Karlsruhe Institute of Technology

INSTITUTE OF INFORMATION SYSTEMS AND MARKETING

Designing an Interactive Click Segmentation Labeling Tool

Seminar Thesis

Paul Theuer

2053912

At the Department of Economics and Management
at the Institute of Information Systems and Marketing (IISM)
Information Systems 1

Reviewer:	Prof. Dr. Alexander Mädche
Advisor:	Merlin Knäble

14th of March 2023

Contents

1. Introduction

Computer Vision (CV) has become an essential tool in many fields, such as medicine, transportation, and robotics, due to its ability to analyze visual information in real-time. With the rise of Machine Learning (ML), the performance of CV models has been significantly improved, with scaling laws suggesting that the more data used to train a model, the better it performs. However, labeling data for ML models, especially for segmentation tasks, is often costly and tedious.

CV tasks can be broadly categorized into image classification, object detection, semantic segmentation, and instance segmentation. While image classification can assign a label to a whole image, object detection and segmentation aim to localize and distinguish individual objects within an image. Semantic segmentation goes one step further and assigns each pixel in an image a class label.

Image segmentation is a task that can quickly become expensive due to the cost of labeling data. For example, the company "mindkosh" charges 3$ per segmented image[1], which can quickly add up for large datasets. By using an interactive segmentation tool like the one proposed in this thesis, the number of required manual labels can be reduced, potentially leading to significant cost savings. The proposed model, averages around 3.9 clicks per image for segmentation[2]. As an upper bound estimate, this could allow for the segmentation of up to 900 images per hour, resulting in potential cost savings of around 2.700 $ per hour. These numbers demonstrate the importance of developing efficient and effective tools for image segmentation.

Although there exist numerous tools for CV tasks, many lack usability, model accuracy, or both, creating research gaps in this area. Moreover, many existing tools are challenging to use for non-machine learning experts, requiring a high level of technical knowledge. To address these issues, the goal of this seminar thesis is to build an interactive web application for image segmentation using click-based annotation.

The research question addressed in this thesis is, "How to design an interactive segmentation tool to increase labeling efficiency while demanding minimal mental workload?" This thesis aims to explore various design strategies and interface features that could potentially increase labeling efficiency while reducing the cognitive load of the user. Through a lightweight user studies and evaluations, we seek to determine the optimal design of an interactive segmentation tool that can improve the labeling efficiency of CV tasks.

To answer this research question, we propose a solution that focuses on a strong baseline model, with high quality training data, that will carry our segmentation task. We also suggest keeping the interface plain and simple, hiding some of the options, and making easy options available. This way, non-machine learning experts will be able to use the tool without extensive training. By implementing these design choices, we aim to create a user-friendly tool that will increase the efficiency of image segmentation labeling tasks while requiring minimal mental workload.

[1] `https://mindkosh.com/annotation-services/cost-estimator.html`
[2] Average taken as total average from Figure 2.5

2. Related Work

In the following sections we will dive into some machine learning models that provide the basis for our application. In doing so, basic concepts will be highlighted and the state of research will be summarized for some key areas in computer vision. As we outline existing models and software artifacts, a number of research gaps and limitations will be addressed.

2.1 Interactive Click Image Segmentation

Interactive click-based image segmentation is a technique for separating objects from their background in an image. In literature we find a broad number of approaches and models in order to tackle this specific segmentation task. However, one can observe common trends when it comes to involving users. As the name suggests, user interaction is often realized with simple clicks.

The basic idea revolves around an interactive and iterative masking task performed by the machine, whilst having the user place positive or negative clicks on the target object. Positive clicks signify the inclusion of a specific region in the segmentation mask, while negative clicks convey the opposite. By involving this user driven feedback loop the system adjust the boundaries of the object until the desired segmentation is achieved. This process is repeated until the user is satisfied with the results.

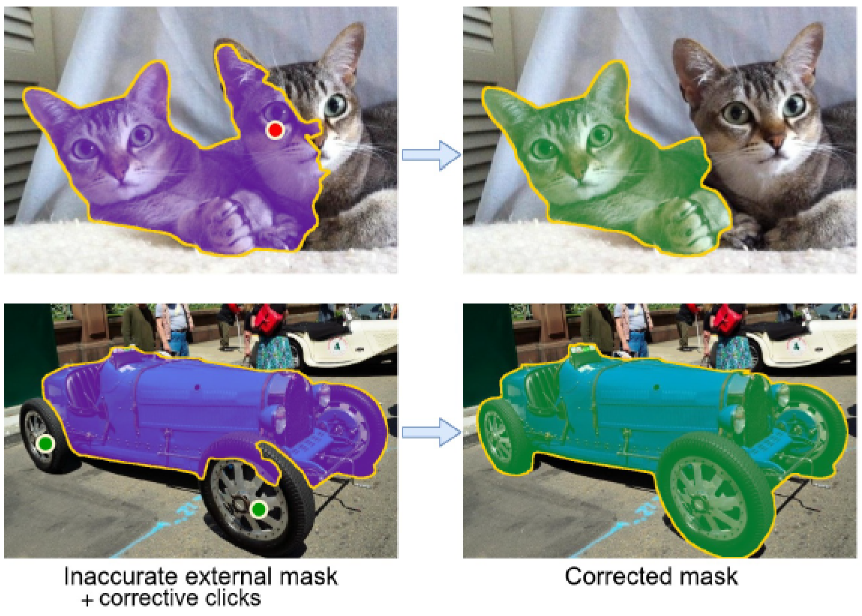


Figure 2.1: Positive (green) and negative clicks (red) Sofiiuk et al. (2022)

As seen in figure 2.1, by employing this strategy the user can fix “false negative and false positive regions with positive and negative clicks respectively” (Sofiiuk et al., 2022, p. 2).

One could argue that click based interaction methods are quite effective when it comes to user workload, as they employ quite an intuitive and simple way to specify the desired object. On the one hand there is no necessity for exhaustive pixel by pixel labeling, on the other hand the concept of positive and negative clicks is quite easy to understand even for non machine learning experts.

Although different models tend to be sharing a number of characteristics, we can spot differences when it comes to implementing different modeling strategies. We can partly attribute this to the nature of being a research heavy field, but also to the fact that there exists no “silver bullet solution”.

Chen et al. (2022) attempt to address deficiencies of previous models in two areas of weakness. Since many preexisting models are not efficient enough, they focus on making their model efficient on low power-devices. “A good annotation tool is expected to produce fine masks with quick responses. Most previous works only focus on accuracy and take advantage of big models and high-resolution inputs” (Chen et al., 2022, p. 1). As a result, these segmentation tools can not be used on devices with less computing power, such as laptops, edge devices or even web applications, which subsequently eliminates a vast amount of potential users by default.

The second deficiency most models tend to show is ignoring preexisting masks. Chen et al. (2022) argue that preexisting masks are often provided by offline models or other forms of pre-processing. Making modifications on these masks can facilitate the annotation process.

With “Focal Click” Chen et al. (2022) address both issues by predicting and updating the mask in localized areas, and by decomposing the slow prediction process into two fast inferences on small crops. The approach also includes a sub-task called Interactive Mask Correction that enables the refinement of preexisting masks, using a technique called Progressive Merge that takes into account morphological information (characteristics and properties of shapes and structures within an image) to decide where to preserve and where to update.

Figure 2.2 illustrates this process. Upon receiving a new click, Focal Click proceeds with a two-step process. Firstly, it selects a Target Crop (indicated by the yellow box) based on the previous mask and performs coarse segmentation. The coarse segmentation step is intended to provide an initial estimate of the target object’s rough shape and location. Following this, it identifies a Focus Crop (indicated by the red box) based on the maximum difference region between the coarse prediction and the previous mask and executes refinement on it. Finally, Progressive Merge is used to update part of the new prediction Chen et al. (2022).

As a result the authors achieve competitive results with significantly smaller FLOPs (floating-point operations per second) compared to state-of-the-art methods and shows significant superiority in correcting preexisting masks Chen et al. (2022).

Z. Lin, Zhang, Chen, Cheng, and Lu (2020) focus on a different aspect in interactiave image segmentation. The authors discuss how current methods treat all interaction points equally, disregarding the importance of the first click. The first click is critical for providing the location and main body information of the target object. To address this issue, they propose a deep learning framework called First Click Attention Network (FCA-Net), which

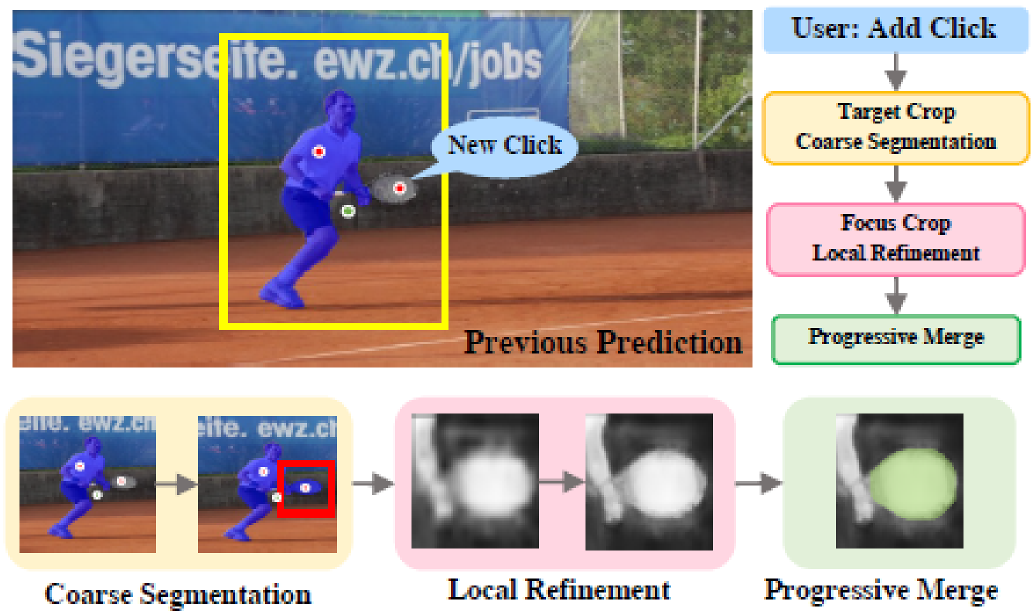


Figure 2.2: Focal Click (Chen et al., 2022)

makes better use of the first click to improve the interactive segmentation result Z. Lin et al. (2020).

This approach seems very intuitive, since the first click the user places will always be on the target object itself. By giving this click extra weight the first click can serve “as a segmentation anchor, while the other points play an auxiliary role for detail repair” (Z. Lin et al., 2020, p. 13339).

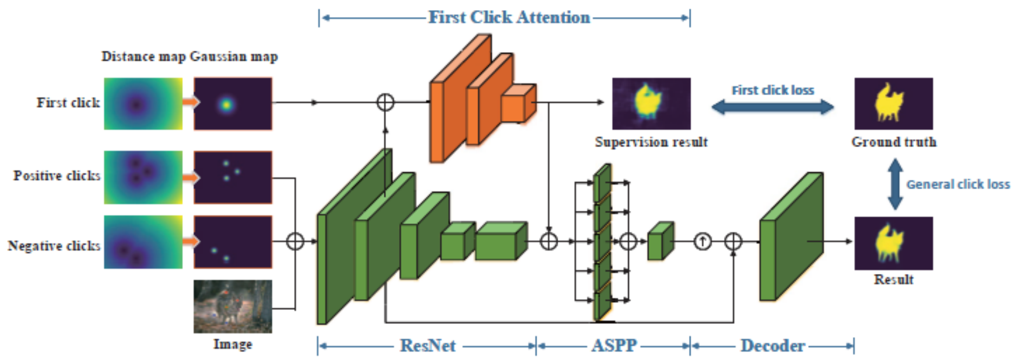


Figure 2.3: FCA Net (Z. Lin et al., 2020)

Figure 2.3 depicts the overall architecture of the FCA-Net. It is divided into two parts: the basic segmentation network (green) and the FCA-Module (orange). Besides the image itself, positive and negative clicks are used as inputs.

One common approach to encoding user clicks are Gaussian distance maps. For each user click, a Gaussian function is defined around the clicked location, where the center of the Gaussian corresponds to the clicked location, and the standard deviation of the Gaussian determines the size of the region around the click. The Gaussian function is then evaluated at every pixel location in the image, producing a distance map that encodes the distance of each pixel to the nearest user click. Pixels that are closer to a click have higher values in the distance map, while pixels that are farther away have lower values. The distance maps from multiple clicks are then combined to create a final probability map that can be used as input to a segmentation backbone. This probability map assigns high probabilities to pixels that are close to user clicks and low probabilities to pixels that are far away, providing guidance to the segmentation algorithm.

The network highlighted in green consists of three modules. The ResNet module is a type of convolutional neural network (CNN) that is regularly used as a backbone in many deep learning models, including segmentation networks. As the name indicates, ResNet uses residual connections to improve the flow of gradients during backpropagation, allowing for deeper networks to be trained more effectively. The ASPP (Atrous Spatial Pyramid Pooling) module is used to capture multi-scale context information by applying filters at different dilation rates to the input feature map. This allows the network to capture both fine and coarse details in the input image. Finally the Decoder module takes the output of the ASPP module and upsamples it to the same size as the input image. The decoder is typically composed of one or more upsampling layers followed by a series of convolutional layers that learn to map the low-resolution feature maps to high-resolution segmentation masks.

The heart of the FCA-Net lies in its First Click Attention Module. It is designed to utilize the guidance information of the first click in interactive image segmentation. It takes the low-level features and the Gaussian map centered on the first click as input, and uses a series of convolutional layers to extract features that focus on the pixels around the first click. The output features are merged into the basic segmentation network before the ASPP module. Additionally, the module is supervised with a first click loss, which focuses on the pixels around the first point. It is worth noting that the FCA Module is distinct from the Attention Mechanism commonly employed in language models.

The authors show how the FCA-Net provides the following benefits:

Focus invariance: The FCA-Net's first click attention module is designed to be focus invariant. This means that the model is able to maintain accuracy even if the user clicks on a different location within the object to be segmented. Since the module is designed to prioritize the first click, it is able to maintain focus on the object regardless of the specific location of the click.

Location guidance: The FCA-Net provides location guidance by using the user's first click to create a Gaussian distance map that guides the segmentation process. This enables the model to take into account the user's intentions and preferences, resulting in more accurate and personalized segmentation results.

Error-tolerant ability: The FCA-Net is designed to be error-tolerant, meaning that it is able to maintain accuracy even in the presence of user error. For example, if the user accidentally clicks on a background region instead of the object of interest, the FCA-Net is still able to produce accurate segmentation results by relying on the other clicks and the Gaussian distance map generated from the first click. This error-tolerant ability enables the model to provide more reliable and consistent segmentation results, even in the presence of user error Z. Lin et al. (2020).

One of the most recent studies Sofiiuk et al. (2022) has been able to achieve higher accuracy scores than the FCA-Net model by implementing some straightforward yet effective strategies. Firstly they find that choosing a strong baseline model greatly improves the performance for segmentation results. By using "diverse large datasets with fine masks for training" (Sofiiuk et al., 2022, p. 2) the authors are able to boost the models' stability. Obtaining this dataset is achieved by combining the LVIS and COCO dataset. The LVIS dataset is a good choice for training models, but it is lacking in general object categories due to being long-tailed (distribution of examples across different categories or classes are highly imbalanced), which can affect the accuracy and generalization of the trained model. To address this, the authors propose augmenting LVIS labels with masks from the COCO dataset, which contains more common and general objects. The authors describe a procedure to construct the combined COCO+LVIS dataset with diverse object classes. The procedure involves "joining all masks from both datasets except for those masks from COCO that have a corresponding mask from LVIS with an intersection over union (IoU) score larger than 80%" (Sofiiuk et al., 2022, p. 6), in which case they only keep the mask from LVIS. This results in a dataset with 104k images and 1.6M instance-level masks Sofiiuk et al. (2022).

Just like Z. Lin et al. (2020), Sofiiuk et al. (2022) chose to encode the user input clicks as gaussian disks, as ablation studies show that this form of click encoding outperforms other methods (e.g. distance maps). However, they chose a different method when it comes to feeding the user input to the segmentation backbone. Typically, these backbones are pre-trained on ImageNet and take only RGB images as input. One common approach is to modify the first convolutional layer to accept N-channel input. In this work, the authors propose a new approach called Conv1S, which introduces a convolutional block that outputs a tensor of the same shape as the first convolutional block in the backbone, which is then summed element-wise with the output of the first backbone convolutional layer. This modification allows for a different learning rate for new weights without affecting pre-trained backbone weights. The proposed approach is illustrated in Figure 2.4.

According to the suggestion made by Chen et al. (2022), the inclusion of preexisting masks can effectively enhance the segmentation process as they provide additional information. In addition, Sofiiuk et al. (2022) further extend this approach by combining result masks from prior clicks and using them as input for the next predicted mask.

For their loss function Chen et al. (2022) propose normalized focal loss (NFL). Focal loss was introduced by T.-Y. Lin, Goyal, Girshick, He, and Dollár (2017). The basic idea behind focal loss (2.1) is to give more weight to misclassified examples that are difficult to classify correctly, while reducing the weight of easy examples. This helps to address the issue of class imbalance in datasets, where the number of examples in one class may

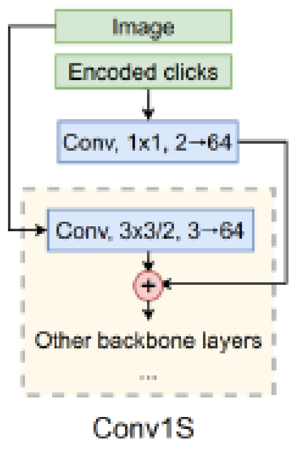


Figure 2.4: Conv1S (Sofiiuk et al., 2022)

be much larger or smaller than that of another class and can subsequently speed up the training process.

$$FL(i, j) = -(1 - p_{i,j})^y * log(p_{i,j}) \tag{2.1}$$

$P(i, j)$ denotes the confidence of prediction at the point (i, j) and y is a focusing parameter that controls the degree of emphasis given to hard examples. However, “one can notice that the total weight $P(\hat{M}) = \sum_{i,j}(1-p_{i,j})^y$ decreases when the accuracy of the prediction increases” (Chen et al., 2022, 5). To tackle this problem the authors introduce a constant to the formula (2.2) that aims to normalize the gradient which in turn allows for faster convergence and better accuracy.

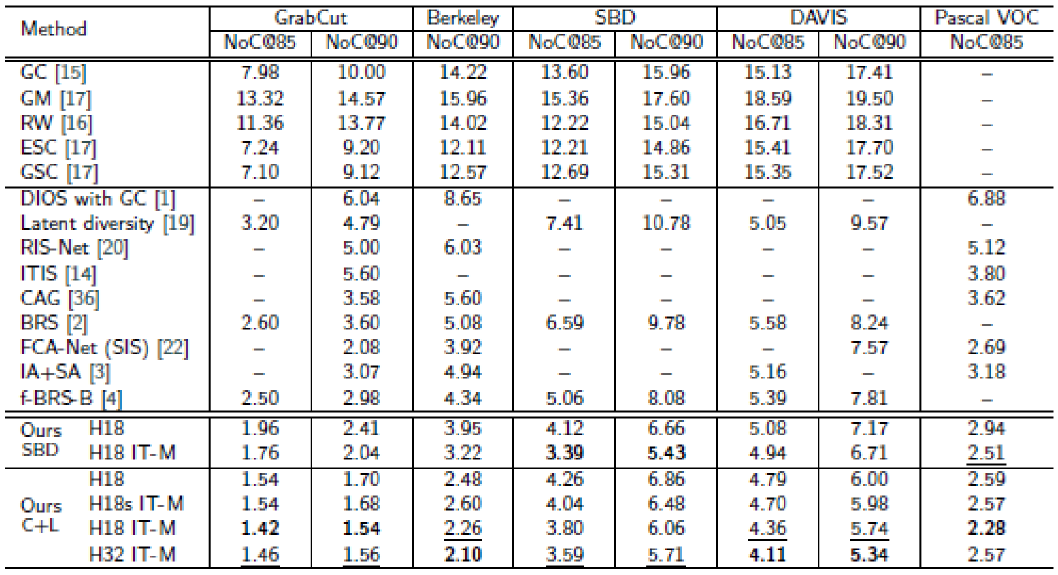
$$NFL(i, j) = \frac{1}{P(\hat{M})} - (1 - p_{i,j})^y * log(p_{i,j}) \tag{2.2}$$

The segmentation results depicted in figure 2.5 show that these rather simple adjustments to the model prove to be quite effective.

Evaluation was perfomed on the Datasets GrabCut, Berkeley, SBD (Semantic Boundaries Dataset), DAVIS and Pascal VOC. The evaluation metrics NoC85 and NoC90, are commonly used and describe the average Number of Clicks needed for an IoU of 85% and 90% respectively. Different models with a number of segmentation backbones (HrNet18, HrNet32, etc.) and training datasets (SBD, COCO+LVIS) are compared to other existing interactive click models, such as the FCA-Net described earlier. As illustrated by the bold numbers, different iteration of their model tend to outperform others across all datasets.

Adding to their model, the authors furthermore provide an Interactive Segmentation Demo Application depicted in 2.6. The App is based on TkInter library and its Python bindings.

Although the application performs well for interactive click segmentation, we have identified some weaknesses related to affordance, intuition, and user involvement. Firstly, the interface’s overwhelming number of buttons and customization options on the right side



Method		GrabCut		Berkeley	SBD		DAVIS		Pascal VOC
		NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85
GC [15]		7.98	10.00	14.22	13.60	15.96	15.13	17.41	–
GM [17]		13.32	14.57	15.96	15.36	17.60	18.59	19.50	–
RW [16]		11.36	13.77	14.02	12.22	15.04	16.71	18.31	–
ESC [17]		7.24	9.20	12.11	12.21	14.86	15.41	17.70	–
GSC [17]		7.10	9.12	12.57	12.69	15.31	15.35	17.52	–
DIOS with GC [1]		–	6.04	8.65	–	–	–	–	6.88
Latent diversity [19]		3.20	4.79	–	7.41	10.78	5.05	9.57	–
RIS-Net [20]		–	5.00	6.03	–	–	–	–	5.12
ITIS [14]		–	5.60	–	–	–	–	–	3.80
CAG [36]		–	3.58	5.60	–	–	–	–	3.62
BRS [2]		2.60	3.60	5.08	6.59	9.78	5.58	8.24	–
FCA-Net (SIS) [22]		–	2.08	3.92	–	–	–	7.57	2.69
IA+SA [3]		–	3.07	4.94	–	–	5.16	–	3.18
f-BRS-B [4]		2.50	2.98	4.34	5.06	8.08	5.39	7.81	–
Ours SBD	H18	1.96	2.41	3.95	4.12	6.66	5.08	7.17	2.94
	H18 IT-M	1.76	2.04	3.22	**3.39**	**5.43**	4.94	6.71	2.51
Ours C+L	H18	1.54	1.70	2.48	4.26	6.86	4.79	6.00	2.59
	H18s IT-M	1.54	1.68	2.60	4.04	6.48	4.70	5.98	2.57
	H18 IT-M	**1.42**	**1.54**	2.26	3.80	6.06	4.36	5.74	**2.28**
	H32 IT-M	1.46	1.56	**2.10**	3.59	5.71	**4.11**	**5.34**	2.57

Figure 2.5: Model performance (Sofiiuk et al., 2022)

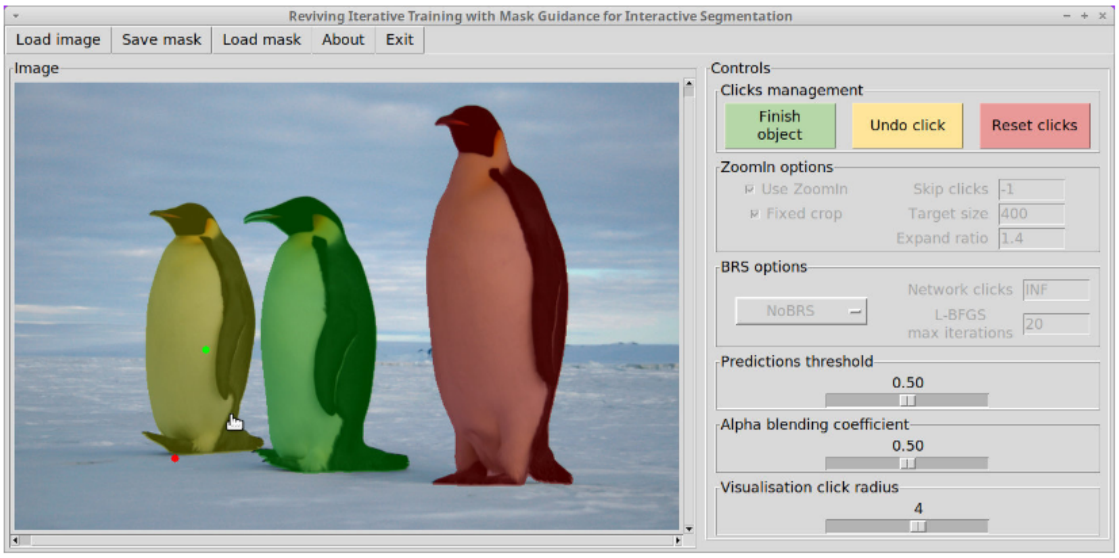


Figure 2.6: TKinter Interactive Segmentation Demo
`https://github.com/SamsungLabs/ritm_interactive_segmentation`

can confuse the average user, despite being helpful for machine learning experts. To cater to a broader user group, most customization modules should be hidden. Secondly, the application only allows users to load one image at a time, making the segmentation process tedious when dealing with multiple images. We aim to simplify the process by enabling users to upload and segment multiple images easily. Additionally, while the demo provides an option to segment images and save their masks, most machine learning models require instance labels, which we plan to implement by allowing users to choose labels from classification models and add them to their segmentation results to enable semantic image segmentation.

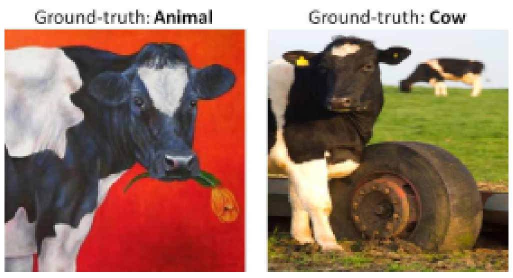
Our goal is to build on the demo application and maintaining the model's performance while ensuring that users can easily participate without requiring expertise in the field of machine learning.

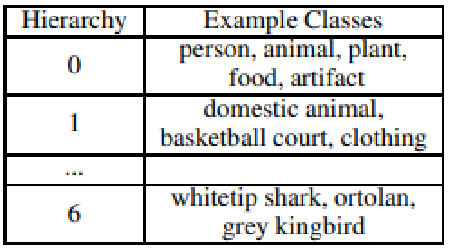
2.2 Image Classification

In order to perform semantic image segmentation, some sort of object labeling model has to be employed. Although we find several good open source image classification models like ResNet, Inception, MobileNet, etc. Ridnik, Ben-Baruch, Noy, and Zelnik-Manor (2021) show some promising results with models pretrained on the ImageNet21k Dataset. The ImageNet21K dataset is a large-scale image dataset containing over 12 million images belonging to around 11.000 classes. It is an extension of the popular ImageNet dataset, which contains about 1.2 million images belonging to 1.000 classes. The images in the ImageNet21K dataset were collected from the web and cover a wide range of visual concepts, such as animals, vehicles, natural scenery, and human activities.

Although the larger and more diverse ImageNet-21K dataset has the potential to be even more effective for pretraining, it is not used as often due to its complexity, limited accessibility, and underestimation of its value Ridnik et al. (2021). The authors bridge that gap by creating an efficient and accessible pretraining process for ImageNet-21K that uses "WordNet hierarchical structures and a novel training scheme called semantic softmax" (Ridnik et al., 2021, 1).

A major problem with the ImageNet21k dataset is it's inconsistency when it comes to class labels in hierachical structures. The tagging methodology does not guarantee that each image is labeled at the highest possible hierarchy. For example, an image of a cow may be labeled as "cow" or "animal," even though "animal" is a semantic ancestor of "cow" (Figure 2.7). This incomplete tagging methodology is common in large datasets, and it makes the training process more difficult Ridnik et al. (2021).





Hierarchy	Example Classes
0	person, animal, plant, food, artifact
1	domestic animal, basketball court, clothing
...	
6	whitetip shark, ortolan, grey kingbird

Figure 2.7: Inconsistent labels due to class hierarchies Ridnik et al. (2021)

In order to address this problem Ridnik et al. (2021) introduce the "Semantic Softmax Training Scheme". In this method, instead of using just one softmax to label an image, they use 11 different softmax layers, each corresponding to a different level of a hierarchy of labels. This allows the network to better understand the different labels that can be applied to an image. However, since not every image is labeled for every level of the hierarchy, they only activate the relevant softmax layers for each image, to avoid confusion. For example, if an image is labeled for hierarchy 5, they will only activate the first 6 softmax layers.

By employing this innovative approach, the authors were able to attain state-of-the-art performance in image classification.

2.3 Interactive Bounding Box Labeling

The application in the master thesis written by Elbarbary (2021) serves as another reference point for our project, as it shares similar modules and architecture. In his work Elbarbary (2021) proposes an interactive labelling system based on bounding boxes. Bounding boxes are rectangular boxes that are drawn around objects in images or videos to indicate their location and size. They are commonly used in object detection tasks in computer vision and machine learning. A bounding box typically consists of four coordinates: the x and y coordinates of the top-left corner of the box, and the width and height of the box. Bounding boxes can help algorithms identify and locate specific objects within an image or video frame, as frequently showcased in autonomous driving tasks. The application suggests bounding boxes on images on some predefined labels and lets the user decide if he wants to apply them. Additionally the user is able to generate bounding boxes by clicking on the target object or drawing the bounding box manually.

Alongside the basic object detection segmentation backbone that is able to predict bounding boxes on known object categories Elbarbary (2021) amplifies his application using a One-shot object detection model. One-shot object detection is a computer vision technique that aims to detect and localize objects in an image with just a single training example per object category. It involves using a model that is pre-trained on a large dataset, and then fine-tuning it on a smaller dataset of one or a few examples per category. This is useful in situations where there is limited training data available, or where it is not feasible to collect large amounts of annotated data for every object category of interest.

By introducing this strategy the user is able to label previously unknown object instances using a single example. Obviously this can significantly reduce workload because it bypasses time consuming data collection as well as manual labeling effort.

During the design process Elbarbary (2021) directs his design decisions along Norman (2002). Norman's Design Principles are a set of guidelines for designing user-friendly products and interfaces. There are six principles, which are summarized in table 2.3.

Just like Elbarbary (2021) we will be mindful of these principles and adhere to them accordingly during the design of our application.

Although the system designed by Elbarbary (2021) reduces the need for manual labeling, we have identified some limitations that we aim to address in our application. Depending on hardware, retraining the model with the one-shot learner can still take up a few hours, depending on the amount of unknown object instances that need retraining. This can quite add up and make the application less effective. We aim to bypass these wait times by using a model with strong baselines and lots of generalization abilities described in chapter 2.1. Additionally, the user interaction serves as an input, which helps the model focus on the correct parts of the image even if the object class is previously unseen. According to Elbarbary (2021), the bounding box segmentation model encounters difficulties

Design Principle	Definition
Visibility	Users should be able to see the options available to them
Feedback	Users should receive feedback on their actions so they know what's happening
Constraints	Limiting the possible actions a user can take can help prevent errors
Mapping	The relationship between controls and their effects should be clear
Consistency	Similar actions should be achieved in similar ways across the system
Affordances	Objects should suggest their function and how they can be used

Table 2.1: Norman's Design Principles Norman (2002)

in distinguishing between objects with similar features, leading to potential confusion and incorrect suggestions for the user. Once again, we aim to address this problem by enabling the user to select the areas for segmentation and those to be ignored through their clicks.

2.4 Related Tools

The following section highlights some related tools as mentioned by Elbarbary (2021) and their applicability to image segmentation tasks. Some of them may be more specialized or easier to use for this task than others.

Prodigy: Prodigy is a data annotation tool that offers a range of annotation types, including image segmentation. It allows users to draw polygon or bounding box shapes around objects of interest and can be used for both binary and multi-class segmentation tasks.

Labelbox: Labelbox is a cloud-based platform for data annotation that supports image segmentation, among other annotation types. It offers a range of segmentation tools, including brush and polygon tools, and supports the creation of pixel-wise masks.

LabelImg: LabelImg is an open-source graphical image annotation tool that supports bounding box, polygon, and line annotation, but is primarily designed for object detection tasks. It can be used for image segmentation by drawing polygon shapes around regions of interest, but is not specifically optimized for this task.

CVAT: CVAT (Computer Vision Annotation Tool) is an open-source annotation tool that supports a wide range of annotation types, including image segmentation. It offers a range of segmentation tools, including polygon, brush, and magic wand tools, and supports the creation of pixel-wise masks.

Labelme: Labelme is an open-source graphical image annotation tool that supports polygon, point, and line annotation, but is primarily designed for object detection tasks. It can be used for image segmentation by drawing polygon shapes around regions of interest, but is not specifically optimized for this task.

VoTT (Visual Object Tagging Tool): VoTT is a Microsoft tool for annotating images and videos, primarily designed for object detection tasks. It offers a range of annotation types, including bounding boxes and polygons, and can be used for image segmentation by drawing polygon shapes around regions of interest.

makesense.ai: makesense.ai is a web-based annotation tool that supports a wide range of annotation types, including image segmentation. It offers a range of segmentation tools, including brush, polygon, and lasso tools, and supports the creation of pixel-wise masks.

Label Studio: Label Studio is an open-source platform for data labeling and annotation that supports a range of annotation types, including image segmentation. It offers a range of segmentation tools, including brush, polygon, and lasso tools, and supports the creation of pixel-wise masks.

Where almost all applications support image segmentation, for most of them the user has to manually mark the boundaries of an object or region in an image by creating a closed polygon shape around it. While this process can lead to the desired outcome, we argue, that our click based application is superior when it comes to efficiency and ease of use for non machine learning experts.

3. System Design

This chapter provides an overview of the system design for a web-based interactive click image segmentation application. It covers the tech stack employed, as well as the design considerations for both the frontend and backend components of the system.

3.1 Development Process

The development process of a web-based interactive click image segmentation application presented a range of challenges. One of the primary difficulties was navigating different codebases and understanding their nuances, which required a steep learning curve. Even though some modules were containerized in a docker environment, getting existing applications up and running was a long and tedious process due to issues with package versions and operating system compatibility. Building the application felt like assembling a puzzle from various pieces, with some elements not fitting seamlessly and requiring the creation or discovery of new components to ensure a cohesive whole. Despite these obstacles, we argue that we succeeded in creating an interactive and user-friendly image segmentation application.

3.2 Techstack/Architecture

The tech stack for application includes a standard client-server architecture (Figure 3.1). The frontend is built using Vue.js, a popular JavaScript framework, with Pinia as the state management library. State management is a technique for managing and synchronizing the application's state across components, making it easier to reason about and maintain.

The backend is written in Python and uses SQLite as the database to store metadata, such as image paths and masks. For the click-based image segmentation, a pretrained network was used. The controller class was modified, abstracted from the Tkinter demo interface, which presented a significant challenge, since it required understanding of the TKinter package and its python bindings as well as custom built neuronal network modules.

Proper version management is essential for the successful development and deployment of any software application. In this project, the use of correct version management was critical in ensuring the compatibility of the various dependencies used in the application. For Python, we utilized the mamba package manager. Mamba is a high-performance package manager for the Python programming language that aims to be a drop-in replacement for the popular Conda package manager. Like Conda, Mamba allows for the creation of isolated environments and simplifies the installation of complex libraries and packages. However, Mamba is built on top of the faster and more memory-efficient C++ library, whereas Conda is built in Python. This allows Mamba to be significantly faster than Conda when it comes to installing and updating packages. Similarly, for Node.js, the node version management (nvm) tool was used to manage the different versions of the runtime environment. By using these tools, we ensured that the correct versions of packages and libraries were used, avoiding issues such as conflicts between dependencies or compatibility issues with the operating system. Overall, proper version management played a crucial role in the development process, allowing for a smoother and more streamlined workflow.

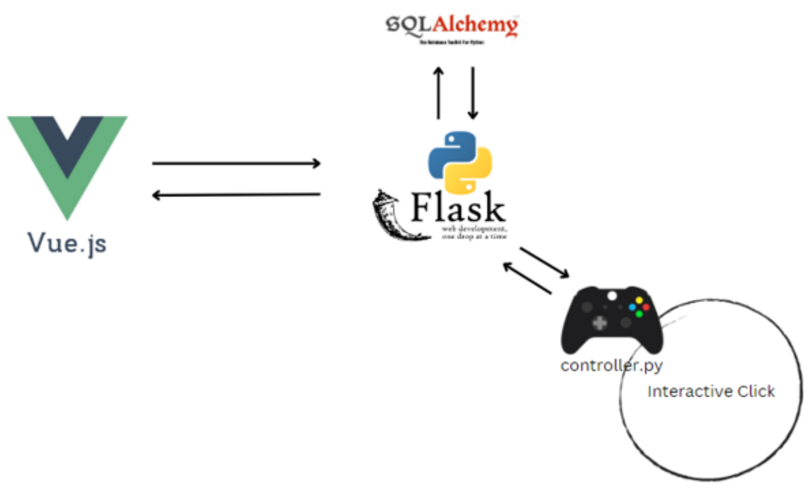


Figure 3.1: Techstack

3.3 Backend

The backend of the application was designed to handle various tasks, including processing user input, performing semantic image segmentation, and storing metadata in a database. The backend was built using Python and utilized the Flask web framework to provide a RESTful API for communication with the frontend, including routes for initializing models, uploading images, performing image segmentation, and retrieving segmentation results. These routes ensure that the frontend can easily communicate with the backend and perform the necessary tasks to provide a user-friendly image segmentation tool.

The Controller Class was adapted from a Tkinter demo interface to be used for the web application. The Controller Class had to be modified to work with the frontend, including receiving click coordinates from the user and manipulating the image based on the user's input. The frontend sends the click coordinates to the backend, which then processes the image and applies the click through the Controller Class. The backend ensures that the picture is correctly shaped and that the click is correctly applied. Metadata, including image paths and masks, get stored in a SQLite database. This allows for efficient retrieval and ensures that the application can quickly and accurately access the necessary data for image segmentation and classification.

To perform image classification, the application uses the timm framework and loads a pre-trained image classifier. The weights for the image classifier are downloaded on application start from a remote server, ensuring that the latest and most accurate weights are used in the classification process. The framework is then used to perform classification on each picture and suggests a minimum number of three labels.

3.4 Frontend

The frontend implementation of the web-based interactive click image segmentation application was designed to be as simple and user-friendly as possible, while still providing the necessary functionality for image segmentation. The implementation utilizes existing infrastructure and state management concepts that have been implemented by Elbarbary (2021), along with the Bootstrap Vue design framework to ensure visual consistency.

To keep the implementation modular and easy to manage, the frontend was designed to only include a single main page after uploading zipped images. This approach simplifies the user interface and allows for easy navigation and interaction.

To further simplify the user experience, only simple interaction possibilities are provided, with a focus on affordance. This approach ensures that users can easily understand the available actions and perform image segmentation tasks efficiently.

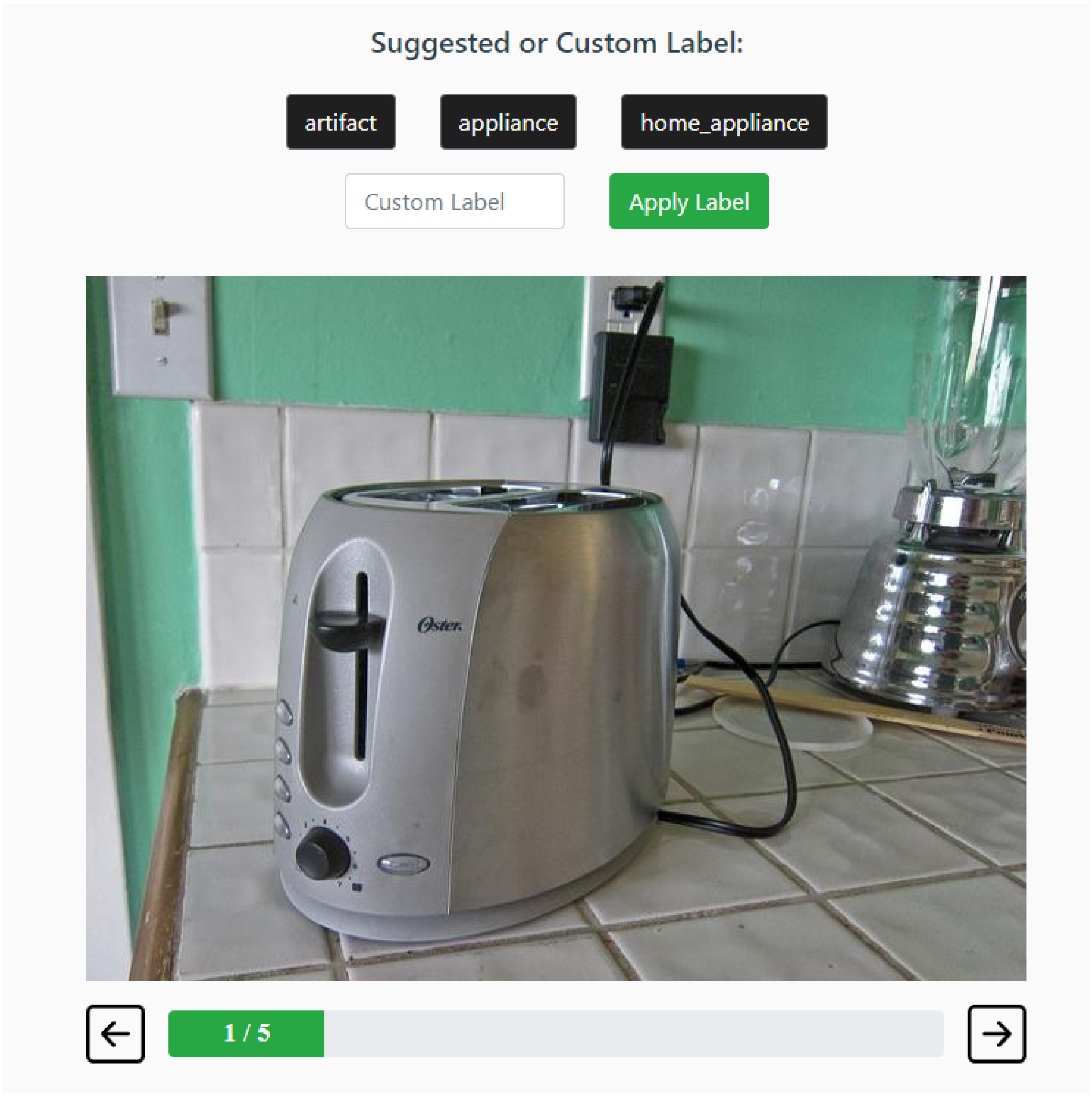


Figure 3.2: Main View

The central view in Figure 3.2 displays the uploaded image on the image canvas, which serves as the core of our application. Users can segment the image by clicking on it, with left-click for positive clicks and ALT + left-click for negative clicks. Multiple suggested labels are provided at the top of the page, and users can also add custom labels if necessary.

The progress bar at the bottom allows users to easily switch between uploaded images.

Once a user places a click on the image, the frontend displays the predicted mask along with the corresponding image, as shown in Figure 3.3. The predicted masks are represented in red and blue colors, with blue indicating areas where the model has less confidence. These areas are typically located at the border of the object instance, where confidence scores are lower. In contrast to the TKinter demo application, we have included additional information in our application as it can be useful for users to make more precise segmentation decisions. These areas are often overlooked and therefore providing this information can improve the accuracy of the segmentation. To provide users with instant feedback during the calculation, which can take up to three seconds depending on the hardware, we have implemented a placeholder click that is instantly placed on the image.

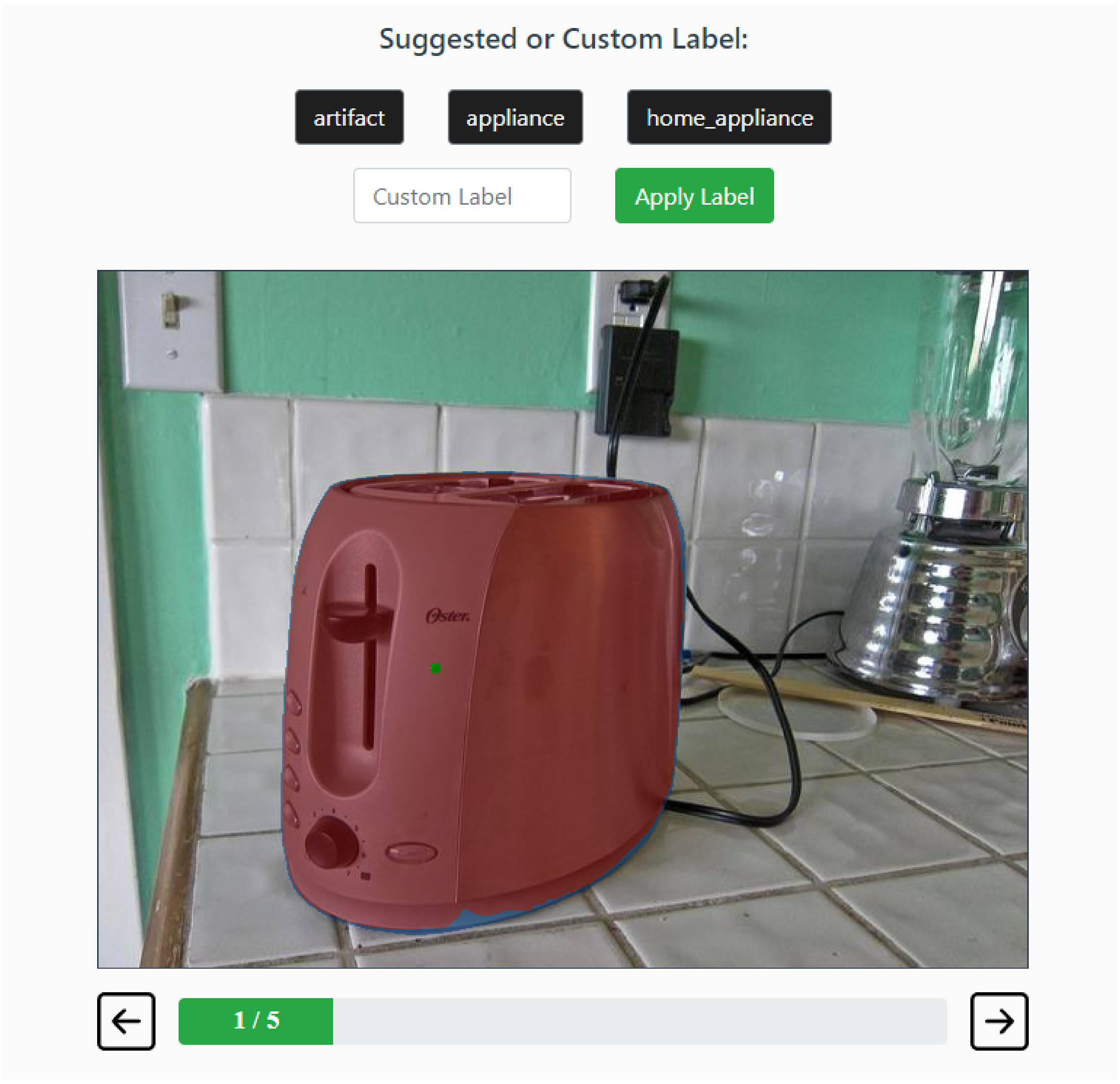


Figure 3.3: Main View with Positive Click

We have included several quality-of-life features on the right-hand side of the page (Figure 3.4), partly adapted from the TKinter demo app. By clicking on “Save Mask”, the user can save the current mask and label for the image. “Undo Click” removes the last click and its corresponding mask from the image, while “Reset Clicks” removes all clicks and returns the image to its original state. By clicking on “Export Data”, all saved masks and images are immediately zipped and downloaded. Users can adjust the click radius by using the slider located at the bottom of the tools.

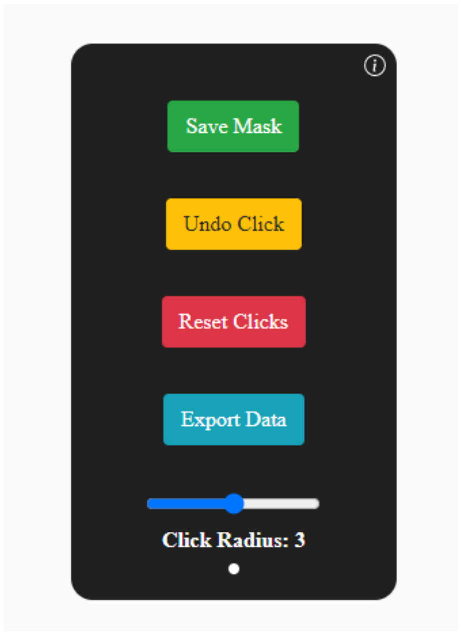
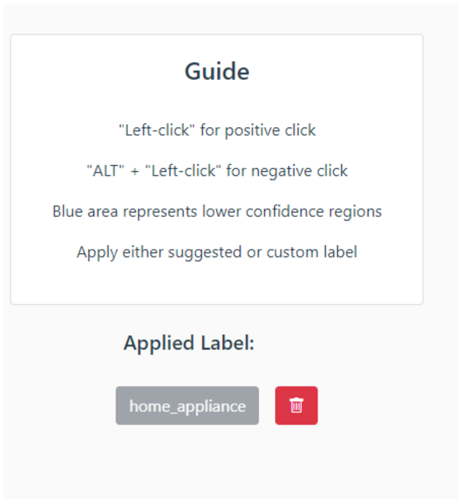


Figure 3.4: Tools

Finally, on the left-hand side (Figure 3.5), users can view the applied label and potentially remove it if necessary. Additionally, they can access helpful information about the application and how it works.



Guide

"Left-click" for positive click

"ALT" + "Left-click" for negative click

Blue area represents lower confidence regions

Apply either suggested or custom label

Applied Label:

home_appliance

Figure 3.5: Guide and Applied Labels

4. Evaluation

The following chapter will present a concise and preliminary usability assessment study, aimed at obtaining a general idea of the effectiveness of our developed application.

The interviews were performed using the Think-Aloud User Interview format. In a Think-Aloud User Interview, the participant is asked to perform a specific task or set of tasks using our developed prototype while continuously thinking aloud and sharing their thought process. The participants are encouraged to verbalize any confusion, questions, or feedback they have throughout the process. This provides valuable insights into the user's experience with the application and can reveal usability issues or areas for improvement. The session were recorded and used to identify pain points in the application.

To accommodate the scope of this thesis, we have chosen three participants, referred to as Interviewee 1, Interviewee 2, and Interviewee 3, who possess varying levels of background knowledge in machine learning. Interviewee 1 has no prior knowledge, Interviewee 2 has some knowledge, and Interviewee 3 has extensive knowledge in this field. The task was to segment 5 images each using the Tkinter demo application and the developed web application. All images were randomly taken from the COCO Dataset. Prior to the interviews, a pre-interview session was conducted where all participants were provided with basic instructions on the functionality of both applications, without disclosing their core features. Moreover, they were presented with their tasks and informed that the objective is to segment all images with maximum accuracy, while minimizing the number of clicks required.

All three participants successfully completed the tasks for both applications, but the demo application took up about 2/3 of the interview time due to its lack of a convenient way to load multiple images. Even segmenting just five images was time-consuming. As anticipated, interviewees 1 and 2 were both overwhelmed by the vast array of customization options available in the demo app, while only interviewee 3 attempted to use them effectively. Although the customization options were somewhat helpful, after experimenting with them for a while, it became clear that they were not necessary to complete the tasks. Although we have been able to identify some pain points, in general the web application was perceived as more convenient, easier to use and more visually appealing.

Interviewee	ML Knowledge	Average NoC	Average IoU
Interviewee 1	no prior knowledge	3	94.5%
Interviewee 2	some knowledge	5.6	94.6%
Interviewee 3	extensive knowledge	2.2	95.8%

Table 4.1: Task Results Web Application

The results of the task for each participant are presented in Table 4.1. Despite using a small sample size, our application demonstrates promising results in terms of NoC and IoU scores. This reinforces our model selection and interface design decisions.

5. Discussion

The purpose of this chapter is to present a set of discussion points and draw attention to the limitations and areas for future development of the web application. Again, regarding the scope of this thesis, these findings can be partly subjective and do not qualify as broad general statements.

5.1 The First Click

According to Z. Lin et al. (2020), the initial click made by users in interactive segmentation tasks is of great importance. Our evaluation has shown signs that this first click can significantly decrease mental stress associated with labeling and enable the user to build trust in the application.

(After placing first click) *"Wow, I feel like this mask is already perfect."* (Interviewee 3)

Even though our model is not using the FCA-Net Model, the strong segmentation baseline model introduced by Sofiiuk et al. (2022) achieves great first-click segmentation accuracy. Moreover, the significance of utilizing prior masks for subsequent prediction steps is emphasized by both Sofiiuk et al. (2022) and Chen et al. (2022) as highlighted in their respective studies. By incorporating this approach with a robust initial click, our evaluation has shown, how users can effectively refine an existing mask without corrupting their prior clicks and thereby limiting the amount frustrating "do-overs".

5.2 Simplicity in Design

As extensively discussed in his master's thesis (Elbarbary, 2021), adhering to fundamental design principles can result in a more efficient workflow and encourage users to engage with the system with less effort. From the user's perspective, one could argue that the more complex and cluttered the interface is, the more difficult their task becomes. After being asked about the customization options in the Tkinter demo application. Interviewee 2 said the following.

"I don't know what these things mean, so I won't touch them." (Interviewee 2)

We tried to design our application mainly by taking into account the affordance design principle introduced by Norman (2002), making the application as simple as possible. Once again, we do not want to exclude specific user groups just because they are missing knowledge in a particular field. As a result, we noticed how the users experienced an increase in confidence when interacting with the system.

5.3 Limitations & Future Work

As already touched on in chapter 4 we have identified some pain points and limitations in our application.

Zoom

To start, we observed that two out of the three users attempted to zoom in on the images for a clearer view of the recommended mask and to make more precise clicks. While our application currently lacks zoom in capabilities, we could potentially incorporate such functionality through basic javascript scroll wheel bindings.

Colors

We've opted to use two colors to exhibit the predicted masks - light red for areas with high confidence and blue for those with lower confidence. Likewise, we've selected green to represent positive clicks and dark red for negative clicks. However, these colors may not be easily distinguishable on certain backgrounds and do not cater to users with color blindness or red-green weakness. To address this, we could provide users with a color palette to personalize these colors according to their preferences.

Components Framework

Despite our interviewees' comments about our application being more visually appealing than the TKinter demo application, the Vue Bootstrap design we've employed is not the most cutting-edge component framework available. To enhance the aesthetics further, we could consider refactoring our application using the Vuetify framework, which incorporates the widely recognized Material Google components.

Image Classification

To facilitate semantic image segmentation, we have included an image classification model. However, despite being trained on numerous object instances and classes, we have observed that the model encounters difficultiies when predicting certain instance labels accurately. To address this issue, we could potentially introduce fine-tuned classification models tailored to the domain of the images to be segmented.

6. Conclusion

The rise of Artificial Intelligence and Computer Vision has made efficient and accurate image labeling an increasingly important task in various fields such as medicine and transportation. Interactive image segmentation has emerged as a promising technique to overcome the tedious and costly process of manual labeling. In this thesis, we have managed to design and implement a click-based web application for interactive image segmentation.

The process involved overcoming a set of obstacles and understanding how to harmonize different code bases. However, through this process, we were able to answer the research question of how to design an interactive segmentation tool to increase labeling efficiency while demanding minimal mental workload.

We found that the core functionality of the application should be on point, which in our case, is a strong and working segmentation model. Moreover, the focus should be on simplicity, as affordance is key in making the application easy to use even for non-machine learning experts.

Finally, through this thesis, we have demonstrated that interactive image segmentation is an effective and efficient method for image labeling. With the knowledge and experience gained from building this application, we can continue to build and improve upon this system in the future.

7. Declaration

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 14. 03. 2023 [illegible]

Paul Theuer

References

Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., & Zhao, H. (2022). Focalclick: towards practical interactive image segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1300–1309).

Elbarbary, Y. (2021). *Designing interactive bounding box labelling systems for image segmentation* (Unpublished master's thesis). Karlsruhe Institute of Technology.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the ieee international conference on computer vision* (pp. 2980–2988).

Lin, Z., Zhang, Z., Chen, L.-Z., Cheng, M.-M., & Lu, S.-P. (2020). Interactive image segmentation with first click attention. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13339–13348).

Norman, D. (2002). Emotion & design: attractive things work better. *interactions*, *9*(4), 36–42.

Ridnik, T., Ben-Baruch, E., Noy, A., & Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*.

Sofiiuk, K., Petrov, I. A., & Konushin, A. (2022). Reviving iterative training with mask guidance for interactive segmentation. In *2022 ieee international conference on image processing (icip)* (pp. 3141–3145).