

# Modular Deep Active Learning Framework for Image Annotation: A Technical Report for the Ophthalmo-AI Project

Md Abdul Kadir<sup>a</sup>, Hasan Md Tusfiqur Alam<sup>a</sup>, Pascale Maul<sup>a</sup>, Hans-Jürgen Profitlich<sup>a</sup>, Moritz Wolf<sup>a</sup>, Daniel Sonntag<sup>a,b</sup>

<sup>a</sup>*German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany*  
<sup>b</sup>*Oldenburg University, Oldenburg, Germany*

---

## Abstract

Image annotation is one of the most essential tasks for guaranteeing proper treatment for patients and tracking progress over the course of therapy in the field of medical imaging and disease diagnosis. However, manually annotating a lot of 2D and 3D imaging data can be extremely tedious. Deep Learning (DL) based segmentation algorithms have completely transformed this process and made it possible to automate image segmentation. By accurately segmenting medical images, these algorithms can greatly minimize the time and effort necessary for manual annotation. Additionally, by incorporating Active Learning (AL) methods, these segmentation algorithms can perform far more effectively with a smaller amount of ground truth data. We introduce MedDeepCyleAL, an end-to-end framework implementing the complete AL cycle. It provides researchers with the flexibility to choose the type of deep learning model they wish to employ and includes an annotation tool that supports the classification and segmentation of medical images. The user-friendly interface allows for easy alteration of the AL and DL model settings through a configuration file, requiring no prior programming experience. While MedDeepCyleAL can be applied to any kind of image data, we have specifically applied it to ophthalmology data in this project.

---

## 1. Introduction

Optical Coherence Tomography (OCT) has been widely used in biomedical imaging. Ophthalmologists use the segmentation of ocular OCT images for the diagnosis and treatment of eye diseases such as age-related macular degeneration (AMD), diabetic retinopathy (DR), and diabetic macular edema (DME) [1]. In recent years, Deep Learning (DL) based methods have achieved considerable success in medical image segmentation [2]. However, their progress has often been constrained as they require large data sets. Labeling and segmenting medical image data is a time-consuming and expensive process as domain experts are required to annotate them. Active learning can be beneficial for medical image segmentation since it reduces the burden of extensive annotation effort [3].

Active Learning (AL) is a paradigm in supervised Machine Learning (ML) where the model interacts with a user to label new data points. It is often used in scenarios with a large pool of unlabeled data where the labeling process is expensive. By using the ML model to select which examples to learn from, the algorithm can learn a concept with fewer examples than traditional supervised learning [4]. This work focuses on pool-based AL, i.e. uses a fixed data pool to select samples for annotation. Active learning is an iterative process (Figure 1) that starts with a small, annotated data set to train the ML model. This model is then used to query the remaining data from the unlabeled pool. Querying refers to the process of assigning scores to individual data points, according to how informative they are. The most informative data points are then selected for the next annotation round and added to the training data set. The cycle is repeated until the model reaches a given target performance.

The objectives of this work are to create an end-to-end modular AL system for deep learning models, that supports the annotation process, data handling, and the AL iterations. It has been developed in the context of the **Ophthalmo-AI** project (BMBF, see <https://www.dfki.de/en/web/research/projects-and-publications/>)

project/ophthalmo-ai), which focuses on **Intelligent, Cooperative Medical Decision Support in Ophthalmology**<sup>1</sup>. To that end, the authors trained a model for the segmentation of retinal OCT images. We employed an uncertainty score for the querying, since the available unannotated data contains many healthy eyes and a large set of medical markers for the diseases. Previous projects in this area do not include an end-to-end solution, do not focus on deep learning, and have to be integrated programmatically. Moreover they focus on classification, while our framework offers support for segmentation.

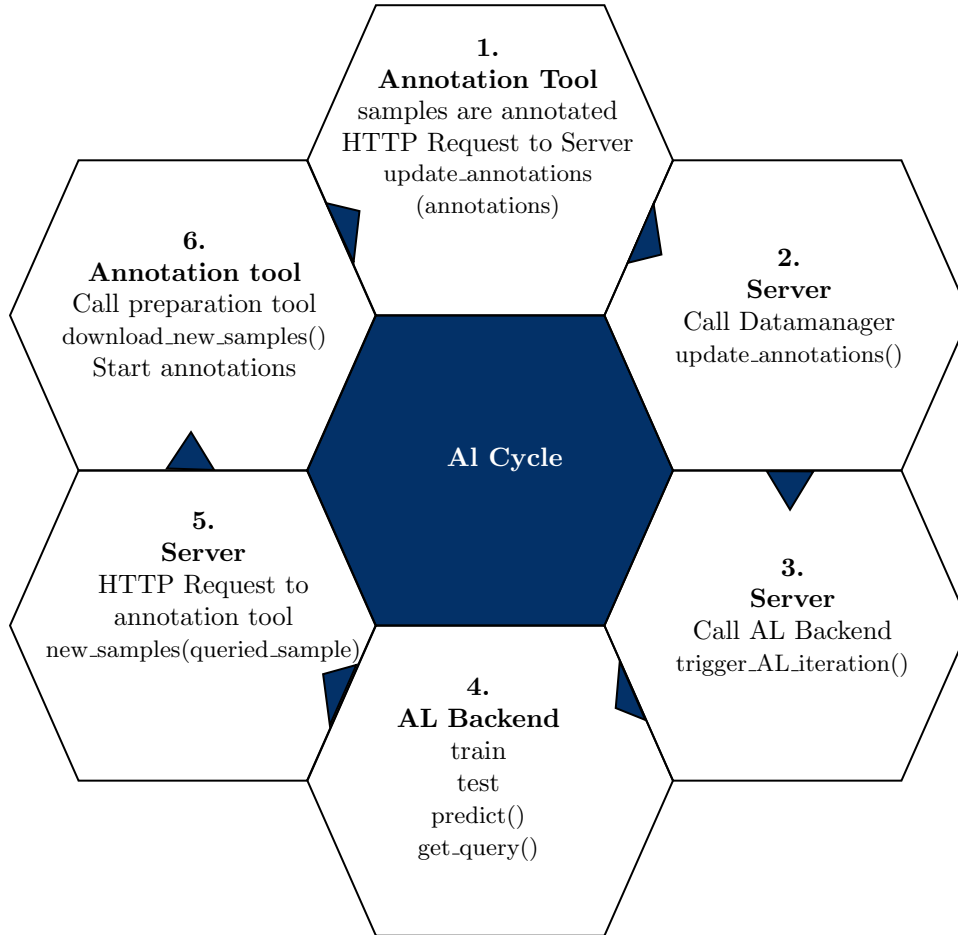


Figure 1: Design of the AL cycle and the functionality of the individual components: The first step consists of the annotation of an initial data set through the Annotation Tool (AT). The annotation information is then forwarded to a Controller, which initiates the update of the data set (2.) and triggers the next AL learning cycle (3.). The back-end then trains a deep learning model and performs the querying of the non-annotated data (4.). The top  $n$  samples resulting from the querying are then sent to the controller, which in turn sends the information to the AT (5). With the help of a data preparation tool, the AT then downloads new samples (6.) and generates preliminary annotations with a pre-trained model. Those preliminary annotations serve to facilitate the annotation process by proposing labels/segmentation, thus reducing annotation costs.

### 1.1. The Ophthalmo-AI Project

The developments in the Ophthalmo-AI project are intended to support ophthalmologists with an intelligent assistance system that can help them make the best possible therapy decision by making a correct diagnosis based on both image and clinical data. To generate comprehensible suggestions for medical staff, the AI system will first label biological structures and pathological features in the image data. Then special AI models derive diagnoses from the image findings and other information from the patient file, make therapy suggestions and predict the success of the therapy. Interactive machine-learning methods are used to integrate the doctors' knowledge of the case in question into the process. A large volume of treatment data collected and processed on the special data integration platform "XplOit" [5] is used for system development. The resulting augmented

<sup>1</sup><https://www.dfki.de/en/web/news/ohthalmo-ai-started>

intelligence system will be tested in clinical demonstrators for its practical suitability for macular degeneration and diabetic retinopathy.

The project focuses on the two most common causes of blindness in people over fifty: age-related macular degeneration and diabetic retinopathy. AMD is caused by aging processes in the central retinal cells (macula = point of sharpest vision). Everyone has a buildup of waste products in this area throughout life, and some have irregular blood vessels growing from the choroid under the retina. They leak fluid and blood, causing symptoms of "distortions" and progressive vision loss ("wet" form of "late" AMD). Diabetic retinopathy can occur as a result of diabetes. Here, too, blood components can escape into the retinal tissue. From this, pathological new blood vessels – so-called "proliferations" – can grow on the retina. Heavy bleeding into the eye can occur from these very vulnerable vessels, which can cause retinal detachment. Retinopathy often leads to DME, a swelling in the area of the macula, which can lead to visual disturbances. The layered representation of the retina, the "optical coherence tomography" (OCT), makes it possible to detect the finest changes and swellings, such as the presence of irregular blood vessels in the area of the macula. This is important for the diagnosis and therapy monitoring of both diseases. New drugs are now available that suppress the growth of abnormal blood vessels and reduce swelling in the macula. These drugs are injected directly into the vitreous cavity of the eye (intravitreal drug delivery, IVOM). The IVOM must be repeated at different intervals, usually monthly, until the vision has stabilized.

In the first project phase, both textual annotations in the patient files and annotations of the retinal structures (especially layers and drusen) are made using specially developed tools. To optimize the required working time, a tool already supports the doctors with segmentation models and special graphic algorithms. To further reduce the time-consuming annotation process, AL is used, which specifically selects those images that promise the greatest benefit in the continuous improvement of the models. In the second step, the resulting neural networks serve to support diagnoses and therapy suggestions and predict the success of the therapy. In order to enable the doctors to understand the system's decisions, in addition to the assessment of an existing disease, explanations of the results are also given in the form of interactive segmentation maps of the characteristics relevant to the diagnosis.

## 2. Related Work

Active learning is a strategy that focuses on the annotation of highly informative samples to improve model performance in a cost-effective manner. This selection is based on factors such as uncertainty [6], data distribution [7], expected model change [8], among others [9]. Uncertainty can be measured through probabilistic analysis of predictions by selecting instances with the lowest confidence [6, 10], or by calculating class entropy [11].

Applications of uncertainty-based approaches in deep neural networks have been explored in a few studies [12]. Notably, Gal et al. [13] proposed an uncertainty estimation method using dropout-based Monte Carlo (MC) sampling. This approach involves multiple forward passes with dropout throughout various layers, generating uncertainty during inference. Ensemble-based methods, which measure uncertainty based on the variance between predictions from numerous models, have also been utilized [14, 15, 16].

Active learning methods have been utilized for a range of segmentation tasks. Gorriz et al. [17] extended the Cost-Effective Active Learning (CEAL) algorithm [18] for Melanoma segmentation, wherein they chose both high and low-confidence samples for annotation. Mackowiak et al. [19] employed a region-based selection approach with MC dropout to estimate model uncertainty, thereby reducing annotation costs. Moreover, Nath et al. [14] proposed an ensemble-based approach, using multiple active learning frameworks that are jointly optimized. They adopted a query-by-committee strategy for sample selection. Siddiqui et al. [12] also suggested an active learning framework for multi-view dataset segmentation, quantifying model uncertainty based on the Kullback-Leibler divergence of posterior probability distributions.

Gaillochet et al. [20] proposed a representative learning-based active learning strategy for MRI prostate segmentation, where samples are selected for annotation at the batch level during training, rather than using

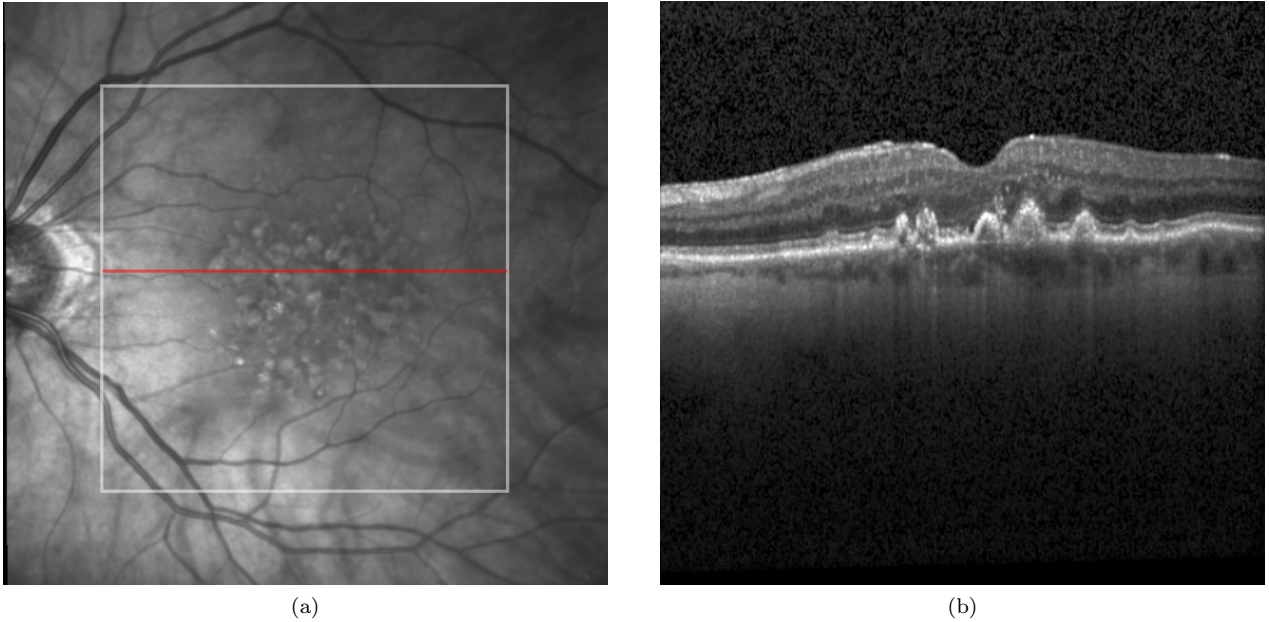


Figure 2: OCT: a) SLO with the positioning of the exposure on the retina (white square) and the position of a slice (red line), and b) corresponding slice

typical stochastic batches. This innovative approach contributes to the speed of training convergence and increases model robustness by concentrating on specific data points. Despite the progress in Active Learning for various applications, there remains a scarcity of comprehensive studies dedicated to Active Learning for OCT segmentation, with [21] being one of the few one can mention. This work by Li et al. [21] necessitates the use of foundation models, as per Khan et al. [22], requiring pre-training on large-scale datasets from similar domains. However, the practical application of this methodology could be challenged due to the difficulty in data collection, particularly considering the strict constraints tied to data privacy. Kadir et al. [23] propose EdgeAL technique, which uses edge information from OCT images as an a priori to improve the quality of queries in Active Learning for Deep Segmentation model training. The main concept is to take advantage of predictions' uncertainty as they cross semantic region borders in the input images.

The active learning framework proposed in this paper is designed to accommodate and integrate all previously mentioned active learning algorithms effectively. This encompassing approach facilitates seamless integration with a wide range of deep learning architectures and configurations, contributing to its versatility and broad applicability in the field.

### 3. OCT: Optical Coherence Tomography

OCT (optical coherence tomography) is a technique used to get a layered representation of the retina. This enables sectional images of the back of the eye to be taken in high resolution by using a laser light. The retina reflects this weak laser light differently; these reflections are measured and converted into detailed images by a computer system. These images provide a precise insight into the finest structures and changes in the retinal layers.

Besides a top-level view (called SLO) a number of slices (called b-scans) are recorded that allow a view under the surface of the retina. Each of these slices shows the individual layers of the retina, whose changes are essential for the assessment of the diseases under consideration and the choice of the right therapy (Figure 2). The implemented tool supports the annotation of up to 20 features within each slice, including the different layers, fluid collections and other bio-markers (for more details see section 6). The SLO image is saved in a file format called VOL together with the slices and metadata about the procedure, including a patient identifier, the number of slices, distance to the eye, software or hardware versions, etc.

#### 4. Project Specific Objectives

The objective of this project phase is to create an AL end-to-end system that can easily be configured and extended by AL users of all experience levels. To that end we build our architecture on top of frameworks that are used by expert machine learning practitioners (PyTorch), but can easily be configured by non-experts. The tool does not only handle training but is also connected to the data annotation tool [24] and handles the whole process from data set compilation with the help of AL, data annotation up to model training. The benefits for the users are: No manual integration of system components from very different backgrounds (e.g., data annotation vs. model training) is needed. Moreover, the user has an easily scalable system at hand that can be used for online and offline AL experiments. Since this framework was built as a decentralized system, individual components can be replaced if necessary.

#### 5. Architecture

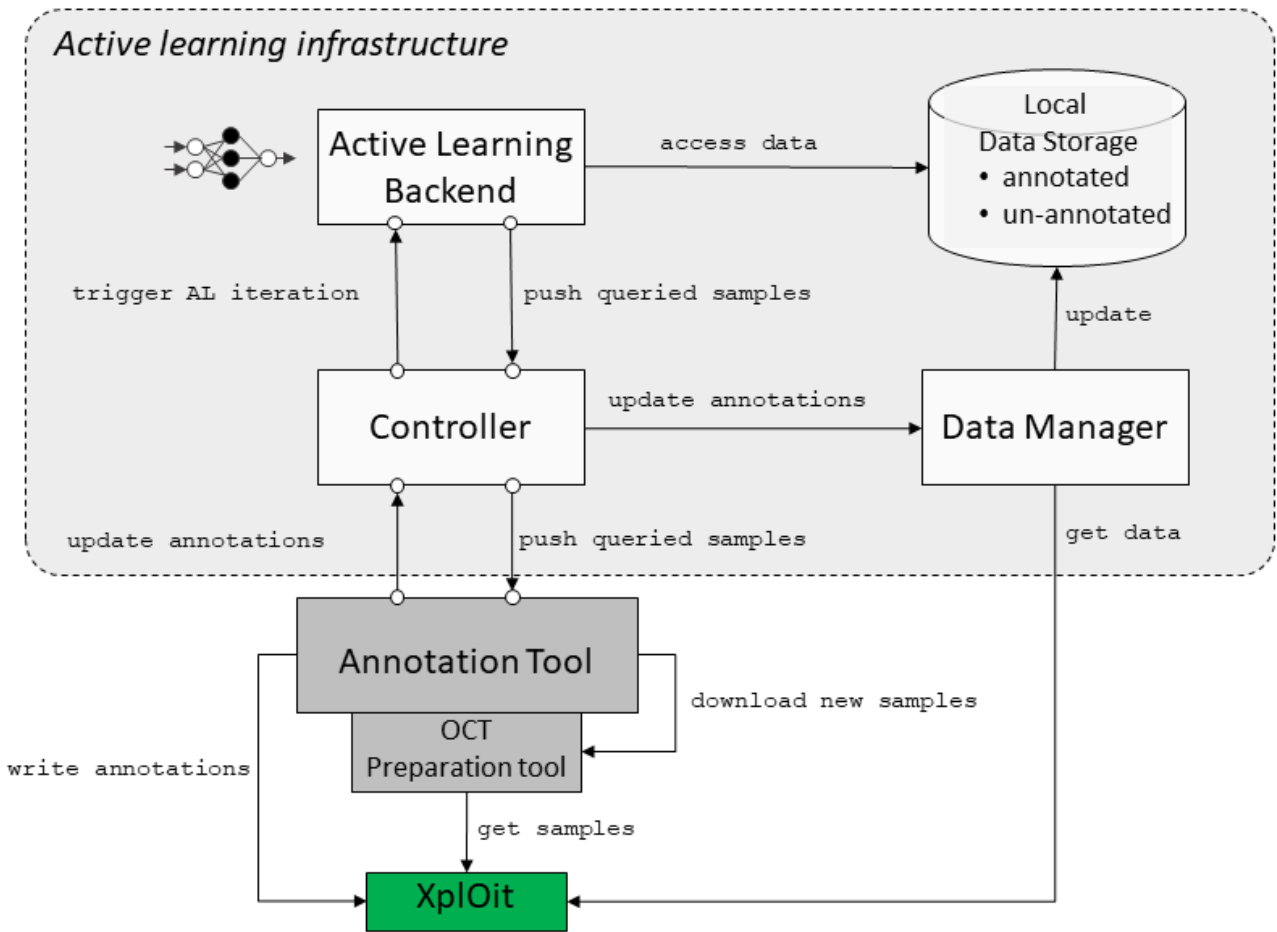


Figure 3: Architecture of the Active Learning Infrastructure

MedDeepCycleAL is designed to optimize the process of annotating and training deep learning models. The system’s four components (Figure 3), namely the Annotation Tool (AT), Controller, Data Manager, and Active Learning Backend, work together seamlessly to create a decentralized and powerful platform for enhancing the development of interactive deep learning tools.

The initial dataset is annotated with the help of an expert, without employing any AL strategy. This annotated dataset is then forwarded to the Controller. It plays a pivotal role in ensuring the proper execution of the AL cycle by coordinating and initiating the various modules within the MedDeepCycleAL system.

Upon receiving the annotated dataset, the Controller activates the Data Manager, which is responsible for downloading and managing the dataset used for model training, as well as maintaining the pool of unannotated

data. This process ensures that the training data stays up-to-date and relevant to the data available in the data warehouse, allowing for more accurate model development.

Once the dataset is prepared, the Active Learning Backend proceeds to train a deep learning model based on the model configuration. The trained model is then employed to rank the remaining unannotated data according to the selected AL query method. The Controller picks the most important sample for further annotation based on the ranking and presents it to the expert annotation tool interface for annotation. After the expert annotates the queried samples, the controller initiates the training cycle again. This strategic approach to data selection enables a more efficient use of resources and improved model performance with a limited amount of annotated data.

### 5.1. Controller

The controller is in charge of coordinating all other components of MedDeepCycleAL. The controller is a Flask server that, in line with the typical AL process, waits for the annotation tool to deliver some annotations in order to trigger the data manager to create the initial data set. With the initial data set, the controller then triggers the *active learning backend* to train a DL model on the initial data set, which is then used to rank the remaining data in the pool of unannotated data points. The top  $n$  results from that list are then sent to the annotation tool for the next AL iteration. With the resultant annotated data, the *controller* forwards to the next active learning cycle.

### 5.2. Data-Manager

The Data Manager is developed as an independent module to support the annotation tool and machine learning backend architecture. It provides APIs to access and process patient's treatment data from the `Xploit`<sup>2</sup> data server. The *controller* module accesses the data manager, and its functionalities can be divided into three main parts: 1) Securely downloading required patient treatment data from the server to the local development machine, 2) process the raw data, which are stored as a large Volume file, and 3) extract unannotated OCT scan slices and other metadata, including patients' health records (EHRs), diagnosis, and treatments. The Data Manager also contains functionalities to store and access the annotations provided by the *annotation tool* for OCT images. This module also provides APIs to access training/validation/test data sets, which are accessed by *active learning backend* via the controller module. The main APIs of the Data Manager class and their functionalities are briefly described below:

- `initialize(root_dir:Path, database:DB)`: Initializes the `DataManager` class using the tables in the database containing patient attributes. Required patient files are stored in Volume format (`.VOL`) are downloaded using the `Xploit` REST API.
- `initialize_with_files(file_path:Path) -> None`: The `DataManager` class can also be initialized using this function, if the required files are already downloaded from the server and stored in the local system.
- `get_unannotated_set() -> List[Dict]` : Returns a list of dictionary items containing the path of unannotated OCT images and associated meta data.
- `remove_from_unannotated_set(file_list:List[Dict]) -> None`: Removes an object from the unannotated set when a batch of OCT slices are annotated.
- `update_annotations(annotations:Dict[Dict]) -> None`: Given the annotation values from the Annotation Tool for an OCT image, this function produces the associated semantic masks and stores it in an image file.
- `get_annotated_set() -> List[Dict]`: Returns a list of dictionary items containing the path to annotated OCT images and their corresponding annotated masks.

---

<sup>2</sup><https://ophthalmo-ai.fraunhofer.de>

- `get_data_loader(DATA_ENUM:str) -> list[Dict]`: Returns the pre-processed data set for machine learning model training. Input argument is a literal of three choices `DATA_ENUM=['train', 'validation', 'test']`.

### 5.3. Active-Learning Backend

We utilized a versatile deep learning backend tool, developed on the PyTorch-based framework VISSL, that can be easily reconfigured and reused for training, testing, validation, and prediction. There are multiple benefits to using this framework, as it enables us to construct a deep learning model, establish hyperparameters, and train the model with a structured and adaptable YAML (Listings 1 and 2 are two example snippets taken from the YAML file) blueprint file. To enable active learning in our machine learning model, we have integrated the modAL framework, which is specifically designed for active learning. However, since our model also requires segmentation functionality, we have added an additional layer on top of the VISSL and modAL integration. This is necessary because VISSL was originally developed for unsupervised model training, and does not include functionalities for segmentation model training. Additionally, the algorithms built into modAL only support classification tasks, and not all active learning algorithms.

To enable segmentation functionality in our model, we have built additional features on top of the VISSL and modAL integration. This enables our model to perform active learning for segmentation tasks, which was not previously possible with the existing frameworks. By combining the strengths of modAL and VISSL with our new segmentation features, we are able to create a powerful and flexible machine learning model that can be trained using active learning techniques.

Listing 1: Snippet of the YAML configuration file for the model, loss function and optimizer initialization

```
MODEL:
  FEATURE_EVAL_SETTINGS:
    EVAL_MODE_ON: False
  TRUNK:
    NAME: unet
    UNET:
      n_channels: 1
      bilinear: True
  WEIGHTS_INIT:
    PARAMS_FILE: data/unet_best.pth

LOSS:
  name: dice_loss
  dice_loss:
    softmax: True
    ignore_index: -1

OPTIMIZER:
  name: sgd
  momentum: 0.9
```

Listing 2: Snippet of the YAML configuration file for data loading, transformation and augmentation

```
DATA:
  NUM_DATALOADER_WORKERS: 1
  TRAIN:
    DATA_SOURCES: [disk_filelist]
    LABEL_SOURCES: [disk_filelist]
    DATASET_NAMES: [seg_data]
    BATCHSIZE_PER_REPLICA: 3
    TRANSFORMS:
      - name: RandomResizedCrop
        size: 224
      - name: ToTensor
      - name: Normalize
        mean: [0.485]
        std: [0.229]
    MMAP_MODE: True
    COPY_TO_LOCAL_DISK: False
    COPY_DESTINATION_DIR: ""
    DATA_LIMIT: 3
    COLLATE_FUNCTION: msk_collator
```

Our YAML configuration file outlines six essential aspects of our active learning tool: ACTIVE LEARNING, MODEL, DATA, METERS, OPTIMIZER, DISTRIBUTED, and MACHINE.

1. ACTIVE LEARNING defines the AL algorithm and hyperparameters to be used for the tool.
2. MODEL specifies the deep learning model and its associated hyperparameters.
3. DATA outlines the data loading approach and the transformations to be considered during training and testing.
4. OPTIMIZER specifies the optimization algorithm and its additional hyperparameters.
5. METERS determines the performance metrics to be assessed during evaluation.

6. DISTRIBUTED and MACHINE define the hardware configuration for training and testing.

This modular configuration of the YAML file allows us to flexibly add new functionalities to our active learning tool, streamlining the process of integrating updates and improvements.

### 5.3.1. Segmentation network

Deep learning models requires a large set of annotated data for training. Instead of training a segmentation model from the scratch we have adopted transfer learning [25] strategy for our segmentation model. We trained segmentation model using a publicly available OCT dataset of similar task namely *Annotated Retinal OCT Images Database* (AROI) [26]. The dataset consist of 1136 annotated OCT scans of 24 patients suffering from age-related macular degeneration disease. Each scans has annotations for three retinal layers and three retinal fluid layers. We train different state-of-the art foundation models for segmentation using AROI dataset including U-Net [27], U-Net++ [28], Attention U-Net [29] and Y-Net [1]. Table 1 compares the segmentation results for different architectures on AROI dataset.

The segmentation network utilized in this tool is based on the U-Net (Figure 4) architecture, as described by Ronneberger et al. [27], Zhou et al. [28]. The network comes with pre-trained weights from Melinscak et al. [30]. In addition to U-net, we also conducted experiments using Y-net (YN)[31] and DeepLab-V3 (DP-V3) [12] with ResNet and MobileNet backbones [10]. Table 2 presents the performance of various active learning algorithms on 12% annotated data for different architectures.

Table 1: Class-wise Results for different Segmentation Models on AROI dataset [26]

Architecture	Above ILM	ILM-IPL/INL	IPL/INL-RPE	RPE-BM	Under BM	PED	SRF	IRF
U-Net	0.995	0.950	0.923	0.669	0.988	0.638	0.513	0.480
Attention U-Net	0.995	0.899	0.890	0.476	0.988	0.533	0.372	0.037
U-Net ++	0.992	0.944	0.924	0.641	0.986	0.622	0.487	0.419
Y-Net	1.000	0.970	0.950	0.640	0.990	0.630	0.520	0.980

### 5.3.2. Sampling

Annotating large amounts of segmentation data can be tedious. To reduce annotation effort, active learning is necessary. Several algorithms exist for active learning selection methods in semantic segmentation, including softmax margin (**MAR**) [10], softmax confidence (**CONF**) [18], softmax entropy (**ENT**) [11], MC dropout entropy (**MCDR**) [13], Core-set selection (**CORESET**) [16], (**CEAL**) [17], and regional MC dropout entropy (**RMCDR**) [19], maximum representations (**MAXRPR**) [15], and random selection (**Random**).

Table 2: The Table summarizes 5-fold cross-validation results (mean dice) for active learning methods and EdgeAL on the Duke dataset. EdgeAL outperforms other methods, achieving 99% performance with just 12% annotated data.

GT(%)	RMCDR	CEAL	CORESET	EdgeAL	MAR	MAXRPR
2%	<b>0.40 ±0.05</b>	<b>0.40 ±0.05</b>	0.38 ±0.04	<b>0.40 ±0.05</b>	<b>0.40 ±0.09</b>	<b>0.41 ±0.04</b>
12%	0.44 ±0.04	0.54 ±0.04	0.44 ±0.05	<b>0.82 ±0.03</b>	0.44 ±0.03	0.54 ±0.09
22%	0.63 ±0.05	0.54 ±0.04	0.62 ±0.04	<b>0.83 ±0.03</b>	0.58 ±0.04	0.67 ±0.07
33%	0.58 ±0.07	0.55 ±0.06	0.57 ±0.04	<b>0.81 ±0.04</b>	0.67 ±0.03	0.61 ±0.03
43%	0.70 ±0.03	0.79 ±0.03	0.69 ±0.03	<b>0.83 ±0.02</b>	0.70 ±0.04	0.80 ±0.04
100%	0.82 ±0.03	0.82 ±0.03	0.82 ±0.03	0.82 ±0.02	0.83 ±0.02	0.83 ±0.02

We conducted a five-fold cross-validation to evaluate the performance of each active learning algorithm. Our findings suggest that EdgeAL and MAXRPR are the most effective in selecting the best queries for training with a minimal amount of annotation. We presented the results in Table 2.

### 5.4. Annotation Tool

After the backend finishes training and querying, the controller sends the queried samples to the annotation tool. The annotation tool calls the OCT preparation tool which downloads OCTs from XplOit, adding additional random OCTs and loads data into annotation tool. 3 slices per OCT are chosen, the middle and two randomly on either half of the OCT. Doctors are then annotating these slices with the tool. Finished annotations are sent back to the controller. They are also sent to XplOit for storage and access for use by project partners.



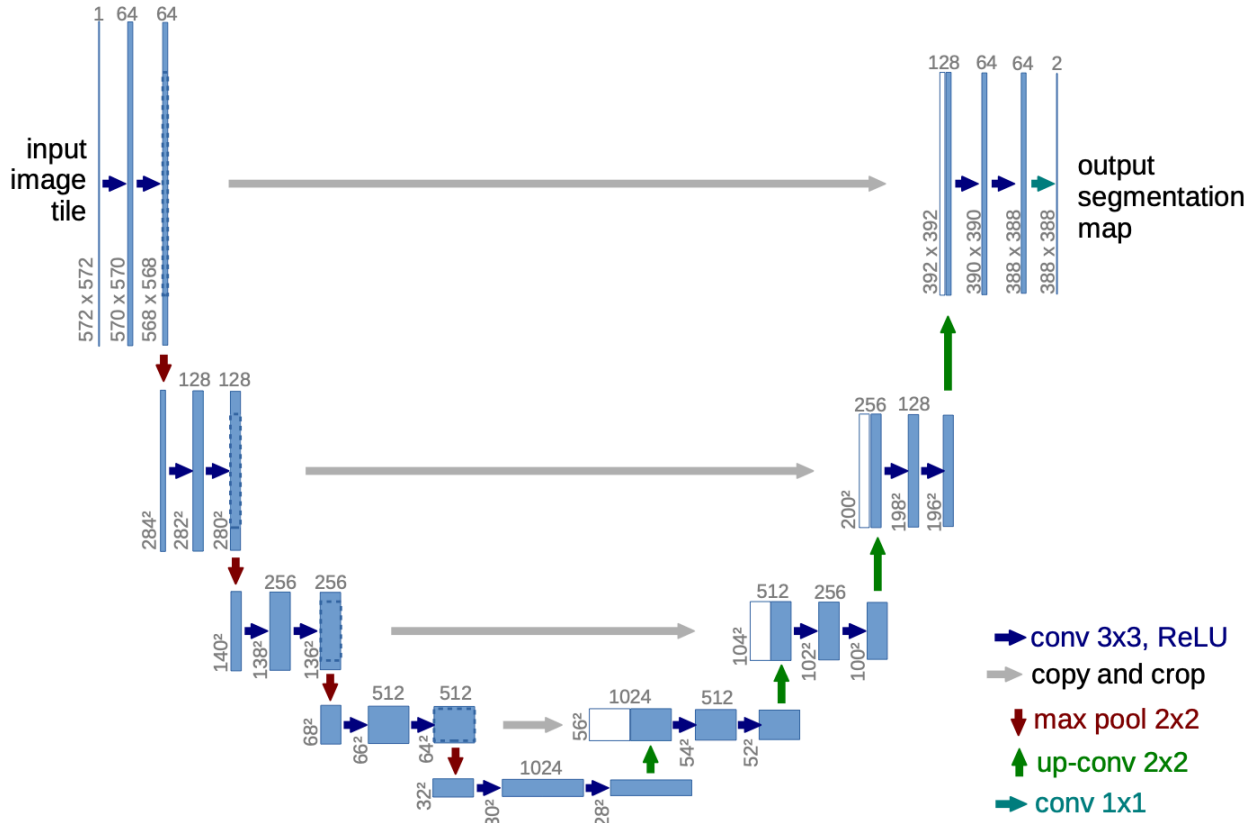


Figure 4: Visual representation of U-net architecture [27]

## 6. Intelligent User Interfaces

### 6.1. The Annotation Tool

The annotation tool that is used for image annotations is a fork of HUMAN [24]. Its main goal was to be as flexible and adaptable to a variety of annotation tasks as possible. The name **HUMAN** stands for **Hierarchical Universal Modular ANnotator**, reflecting the main features of this tool: It provides a variety of modular question types like multiple-choice, yes-no or setting bounding boxes that can be arranged via an internal state machine. These modules can be self contained or rely on each other, i.e. there can be a follow up task depending on a previous annotation. E.g. for semantic image labeling, one task could be drawing one or multiple bounding boxes, the next task could be labeling these with a predefined object name. The architecture (Figure 5) also allows the input for a task to come from another source, e.g. raw data or a classification model. For the example above a segmentation model could then provide the bounding box coordinates for the bounding box labeling task. The tool is implemented as a native JavaScript web application (Figure 6) with a Flask server backend. This has the advantage of better compatibility with various operating systems, parallel annotations and automated annotation distribution to different users. The procedures for setting up a data annotation protocol and annotating data instances are described in algorithm 1. The administrator must launch a server and create a state machine protocol before the algorithm may begin. In the Appendix A, a sample state machine protocol is given. All the annotated information is stored in an SQLite database. Appendix B provides an overview of the tool’s workflow along with some example annotations.

---

**Algorithm 1:** Protocol Setup and Data Annotation

---

**Result:** Annotated data instances

- 1 **Step 1:** Start the server;
- 2 **Step 2:** Administrator defines a state machine protocol;
- 3     *YAML file with task configuration and chaining;*
- 4 **Step 3:** Upload raw data into the database;
- 5     *Manually via Upload Console or automatically via OCT Preparation Tool;*
- 6 **Step 4:** Register users for data annotation;
- 7 **while** *Not all data instances are annotated* **do**
- 8     **Annotation Process;**
- 9         User logs in;
- 10        State machine assigns a data instance;
- 11        Perform annotations;
- 12        State machine transitions until end state;
- 13        Annotations are saved in the database;
- 14        Assign new data instances for annotation;
- 15 **end**

---

### *OCT Specific Changes*

For efficient annotations of OCT images, some changes were needed. For OCT annotations it is often necessary to jump back and forth between tasks. This made it necessary to abandon the hierarchical features of HUMAN in favor of the possibility to jump between states.

### *6.2. Diagnostic Decision Support Prototype*

A study by Brinkman et al [32] shows that there is a significant risk of related diseases being accidentally diagnosed as AMD. In order to provide diagnostic support for AMD to healthcare professionals, one of the project goals is to provide a diagnostic support system (DSS). The project's the clinical partners identified five diseases that are at risk of being diagnosed as AMD. To support prototype development for each disease 20 images have been provided by the partners in the data warehouse. Up to date a preliminary prototype has been developed. It evaluates medical markers that indicate an alternate diagnosis with the help of the segmentation of the retinal layers, provided by a DL model, and classical computer vision methods, that operate on top of the segmentation. F.ex. to compute the choroidal thickness, the segmentation of the choroidal layer at the center of the image can be used. In order to improve the prototype, it is desirable to incorporate more interactivity.

In [33] Bhattacharya et. al explore different approaches to a data-centric design in medical DSS. Here, the authors developed a dashboard for monitoring diabetes onset, aimed at healthcare professionals and patients. While the main focus of this work lies on exploring the impact of design choices onto the usefulness of the application, one notable feature are the what-if explorations. These allow the users to explore the effects of the input parameters (such as waist circumference, alcohol intake, etc.) onto the predicted diabetes onset.

Our prototype providing risk assessment for misdiagnosing AMD is aimed at healthcare professionals only. It is founded on marker-based reasoning, thus providing a reasoning-framework that healthcare professionals are accustomed to. For the new prototype version it is planned to incorporate a marker ranking, that can be re-ordered (Figure 7 visual component number 3). This allows the user to better understand the reasoning behind the risk prediction, which can be adjusted through drag-and-drop by the user, incorporating their feedback and promoting the interactive aspect of the tool.

## **7. Discussion**

Our AL framework MedDeepCyleAL provides a framework for AL experiments that supports segmentation and classification in medical images. It can easily be used with different deep learning architectures and self-supervised models can be incorporated. Different experiments are planned with regard to AL.

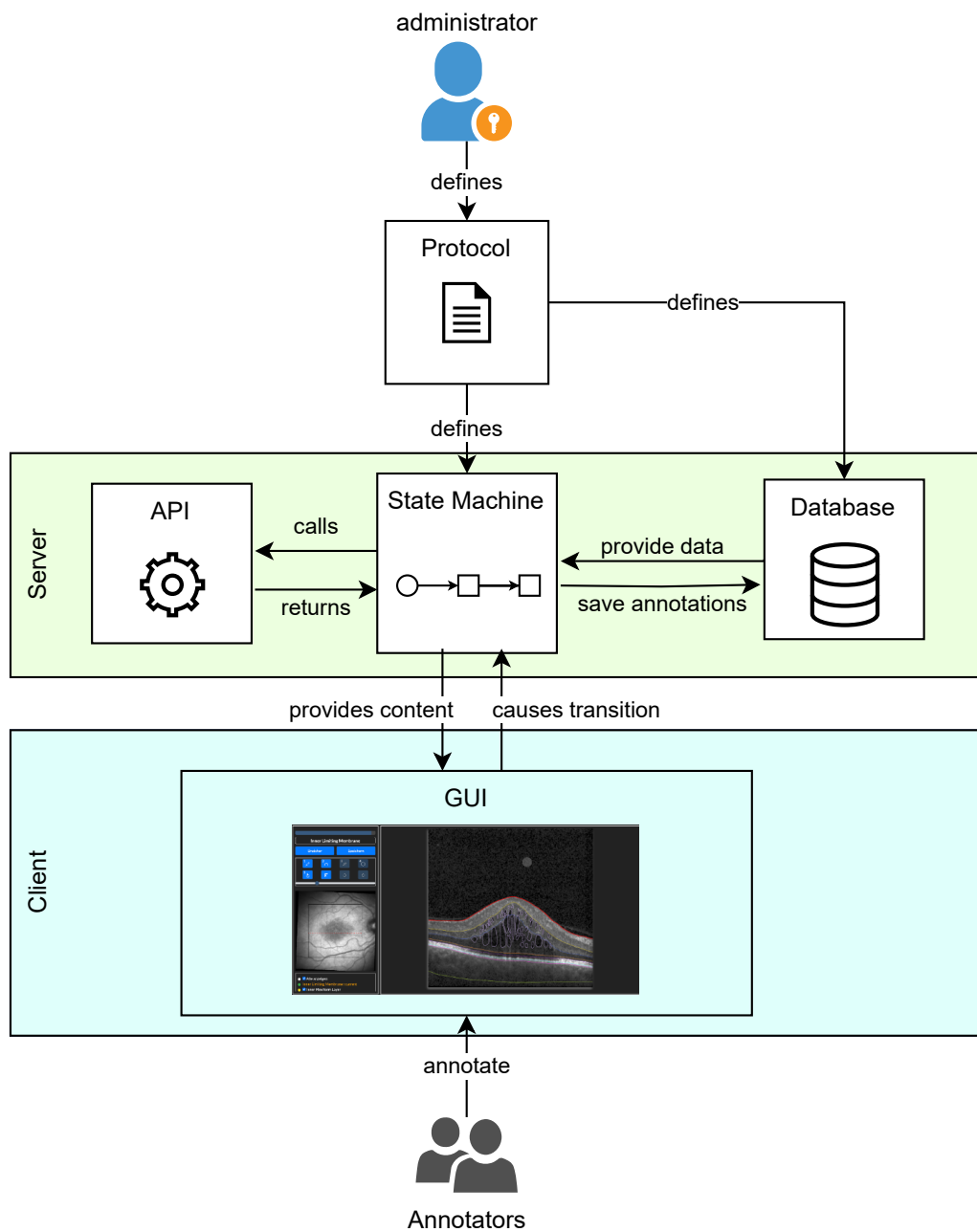


Figure 5: Basic architecture of HUMAN: Administrators define a protocol that is translated into a state machine. The State Machine interacts with API, Database and Client GUI. Annotators annotate using the GUI.

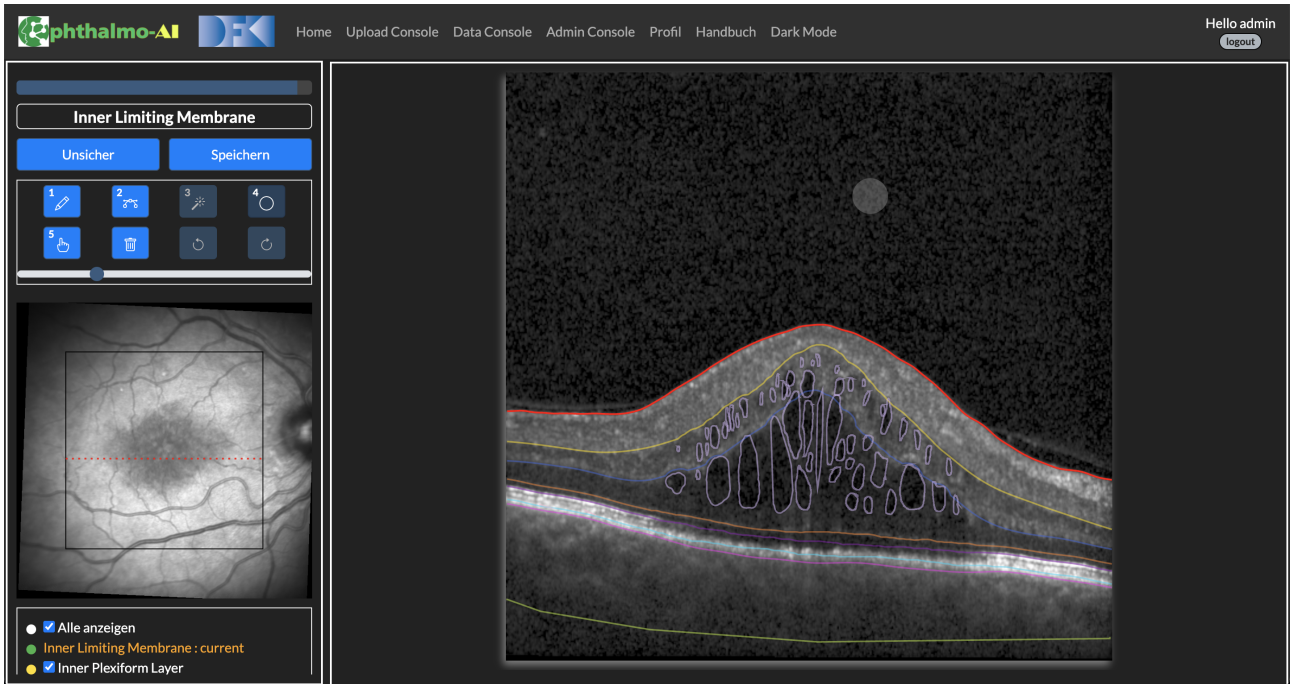


Figure 6: Example OCT annotation in HUMAN. Left side, from top to bottom: a progress bar, the current layer or fluid to be annotated, continue and uncertain buttons, drawing tools, the SLO and a list summarizing all annotations. Right side: The slice in which the annotation is to be done. The current annotation is marked red, the others are transparent and colored corresponding to the mark in the summary list.

### 7.1. Partial Labeling

Here, we hypothesize that some parts of the retinal images of AMD patients are easier to segment than others. While the upper-most layer is easily distinguished from the background/vitreous eye, distinguishing neighboring layers is more challenging. We are training a DL model on retinal OCT images, whose input is a 2D image of the retina and that creates a (semantic) segmentation of the input image. While the common approach involves training a model on a fully labeled image, our approach only requires labeling in specific areas. Which areas will be presented to the annotator for labeling is determined through AL. Note that in most AL approaches, the AL algorithm would be used to compile a training set for a deep learning model. In this experiment, we use a pre-defined, fixed set of images. With the help of AL, it is determined what layers/medical markers of an individual image will be annotated. Instead of training the model with a fully annotated ground truth, we use a weighted loss that effectively masks out non-annotated areas. Thus, we reduce annotation cost not by reducing the required data set size to attain a certain score but reduces the amount of annotations that need to be added to individual images.

### 7.2. Active Selection with Self-Supervised Learning

Along with active learning, another paradigm to reduce the annotation effort is self-supervised training that learns from a large pool of unlabeled data in an unsupervised way. Self-supervised learning (SSL) has gained attention as it can utilize large amounts of unlabeled data, which are often readily available. It leverages the inherent structure or information present in the input data itself to create supervised learning tasks. The model learns to predict certain aspects of the data, such as filling in missing parts, generating transformed versions, or clustering group of similar samples closer and diverse samples afar (contrastive learning) [34]. Recent studies suggest using self-supervised learning with active learning can be a powerful combination that leverages the benefits of both approaches [35]. Neural models for our task can be pre-trained using large pool of unlabeled OCT slices by applying self-supervised tasks in order to learn meaningful representations of the data. And by integrating SSL optimization in the sample selection of our active learning framework, we plan to evaluate their performance on medical imaging domain.

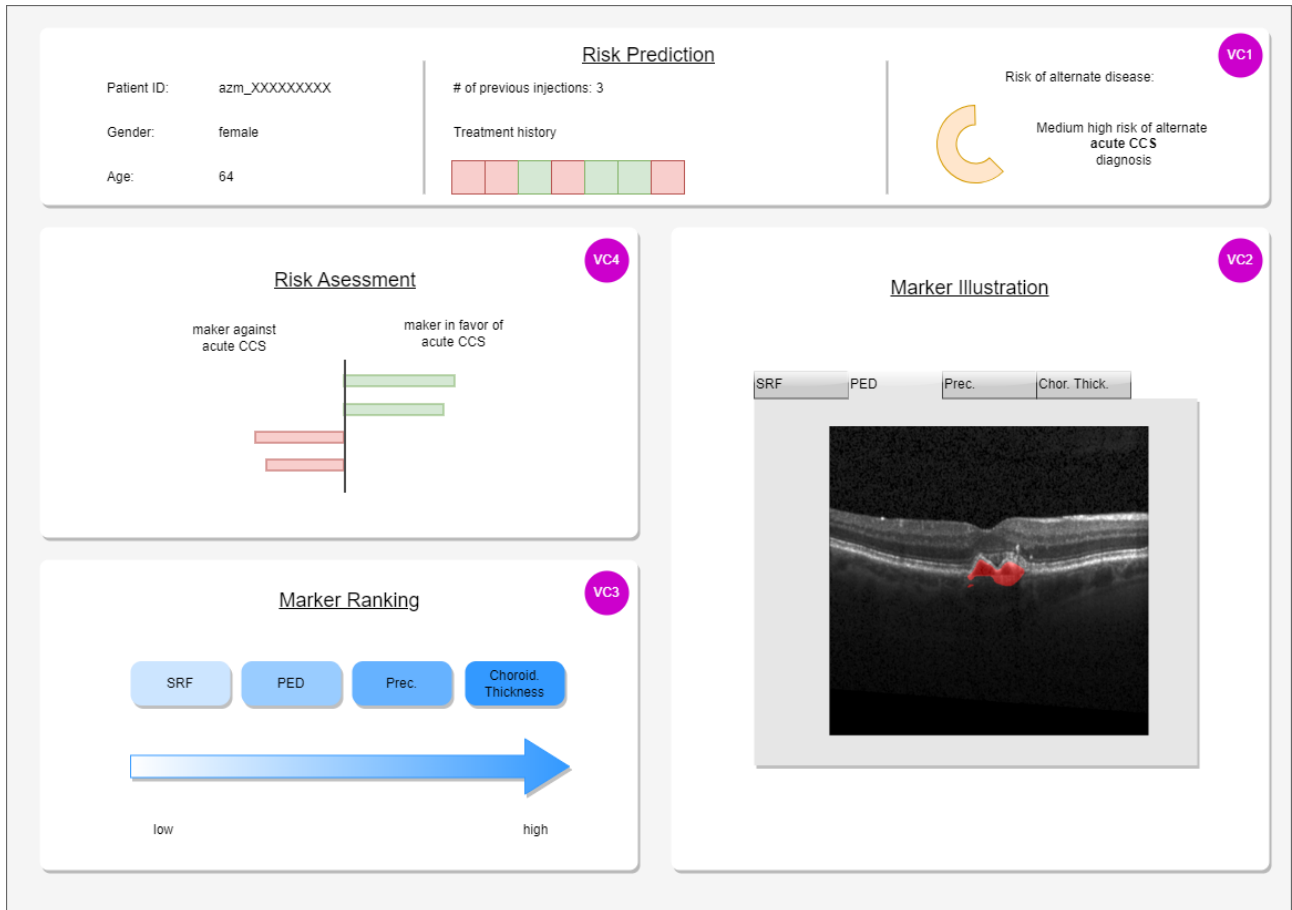


Figure 7: Proposed interactive DSS prototype to differentiate between AMD and other mis-diagnosed diseases, the pink circles indicate individual visual components (VC). VC1 provides an overall patient overview, including treatment history (which consists of injections vs no injections for AMD treatment) and an risk evaluation indicating how likely it is that the patient has been misdiagnosed. VC2 illustrates the individual markers with their evaluation through the system. This allows the user to investigate individual markers in-depth. VC4 provides a data-centric reasoning panel to illustrate the contribution of the individual markers to the risk prediction. VC3 is the interactive component of the demo. Here, the user may re-order the marker ranking which leads to a retraining of the risk assessment system in the back-end.

### *7.3. The Benefit of Active Learning for the Annotation Process*

In AL, the models that are gradually trained, are sometimes used to provide suggestions to the annotators in order to speed up the annotation process. In this experiment, we measure the time-wise benefit of such an approach for the segmentation of retinal layers. The underlying hypothesis is that, while the models trained on the partial data set may not be as accurate, suggestions generated with their help still provide a speed-up in the annotation process. Since the purpose of uncertainty sampling in AL is to provide challenging examples, this benefit has to be present for challenging data points.

We expect this effect to take place the moment the provided suggestions are exact enough, such that they can be corrected with an effort that falls below the creation of the same annotation from the very beginning, or that the suggestions are good enough for some annotation classes, such that the annotator may skip the particular example at all. A known challenge from previous studies is the distinction between secondary effects and the actual benefit of the described method. For one it is difficult to measure the inter-annotator difference and then it provided to be difficult to differentiate between the learning effect of performing the annotation process (f.ex. learning to use the tool) and benefit of the described method. We are countering the former by choosing an appropriate amount of annotators and the latter by giving the annotators some example annotations to get to know the tool.

### **Acknowledgement**

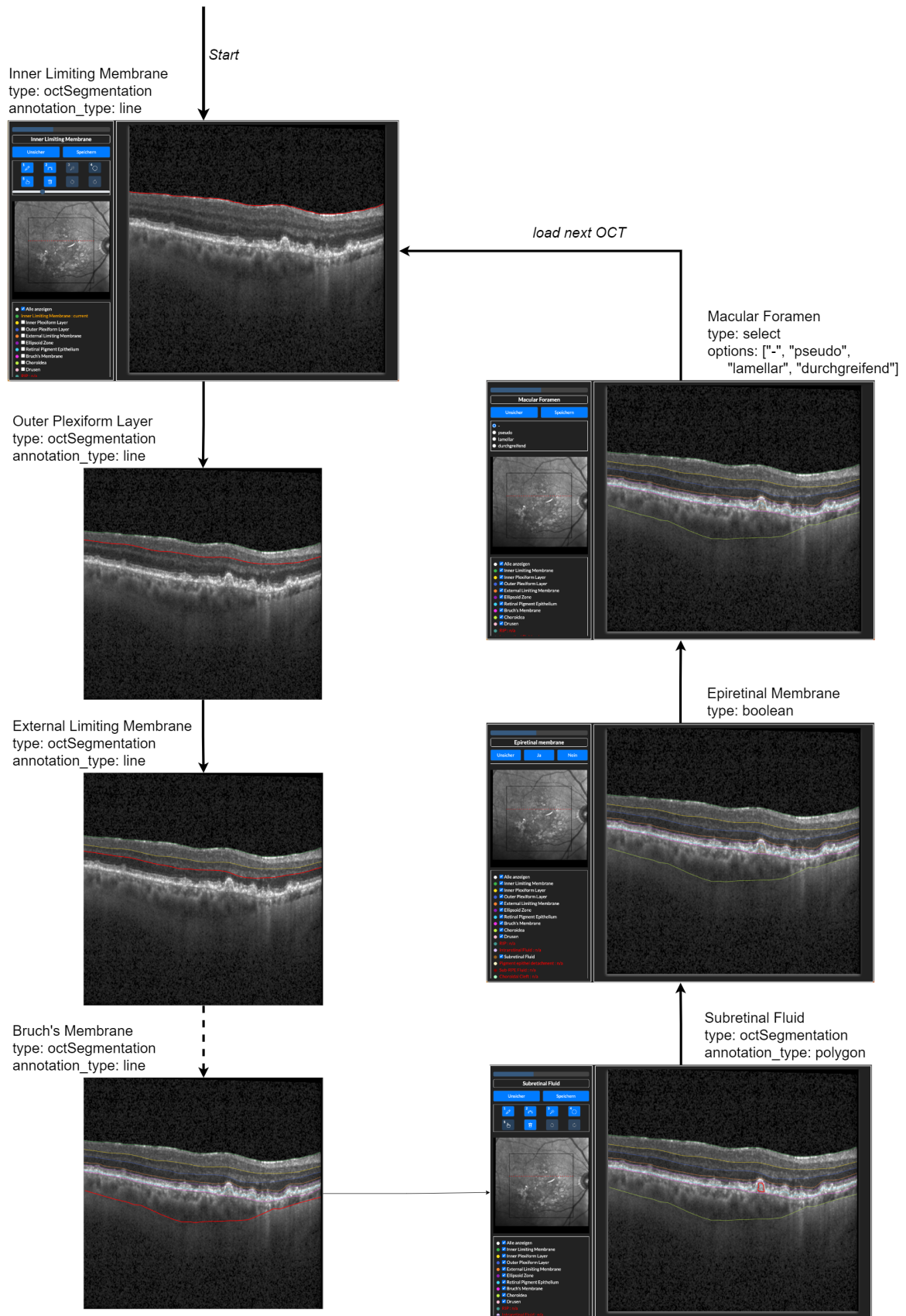
The funding for this project was provided in part by the German Federal Ministry of Education and Research (BMBF) under grant number 16SV8639 (Ophthalmology-AI). Additional support was received from the Lower Saxony Ministry of Science and Culture and the Endowed Chair of Applied Artificial Intelligence (AAI) at the University of Oldenburg.

## Appendix A. Annotation Tool Protocol for Annotating the Inner Limiting Membrane Layer

Listing 3: Example protocol consisting of a load state loading an OCT an image segmentation task for the Inner Limiting Membrane, a categorial question for the type of Macular Foramen (none, pseudo lammelar or full-thickness), and a summary state providing a summary of all annotations at the end.

```
start:
  type: load
  dataloader: OCTLoader
  transitions:
    - next:
      target: seg_ilm
seg_ilm:
  type: octSegmentation
  question: Inner Limiting Membrane
  annotation_type: line
  transitions:
    - "*":
      target: macular_foramen
macular_foramen:
  question: Macular Foramen
  type: select
  options:
    - ' - '
    - pseudo
    - lamellar
    - full-thickness
  transitions:
    - '*':
      target: summary
summary:
  type: summary_oct
  question: Summary
  transitions:
    - '*':
      target: end
```

## Appendix B. Example Annotation Workflow





## References

- [1] A. Farshad, Y. Yeganeh, P. Gehlbach, and N. Navab, “Y-Net: A Spatiospectral Dual-Encoder Network for Medical Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 582–592.
- [2] W. Yuan, D. Lu, D. Wei, M. Ning, and Y. Zheng, “Multiscale Unsupervised Retinal Edema Area Segmentation in OCT Images,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 667–676.
- [3] V. Nath, D. Yang, H. R. Roth, and D. Xu, “Warm Start Active Learning with Proxy Labels and Selection via Semi-supervised Fine-Tuning,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 297–308.
- [4] B. Settles, “Active Learning Literature Survey,” *University of Wisconsin, Madison*, vol. 52, 07 2010.
- [5] G. Weiler, U. Schwarz, J. Rauch, K. Rohm, T. Lehr, S. Theobald, S. Kiefer, K. Götz, K. Och, N. Pfeifer, L. Handl, S. Smola, M. Ihle, A. T. Turki, D. W. Beelen, J. Rissland, J. Bittenbring, and N. M. Graf, “XplOit: An Ontology-Based Data Integration Platform Supporting the Development of Predictive Models for Personalized Medicine,” in *Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth - Proceedings of MIE 2018, Medical Informatics Europe, Gothenburg, Sweden, April 24-26, 2018*, ser. Studies in Health Technology and Informatics, A. Ugon, D. Karlsson, G. O. Klein, and A. Moen, Eds., vol. 247. IOS Press, 2018, pp. 21–25. [Online]. Available: <https://doi.org/10.3233/978-1-61499-852-5-21>
- [6] B. Lee and K. Paeng, “A Robust and Effective Approach Towards Accurate Metastasis Detection and pN-stage Classification in Breast Cancer,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*. Springer, 2018, pp. 841–850.
- [7] S. Samrath, E. Sayna, and D. Trevor, “Variational Adversarial Active Learning,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.
- [8] C. Dai, S. Wang, Y. Mo, K. Zhou *et al.*, “Suggestive Annotation of Brain Tumour Images with Gradient-guided Sampling,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*. Springer, 2020, pp. 156–165.
- [9] F. Bai, X. Xing, Y. Shen, H. Ma *et al.*, “Discrepancy-Based Active Learning for Weakly Supervised Bleeding Segmentation in Wireless Capsule Endoscopy Images,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. Springer, 2022, pp. 24–34.
- [10] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-class Active Learning for Image Classification,” in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 2372–2379.
- [11] W. Luo, A. Schwing, and R. Urtasun, “Latent Structured Active Learning,” *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [12] Y. Siddiqui, J. Valentin, and M. Nießner, “Viewal: Active learning with viewpoint entropy for semantic segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9433–9443.

- [13] Y. Gal, R. Islam, and Z. Ghahramani, “Deep Bayesian Active Learning with Image Data,” in *International conference on machine learning*. PMLR, 2017, pp. 1183–1192.
- [14] V. Nath, D. Yang, B. A. Landman, D. Xu, and H. R. Roth, “Diminishing Uncertainty Within the Training Pool: Active Learning for Medical image Segmentation,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 10, pp. 2534–2547, 2020.
- [15] L. Yang, Y. Zhang, J. Chen, S. Zhang *et al.*, “Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11–13, 2017, Proceedings, Part III 20*. Springer, 2017, pp. 399–407.
- [16] O. Sener and S. Savarese, “Active Learning for Convolutional Neural Networks: A Core-set Approach,” *arXiv preprint arXiv:1708.00489*, 2017.
- [17] M. Gorriz, A. Carlier, E. Faure, and X. Giro-i Nieto, “Cost-effective active learning for melanoma segmentation,” *arXiv preprint arXiv:1711.09168*, 2017.
- [18] K. Wang, D. Zhang, Y. Li *et al.*, “Cost-effective Active Learning for Deep Image Classification,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.
- [19] R. Mackowiak, P. Lenz, O. Ghori, F. Diego *et al.*, “Cereals-Cost-effective Region-based Active Learning for Semantic Segmentation,” *arXiv preprint arXiv:1810.09726*, 2018.
- [20] M. Gaillochet, C. Desrosiers, and H. Lombaert, “Active Learning for Medical Image Segmentation with Stochastic Batches,” *arXiv preprint arXiv:2301.07670*, 2023.
- [21] X. Li, S. Niu, X. Gao, T. Liu *et al.*, “Unsupervised Domain Adaptation with Self-selected Active Learning for Cross-domain OCT Image Segmentation,” in *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II 28*. Springer, 2021, pp. 585–596.
- [22] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks,” *Artificial intelligence review*, vol. 53, pp. 5455–5516, 2020.
- [23] M. A. Kadir, H. M. T. Alam, and D. Sonntag, “EdgeAL: An Edge Estimation Based Active Learning Approach for OCT Segmentation,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds. Cham: Springer Nature Switzerland, 2023, pp. 79–89.
- [24] M. Wolf, D. Ruiter, A. G. D’Sa, L. Reiners, J. Alexandersson, and D. Klakow, “HUMAN: Hierarchical Universal Modular ANnotator,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 55–61. [Online]. Available: <https://aclanthology.org/2020.emnlp-demos.8>
- [25] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A Survey of Transfer Learning,” *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [26] M. Melinščak, M. Radmilović, Z. Vatauvuk, and S. Lončarić, “Annotated Retinal Optical Coherence Tomography Images (AROI) Database for Joint Retinal Layer and Fluid Segmentation,” *Automatika*, vol. 62, no. 3-4, pp. 375–385, 2021.
- [27] 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*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015.

- [28] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds. Cham: Springer International Publishing, 2018, pp. 3–11.
- [29] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, “Attention U-net: Learning where to Look for the Pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.
- [30] M. Melinscak, M. Radmilović, S. Loncaric, and Z. Vatauvuk, “Annotated Retinal Optical Coherence Tomography Images (AROI) Database for Joint Retinal Layer and Fluid Segmentation,” *Automatika*, vol. 62, pp. 375–385, 08 2021.
- [31] A. Farshad, Y. Yeganeh, P. Gehlbach, and N. Navab, “Y-net: A spatio-spectral dual-encoder network for medical image segmentation,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*. Springer, 2022, pp. 582–592.
- [32] C. K. Brinkmann, P. Chang, T. Schick, B. Heimes, J. Vögeler, B. Haegele, B. Kirchhof, F. G. Holz, D. Pauleikhoff, F. Ziemssen, S. Liakopoulos, G. Spital, and S. Schmitz-Valckenberg, “Initiale Diagnostik und Indikationsstellung zur Anti-Vascular-Endothelial Growth-Factor-Therapie bei Netzhauterkrankungen: Vergleich der Befundung durch Studienarzt versus Reading Center (ORCA/OCEAN-Studie),” *Der Ophthalmologe*, vol. 116, no. 8, pp. 753–765, Aug. 2019. [Online]. Available: <http://link.springer.com/10.1007/s00347-018-0805-y>
- [33] A. Bhattacharya, J. Ooge, G. Stiglic, and K. Verbert, “Directive Explanations for Monitoring the Risk of Diabetes Onset: Introducing Directive Data-Centric Explanations and Combinations to Support What-If Explorations,” in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, Mar. 2023, pp. 204–219, arXiv:2302.10671 [cs]. [Online]. Available: <http://arxiv.org/abs/2302.10671>
- [34] R. Balestrieri, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian *et al.*, “A Cookbook of Self-supervised Learning,” *arXiv preprint arXiv:2304.12210*, 2023.
- [35] J. Z. Bengar, J. van de Weijer, B. Twardowski, and B. Raducanu, “Reducing Label Effort: Self-supervised Meets Active Learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1631–1639.