Report

Machine Vision

Remove & add Background

Hadeer Elkady	42010436
Mustafa Mahmoud	42010544

Supervised Eng/

Noura Tarek

Introduction

The code presents a Graphical User Interface (GUI) application developed in Python using the Tkinter library. Tkinter facilitates the creation of desktop applications with a simple and intuitive interface. This particular application allows users to manipulate images by removing their backgrounds, adding new backgrounds, and saving the processed images.

Code Structure

The code is organized into distinct sections:

Import Statements:

The initial part of the code includes import statements, bringing in essential libraries and modules required for image processing, GUI development, and file handling.

Function Definitions:

Following the import statements are the definitions of various functions:

upload_image(): Prompts the user to select an image file from their device.

upload_background(): Similar to "upload_image()" , but for selecting background images.

process_images(): Processes the uploaded image(s) by removing the background and optionally adding a new background.

save_image(): Allows the user to specify a location to save the processed image.

GUI Layout:

The GUI layout is constructed using Tkinter widgets. It comprises buttons for uploading images and backgrounds, a check button to toggle background addition, buttons for image processing and saving, and a label to display the processed image.

Functionality:

Image and Background Upload:

Users can select image and background files from their system using the respective upload buttons. Tkinter's “filedialog” module facilitates this interaction.

Image Processing:

Upon selecting an image, the application processes it by removing the background using the “rembg” library. If the user opts to add a background, the processed image is composited with the selected background. The resulting image is then displayed in the GUI.

Image Saving:

Users have the option to save the processed image to their device. The “save_image()” function prompts them to specify a file path and format (defaulting to PNG) for saving the image.

User Interaction:

The GUI offers intuitive interaction for users:

They can upload images and backgrounds, toggle background addition, process images, and save the processed images using the corresponding button.

Error messages are displayed if users attempt to process images without uploading them or if they try to add a background without selecting one.

Conclusion

In summary, the provided code delivers a straightforward yet effective image manipulation application. It leverages Tkinter's capabilities to create a user-friendly interface for image processing tasks. Future enhancements could focus on refining the user experience, incorporating additional image editing features, and optimizing performance for larger datasets.