User guide

Due to file size limitations on Moodle and GitHub, our model weight files are stored on Dropbox. Before running the system, you need to download these weights and place them in the **weightModel** folder. For more installation details, please refer to the **README.md** file. Additionally, we have provided the results of unit tests and UI tests conducted before delivery.

This compressed package contains all other files required to run the project. If you need a backup, you can download it from GitHub (please note that the backup is on the **Final** branch, not the main branch).

Dropbox link: https://www.dropbox.com/scl/fo/1k1hztmtpecfgvybjbrnw/AAVFRmIT07-TVkHjdNefahA?rlkey=4jwrklogub942nh675vx1cc1h&st=5ktbm02a&dl=0

Github link: https://github.com/unsw-cse-comp99-3900/capstone-project-2024-t3-9900f16asuperlu/tree/Final

Project Structure and Introduction

1. The **Docker_laddernet** folder contains the **LadderNet** model along with its training and prediction functionalities. It includes **app.py** to enable the prediction function and a **Dockerfile** for Docker packaging. When the model is invoked, it returns the corresponding prediction result in PNG format.

2. The **labels** folder contains the manually annotated labels for all images in the DRIVE test set. These labels are used to compare the predictions generated by the model with the manual annotations for metric calculation.

3. The **modelWeight** folder contains the weight files for two models.

4. The **processed_zips** folder stores the prediction results in ZIP format.

5. The **static** folder follows the traditional Flask framework structure. It contains the **CSS files** for the index webpage, a **results folder** for saving individual prediction results, and the platform's logo image.

6. The **templates** folder contains all frontend pages, including the HTML files for login, registration, homepage, prediction history, prediction results, and more.

7. The **uploads** folder stores images uploaded by users.

8. The **config** file contains the database configuration settings.

9. The **databaseModel** defines the database structure, generating tables for **user** and **history**.

10. The **Unet_Model** folder includes the structure of the **AggreUnet model**, along with its **prediction and metric calculation** functionalities.

11. The **docker-compose** file packages the system and the new **LadderNet** model into containers.

12. The **user guide** folder contains the **User Guide** document, the **README** file, and the results of unit tests and UI tests.