Université Nice Côte d'Azur
Enrico FORMENTI

Année 2021/22
M1 Info
NNL

Project Exam
Part 1/2

Facial recognition: the mask/no mask case

Contents

1 Before starting

This project needs to work in an adapted environment. You are supposed to program in Python, hence first of all please update your installation to the last version (`3.10`). You need to install the following libraries if it is not already done:

- `PIL`: this is an handy library for quick image manipulation. I suggest to use it for loading images in a number of formats and to convert them to the internal bitmap model used for our purposes. Since the former version has been discontinued, there is a well-established new port and update which comes with the `pillow` package.
- `tkinter`: this is a well-known GUI system for Python. It is pretty simple and perfectly fits our purposes. You can find a number of tutorial on web. This is a nice wrap-around for the famous `Tcl-Tk` library used in many open projects all over the world. You can therefore find a huge amount of explanations and usage examples. The only drawback of `tkinter` is when working with mouse device and the `tkinter` event system. Indeed, I spent quite a long time before understanding that double-clicks events are indeed literally treated as two successive mouse-clicks and hence three events were dispatched: two button clicks and a double-click. Take this into account if you decide to used double-clicks related events. Feel free to use another GUI system if you don't like `tkinter`.
- `Shapely`: this is a package dedicated to computational geometry. I suggest to used it to compute objects intersections during image annotation. It is pretty precise and reasonably fast. Here is the package documentation and an explanation of the API.

- `OpenCV`: an open library for advanced image manipulation that it is suggested to used for more precise image resizing. You can find the library main website here. The website also containing a quite handful set of tutorials. Even if we are going to use a very limited subset of the API, maybe following the first tutorials is a good investment for the future.

2 The first part of the project

You are supposed to conceive an image annotation system which will prepare the data for the second part of the project. An image annotation system is a GUI program that takes an image and allows the user to manually detect a specific set of objects for later use. For each of these objects, the used will attribute a category. This information and the objects spatial coordinates will be saved on a file for later use.

We are going to describe the program and all the features that you are supposed to develop to assist an operator in the image annotation task.

Call the program `ImageAnnotator`. First of all the program should allow to load images and show the to the user/operator for starting annotating them. The following features should be implemented:

box selection: when an image is shown, by using its mouse the operator will select a series of boxes on the image. For each box he will attribute a category chosen from a given list. I suggest to implement this action using mouse events. For example, you can wait for a mouse-click on the image and for the following mousing motion event. At this point you can start drawing a rectangular box from the initial point of the mouse click event to the current mouse position and when the user releases the mouse button, you can fire a popup to inform the user about the box (points, size, etc) and to let him choose the category from a list.

box editing: when the user double-clicks on an existing box, the system should fire a popup allowing the user to get information about the selected box and to allow him to update the category if needed.

annotation saving: the user should be able to save his annotations and all the related information on a file for later usage. See Section 2.2 for the file formats to use.

categories add/replace: during the annotation process the user should be allowed to add new categories to the current list or to replace a category name with another. Of course, in the case of a category replacements, the new category name should be also consistently replaced in all the boxes currently being edited in the program.

importing categories: the program should allow to import a list of categories from other files formats than `.json`. A minima, the `.csv` file format should be supported.

2.1 Boxes intersection caveats

When an image contains many sensible details, it is quite possible that one would select boxes which have large intersections with each others. However, allowing too large intersections can mess out learning algorithms. The same it is if the box area is too and hence it contains too few informations to distinguish it. For this reason, the program should reject if a box surface is less than 40 pixels in total or spatial dimensions (height and width) are too small for example less than 5 pixels. Moreover, if a box has an intersection with other boxes for more than 20% of its surface should be discarded. One should also discard a box if it completely covers a pre-existing box or it is completely contained in a pre-existing one.

Pay attention that in zones with a high density of boxes, computing the total surface of intersections with other boxes can be quite messy. For this reason, you should consider the primitives in the `shapely` package. Remark that this package is quite handy also to find the point-to-box intersections.

2.2 File formats

Images: a minima, the program should be able to load images in the `.jpg` and `.png` file formats. It is mandatory to save the images extracted in `.png` format.

Annotations: all the annotations should be saved in a single file in `.json` format.

Categories: the categories should be loaded from (resp., saved in) a file in `.json` format.

Figure 1: A screenshot of the Image annotator in action. Remark that some of the boxes have non-empty intersections with each other.

2.3 Bonus points

The requirement described above are minimal requirements. All improvement and addition to them can earn you more points. You can add, for example, a help system, a preference pane to configure the program, categories removal, improved exploitation of mouse buttons (we did not you one of the buttons...), *etc.*

3 How to produce data and how much data we need?

There is a number of way on how to produce data for this project:

1. Just visit Kaggle website and search for images with people with mask/no mask.
2. Take your smartphone or a digital camera (if you have one) and take shots of you and your team mates in different positions and in different backgrounds.
3. The same as above but just walking outside in the streets or in stores.

We don't need an excessive amount of data for this project (since we will take some shortcut as we will see in the next part of the project). About a hundred pictures should be enough for our purposes.

4 Rating

The project want to be a representative example of a real application case of the notions seen in our course. Therefore, the ration will consist in three marks:

(C) the source code;

(P) the performances;

(R) the report.

Each of this rating criteria will be detailed in the next sections.

4.1 Code (C)

As all project in informatics, you are supposed to produce a program which satisfies the requirements. In our case, they are detailed in Section 2. What we are going to rate here is at which degree your solution satisfies the requirements, the quality of the source code, the re-usability, the genericity and the code documentation.

4.2 Performances (P)

There are no huge requirements concerning the computational performances in this first part of the project. However, we will rate here how responsive is your solution to the GUI actions. The quality of the produced annotation files and their production time, *etc.*.

4.3 Report (R)

The final mark is on the quality of the report. Indeed, you are supposed to produce a short report (preferably in LaTeX) containing *a minima*:

- a short description of the project and its goals;
- a short description of how you produced data and how they were adapted for the usage in the project (if any adaptation was necessary);
- the methodology adopted, the technologies used, and the choices made to meet the requirements;
- a short section containing conclusions, perspectives, further possible improvements, *etc.*;
- a short presentation of the members of the group and of their role in the project.

4.4 Final rating

Each part of the project will be rated on 5 points and it will provide a final mark $C + P + R$ to which we will add the mark for the TPs (which is rated on 5 points).

Beware: antiplagiarism tools will be used in the evaluation of (C) and (R) for detecting plagiarism *w.r.t.* other projects or *w.r.t.* code found on the web. Actions will be taken in the event of apparent plagiarism.

5 Teams

Each team is made of at most two or three students. All other team compositions are forbidden. To register your team, please go on the Moodle website for our course. Choose the tab `Project` and look for the activity `Teams formation`.

Attention: the activity `Teams formation` has a limited duration. It will end by **December 15th, 2021 at 23 o'clock**.

6 What you should submit and when

You should submit a compressed (only `.zip` format will be accepted) file with a name `NNL-2021-XX.zip` where `XX` is the ID of your team. This archive **must** contain the following four directories: `pre`, `src`, `img` et `rep`. These directories are supposed to contain, respectively:

- the sources of the scripts or the programs used for data pre-treatment if any;
- the Python scripts for your project;
- the graphical data or any other image used in the project;
- the report (if you make in LaTeX, please provide full source, tables and figures to be able to produce a pdf from sources).

Deadline: submit your project via the Moodle website of the course before **December 28th, 2021 at 23 o'clock**.

N.B.: at submission time, the approval of a single member of the team will be enough to complete the submission activity.

6.1 Late penalties and deadline extension requests

If the project is not submitted within the deadline, you should expect penalties on the rating. The penalties will be proportional to the delay. No deadline extension without penalties is allowed, except for exceptional cases which will be judged case by case.

7 Help!

If during the project you need help for some reason, please try to apply the usual algorithmic solution:

1. double your efforts during a finite lapse of time (max 1 day);
2. if step 1. is without results ask your classmates;
3. if step 2. doesn't work ask classmates of the previous year (they are in M2 at present);
4. if also the previous step is unfruitful then contact the teacher urgently!

Beware that step 1. to 3. should no take more than three days!

Good luck to you all!