

# Python GUI

## Abstract

Developed a Graphical User interface (GUI) in python. The GUI can be used to select the raster file in TIFF (.tif) format. By selecting the type of image segmentation, Object Based Image Analysis can be done through k means clustering segmentation algorithm, Chessboard, QuickShift, QuickShift\_ratio and SLIC. Currently, segmentation can be done through the GUI. The best means for OBIA implemented in the GUI is K means clustering. The show raster and show vector functions and the canvas was for the purpose of displaying geographical data, however it is not implemented currently. All other features work successfully. The main purpose of GUI was to perform Object Based Image segmentation that can be done through GUI in four different types of segmentation. The sample input file and output file can be found in corresponding folders within the GitHub repository that has been used for the test purpose. Using pyinstaller library there was an attempt to convert jupyter notebook file into exe file. Opening the executable file, the GUI works fine within the same user as it was created. However, transferring .exe file into other computer does not work without installing the dependencies. Therefore, for making the code standalone other means would be preferred e.g. Docker.

## Requirements

Following are the packages used for creation of Graphical User Interface along with their purpose of use.

-Tkinter

For Graphical User Interface

-Opencv

For Kmeans segmentation

-Skimage

For other four types of segmentation

-GDAL

Working with raster data

-Numpy

For rasters to be processed as numpy arrays

-Time

For calculating time on each process.

## Literature Background:

### GDAL

GDAL, also known as GDAL/OGR, is a library of tools used for manipulating geospatial data. GDAL works on both [raster and vector](#) data types, and is an incredible useful tool to be familiar with when working with geospatial data. While the GDAL library can be used programmatically, GDAL also includes a CLI (Command Line Interface).

#### *A note about names*

Traditionally, the name "GDAL" was used to refer to the raster-related half of the library, while "OGR" referred to the vector part. You may commonly hear people use both "GDAL" and "GDAL/OGR" to refer to this library - but in all cases, the toolset being referenced is the same.

#### Commonly-used commands

- ogrinfo - Get information about a vector dataset
- gdalinfo - Get information about a raster dataset
- ogr2ogr - Convert vector data between file formats
- gdal\_translate - Convert raster data between file formats

Python provides different modules and libraries to make the projects interactive. One such attribute is the Graphical User Interface (GUI), which can be added to the projects using the Python libraries like Tkinter, PyQt5, Kivy and more. The Tkinter library is among the most common libraries for designing interactive GUI applications. It is lightweight and easy to use. We can operate this library on different platforms or Operating Systems like Windows, Linux, and macOS.

### TKINTER PROGRAMMING

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps:

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

## 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	<a href="#"><u>Button</u></a> The Button widget is used to display buttons in your application.
2	<a href="#"><u>Canvas</u></a> The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	<a href="#"><u>Checkbutton</u></a> The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	<a href="#"><u>Entry</u></a> The Entry widget is used to display a single-line text field for accepting values from a user.
5	<a href="#"><u>Frame</u></a> The Frame widget is used as a container widget to organize other widgets.
6	<a href="#"><u>Label</u></a> The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	<a href="#"><u>Listbox</u></a> The Listbox widget is used to provide a list of options to a user.
8	<a href="#"><u>Menubutton</u></a> The Menubutton widget is used to display menus in your application.
9	<a href="#"><u>Menu</u></a> The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	<a href="#"><u>Message</u></a> The Message widget is used to display multiline text fields for accepting values from a user.
11	<a href="#"><u>Radiobutton</u></a> 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	<a href="#"><u>Scale</u></a> The Scale widget is used to provide a slider widget.
13	<a href="#"><u>Scrollbar</u></a> The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	<a href="#"><u>Text</u></a> The Text widget is used to display text in multiple lines.
15	<a href="#"><u>Toplevel</u></a> The Toplevel widget is used to provide a separate window container.
16	<a href="#"><u>Spinbox</u></a> 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	<a href="#"><u>PanedWindow</u></a> A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	<a href="#"><u>LabelFrame</u></a> A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	<a href="#"><u>tkMessageBox</u></a> This module is used to display message boxes in your applications.

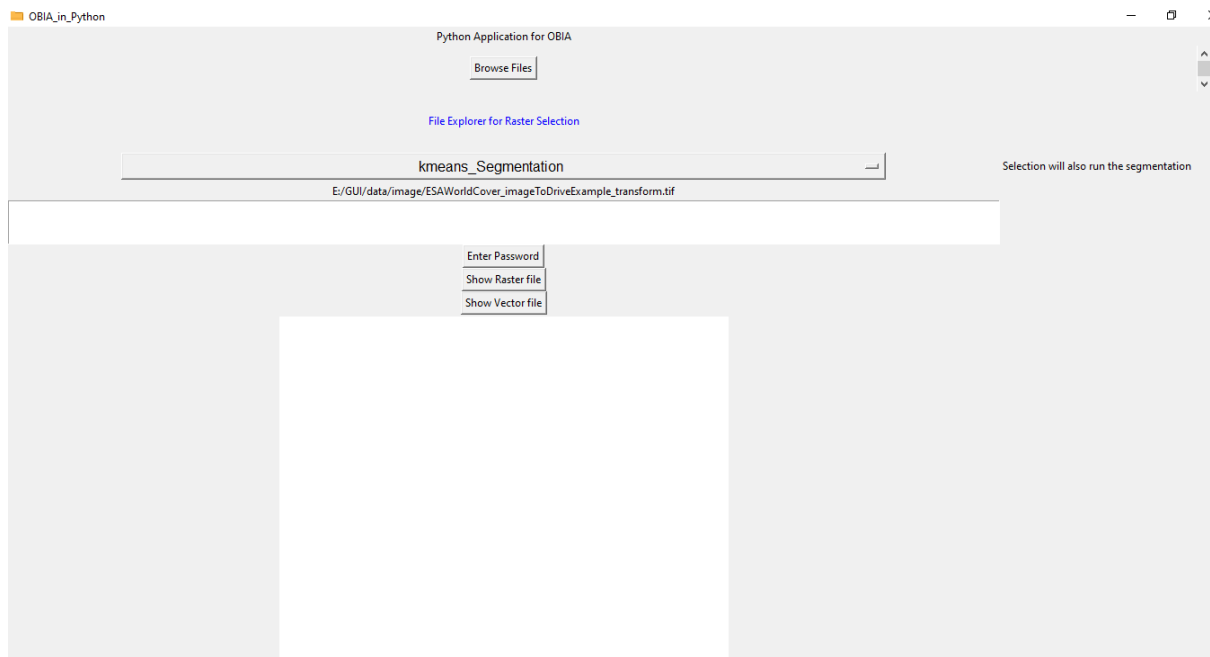
## Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

- The pack() Method – This geometry manager organizes widgets in blocks before placing them in the parent widget.
- The grid() Method – This geometry manager organizes widgets in a table-like structure in the parent widget.
- The place() Method – This geometry manager organizes widgets by placing them in a specific position in the parent widget.

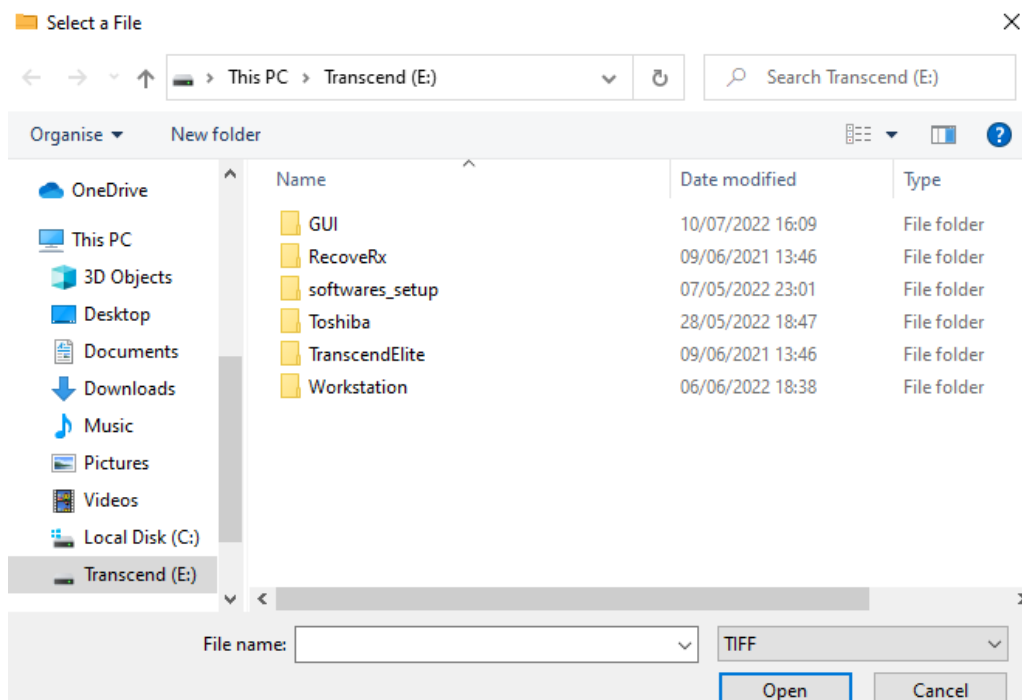
For the created GUI grid method was used for organizing the widget.

## Result:



## Graphical User Interface

Browse the image to be segmented through Browse File button

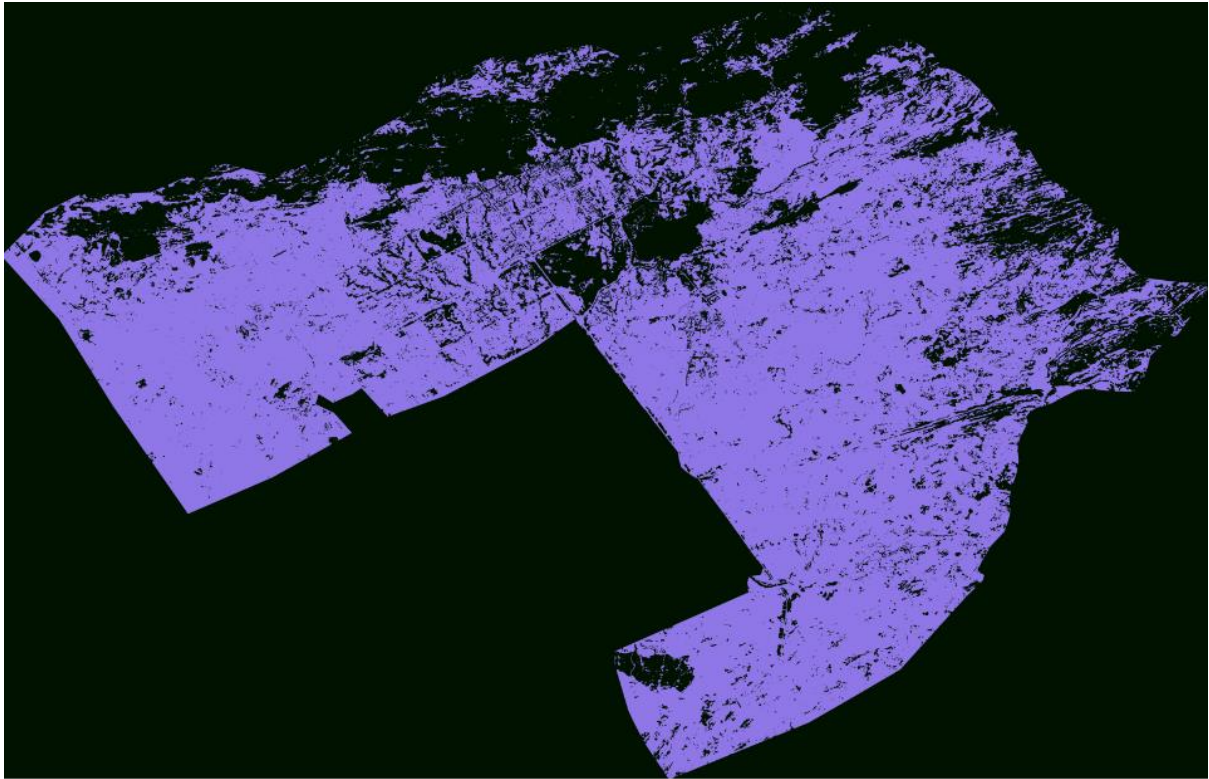


Click on the type of segmentation you want to run through option list, dropdown menu.

Currently, kmeans\_Segmentation was selected.

After the selection, output folder window appears where one can select through file explorer. When the folder is selected, segmentation run's at the backend automatically.

Kmeans Segmentation output (generated through the GUI)



The segmentation can be viewed through python third party software such as Photos or QGIS.