

**REPORT WRITING**

**NAME**- ANWESHA PRAMANIK

**ROLL NUMBER**- UG/02/BCA/2020/011

**ENROLLMENT NUMBER**-AU/2020/0004483

**NAME OF THE PROJECT**- IMAGE COLLAGER USING TKINTER

**PROJECT GUIDE**- MR. PABAK INDU

TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1. Introduction | 3 |
| 1. Objective | 3 |
| 1. Existing Processes  * Paste Method * Image Batch * Converting images to NumPy arrays * Using NumPy and OpenCV | 4  4  5  7  7 |
| 1. Proposed Method  * Algorithm * Flowchart | 11  11  13 |
| 1. Results  * Output | 15  15 |
| 1. Comparison | 17 |
| 1. References | 18 |
| 1. Acknowledgement | 19 |

1. **Introduction:**

For creating Graphical User Interface (GUI), Python provides a variety of frameworks, including PyQt5, Tkinter, Kivy, wxPython, Libavg, Pysimple, PyForms, and Wax. Tkinter is the most widely used GUI technique, and we employed it in the creation of our application. The Tk GUI toolkit bundled with Python has a common interface called Tkinter. The quickest and simplest method for developing GUI apps is using Python with Tkinter.

In our project, we have used different modules of Python Standard Library like the Filedialogue module, ImageTk module**,** shutil module, Tempfile module, and Python Imaging Library (PIL).

1. **Objective:**

Collages are an excellent way to tell a story with your photos and images. Photo collages can be used on a variety of platforms. Collages can be used to display a collection of photos, or even an entire mood, on platforms ranging from Facebook to Pinterest. It is an interesting way for viewers to look at photos, especially if they are themed. Whether you want to capture the impact of a trip, explain how to use a product, or capture the essence of your brand, a collage takes viewers on a journey that stays with them in a way that single images cannot.

So, in our project, we attempted to create an image collage application using Python, the most popular and easy-to-learn programming language.

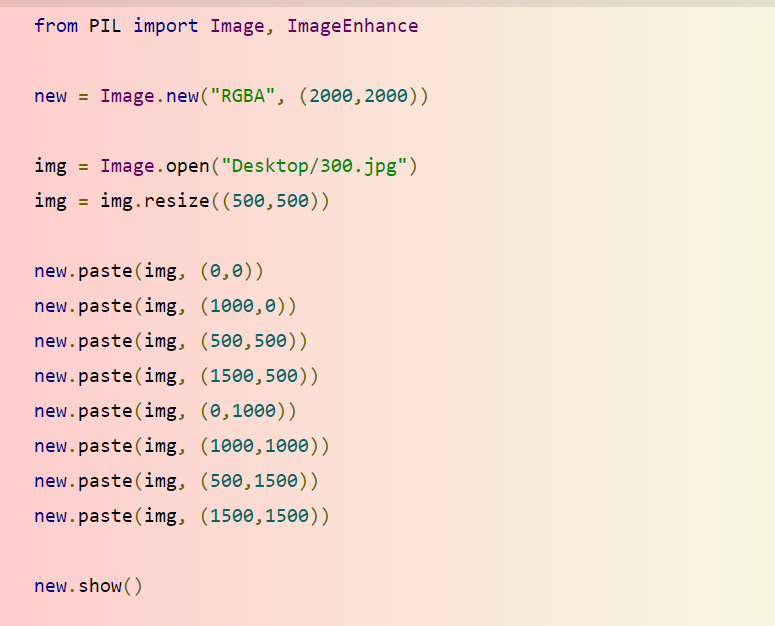
1. **Existing Processes:**

## Paste Method

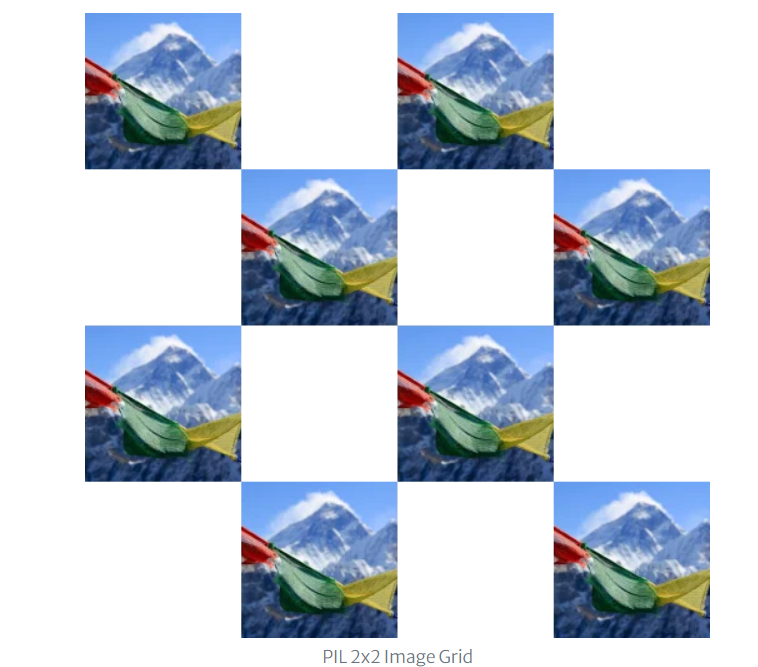
PIL includes a useful method called paste() that can be used to paste one image onto another.

In this code snippet, an image named new\_img is pasted to a new image named collage that we created using Image.new.

Code snippet:



Output:





Advantage:

1. This method is useful when we need to copy the image but also retain the original.

Disadvantage:

1. Time-consuming because we have to paste the images one by one, thus unsuitable for dealing with a large set of data.

**3.2. Image Batch**

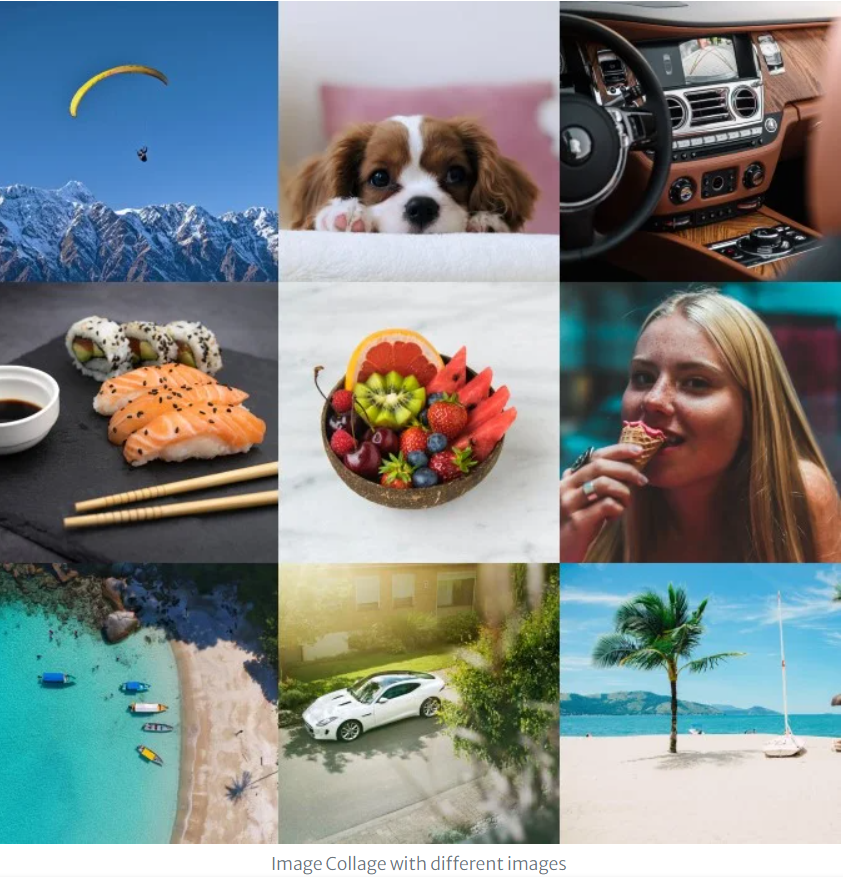
1. This method is used to iterate through various images to create a collage. This is easily accomplished by opening a new image and pasting it into the collage at each iteration.

2. In this example, we will concatenate a sequence of files from the Desktop folder. The listdir() method from the OS library can be used to automatically read image file names from within a folder.

Code snippet:



Output:

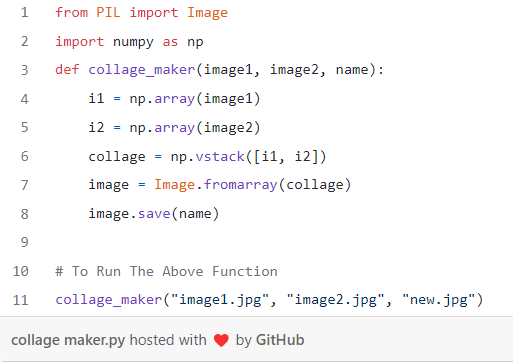


Advantage:

1. Provides code-reusability.
2. Using loops we do not need to write the same code again and again.

**3.3. Converting images to Numpy arrays**

To create a photo collage, we first need to read the images and convert them into arrays before merging them into a collage.

****

To run this code, we first have to input the name of the two images that we want to use to create a collage in the function, and then we have to enter the name we want to save the image as the third parameter. Once the code executes, it will save a collage of our images in the same directory where our Python file is.

Advantage:

1. Numpy arrays take less space. NumPy’s arrays are smaller in size than Python lists. It consumes less memory. It is fast compared to the python List. It is convenient to use.

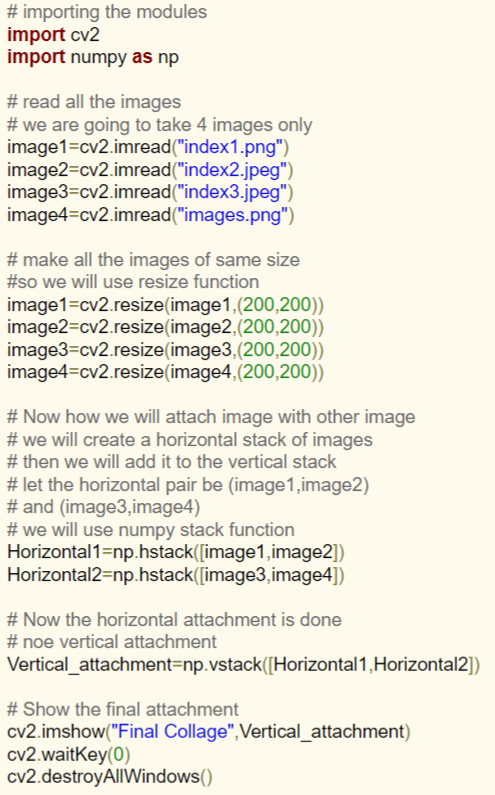
Disadvantage:

1. When working with a big data set, it is very hard to keep track of which column is which, because we need to remember which variable corresponds to which.

**3.4.** **Using NumPy and OpenCV**

**The images are joined with the help of NumPy functions** and OpenCV(cv2) is used for reading and getting the data of the images. We will create a horizontal stack of images then after that, we will join them in a vertical stack of images.

Code snippet:



Output:



Advantage:

1. First and foremost, OpenCV is available free of cost.
2. Since OpenCV library is written in C/C++ it is quite fast
3. Low RAM usage.
4. It is portable as OpenCV can run on any device than can run C.

Disadvantage:

1. Takes more time to display image compare to matplotlib.
2. **Proposed Method**
   1. **Algorithm:**

Step 1: Start

Step 3: Create the parent window and set the geometry.

Step 5: Create a frame inside the parent window and set the dimensions of the frame using grid() method.

Step 6: Create two buttons named “upload” and “create collage” inside the frame.Set their position using grid() method. They are invoked using function upload\_file() and photo\_collage() respectively.

Step 8: A label is created inside the parent window called ‘root’ above the frame.

Step 9: Declare a list named lst[] globally.

Step 10: Define a function upload\_file()

Step 11: Take input of files of any format(JPG,PNG or JPEG) from the user using askopenfilename() and store the files in the variable name filename.

Step 12: One copy of each input file will be stored in the temp folder locally.

*try:*

*tmp\_dir = tempfile.mkdtemp()*

*finally:*

*try:*

*shutil.rmtree(tmp\_dir)*

*new\_file = "img" + str(j)*

*ext = ".jpg"*

*new\_file = str(j)*

*new\_file = new\_file + ext*

*print(new\_file)*

*img.save(new\_file)*

*lst.append(new\_file)*

*j = j + 1*

*except OSError as exc:*

*if exc.errno != errno.ENOENT:*

*raise*

Step 13: These temp files will be resized and then finally displayed on the output screen using function PhotoImage.

Step 14: Set the position of the images on the output screen.

col = 1

row = 4

j = 0

if col == 3:

row = row + 50

col = 1

else:

col = col + 90

Step 15: Define another function photo\_collage to create the final collage image.

Step 16: Check if the number of inputs is greater than 1 and less than 10, to create the collage. If the number of inputs doesn’t satisfies any of the condition, throw an error message or else create the collage successfully and save at the desired location with the name “collage.png”.

for i in range(0, 1500, 500):

for j in range(0, 1500, 500):

if c < len(lst):

print(lst)

print(len(lst))

print(c)

file = lst[c]

print(file)

photo = Image.open(file)

photo = photo.resize((490, 675))

collage.paste(photo, (i, j))

c += 1

collage.show()

collage.save("C:/Users/Anwesha Pramanik/Documents/collage.png")

Step 17: End.

**4.2 Flowchart:**



**5. Results:**

1. The application is running locally.
2. The GUI application can accept the input from the user.
3. The input should be more than one and less than 10 to create the collage otherwise it will throw an error.
4. It can successfully create the collage of 1<n<10 inputs.
5. The collage is saved in the location

"C:/Users/Anwesha Pramanik/Documents/collage.png".

**5.1. Output:**

Fig 1.1, 1.2, 1.3, and 1.4 are the screenshots of the output screen of our application. It shows how our application is taking input and giving the output.

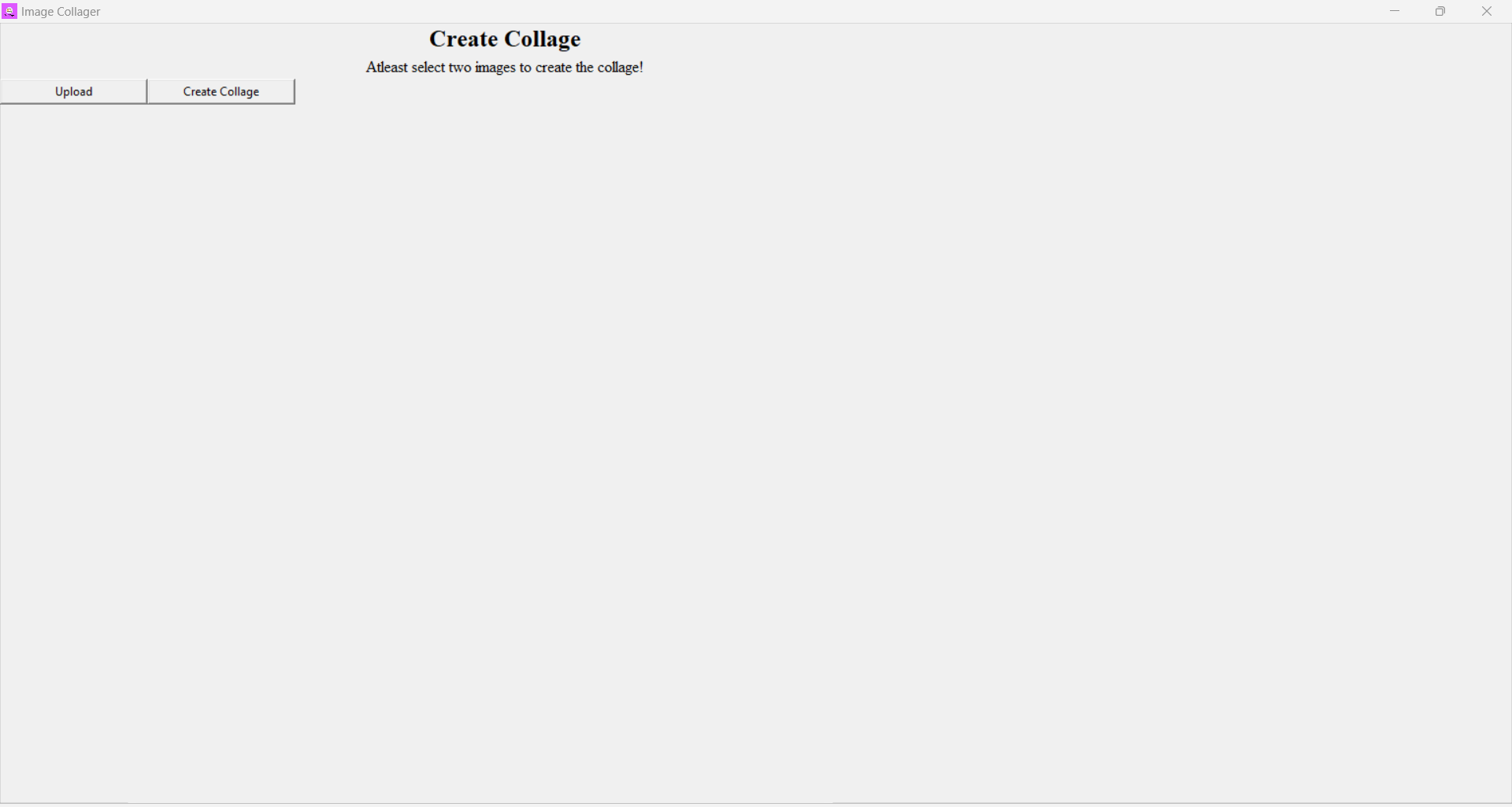


Fig 1.1: Parent Window.

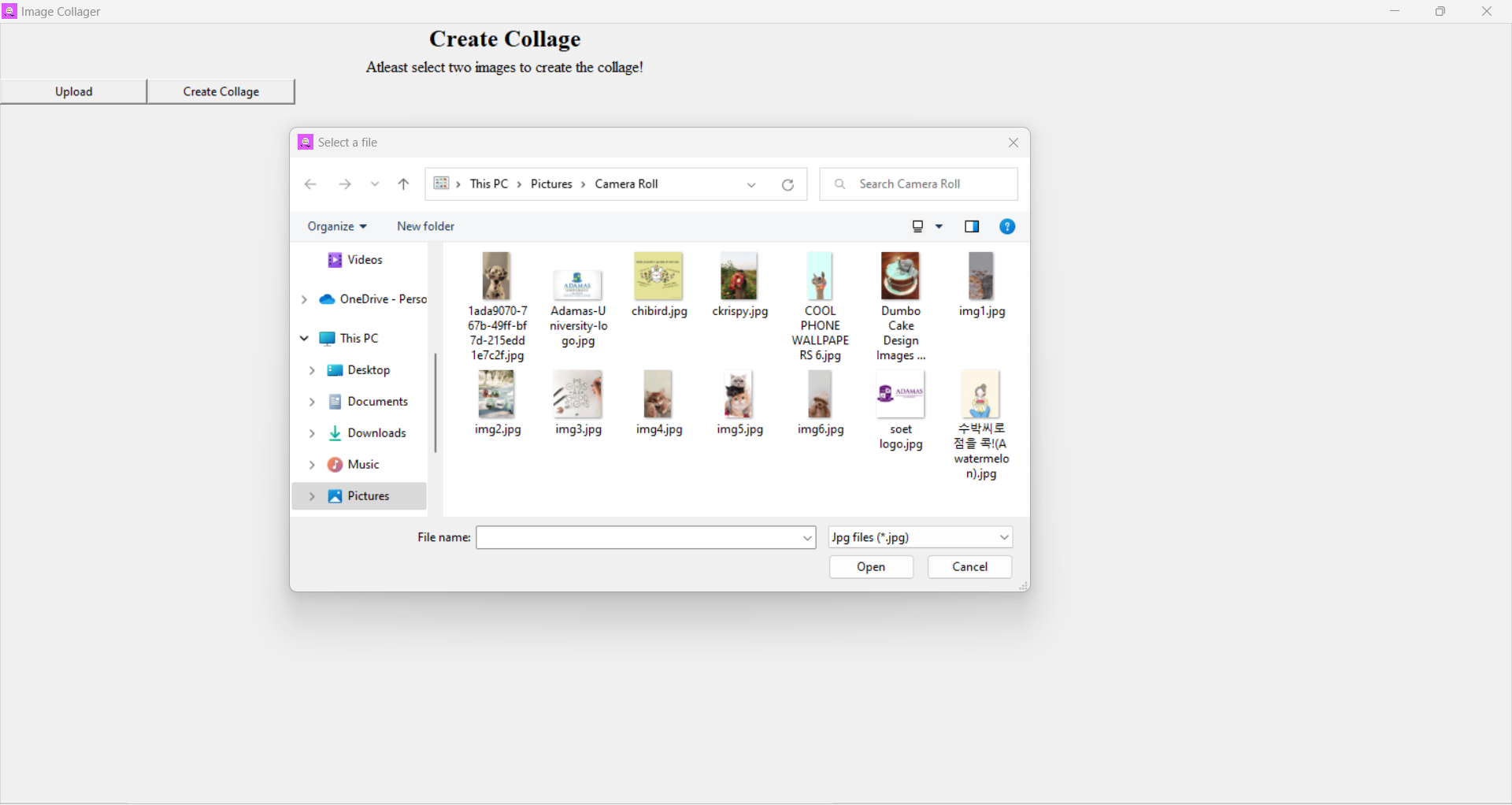


Fig 1.2: “Select a file” dialogue box is opened.

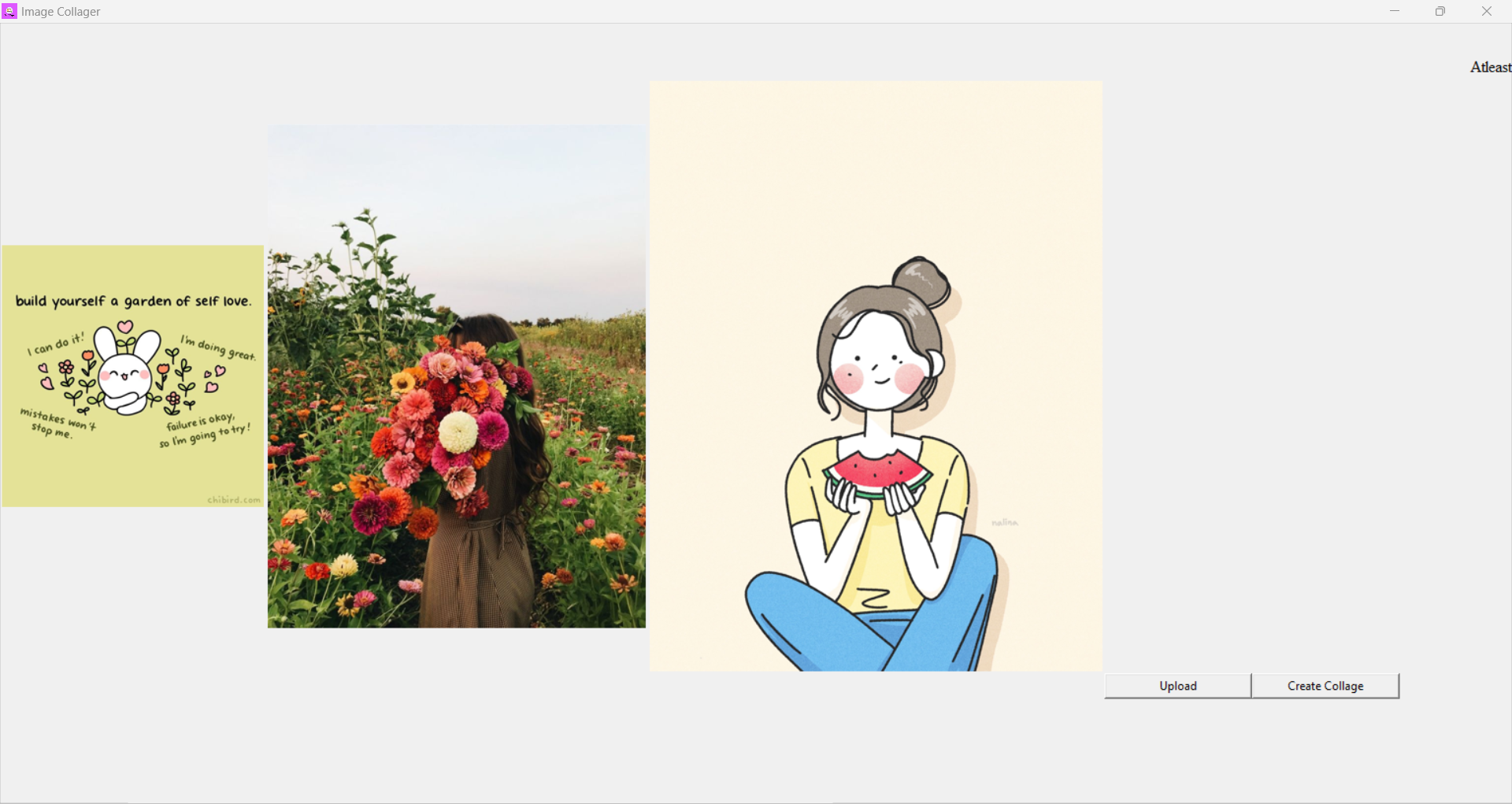


Fig 1.3: 3 images are given as input

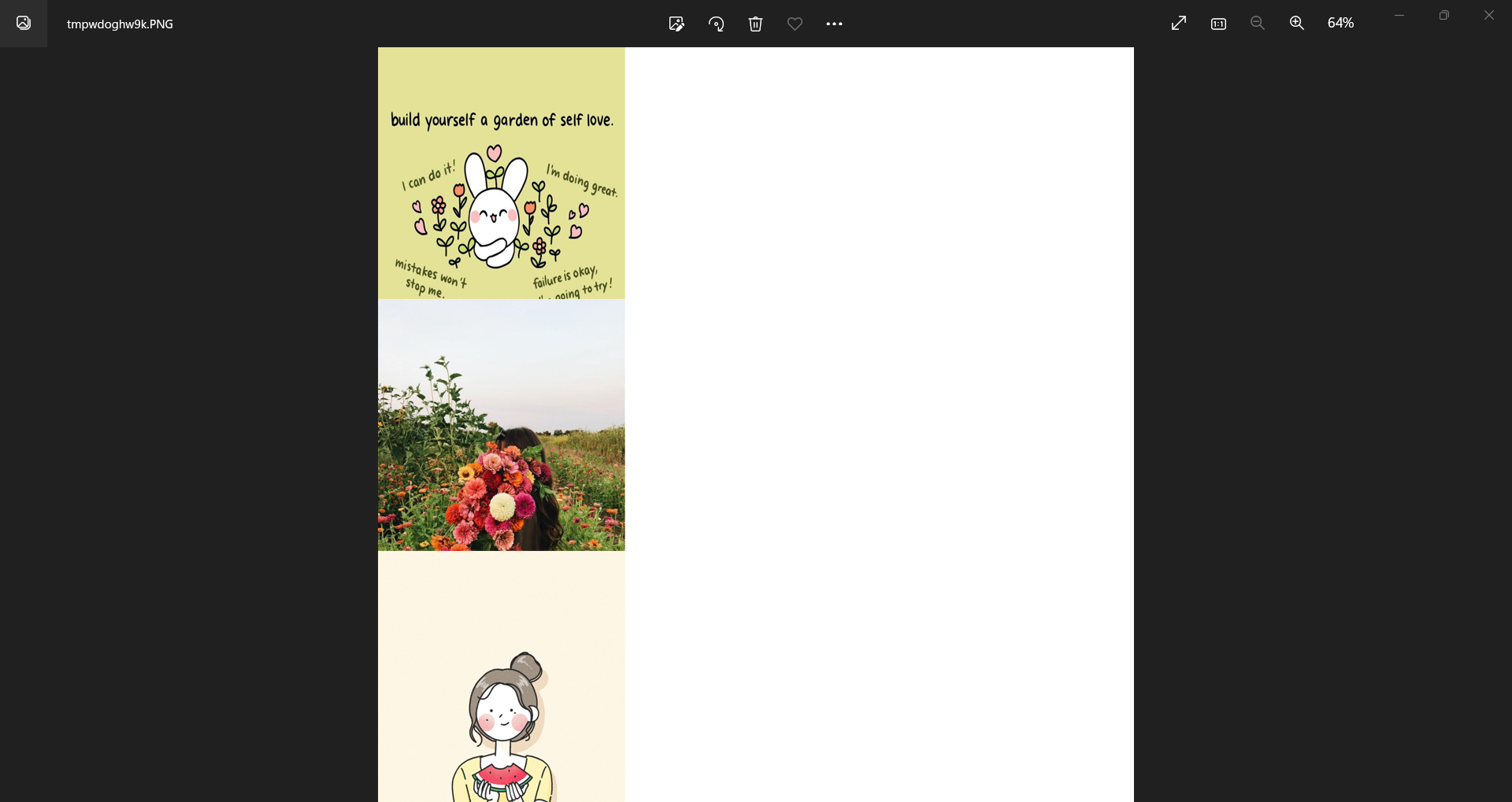


Fig 1.4: Collage is being created

**6. Comparison:**

1. If images of very large dimensions is selected by the user the labels are going out of the screen. The labels are not staying in their position after selecting more than two images so it is not possible for the user to see the two buttons. This problem cannot be solved by the grid method.
2. Collages are done column-wise and not horizontally.
3. Images are collaged after being resized so because of this, some images are wrapped and not looking good.
4. Many temporary files are being created.

**7. References:**

1. <https://www.includehelp.com/python/create-a-collage-of-images-with-the-help-of-numpy-and-python-opencv-cv2.aspx> (7:00 PM,19/09/22)
2. <https://thecleverprogrammer.com/2021/07/31/collage-maker-using-python/> (5:20 PM, 20/09/22)
3. <https://holypython.com/python-pil-tutorial/creating-photo-collages/> (8:34 AM, 19/09/22)
4. <https://www.geeksforgeeks.org/python-copy-and-paste-images-onto-other-image-using-pillow/> (6:48 PM, 30/09/22)
5. <https://www.tutorialspoint.com/python/python_gui_programming.htm> (4:55 PM, 13/10/22)
6. <https://www.geeksforgeeks.org/python-gui-tkinter/> (6:48 PM, 13/10/22)
7. <https://www.youtube.com/watch?v=NoTM8JciWaQ> (3:25 PM, 15/10/22)
8. <https://www.youtube.com/watch?v=yFoCL5b0GZo> (5:55 PM, 17/10/22)
9. <https://www.youtube.com/watch?v=ndUuy_55jho> (3:30 PM, 17/10/22)
10. <https://www.youtube.com/watch?v=ndUuy_55jho&t=1s> (4:16 PM, 16/10/22)
11. <https://stackoverflow.com/questions/8577137/how-can-i-create-a-tmp-file-in-python> (7:15 PM, 12/10/22)

**8. Acknowledgement:**

I'd like to thank everyone who made this project possible. First and foremost, I would like to thank Mr. Pabak Indu, who guided me through the project and provided invaluable assistance. He instructed me how to finish this project, and without him, it would not have been finished.

This project has provided an opportunity to learn and apply our theoretical knowledge in the real world. So I'd like to thank him for his assistance and direction on this project.

I'd also like to thank my parents for always being there for me.