

Design Manual Number Plate Recognition

Ribemont Francois

Supervisor: Nigel Whyte

January 16, 2012

Contents

1	Introduction	2
2	Graphical User Interface	2
	2.1 Overview	2
	2.2 Other widgets	3
	2.3 File chooser widget	4
3	UML diagrams	6
	3.1 Use cases	6
	3.2 Sequence diagrams	7
	3.3 Class diagram	9

1 Introduction

This document presents in a first part the Graphical User Interface that will be created and the different design diagrams, the use case diagram, the sequence diagrams, and finally the class diagram. Every widget of the graphical user interface is fully explained and detailed. About the design diagrams, every method, every diagram is fully documented and detailed as well.

2 Graphical User Interface

2.1 Overview

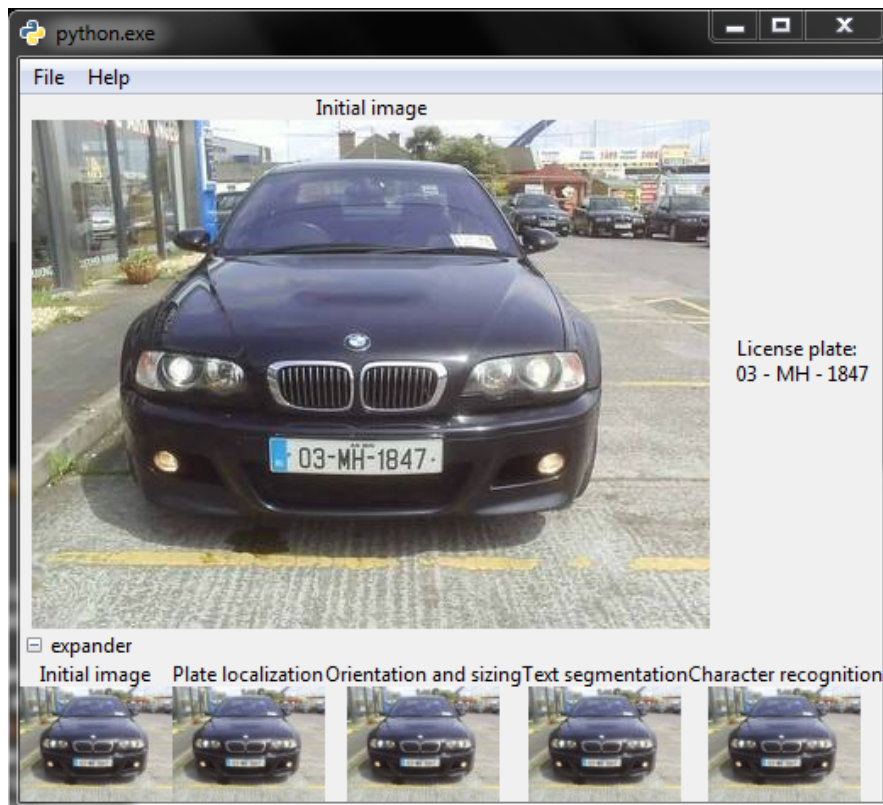


Figure 1: GUI: Main interface

This is the main window. The interface is split into different parts. On the top, there is the menu (see below). Right below this menu, we have a picture,

and a label on the right. The image part shows the one of the different steps of the process, and the label shows the result of this process (the car plate number). Finally, there is an expandable panel where are different thumbnails corresponding to each step of the process. A click on it from the user will change the image in the middle part of the interface by the one the user clicked on.

2.2 Other widgets

File menu

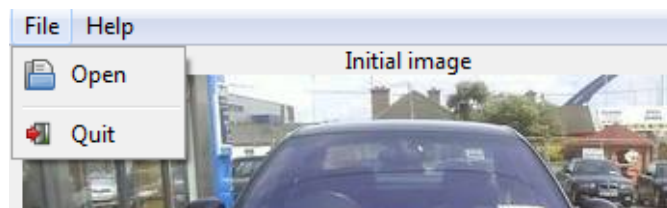


Figure 2: GUI: File menu

In this menu, we have two options: Open and Quit. The first one is used to run the file chooser widget (see Figure 4), and the second one is used to quit the program.

Help menu

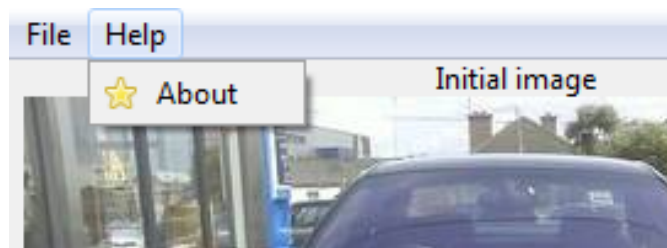


Figure 3: GUI: Help menu

In this menu, there is only one option which is About. Clicking on it will run the about window (see Figure 5).

2.3 File chooser widget

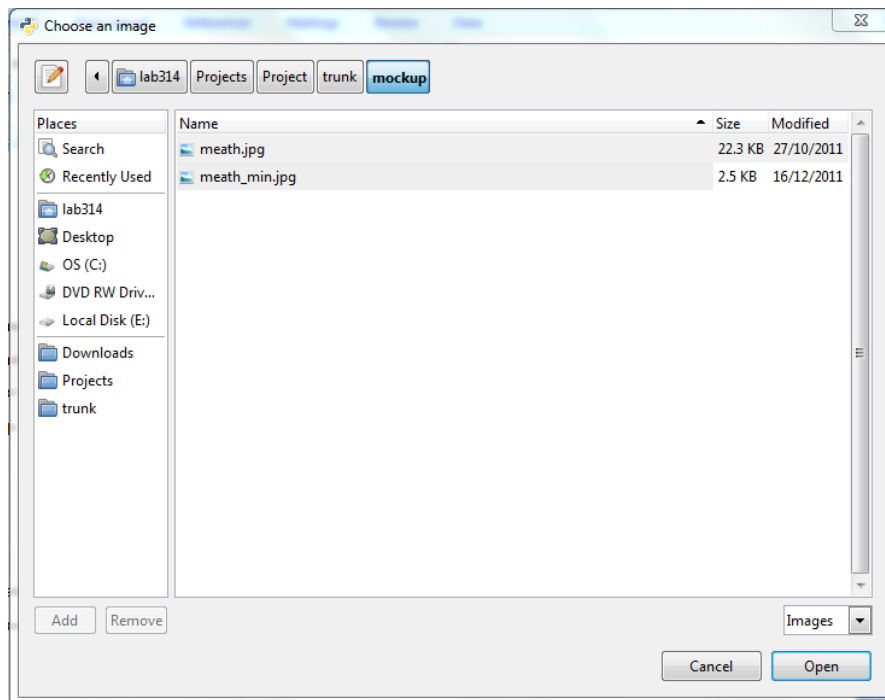


Figure 4: GUI: File chooser window

This widget shows the file system where we can look for the file we want to get the car plate number. This includes a filter on the bottom right. We can filter images (files with a name that ends with .jpeg, .jpg, .png, .bmp) or show all files (in case the name of the image does not end with the image extensions, but is still an image).

About dialog

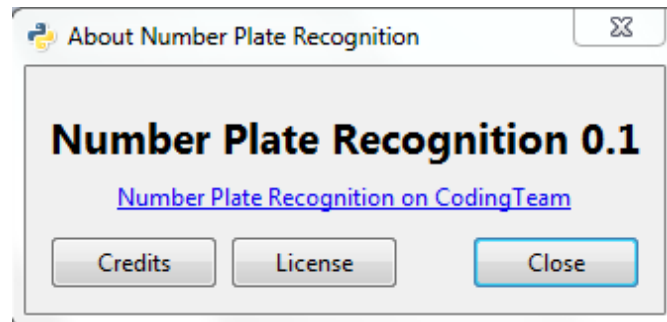


Figure 5: GUI: About dialog

This dialog window shows the credits of the application. The blue link is a link redirecting on the CodingTeam website where the project is hosted, where we can see the code online, report bugs... In the "Credits" part, we can see the author of the project, and the supervisor. Clicking on the "Licence" button shows the licence text. Finally, the "Close" button closes the window.

3 UML diagrams

3.1 Use cases

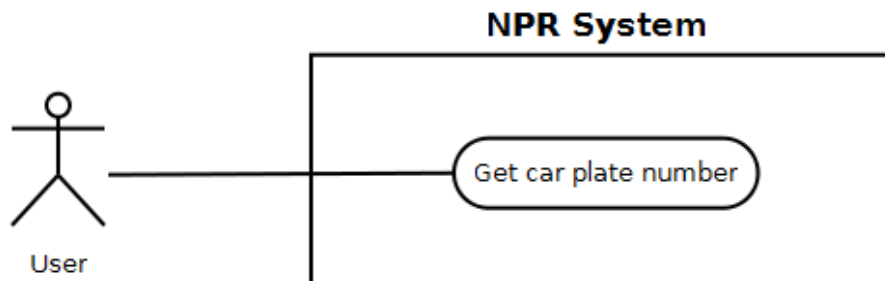


Figure 6: Use cases

Use case: Get car plate number

- Actor: user
- Description: This use case takes place when the user wants to get a car plate number from an image. The user simply uses the graphical interface to do it.

3.2 Sequence diagrams

In this part, the sequences diagrams are presented. There are two main sequence diagrams: Start program and Open Image. The detailed methods are exposed in the Class diagram (see below).

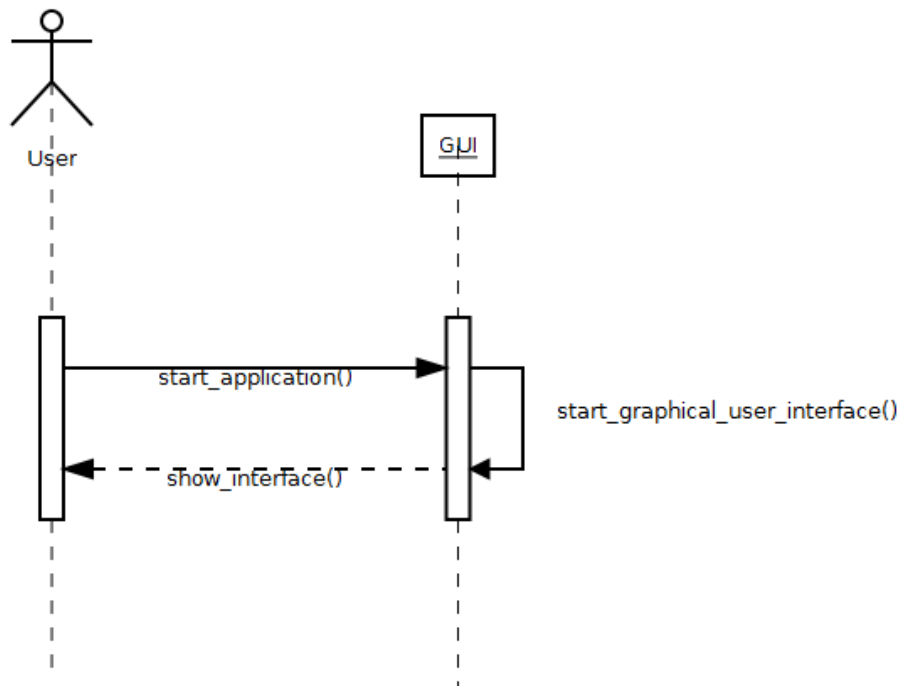


Figure 7: Sequence diagram: Start program

This sequence diagram happens when the user starts the application. First, he launches the software, this is represented with the `start_application()` function. It will call the method `start_graphical_user_interface()`, and show the main window of the program (represented as `show_interface()`).

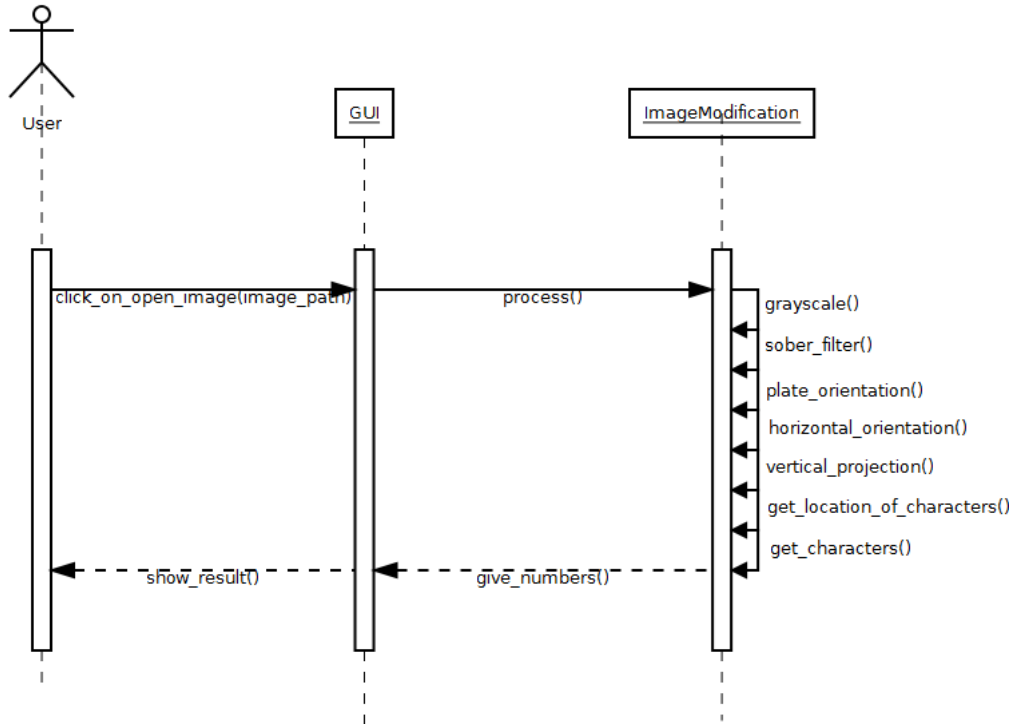


Figure 8: Sequence diagram: Open image

This sequence diagram happens when the user clicks on the "Open image" menu. This will call the `click_on_open_image(image_path)`. Then it will call the GUI's `process` method which will call the following ImageModification's methods:

- `grayscale()`
- `sober_filter()`
- `plate_orientation()`
- `plate_orientation()`
- `horizontal_projection()`
- `vertical_projection()`
- `get_location_of_characters()`
- `get_characters()`

3.3 Class diagram

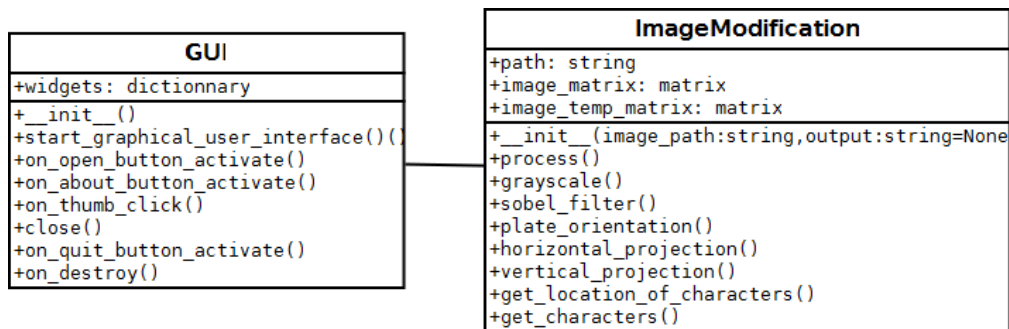


Figure 9: Class diagram

GUI

This class introduces a couple of methods and one attribute. The attribute is called `widgets`. It's a python dictionary containing names of all widgets used in the application associated to the widgets themselves (`gtk.Widget`). The methods are:

- `__init__()`: the constructor of the GUI class. This creates the object, and builds the graphical user interface, but doesn't show it. This method returns nothing
- `start_graphical_user_interface()`: this methods shows the graphical user interface build by the constructor. It returns nothing.
- `on_on_open_button_activate()`: this method is called when the user clicks on the Open image menu. This emits a signal handled by this method. It will create a `ImageModification` object, and call the `ImageModification`'s method called `process`. It modifies the value of different widgets. It returns nothing.
- `on_about.button_activate()`: this methods is called when the user clicks on the About button in the Help menu. This handles the signal emitted by the click. This shows the about dialog. It returns nothing.
- `on_thumb_click()`: this method is called when the user clicks on a thumbnail in the expandable menu. This will replace the image in the center of the main window by the one the user clicked on. So, it will change the value of the image widget in the center of the application. It returns nothing.

- `close()`: this method closes the program. It return an exit code which can be either 0 or 1. 0 if the program closed properly and 1 if there were errors.
- `on_quit_button_activate()`: this methods is called when the user clicks on the Quit button in the File menu. This handles the clicks. It will simply call the `close()` method.
- `on_destroy()`: this method is called when the user clicks on the closing button (the cross on the top-right corner). This will simply call the `close()` method

ImageModification

This class has three different attributes:

- `path`: this is a Python's string containing the path of the image the software is working on.
- `image_matrix`: this is a matrix containing the values of the pixels of the image.
- `image_temp_matrix`: this attributes is also a matrix containing the values of the pixels of the image, but it contains the values of an image before a modification. So, in this case, we still have all the values of the image while modifying the other one.

There is also a couple of methods:

- `__init__(image_path:string, output:string=None)`: this is the constructor of the ImageModification class. This builds the matrix of pixels from an image given by the `image_path` parameter. The `output:string=None` is an optional parameter used to know where to store the images of the different steps.
- `process()`: this method is called by the GUI class, and it will simply call the methods below. It returns nothing.
- `grayscale()`: this method modifies an image into the gray scale mode. It will modify the `image_matrix` attribute. It returns nothing.
- `sobel_filter()`: this method modifies the `image_matrix` using the Sobel Filter. It returns nothing.
- `plate_orientation()`: this method will search where the plate is, and only keep it. It will be easier to work only on the plate rather than the whole image. It modifies the `image_matrix` attribute. It returns nothing

- `horizontal_projection()`: this method processes the horizontal projection in order to split the plate into characters. It returns the projection.
- `vertical_projection()`: same as `horizontal_projection`, but using the vertical projection instead of the horizontal one. The two methods are used to split the characters. It returns the projection.
- `get_location_of_characters()`: this method is used to get the location of the characters using the two different projections. It returns a list of small image
- `get_characters(characters_image:list)`: the last method will process the different image given by `characters_image` which is a list of images. It will try to detect characters using the character template matching. It returns the car plate numbers.

List of Figures

1	GUI: Main interface	2
2	GUI: File menu	3
3	GUI: Help menu	3
4	GUI: File chooser window	4
5	GUI: About dialog	5
6	Use cases	6
7	Sequence diagram: Start program	7
8	Sequence diagram: Open image	8
9	Class diagram	9