listing of only some applications gives an idea of how versatile and functional image segmentation is and what can be achieved with it in the future.

2.3 Interactive Segmentation

Image segmentation takes as input x only the image itself, in contrast interactive segmentation takes beside the image some additional information interactively provided by an user as input. This additional information is especially beneficial, because it is manually picked using the valuable image processing capabilities of human users. Due to this, interactive segmentation networks are provided with a high level guidance about the location of the desired object. Depending on the type of interaction, the receipt of the user input may be more or less elaborately, which leads to a fundamental difficulty of interactive methods in general. To perform the user interactionwith in order to obtain the additional input may be considered expensive. Especially, for deep learning tasks a great quantity of images is required.

Interactive methods are mostly applied for the task of instance segmentation. They mostly focus on extracting one object from an image, rather than segmenting a whole image at once. In the following several concepts of interactive segmentation methods are introduced.

2.3.1 Classical Concepts

Before the upcomming of DL and CNNs, segmentation was already performed with classical image processing. These methods also focus on the extraction of a foreground object from the background by little user interaction.

A prominent algorithm is GrabCut [RKB04] published in 2004. As user interaction GrabCut requires a loose bounding box. Everything outside the bounding box and the borders itself are defined as background, while the inside of the box is segmented based on contrast and color information. Further, the goodness of the result may be enhanced by iteratively defining explicit parts as fore- or background.

Another still relevant method to perform instance segmentation is the Watershed algorithm [NS94]. This method interactively collects fore- and background regions from an user in order to perform segmentation. The Watershed algorithm is part of the benchmark study and elaborately examined in Section 3.2.

These methods may perform very well on certain images, but reach their limitations as they deal with more complex structures. This is due to their rather simple processing of superficial characteristics e.g., edges, textures, contrast and color. On the contrary DL based methods are capable to examine images on a deeper level and so understand more complicated structures.

2.3.2 User Point Concepts

[Xu+16] [MVL18] [MKA20] [Hu+19] [Lie+17] [JG16]

[Man+18] [Zha+20]

2.3.3 Polygon Concepts

[Acu+18] [Lin+19]

2.3.4 Drawing Concepts

3 Methods

3.1 Polygon Drawing

The paper "Deep Extreme Cut: From Extreme Points to Object segmentation" from Manisis et al. published in 2018 introduces another method to perform interactive object segmentation [Man+18].

3.1.1 Method Description

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.1.2 Architecture

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.1.3 Refinement

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no

sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.1.4 Results

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2 Watershed

The paper "Deep Extreme Cut: From Extreme Points to Object segmentation" from Manisis et al. published in 2018 introduces another method to perform interactive object segmentation [Man+18].

3.2.1 Method Description

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2.2 Architecture

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no

sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2.3 Refinement

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2.4 Results

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3 Deep Extreme Cut

The paper "Deep Extreme Cut: From Extreme Points to Object segmentation" from Manisis et al. published in 2018 introduces another method to perform interactive object segmentation [Man+18].

3.3.1 Method Description

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no

sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3.2 Architecture

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3.3 Refinement

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3.4 Results

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.4 IOG

The paper "Interactive Object segmentation with Inside-Outside-Guidance"[Zha+20] published by S. Zhang, J. H. Liew, Wei, et al. *et al*. in 2020 provides a state-of-the-art method to perform interactive object segmentation.

3.4.1 Method Description

The execution of this method outputs a binary segmentation for a single object of interest within an image. To segment multiple objects in one image, the method has t be applied for each of them sequentially.

Inside Outside Guidance (IOG) is an interactive segmentation method and hence requires user input. The input is given by a three mouse clicks on the object's foreground and on its background. The procedure is shown in Figure 3.1 and described in the following: first, in order to form an *"almost-tight bounding box"*[Zha+20, p. 12235] two exterior clicks are set at the two diagonal locations corners of the object (top-left and bottom-right or bottom-left and top-right). Based on these two points the other two corner points are derived, which leads to four points on the background. Second, to define the object inside the bounding box a single click around the center of the desired object is made, this click is processed as foreground point. The background points *"provide "outside" guidance (indicating the background regions) while the interior click gives an "inside" guidance (indicating the foreground region), thus giving the name Inside-Outside-Guidance"*[Zha+20, p. 12235].

These three points are preprocessed before they are input to the actual model. To include context from the surrounding region the bounding box is enlarged by p_{box} pixels. In order to focus on the object of interest the enlarged bounding box is cropped and resized to the size of 512×512 px. For background and foreground points, a separate heatmap is created by centering a 2D Gaussian at each point with

$$Gauss = \frac{\exp -4 * \log 2}{\sigma^2} \tag{3.1}$$

The two heatmaps have the size of 512×512 px and are concatenated with the input RGB image to create a 5-channel input for the model.

3.4.2 Architecture

The architecture of the IOG method is based on a *"coarse-to-fine design"*[Zha+20, p. 12237] (see Figure 3.2), containing two main parts: the CoarseNet and the FineNet.

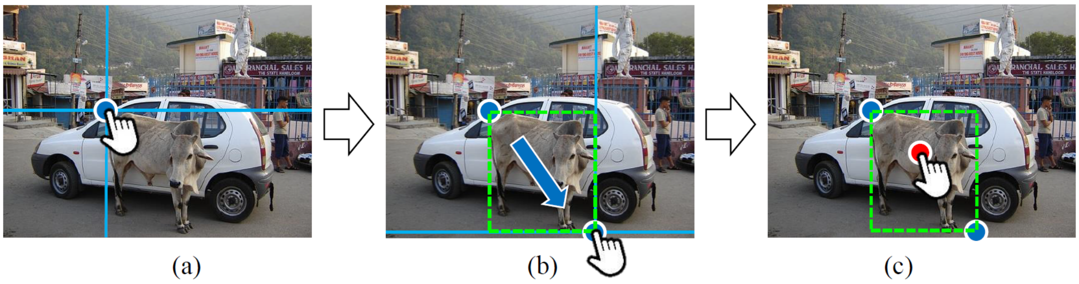

Figure 3.1: Procedure of setting the three IOG clicks [Zha+20]. Set the two background clicks (blue) at the diagonal corner locations of the object. Gather a bounding box based on the background clicks. Set a foreground point (red) at the middle of the object.

CoarseNet The CoarseNet contains the heavy encoder part, that mainly consists of a classifier often referred to as backbone. In IOG a ResNet-101 [He+16] is used . This ResNet-101 is implemented without the head of fully connected layers. It contains four ResNet blocks and the fourth block outputs 2048 feature maps of the size 32×32 px. After the backbone a PSP-network is applied in order to enrich "the representation with global contextual information"[Zha+20]. The coarse prediction from the PSP-Network [Zha+17] has a spatial dimension of 32×32 px with 512 feature maps. From this onward the layers are enlarged by a four staged upsampling process to obtain the original input size of 512×512 px. During the upsampling process activations from the residual parts of the ResNet are transferred from the ResNet using so lateral connections and concatenated with the upsampled feature maps. A benefit of this architecture is the fusion of information from different network stages.

FineNet The FineNet is based on a "multi-scale fusion structure"[Zha+20]. The activations from all four stages of the upsampling process from the CoarseNet are further processed along different paths. Depending on the spatial dimension, a number of additional convolution and upsampling operations are applied in order to use *"features at deeper layers for better trade-off between accuracy and efficiency"* [Zha+20, p. 12237]. These different paths are concatenated to create the networks final layer. A sigmoid is applied to this final layer, which results in a probability map as final prediction of the IOG network. The author shows in an ablation study, that the FineNet enhances the networks IoU by 0.8%. The ablation study is performed with a ResNet-50 as backbone and PASCAL-1k [Eve+10] as dataset.
This architecture especially performs well due to its application of lateral connections from different levels in order to recover local detail. The combination of layers with

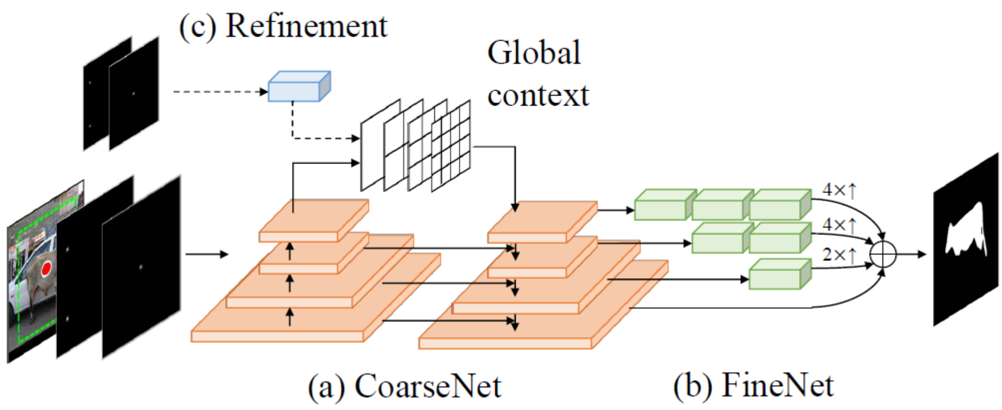

Figure 4. **Network Architecture.** (a)-(b) Our segmentation network adopts a coarse-to-fine structure similar to [14], augmented with a pyramid scene parsing (PSP) module [68] for aggregating global contextual information. (c) We also append a lightweight branch before the PSP module to accept the additional clicks input for interactive refinement.

Figure 3.2: IOG architecture (not final).

high localization detail with the layers, that contain high detection details, is helpful to prevent a information loss during the down- and upsampling process.

3.4.3 Refinement

If a segmentation results does not meet the user's expectations a refinement can be performed iteratively. This is done by an additional user click, which can be a fore- or background click on the region with the greatest error. In the refinement iteration of the model, this new point is processed in the same way as the initial user click positions to create a heatmap for fore- and background. These two heatmaps are combined into a two-channel input, which is processed in a so called "lightweight-branch". In this branch five convolution operations are applied and the result is concatenated with the ResNet's output of the first iteration. Hence, the ResNet does not require another execution and leads to a fast refinement process. Further, the normal IOG process is executed from the PSP-module. Zhang states that the usage of the lightweight-branch performs better than adding the refinement click into the normal 5-channel input.

In their experiments Zhang compares the IOG method to other state-of-the-art methods on different benchmarks, as shown in Figure They also evaluate the generalization abilities of IOG on unseen classes. Zhang claims that IOG outperforms all other methods.

List of Figures

List of Tables

Acronyms

AI Artificial Intellegence.

ASPP Atrous Spatial Pyramid Pooling.

CNN Convolutional Neural Network.

CRF Conditional Random Fields.

DL Deep Learning.

FCN Fully Convolutional Network.

GPU Graphical Processing Unit.

GT Ground Truth.

IOG Inside Outside Guidance.

IoU Intersection over Union.

mIoU mean Intersection over Union.

ML Machine Learning.

OP Overall Pixel.

PC Per-Class.

PSP Pyramid Scene Parsing.

SPP Spatial Pyramid Pooling.

Bibliography

[18] *Detection and segmentation*. May 1, 2018. URL: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf.

[Acu+18] D. Acuna, H. Ling, A. Kar, and S. Fidler. "Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 859–868. DOI: 10.1109/CVPR.2018.00096.

[BB01] M. Banko and E. Brill. "Scaling to Very Very Large Corpora for Natural Language Disambiguation." In: *Annual Meeting of the Association for Computational Linguistics*. 2001, pp. 26–33. DOI: 10.3115/1073012.1073017.

[BKC17] V. Badrinarayanan, A. Kendall, and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[Che+18a] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "DeepLab: segmentation Image segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[Che+18b] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-Decoder with Atrous Separable Convolution for segmentation Image segmentation." In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 833–851. DOI: 10.1007/978-3-030-01234-2_49.

[Che+19] Y. Chen, W. Li, X. Chen, and L. Van Gool. "Learning segmentation segmentation From Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1841–1850. DOI: 10.1109/CVPR.2019.00194.

[Cor+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The Cityscapes Dataset for segmentation Urban Scene Understanding." In: *IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR). 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[CP13] D. Csurka Gabriela andLarlus and F. Perronnin. "What is a good evaluation measure for segmentation segmentation?" In: *British Machine Vision Conference (BMVC)*. 2013, pp. 32.1–32.11. DOI: 10.5244/C.27.32.

[Dai+17] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2432–2443. DOI: 10.1109/CVPR.2017.261.

[Den+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A large-scale hierarchical image database." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[Eve+10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes (VOC) Challenge." In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338.

[GAD17] V. Gudivada, A. Apon, and J. Ding. "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations." In: *International Journal on Advances in Software* 10.1 (2017), pp. 1–20.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016.

[Gér17] A. Géron. *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2017. ISBN: 978-1-491-96229-9.

[He+15] K. He, X. Zhang, S. Ren, and J. Sun. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 37.9 (2015), pp. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824.

[He+16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[HNP09] A. Halevy, P. Norvig, and F. Pereira. "The Unreasonable Effectiveness of Data." In: *IEEE Intelligent Systems* 24.2 (2009), pp. 8–12. DOI: 10.1109/MIS.2009.36.

[Hu+19] Y. Hu, A. Soltoggio, R. Lock, and S. Carter. "A fully convolutional two-stream fusion network for interactive image segmentation." In: *Neural Networks* 109 (2019), pp. 31–42. DOI: 10.1016/j.neunet.2018.10.009.

[Jad20] S. Jadon. "A survey of loss functions for semantic segmentation." In: *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[JG16] S. Jain and K. Grauman. "Click Carving: Segmenting Objects in Video with Point Clicks." In: *AAAI Conference on Human Computation and Crowdsourcing*. Vol. 4. 1. 2016.

[KK11] P. Krähenbühl and V. Koltun. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials." In: *Neural Information Processing Systems (NIPS)*. Vol. 24. Curran Associates, Inc., 2011, pp. 109–117.

[Li+18] W. Li, C. He, J. Fang, and H. Fu. "segmentation segmentation Based Building Extraction Method Using Multi-source GIS Map Datasets and Satellite Imagery." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 233–2333. DOI: 10.1109/CVPRW.2018.00043.

[Lie+17] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. "Regional Interactive Image Segmentation Networks." In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2746–2754. DOI: 10.1109/ICCV.2017.297.

[Lin+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft COCO: Common Objects in Context." In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

[Lin+19] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler. "Fast Interactive Object Annotation With Curve-GCN." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5252–5261. DOI: 10.1109/CVPR.2019.00540.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. "Fully Convolutional Networks for segmentation segmentation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. DOI: 10.1109/CVPR.2015.7298965.

[LSF09] L.-J. Li, R. Socher, and L. Fei-Fei. "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 2036–2043. DOI: 10.1109/CVPR.2009.5206718.

[Man+18] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. “Deep Extreme Cut: From Extreme Points to Object segmentation.” In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 616–625. DOI: 10.1109/CVPR.2018.00071.

[MG15] M. Menze and A. Geiger. “Object scene flow for autonomous vehicles.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3061–3070. DOI: 10.1109/CVPR.2015.7298925.

[MKA20] S. Majumder, A. Khurana, and A. R. and Angela Yao. “Multi-stage Fusion for One-Click Segmentation.” In: *DAGM German Conference on Pattern Recognition (GCPR)*. Vol. 12544. Lecture Notes in Computer Science. 2020, pp. 174–187. DOI: 10.1007/978-3-030-71278-5_13.

[MVL18] S. Mahadevan, P. Voigtlaender, and B. Leibe. “Iteratively Trained Interactive Segmentation.” In: *British Machine Vision Conference (BMVC)*. 2018, p. 212.

[Neu+17] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. “The Mapillary Vistas Dataset for segmentation Understanding of Street Scenes.” In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5000–5009. DOI: 10.1109/ICCV.2017.534.

[NHH15] H. Noh, S. Hong, and B. Han. “Learning Deconvolution Network for segmentation segmentation.” In: *IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1520–1528. DOI: 10.1109/ICCV.2015.178.

[NS94] L. Najman and M. Schmitt. “Watershed of a Continuous Function.” In: *Signal Processing* 38.1 (1994), pp. 99–112. DOI: 10.1016/0165-1684(94)90059-0.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image segmentation.” In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. “"GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts.” In: *ACM SIGGRAPH 2004 Papers*. 2004, pp. 309–314. DOI: 10.1145/1186562.1015720.

[SM18] R. Shanmugamani and S. Moore. *Deep Learning for Computer Vision*. Birmingham, United Kingdom: Packt Publishing, 2018. ISBN: 9781788295628.

[SZ15] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In: *3rd International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. 2015.

[Tra+18] T. Tran, O.-H. Kwon, K.-R. Kwon, S.-H. Lee, and K.-W. Kang. "Blood Cell Images segmentation using Deep Learning segmentation segmentation." In: *IEEE International Conference on Electronics and Communication Engineering (ICECE)*. 2018, pp. 13–16. DOI: 10.1109/ICECOME.2018.8644754.

[Xu+16] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. "Deep Interactive Object Selection." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 373–381. DOI: 10.1109/CVPR.2016.47.

[Zha+17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. "Pyramid Scene Parsing Network." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239. DOI: 10.1109/CVPR.2017.660.

[Zha+20] S. Zhang, J. H. Liew, Y. Wei, S. Wei, and Y. Zhao. "Interactive Object segmentation With Inside-Outside Guidance." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12231–12241. DOI: 10.1109/CVPR42600.2020.01225.

[Zol+19] J. Zolfaghari Bengar, A. Gonzalez-Garcia, G. Villalonga, B. Raducanu, H. Habibi Aghdam, M. Mozerov, A. M. Lopez, and J. van de Weijer. "Temporal Coherence for Active Learning in Videos." In: *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 914–923. DOI: 10.1109/ICCVW.2019.00120.

[Zop+20] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le. "Rethinking Pre-training and Self-training." In: *Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 3833–3845.