



DIGITAL IMAGE PROCESSING

FINAL PROJECT

Submitted By

Qurat-ul-Ain i19-0685

Section

BS (CS)-B

Submitted To

Sir Akhtar Jamil

Date

14th May 2023

Report

Introduction:

There are three aims of the project:

- 1) Image Quilting: Generate a new image by stitching together small patches of existing images maintaining the texture. This process is called Image quilting.
- 2) Texture Synthesis: Texture synthesis refers to the process of generating new texture images or patterns from existing texture samples. It involves creating a larger or different-sized texture image that exhibits similar visual properties and statistical characteristics as the given source texture.
- 3) Texture Transfer: Provides a means to incorporate visually appealing and distinctive texture characteristics from one image to another, enabling creative expression and enhancing the visual quality of images in various applications such as digital art, photo editing, and computer graphics.

Implemented Main Features:

Patch selection: The algorithm selects small patches of an input image that are used to build the final output image.

Patch placement: The algorithm determines how to place the selected patches in the output image in order to create a seamless texture.

Overlapping patches: The patches are overlapped to reduce visible seams and create a more natural-looking texture.

Find Closest Patch: A feature to return a random patch amongst the closest patches to the given patch in the original image.

Find Closest Transfer Patch: A feature to return a random patch amongst the closest patches to the given patch in the original image.

Texture synthesis: The algorithm can generate new textures by randomly sampling patches from the input image and placing them in the output image.

Image stitching: The patches are seamlessly stitched together to create a larger image.

Texture matching: The algorithm finds patches that match the texture of nearby pixels in the output image, ensuring that the overall texture is consistent.

Texture Transfer: maps a particular quantity of the target image (e.g. the intensity, the luminance or blurred luminance) while also ensuring that the texture matches the source texture file.

Friendly GUI: Friendly GUI feature implementation refers to the ease and convenience of designing and developing graphical user interfaces (GUIs) using a framework or library. PyQt, a popular Python binding for the Qt framework, offers a wide range of features that make GUI development more user-friendly

Software Used:

Visual Studio Code: Visual Studio Code provides a lightweight, versatile, and feature-rich environment for Python development. Its strong Python support, integrated terminal, Git integration, extensive extension ecosystem, and customization options make it a popular choice for developers working on Python projects.

Jupyter Notebook: Jupyter Notebook offers an interactive and flexible environment for Python projects, allowing for exploratory coding, rich media support, narrative text, and easy code sharing. Its extensibility, integration capabilities, and integration make it a popular choice for data scientists, researchers, and developers working on data analysis, machine learning etc.

PyQt Designer: PyQt Designer is a visual design tool that simplifies the creation of GUIs for Python projects using PyQt. With its drag-and-drop interface, real-time preview, and signal-slot connections, it enables developers to design visually appealing and interactive UIs. The integration with PyQt and the ability to generate UI files make it easy to combine the design with the application logic.

Deployment:

Web application deployment is intended to be used, it can be deployed on a web server. The application can be built using a web framework like Flask or Django, and the user interface can be created using HTML, CSS, and JavaScript. The application can be hosted on a cloud platform or a web hosting service. Or desktop application with frontend built in PyQt which is a Python binding for Qt, which is a set of C++ libraries and development tools providing platform-independent abstractions for graphical user interfaces (GUIs).

Friendly Graphical User Interface:

Below is attached the UI of my desktop application which is made using PyQt. PyQt provides a robust and feature-rich toolkit for developing GUI applications in Python. Its cross-platform compatibility, extensive documentation, and integration with other libraries make it a popular choice for creating modern and interactive user interfaces.

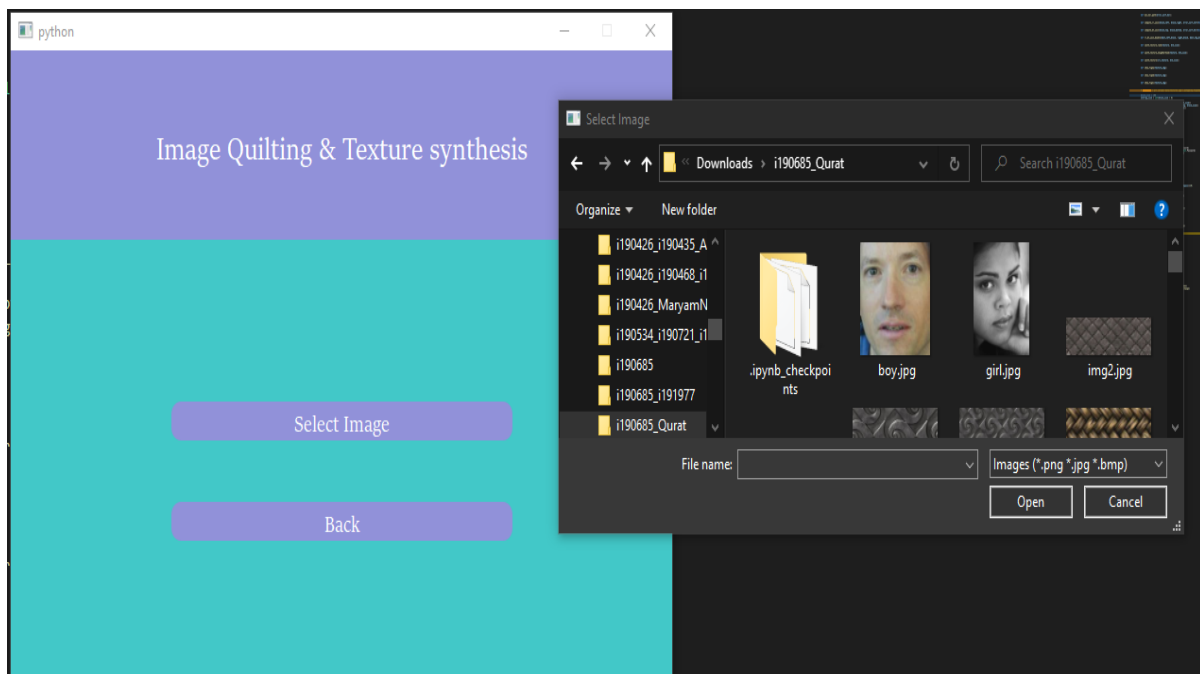
This is the main page of my Image Quilting, Texture Synthesis and Texture Transfer application. This is the Main Menu Page which gives my main features options. From here, user can select any option of own choice and then proceed to the next page.



On the next pager, user needs to input image from any folder in the laptop. Then the image passed to the selected feature on which further processing is being done according to the feature requirements. Also the selected image is displayed in a pop-up window so that user can see and make sure the given image is up to the mark. There is also a back button from where user can be reverted to the Main Menu.



This is the screenshot attached showing how user can select any picture from any folder.



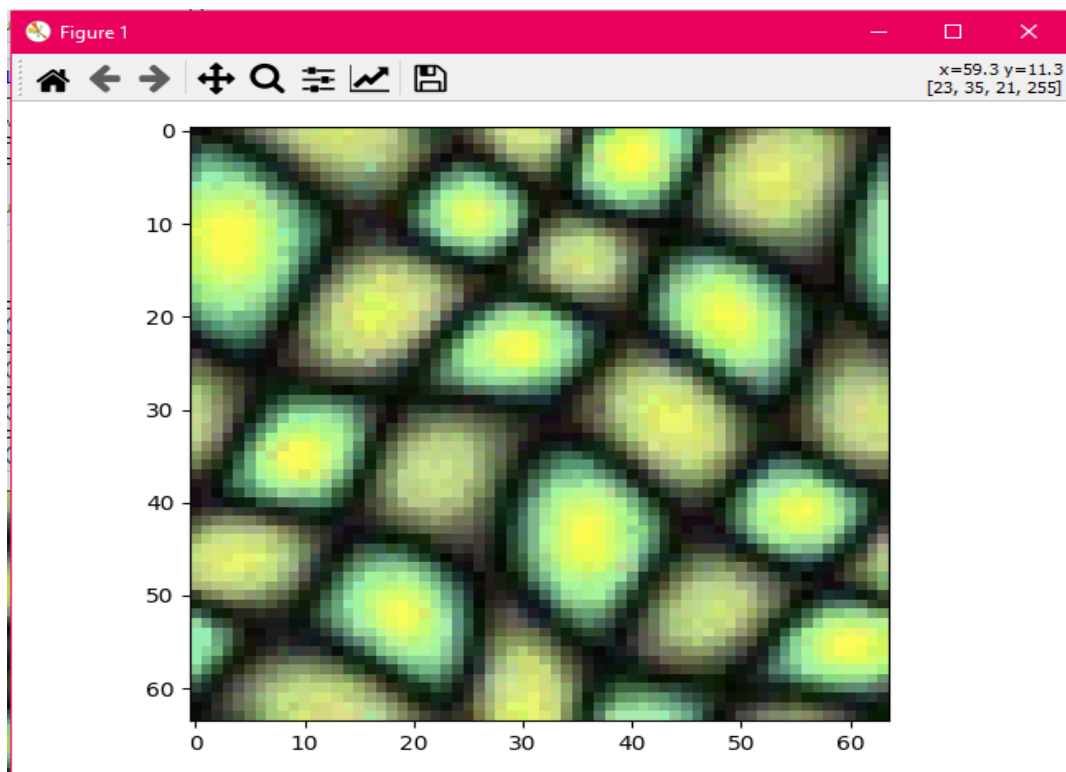
These screenshots are shown for the first feature which is Texture Synthesis. Same is the working for the rest two features that are Image Quilting and Texture Transfer.

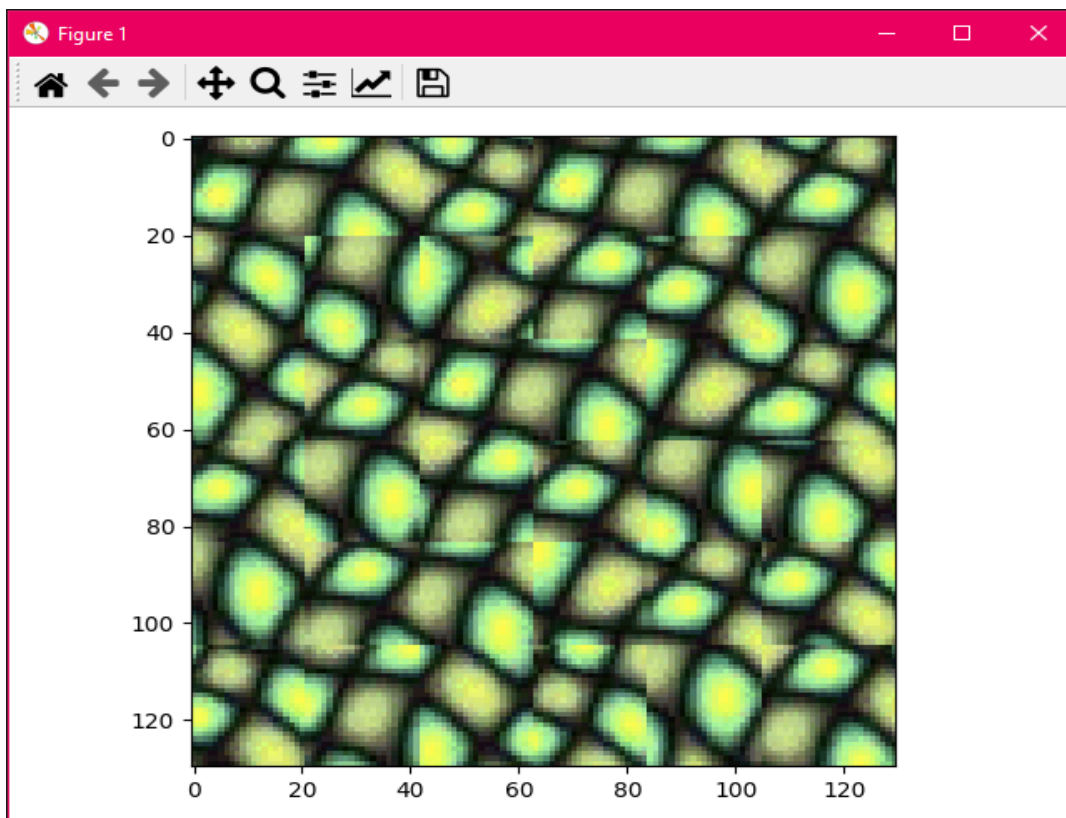
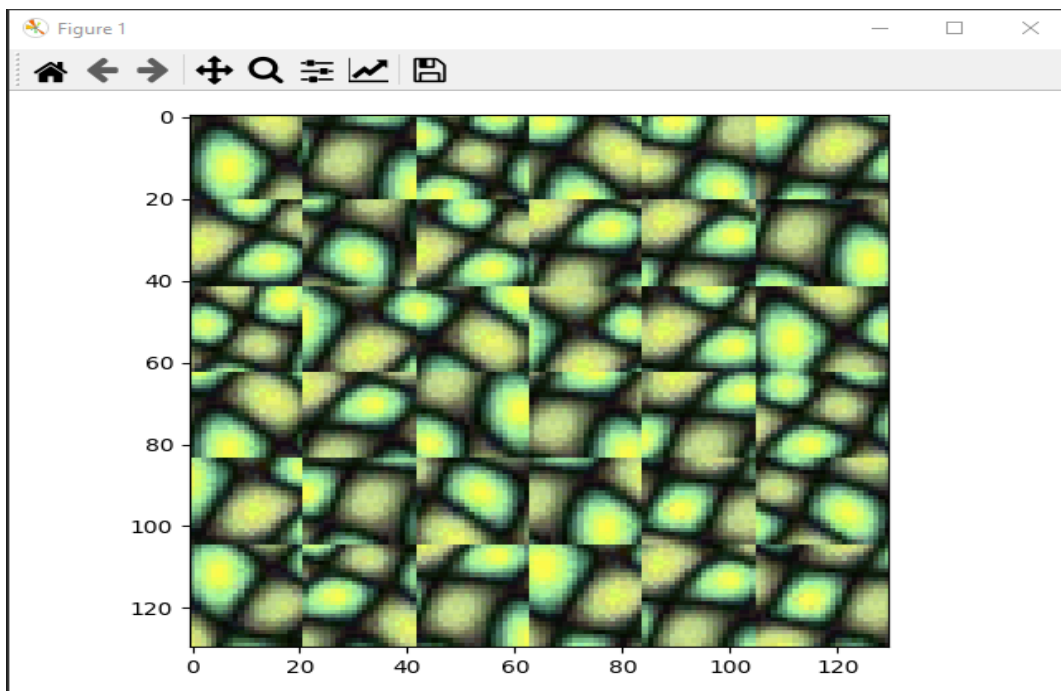
Image Quilting:

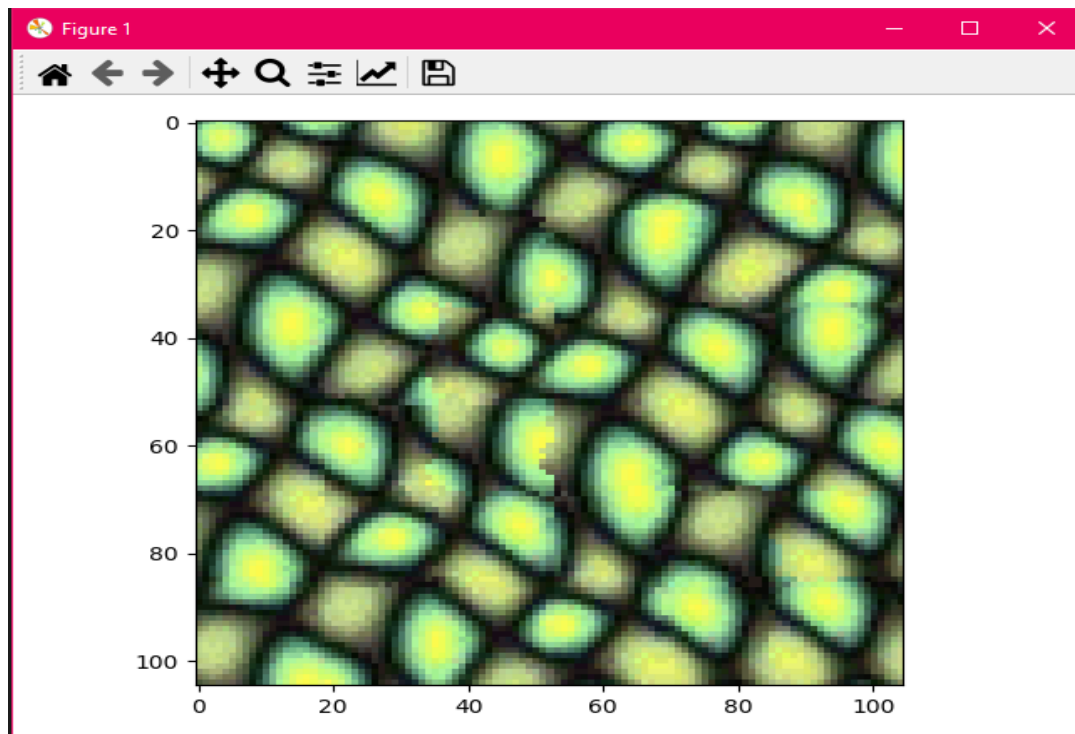
Image quilting is a texture synthesis technique that involves stitching together small overlapping patches from a given input texture to create a larger output image. It aims to generate a visually coherent and seamless texture by reusing and blending these patches in a way that avoids visible seams or repetitions.

Image quilting algorithms are effective in generating textures that exhibit local variations while maintaining overall consistency. They are widely used in computer graphics, image editing, and computer vision applications. Image quilting can be used to create realistic textures for virtual environments, generate image mosaics or collages, remove unwanted objects from images, and more.

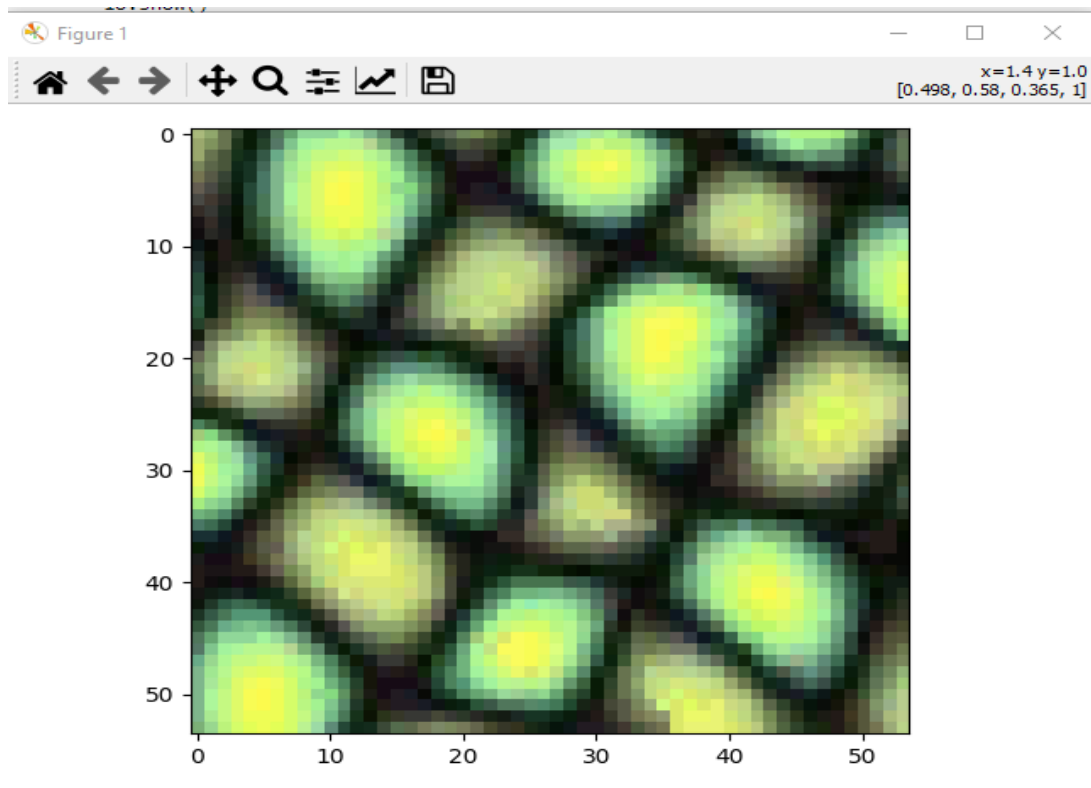
Below are the output attached as well aster every iteration:







This is the final output attached after 4 iterations.



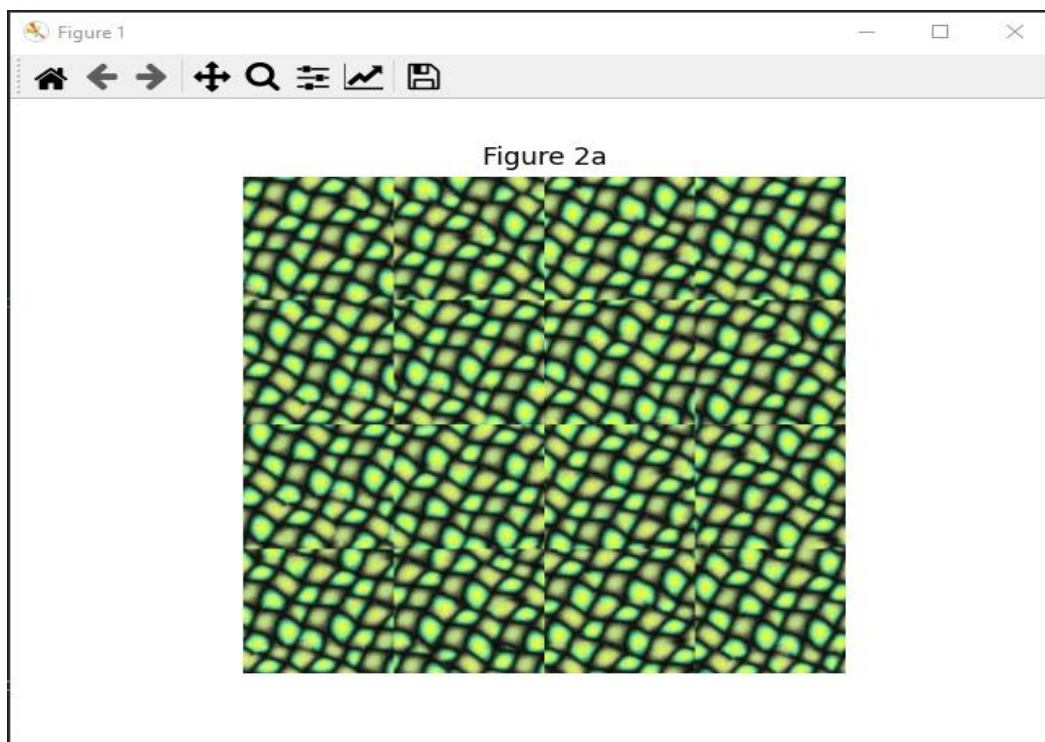
Texture Synthesis:

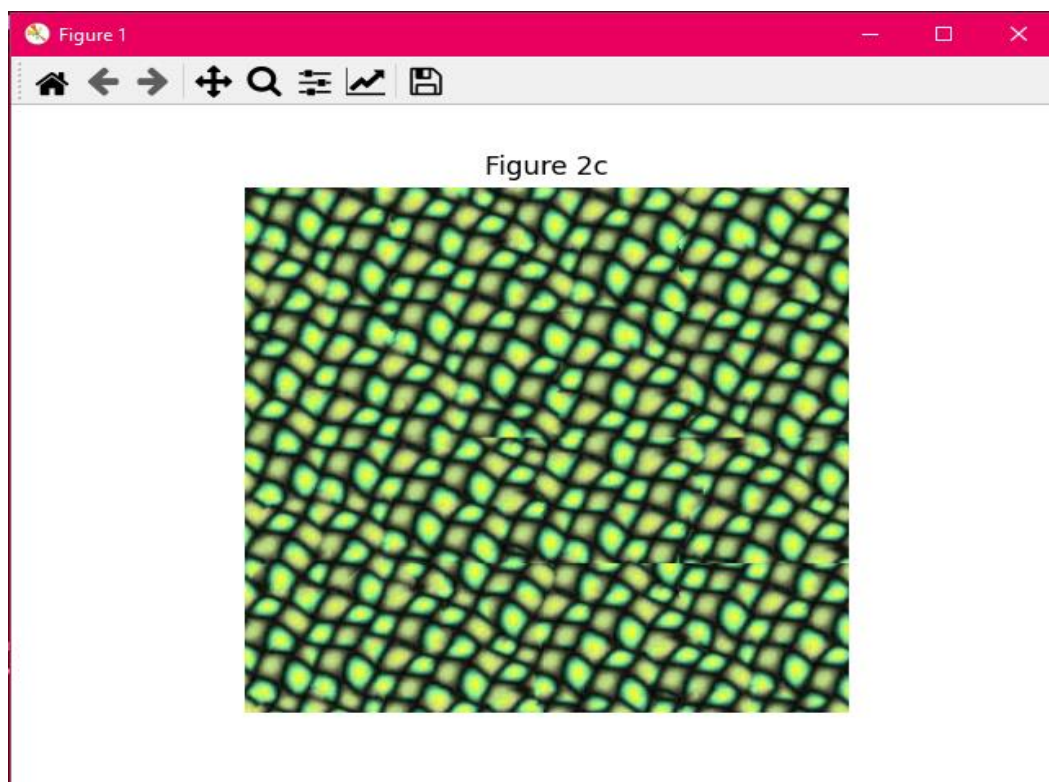
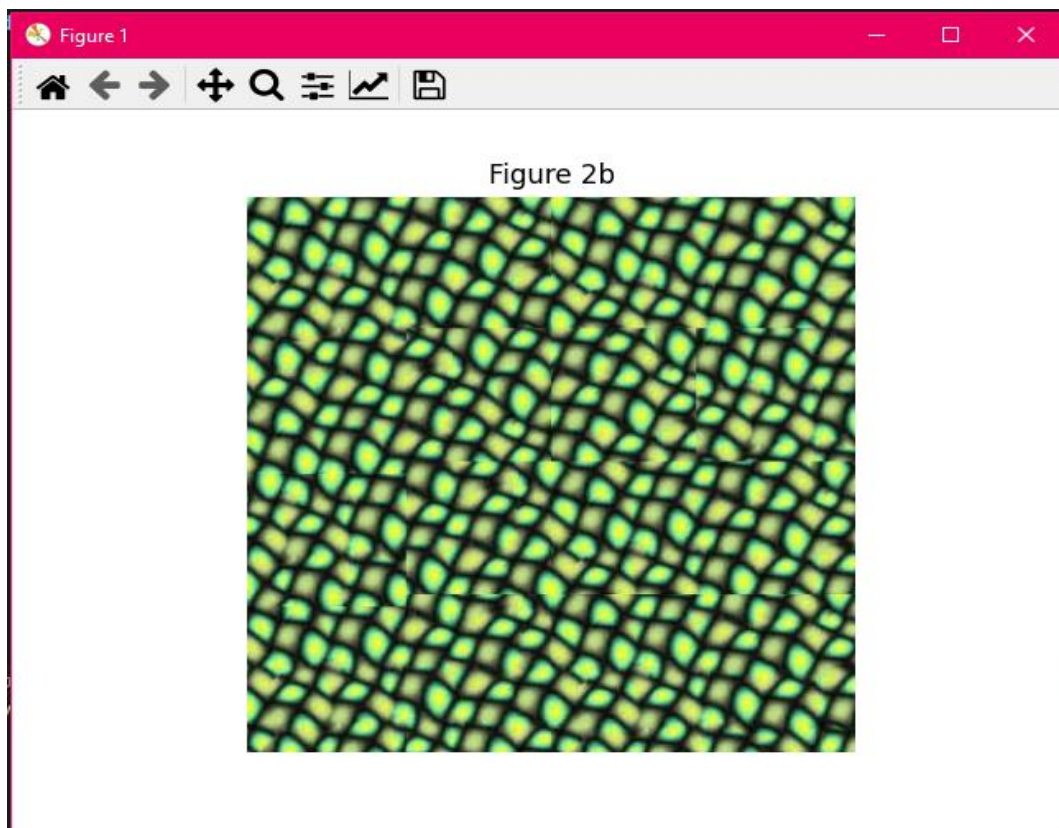
The goal of texture synthesis is to generate high-quality textures that are visually consistent with the input samples while avoiding obvious repetitions or artifacts. This technique is widely used in computer graphics, computer vision, and digital art to create realistic textures for 3D models, generate image backgrounds, or enhance visual effects.

Texture synthesis algorithms can be categorized into deterministic and stochastic methods. Deterministic methods aim to replicate the input texture exactly, while stochastic methods introduce randomness to create variations in the synthesized texture.

Texture synthesis techniques have a wide range of applications, including computer-generated imagery (CGI), virtual reality, video games, and digital design. They enable the generation of visually appealing and diverse textures that can be used to enhance the realism and aesthetics of virtual environments, graphics rendering, and various digital content.

Below are the outputs attached as well:





Texture Transfer:

Texture transfer refers to the process of transferring the visual characteristics or "texture" from one image to another. It involves extracting the texture information from a source image and applying it to a target image while preserving the overall structure and content of the target image.

Below are two images taken for texture transfer and the result is shown after transferring both images with each other.

Input Images:



Output Image:

