DRPF Class Diagram

By: Maurits Krijnen

Version: 1.0

Date: 24/03/22

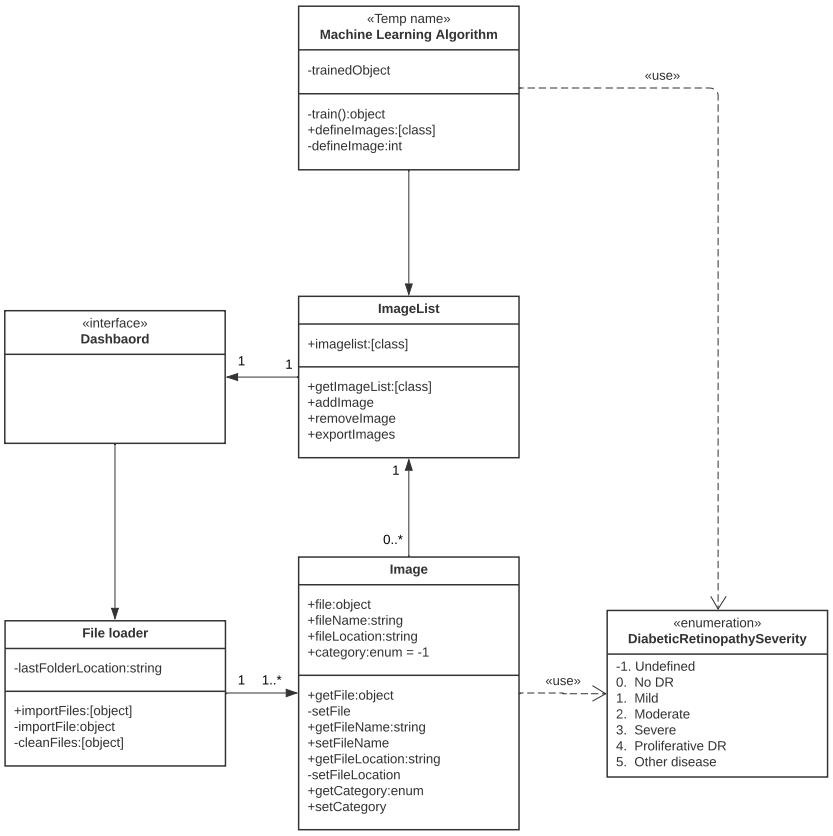
Table of Contents

Description

The UML Class diagram is a graphical notation used to construct and visualize object oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

- classes,
- their attributes,
- operations (or methods),
- and the relationships among objects.

Diagram



Explanation

This class diagram is an extension of the dashboard prototype. It shows a more technical in depth look into the design of the application. Depicted are classes and their relation to each other, and other objects.

Dashboard

This is the frontend shown in the prototype. This is where the user interacts with the application

File Loader

The file loader class handles the loading of file and turning them into Image objects.

Image and Imagelist

The image object is a class containing data including:

- An image file
- Image Name
- Folder location of the file
- An Enumeration indicating the presence of DRP

These images are saved in the Image list object. This class contains functions for adding and removing images.

Enumeration/Diabetic Retinopathy Severity

The enumeration contains categories describing the severity of Diabetic Retinopathy. This object is used in the Image class.

Machine Learning Algorithm

This class will contain functions for determining the Enumeration category of an Image.

Approach

For the class diagram I tried translating the functionality described in the prototype to UML classes. I first created the dashboard object, and afterwards started creating classes for the buttons I created for the prototype. First the file loader for the import function, after which I created the Image and Image list class. I added the enum during the image class creating process and the ML algorithm at the end.

Now I had a good idea of what classes would be required but not how they would work. While considering the functions inside the classes, I wrote down some functions that would be required in the back-end to make the prototype design work.

After making some small adjustments and adding the class connections I felt the initial design was done.

Conclusion

Where creating the prototype helped me gain a better understanding of what kinds of functions would be expected and/or required by a user, the Class diagram helped build out the technical requirements of the project.

Using this diagram I can quickly setup the classes for the dashboard application. It also improves the transferability for future developers looking to get a better understanding of the working of the project.