Iris Fast Labeling System Based on OCT Image

11911737 林小璐 11910311 周一凡 11911425 李怀武

1 Abstract

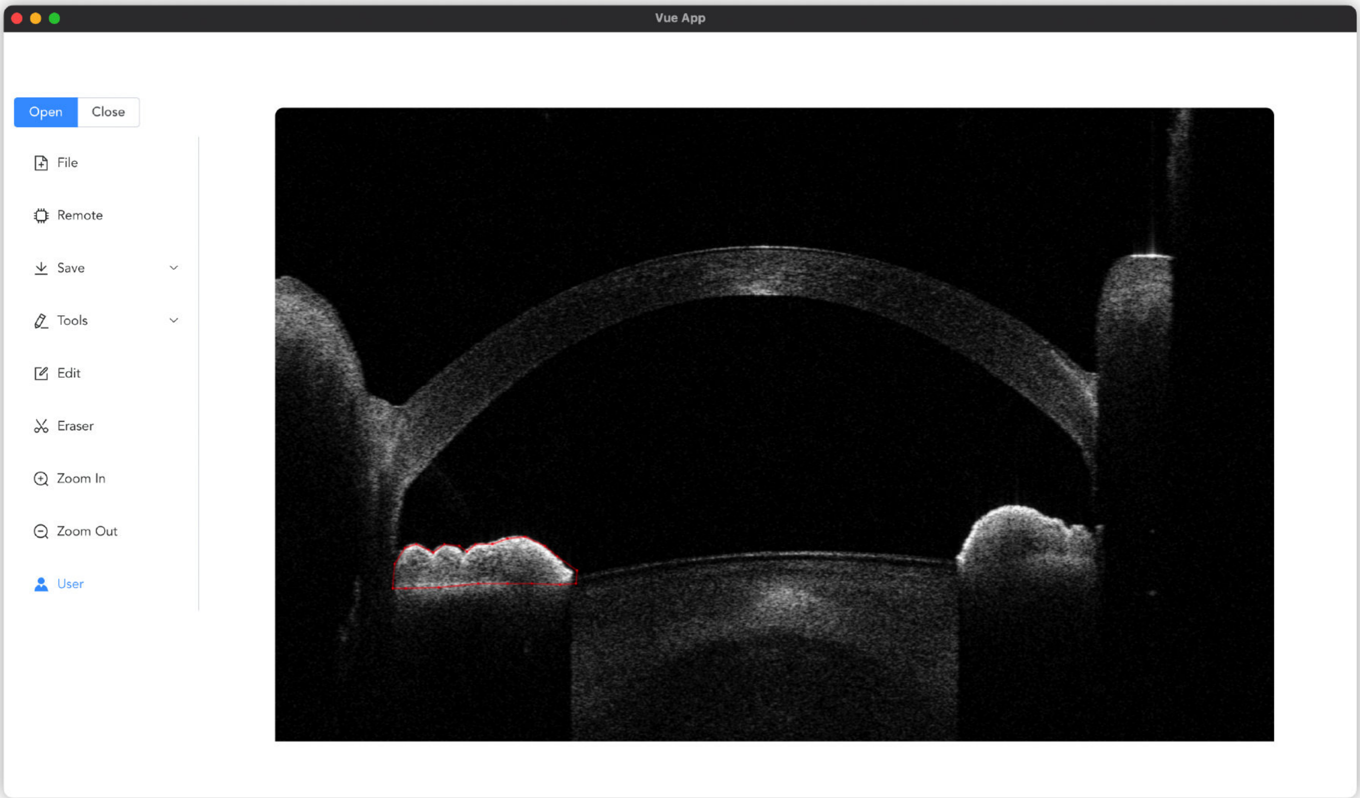
With the development of computer technology, especially the rapid progress of deep learning in recent years, the computer-aided diagnosis system has become a powerful assistant for doctors. However, the effect of deep learning relies on a large number of accurately labeled images, especially in the field of medical images, and the high cost and difficulty of labeling hinders the popularization of deep learning technology, and there is a lack of labeling software on the market that can adapt to the case of data defects. To address this problem, we propose a fast iris annotation system based on OCT images by combining weakly supervised techn iques. The system separates the front and back ends, which is beneficial for project management and data security, and integrates three weakly supervised algorithm models for three data defect cases, and the corrected annotated images are sent back to the back end for model correction. For iris images, a simple and efficient key point annotation method can be used. For other medical images, it can also be used to improve the annotation efficiency of users and reduce the labor cost according to their needs.

2 Project Design

2.1 Front-end Design

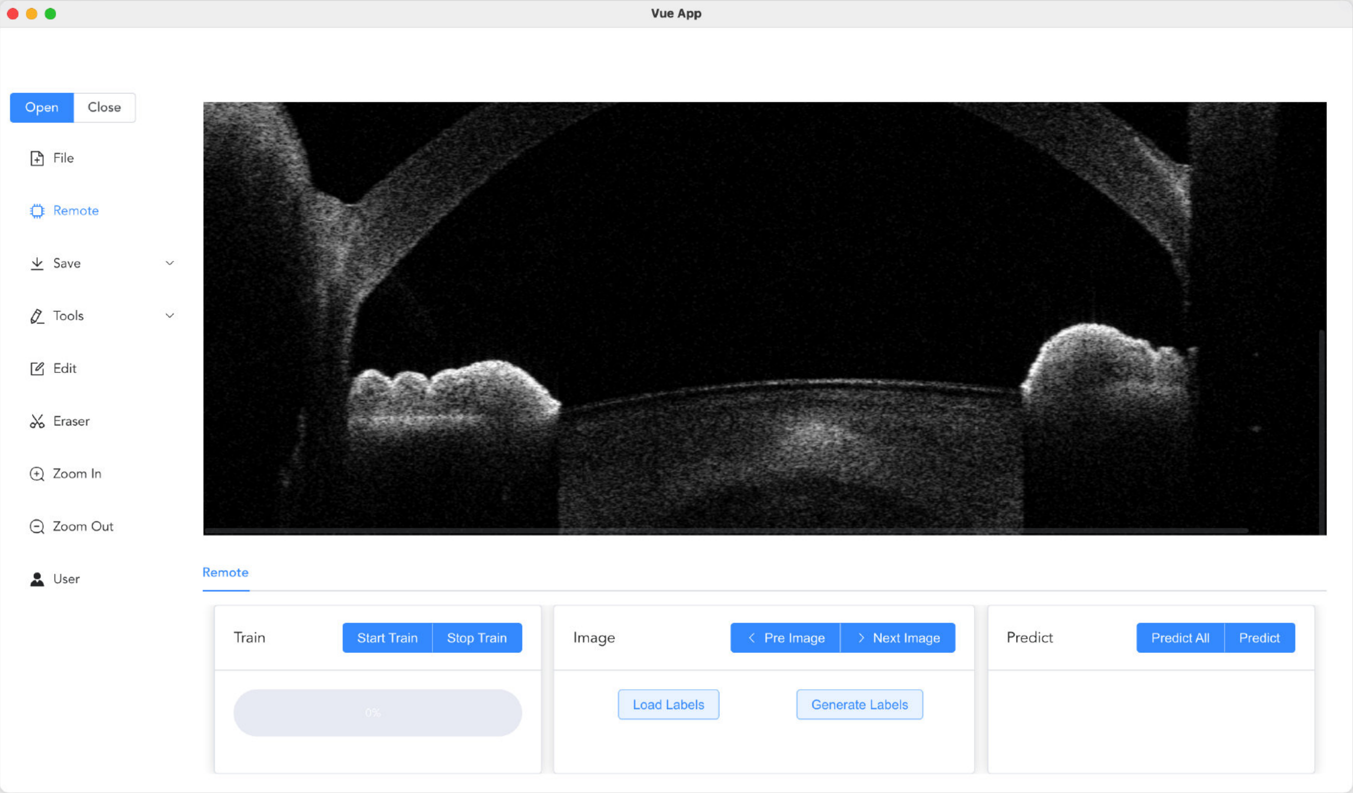
In the front-end design, we use the html+css+js architecture, and the main annotation functions are implemented in js. At the front end, we have implemented basic labeling functions including point, line, polygon and rectangle labeling, data exchange with the back end, temporary storage of labeling data, and prediction results. Now the front-end has supported users to load data from the back-end, annotate data on the front-end, store the annotated data in the back-end or get annotated data from the back-end, and display the back-end prediction results.

2.1.1 Total view:



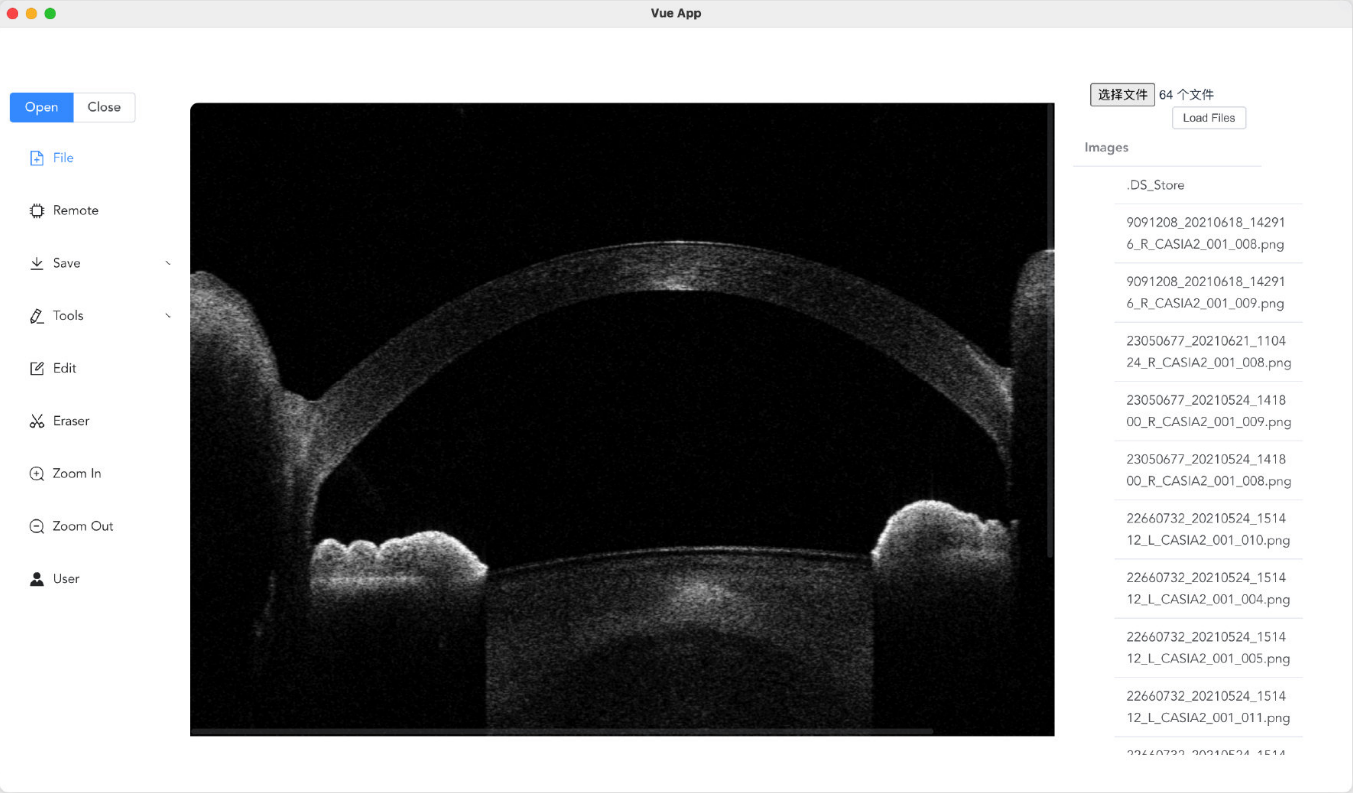
2.1.2 Remote Panel:

Click the "Remote" button and a remote panel will pop up below. In the train module, click "Start Train" to start training, click "Stop Train" to stop training, and there is a scrolling training progress display animation below. In the image module, click "Pre Image" to switch to the previous image, and "Next Image" to switch to the next image. Click the "Load Labels" button to load the label from the back end. Clicking on the "Generate Labels" generates label images that can be trained. In the predict module, click "Predict All"to prediction all the images, click "Predict" to predict current images, and there is a list of models below to choose.



2.1.3 File List:

Click the file button, and the file list will pop up on the right. Click “选择文件” to select a dataset locally. Click the “Load Files" button to load the remote cloud dataset. Scroll through the list of images and tap to select an image.



.DS_Store

9091208_20210618_142916_R_CASIA2_001_008.png

9091208_20210618_142916_R_CASIA2_001_009.png

23050677_20210621_110424_R_CASIA2_001_008.png

23050677_20210524_141800_R_CASIA2_001_009.png

23050677_20210524_141800_R_CASIA2_001_008.png

22660732_20210524_151412_L_CASIA2_001_010.png

22660732_20210524_151412_L_CASIA2_001_004.png

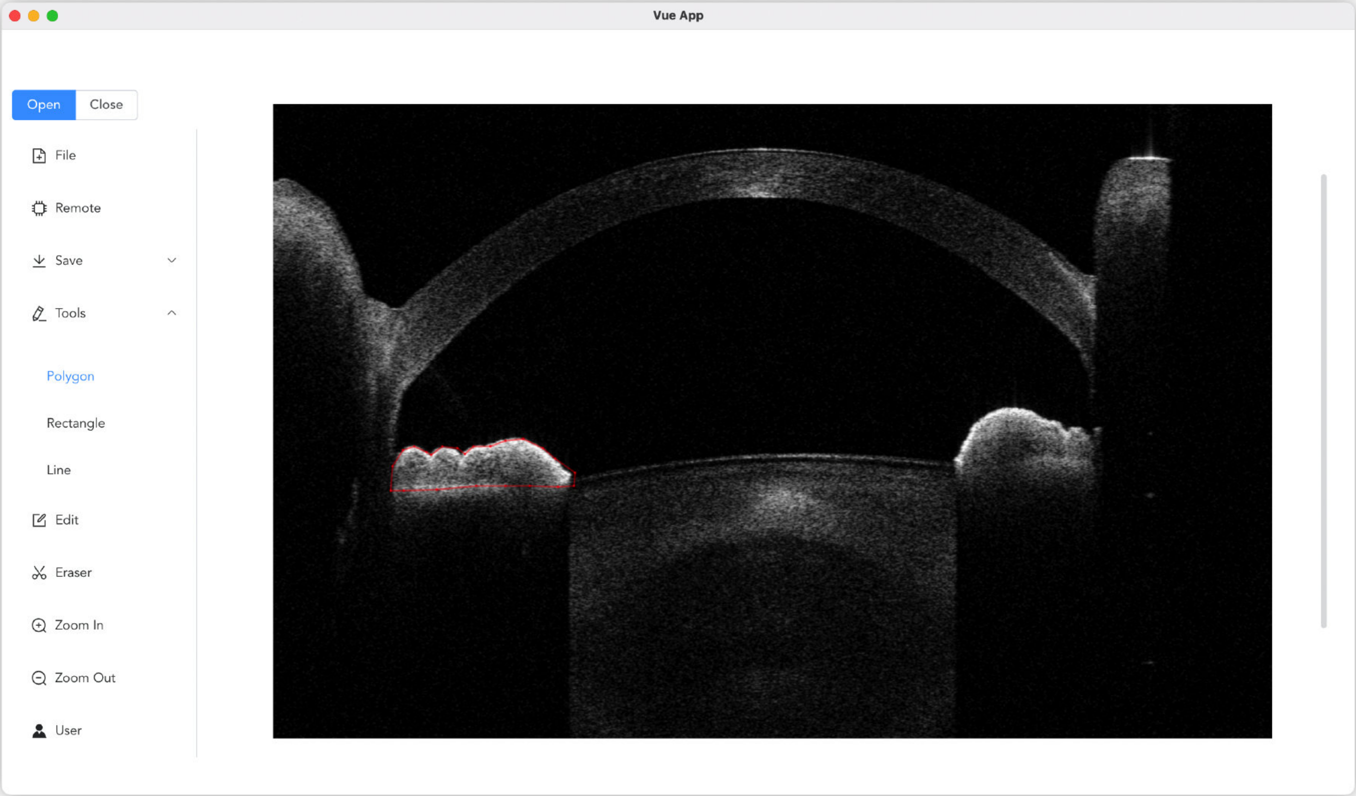
22660732_20210524_151412_L_CASIA2_001_005.png

22660732_20210524_151412_L_CASIA2_001_011.png

22660732_20210524_1514

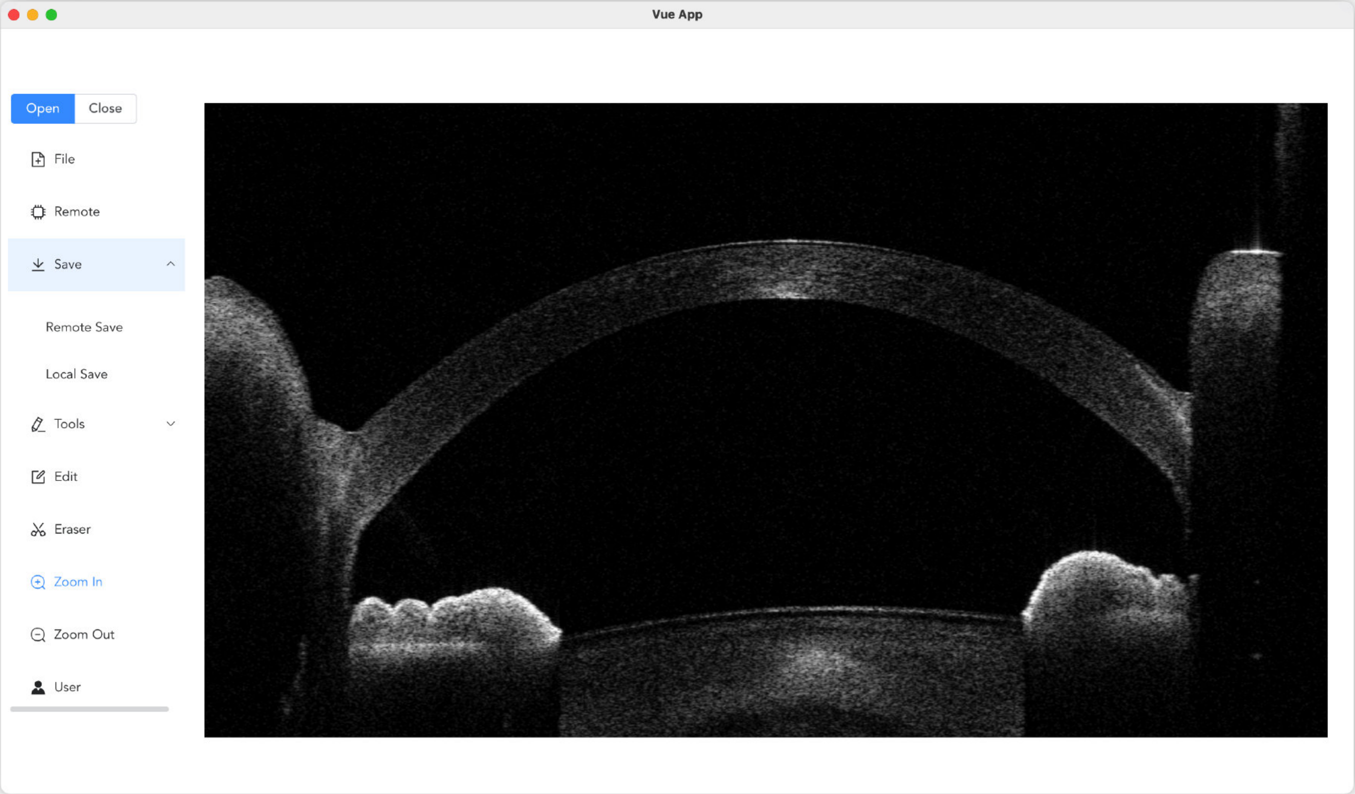
2.1.4 Labeling View:

Clicking the “Tools” button will bring up a selection list for the Labelers. Among them are polygons, rectangles and lines. Select one to annotate on the image. Click the "Edit" button to modify the label. Click the "Eraser" button to erase any of the label points.



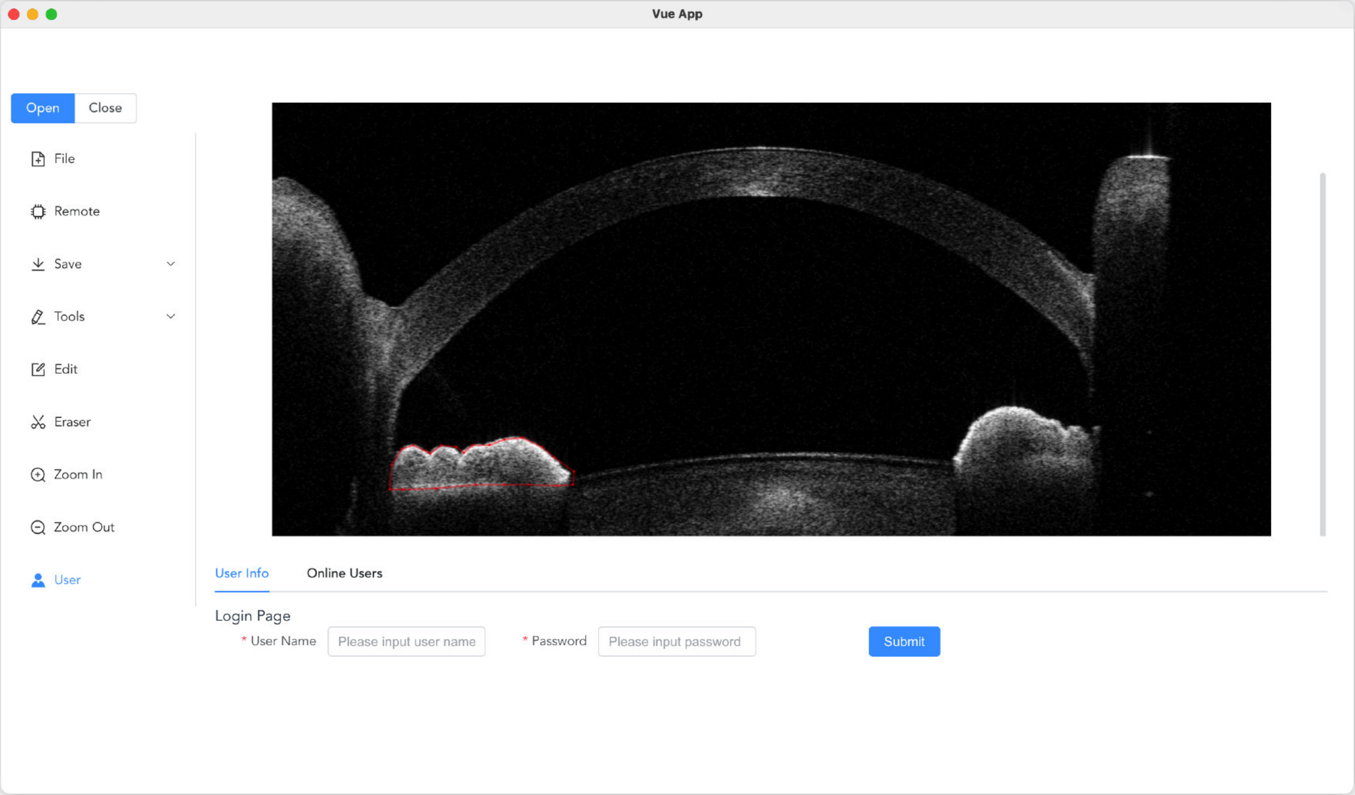
2.1.5 Save Labels:

Click the “Save” button and the save selection box appears. Click “Remote Save” to save the annotations to the cloud server. Click “Local Save” to save the label locally.



2.1.6 User Panel:

Click the "User" button and a user panel will pop up below. The user needs to log in, and the user info module displays the user's information after logging in. Click on the Online Users module to see other users who are online at the same time and their user information.



Login Page

* User Name Please input user name

* Password Please input password

2.2 Back-end Design

2.2.1 Annotation Function

After the back-end receives the annotation information from the front-end, the back-end will store the annotation information from the front-end into the postgres database, and we will put the data into different tables in the database according to the annotation data types. Currently, the polygon data is stored in the polygon table, while the point and line segment data is stored in the point and line segment tables.

The design of polygon table is shown above, divided into three columns which are id, imageid and label information . Id and label are one-to-one correspondence, and then each imageid corresponds to an image stored on the server. To establish the link between the image and imageid, we create the name table on the purpose of finding image by image id.

Since we store the images in a folder, we can get the images stored on the server simply by the name of the image we need to store. When we have the imageid, we can get the image, and when we have the image path, we can find the corresponding imageid in the database and then find the corresponding annotation information.

2.2.2 Training Function

When the user has finished the annotation for one day, the user can click the training button to train the model. If the local data is not found in the database, then we know that the image is not labeled. After obtaining and generating the training labels, we feed all the image data and the labeled data into the training of the weakly supervised model, which is trained and ready to be used when the user performs the next labeling process.

2.2.3 Prediction Function

When the user needs to perform annotation based on the model's prediction, the user will click the predict button. When the predict button is clicked, the backend framework will call the previously trained weakly supervised model to predict the current image. The prediction result is converted into a set of sequential point coordinates and sent to the front-end. The predicted result is presented by the front-end. Since the weakly supervised model has not yet been implemented, our current display system calls the model with the trained fully supervised model UNET.

3 Framework and Functionality

This semester we update our architecture and features, the main change being that we will be updating our front-end architecture and doing new features on that architecture that the previous architecture could not accommodate. We will use Electron + Vue.js instead of html + js + css, the advantage of which is that we can separate the user interface from the server, in the last semester the user markup interface relied on the ssh connection between the client and the server, in this semester we will separate the user markup interface, the advantage of which is as follows.

The first is that when we build our interface and client with electron+vue.js, we have the ability to annotate on multiple platforms. Users from Windows, Linux, or Mac can use the software to annotate.

Secondly, we can complete the offline annotation function from the server, after the user annotation application is ready, we can complete the offline annotation function, so that if the data previously transmitted by the server is cached in the client, the user can annotate according to the cached image data, store the annotation information locally, and wait until we can communicate with the server to synchronize and update the annotation information.

Thirdly, the function of simultaneous annotation by multiple users can be realized. When the client exists independently of the server, many users can work on their personal computers at the same time, and multiple clients can connect to the server at the same time. This way we only need to improve the synchronized update of annotation data under multi-user use to realize the multi-user simultaneous annotation function.

In addition to these benefits of the new architecture, we also improve the communication between the client and the server. Users can communicate with the server in real time to obtain information about the current training situation and labeling progress, as well as real-time control of the server training model. This can solve the problem that the user can not reach the server to control the server state.

4 Model and Dataset

4.1 Normal Weak-supervised Model/Algorithm

- Cross Pseudo Supervision[1]
- Cross Consistency Training[2]
- Mean Teacher[3]
- Deep Adversarial Networks[4]
- Uncertainty Aware Mean Teacher[5]

4.2 Dataset

- Tomey Iris Dataset
- Oxford-IIIT pet Dataset[6]
- COCO Dataset[7]
- Pascal Voc Dataset[8]

5 Application Value and Prospect

This project focuses on solving the problem of "difficult annotation" in the field of medical images in deep learning, using weakly supervised and incremental learning techniques to subtly reduce the annotation threshold, which is conducive to the research and development of deep learning in the field of medicine.

This project is based on iris annotation, but not limited to iris; based on OCT images, but not limited to OCT images. The annotation system implemented in this project can be applied to other scenarios as a more general tool.

This project has completed the design of the overall front-end and back-end system architecture and the initial construction of the front-end and back-end. The front-end functions include the annotation of points, lines and contours in the image data. The back-end functions include the extraction and storage of annotated data in the database, and the construction, training, and prediction of deep learning models.

The project has added the function of relying on iris feature points for deep learning training to the existing deep learning model, which is based on the location of the feature points as a priori knowledge to enhance the model training effect.

6 Future Work Schedule

In mid to late May, we will continue to improve the front-end functionality of the software and try to implement the related multi-player collaborative tagging feature. After that, we will implement a variety of weakly supervised models, and if we have enough time, we will also improve on the models. Our list of models includes Cross Consistency Training, Mean Teacher, Deep Adversarial Networks, Uncertainty Aware Mean Teacher, and others.

In early June, we will deploy the weakly supervised model and compare our experimental results. The results will include three time comparisons: using traditional data labeling, using weakly supervised model-assisted labeling, and using multiple people using weakly supervised assisted labeling. We will analyze the time differences of the approaches and the advantages and disadvantages of each.

7 Expected Outcome and Benefits

After users deploy the system to their servers, experts can annotate it on the web side.

The results hit the pain points of medical image segmentation problems, as the system has the following features:

- A corrected annotation model designed for the problem of difficult annotation and automatically adds the corrected data to the training to reduce the burden of experts.
- A front- and back-end separation model designed for the problem of data security.
- A variety of weakly supervised algorithm models are integrated for data accuracy and precision defects to improve accuracy

Inference

1. Chen, X., Yuan, Y., Zeng, G., & Wang, J. (2021). Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2613-2622.
2. Ouali, Y., Hudelot, C., & Tami, M. (2020). Semi-Supervised Semantic Segmentation With Cross-Consistency Training. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12671-12681.
3. Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NIPS*.
4. Huang, Y., Panahi, A., Krim, H., Yu, Y., & Smith, S.L. (2019). Deep Adversarial Belief Networks. *ArXiv, abs/1909.06134*.
5. Hegde, D., Sindagi, V.A., Kilic, V., Cooper, A.B., Foster, M., & Patel, V. (2021). Uncertainty-aware Mean Teacher for Source-free Unsupervised Domain Adaptive 3D Object Detection. *ArXiv, abs/2109.14651*.
6. Oxford-IIIT pet Dataset from https://www.robots.ox.ac.uk/~vgg/data/pets/
7. COCO Dataset fromhttps://cocodataset.org/#download
8. Pascal voc Dataset fromhttp://host.robots.ox.ac.uk/pascal/VOC/