Image Processing

Image processing is the field of study and application that deals with modifying and analyzing digital images using computer algorithms.

The goal of image processing is to enhance the visual quality of images, extract useful information, and make images suitable for further analysis or interpretation.

Pillow (PIL) - which stands for "Python Imaging Library" : the most commonly used library for image processing in Python

-an open-source library for image processing tasks that requires python programming language. PIL can perform tasks on an image such as reading, rescaling, saving in different image formats. PIL can be used for Image archives, Image processing, Image display.

Graphical User Interfaces (GUI)

A GUI displays text as well as small images (called icons) that represent objects such as directories, files of different types,command buttons, and drop-down menus

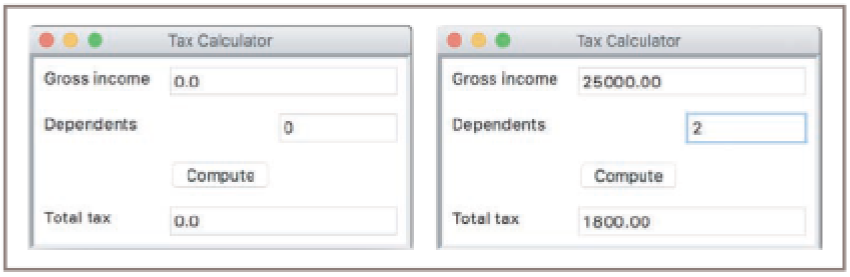
In addition to entering text at the keyboard, the user of a GUI can select an icon with a pointing device, such as a mouse, and move that icon around on the display.

Commands can be activated by pressing the Enter key or Control keys, by pressing a command button, or by selecting a dropdown menu item with the mouse.

Put more simply, a GUI displays all information, including text, graphically to its users, and allows them to manipulate this information directly with a pointing device.

The structure of a GUI program differs significantly from that of a terminal-based program.

Second, a GUI program is event driven, meaning that it is inactive until the user clicks a button or selects a menu option. In contrast, a terminal-based program maintains constant control over the interactions with the user. Put differently, a terminal-based program prompts the user to enter successive inputs, whereas a GUI program allows the user to enter inputs in any order and waits for the user to press a command button or select a menu option.



Gross income	0.0
Dependents	0
Total tax	0.0

Gross income	25000.00
Dependents	2
Total tax	1800.00

Figure A GUI-based tax calculator program

•A title bar at the top of the window. This bar contains the title of the program, “Tax Calculator.” It also contains three colored disks. Each disk is a command button. The user can use the mouse to click the left disk to quit the program, the middle disk to minimize the window, or the right disk to zoom the window. The user can also move the window around the screen by holding the left mouse button on the title bar and dragging the mouse.
• A set of labels along the left side of the window. These are text elements that describe the inputs and outputs. For example, “Gross income” is one label.

• set of entry fields along the right side of the window. These are boxes within which the program can output text or receive it as input from the user. The first two entry fields will be used for inputs, while the last field will be used for the output. At program start-up, the fields contain default values, as shown in the window on the left side of Figure.
• A single command button labeled Compute. When the user uses the mouse to press this button, the program responds by using the data in the two input fields to compute the income tax. This result is then displayed in the output field. Sample input data and the corresponding output are shown in the window on the right side of Figure .
•The user can also alter the size of the window by holding the mouse on its lower-right corner and dragging in any direction. Although this review of features might seem tedious to anyone who regularly uses GUIbased programs, a careful inventory is necessary for the programmer who builds them. Also, a close study of these features reveals the following effects on users:
•The user is not constrained to enter inputs in a particular order. Before she presses the Compute button, she can edit any of the data in the two input fields.
• Running different data sets does not require re-entering all of the data. The user can edit just one value and press the Compute button to observe different results.

Event-Driven Programming
Rather than guide the user through a series of prompts, a GUI-based program opens a window and waits for the user to manipulate window objects with the mouse. These user-generated events, such as mouse clicks, trigger operations in the program to respond by pulling in inputs, processing them, and displaying results. This type of software system is event-driven, and the type of programming used to create it is called **event-driven programming**.

Coding Simple GUI-Based Programs
There are many libraries and toolkits of GUI components available to the Python programmer, but in this we use just two: tkinter and tkMessageBox. Both are standard modules that come with any Python installation.tkinter includes classes for windows and numerous types of window objects.tkMessageBox includes functions for several standard pop-up dialog boxes.

Windows and Labels
Our first demo program defines a class for a main window that displays a greeting.In all of our GUI-based programs, this class extends tkinter’s Frame class. The Frame class provides the basic functionality for any window, such as the command buttons in the title bar.

To create a tkinter app:

- Importing the module – tkinter
- Create the main window (container)
- Add any number of widgets to the main window
- Apply the event Trigger on the widgets.

import tkinter

There are two main methods used which the user needs to remember while creating the Python application with GUI.
1.Tk(screenName=None, baseName=None, className=’Tk’, useTk=1): To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:
m=tkinter.Tk() where m is the name of the main window object

mainloop(): There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the

event as long as the window is not closed.
m.mainloop()

```
import tkinter
m = tkinter.Tk()
'''
widgets are added here
'''
m.mainloop()
```

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

Sr.No.	Operator & Description
1	Button The Button widget is used to display buttons in your application.
2	Canvas The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame The Frame widget is used as a container widget to organize other widgets.
6	Label

	The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	Listbox The Listbox widget is used to provide a list of options to a user.
8	Menubutton The Menubutton widget is used to display menus in your application.
9	Menu The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	Message The Message widget is used to display multiline text fields for accepting values from a user.
11	Radiobutton The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	Scale The Scale widget is used to provide a slider widget.
13	Scrollbar The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	Text The Text widget is used to display text in multiple lines.

15	Toplevel The Toplevel widget is used to provide a separate window container.
16	Spinbox The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	PanedWindow A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	LabelFrame A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	tkMessageBox This module is used to display message boxes in your applications.