

Project's document

1. Personal information

- Title: Picture Annotator
- Creator's name: Thang Phan
- Student number: 1008713
- Study program: BSc. Economics
- Date: 12/05/2023

2. General description

The Picture Annotator is a software application designed for users to easily add annotations to their pictures. With this program, users can easily select an image, add annotations, and export them as a dataset used to train a deep learning model for object detection task. It is user-friendly, intuitive, and efficient in managing images, making it a valuable tool for any individuals.

3. Instructions for the user

The program can be launched by using Python or through the Command Line Interface. More details can be found in the [README.md](#) file located in the project's directory.

4. External libraries

The external libraries used in the projects are:

- PyQt6 for implementing the UI of the program
- Pillow for processing images
- Pytorch for supporting export dataset

5. Structure of the program

The program composes of these main classes: UI, Canvas, CustomGraphicsView, FileList, MenuBar, FilterWidget.

The main class is the UI class and the others are components of it.

6. Algorithms

The program does not have any algorithms implemented.

7. Data structures

The data structures used in the project are mostly built-in Python data structures (list, tuple, set, string, integer, dictionary, boolean). In addition, the semi-structure xml is used to store the annotations of the images.

8. Files

The program has to deal with images and annotations file. Therefore, it must be able to process images file format (.jpeg, .jpg, .png) and annotations file format (.xml).

- In images file format, the data are represented by pixels though different color channels (1 for black-white images and 3 for colorful images). Each pixel have the value from 0 to 255.
- In annotations file format, the data are represented as tree with nodes. Each node can have its tag and text.

9. Testing

The program is covered with import and export tests.

10. The known shortcomings and flaws in the program.

Although the program was implemented in a way that bugs and errors are yet to be discovered, there may still be some issues with the program.

11. 3 best and 3 worst areas

- Best areas:

1. User-friendly interface: The Picture Annotator has a simple and intuitive interface that allows users to easily navigate and make annotations to their pictures.
2. Efficient management of images: The program is efficient in managing images, making it a valuable tool for individuals who need to annotate large numbers of pictures.
3. Documentation of the program: The program is well-documented and highly detailed, making it easy for users to understand how to use it.

- Worst areas:

1. Lack of testing: The program does not contain any tests, which means that there may be bugs or errors that have not been caught.
2. Limited file formats: The program can only process images in .jpeg, .jpg, and .png formats, and annotations in .xml format. This may limit its usefulness for some users.
3. No algorithmic implementation: The Picture Annotator does not have any algorithms implemented, which may limit its functionality for some users who require more advanced image processing capabilities.

12. Changes to the original plan

The program follows closely with the original plan. However, there is still a change from planing to load the images from database to load them from files.

13. Realized order and scheduled

The project was implemented the UI and its components first, then the export functionality, and finally the integration part with Pytorch and documentation.

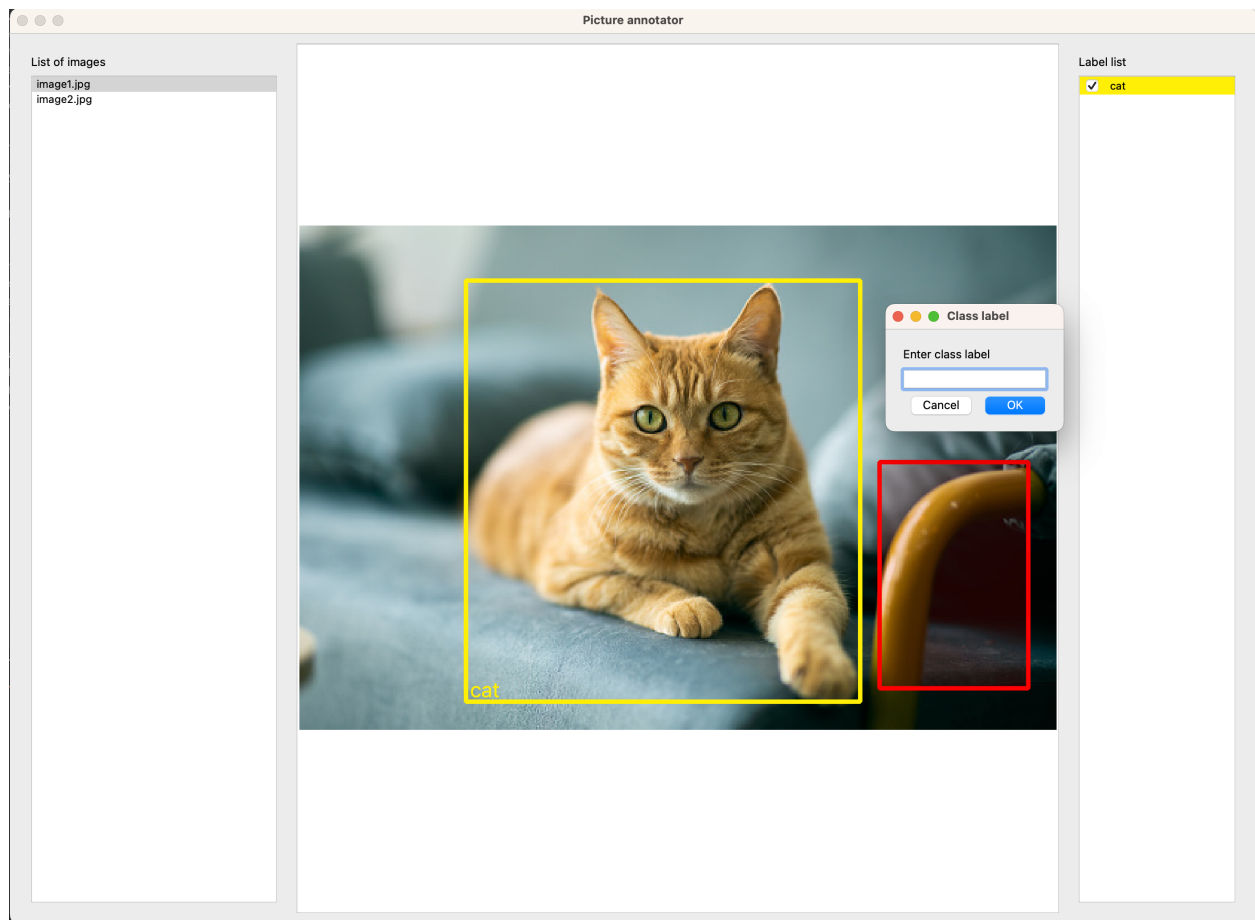
14. Assessment of the final result

The final result of the Picture Annotator is a user-friendly and efficient tool for annotating images and exporting them as a dataset for object detection. However, the lack of testing and limited file format support are significant flaws that may limit its usefulness for some users. In the future, the program needs to have a more comprehensive testing process and support for additional file formats to improve its functionality and usability. Overall, the program closely follows the original plan and satisfies all the functionality requirements.

15. References

- Course material: <https://plus.cs.aalto.fi/y2/2023/>
- PyQt6: <https://www.riverbankcomputing.com/software/pyqt/>
- Pytorch: pytorch.org
- Pillow: <https://pillow.readthedocs.io/en/stable/#>

16. Attachments



The UI of the project