Zocket

Zocket-Product-Photography-with-generative-A.I

Process and code flow

Please follow the below steps to use the product and code for access token and further steps:

Step 1 : Install the necessary python packages from the requirements.txt from the github na d clone repo to local for scripts to run in local

Step 2 : Please create accounts for stability ai if you don't have access to the gpu uwe will be using open source free demo access token from stability ai for loading and inference the background images else login to hugging face to access the token for access stable diffusion model locally to download weights and inference

```
# Sign up for an account at the following link to get an API Key.
# https://platform.stability.ai/

# Click on the following link once you have created an account to be taken to your API Key.
# https://platform.stability.ai/account/keys

# Paste your API Key below after running this cell.

os.environ['STABILITY_KEY'] = getpass.getpass('Enter your API Key')
```

Get Huggingface token

- Visit https://huggingface.co/ and register for an account. You'll need it to get the weights.
- Accept the license terms for https://huggingface.co/CompVis/stable-diffusion-v1-4
- Go to https://huggingface.co/settings/profile and create a new access token for READ. I called it stable diffusion
- Save the token to a file named token.txt in the same folder as this notebook (DO NOT CHECK THIS FILE INTO GITHUB)

Code Flow and working :

The code you've provided for a background change generator using the FastSAM model and the Stable Diffusion method. Let's break down the different components of the code and explain its functionality:

1. Import Statements:
    - The script starts with a series of import statements to bring in various Python libraries and modules, including image processing libraries (PIL, OpenCV), machine learning libraries (PyTorch), and other utilities.
2. Class Definition: BackgroundChangeGenerator
    - This class appears to encapsulate the functionality for generating background changes. It contains several methods for different tasks related to this process.
3. Overlay Mask:
    - overlay_mask is a method that takes an original image and a mask image, and it overlays the mask on the original image with a specified opacity.
4. Get Masks:
    - get_masks method processes annotations to obtain masks and create a unified mask image.
5. Get Image:
    - get_image method seems to extract and process an image from annotations.
6. Generate Background Change:
    - The class contains two methods for generating background changes, one using an API (generate_background_change_api) and another using a custom pipeline (generate_background_change_custom). These methods take a prompt, an image, a mask, and some other parameters to produce a background-changed image.
7. File Handling and User Interaction:
    - The class also has methods to handle file selection and drawing rectangles on images.
8. Concatenating Images:
    - get_concat_h_multi_resize method takes a list of images and concatenates them horizontally while resizing them to have the same height.
9. Mask the Region:
    - mask_the_region method seems to be used for interactively selecting a region in an image, processing it with a model, and displaying the results.

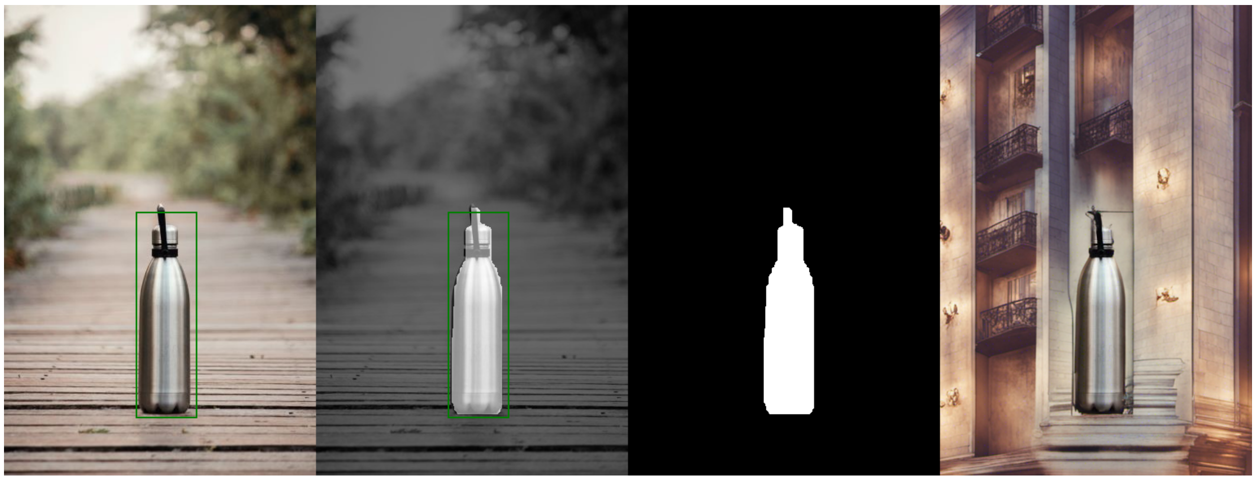
In the main part of the script:

- An instance of the BackgroundChangeGenerator class is created.
- The FastSAM model is loaded.
- An image, annotated image, and mask are obtained by selecting a region in the input image.
- The user is prompted to enter a prompt for background change.
- The script generates a background change using the generate_background_change_api method and displays the result.

The script also includes an option for using the generate_background_change_custom method, which is a GPU-intensive operation and requires certain GPU capabilities.

Results of the prompt is below:

prompt : Change background to hotel



In summary, this script provides an interactive way to select a region in an image, apply a background change using either an API or a custom method, and display the resulting image. It combines several libraries and models for this purpose. To use the script, you would need the required models and API access tokens. Additionally, you might need to install the necessary Python packages. The script is designed for background change tasks and can be useful in creative applications or image editing workflows.

Sample command line to run in terminal:

Coomand_line : python Product_Photography.py --custom --path "C:\Users\Rakshith\Downloads\pexels-gabriel-peter-1188649.jpg"

Testing of the Code flow in tkinter :

The code for a graphical user interface (GUI) application built using the tkinter library. The application seems to be focused on product photography, and it integrates with various other libraries and modules, such as FastSAM, stability AI, and customtkinter for UI styling. I'll explain the code in steps:

1. Importing Libraries and Modules:
   - The script begins by importing several libraries and modules, including tkinter, customtkinter, screeninfo, PIL (Pillow), ultralytics, and others. These libraries are used for creating the GUI, handling image processing, and running AI models.

2. Setting the Appearance Mode:
   - `customtkinter` is used to set the appearance mode and color theme for the GUI. This allows you to customize the look and feel of the application.

3. Creating the App Class:
   - The `App` class is defined, which inherits from `customtkinter.CTk`. This class represents the main application window.

4. Initializing the GUI:
   - The `__init__` method of the `App` class initializes the GUI. It configures the window size and layout using a grid, sets the window title, and creates a sidebar with buttons for various actions.

5. Defining Widgets:
   - Various widgets are defined, such as labels, buttons, entry fields, and canvas elements. These widgets will be used for different purposes within the application.

6. Event Handling Functions:
   - The `print_prompt`, `save_image_to_downloads`, and other functions are defined to handle events triggered by user interactions with the GUI. These functions perform actions like generating AI-based background changes, saving images, and more.

7. Loading Images:
   - The `load_image` function allows users to load an image file into the application. The selected image is displayed in one of the canvas elements.

8. Drawing Boxes:
   - The application allows users to draw boxes on the loaded image using the mouse. The `start_box`, `draw_box`, and `end_box` functions are responsible for enabling this functionality. These boxes can be used for annotations.

9. Clearing Boxes:
   - The `clear_boxes` function clears any drawn boxes on the canvas. A confirmation dialog is displayed to confirm this action.

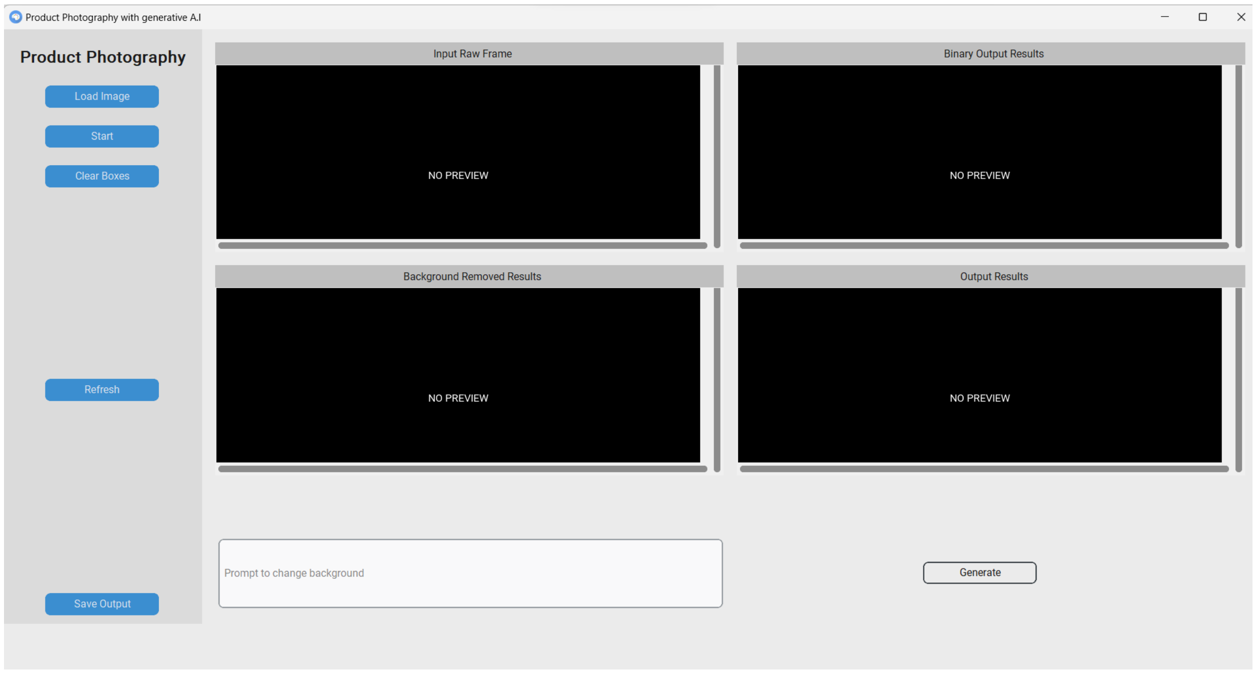
10. Refreshing the Application:
   - The `refresh_all` function resets the application to its initial state. This includes clearing boxes and reloading the default image.

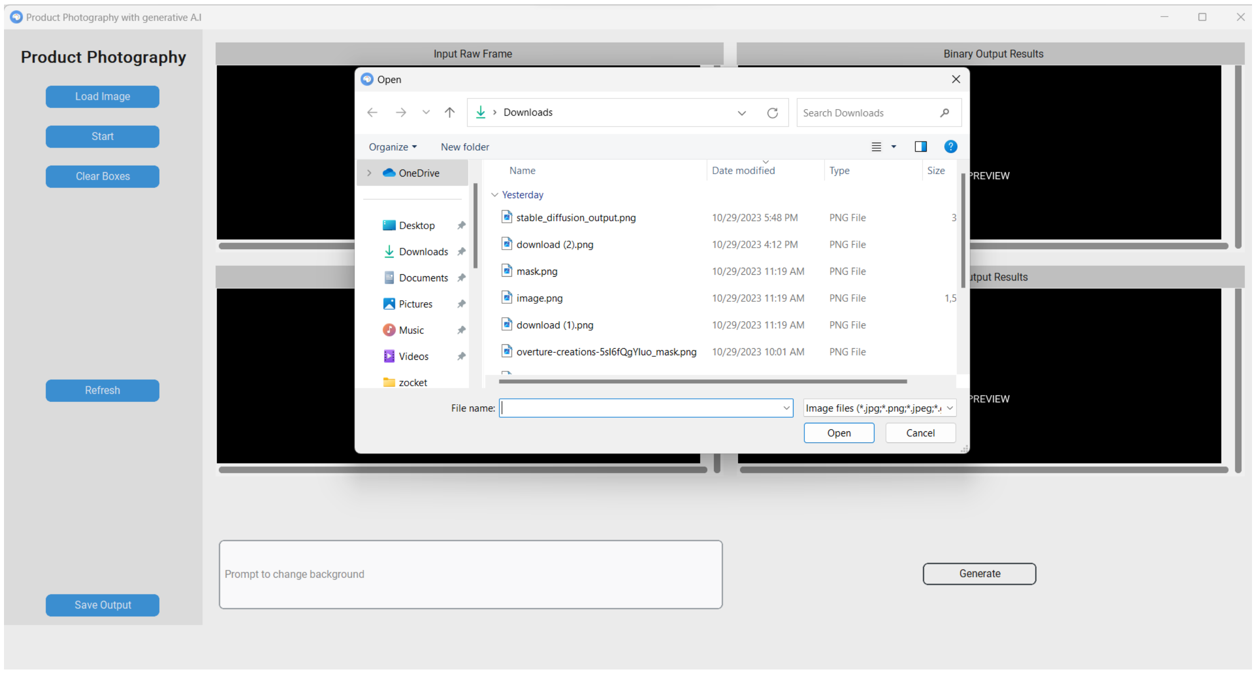
11. Saving Annotations:
   - The `save_annotation` function is used to save annotations made by drawing boxes on the image. It interacts with the FastSAM model to generate annotations and displays the results on different canvas elements.

12. Main Application Loop:
   - The `if __name__ == "__main__"` block creates an instance of the `App` class, sets an application icon, and initializes the FastSAM model. It then enters the main GUI event loop using `app.mainloop()`.

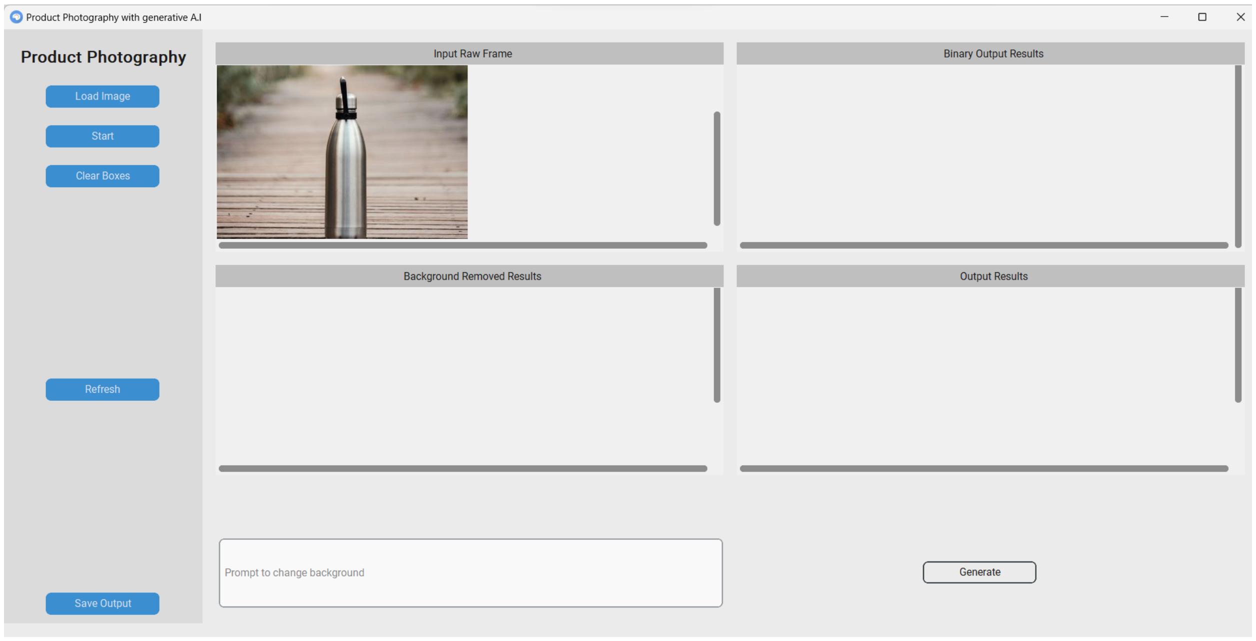
Overall, this code represents a complex GUI application for product photography. It allows users to load images, annotate them by drawing boxes, use an AI model (FastSAM) to generate annotations and change backgrounds, and save the results. The application has various UI components and is designed for an interactive and user-friendly experience in the field of product photography.

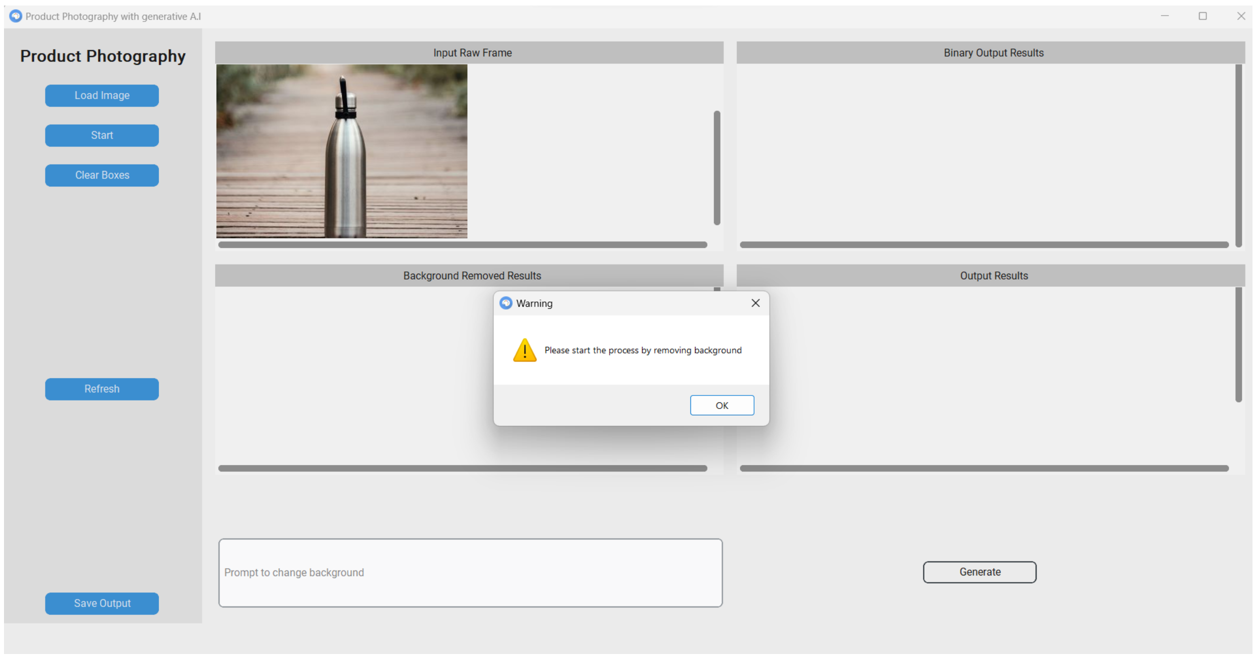
Screen shots to the GUI:



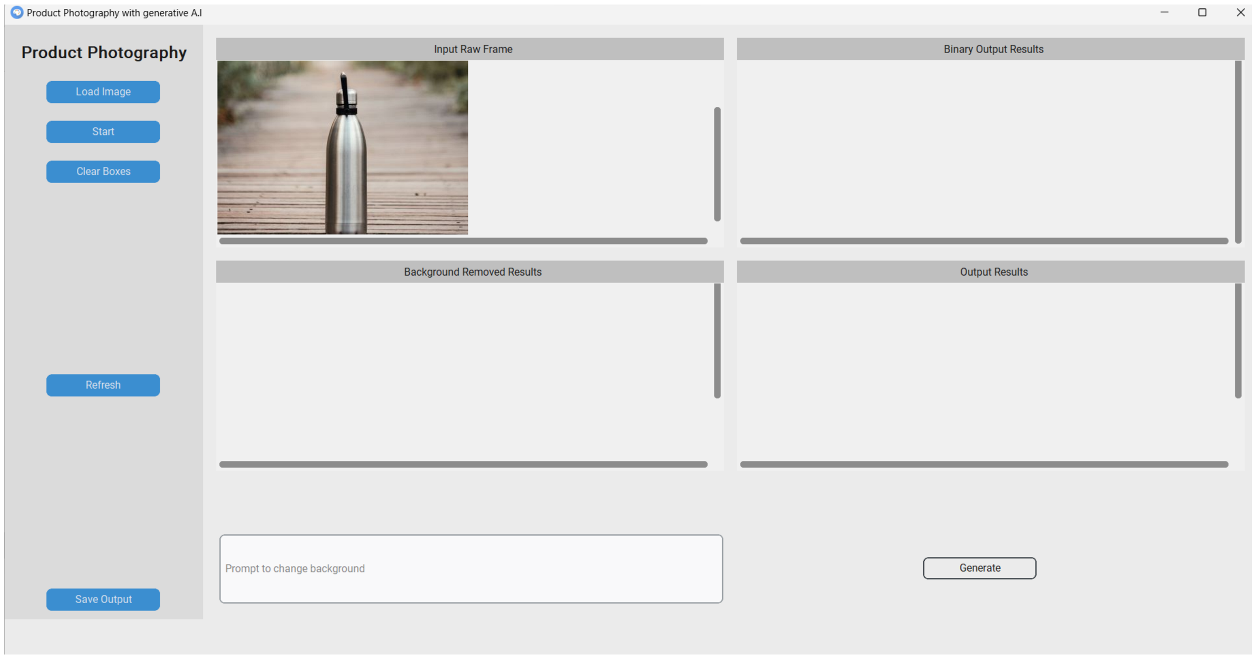


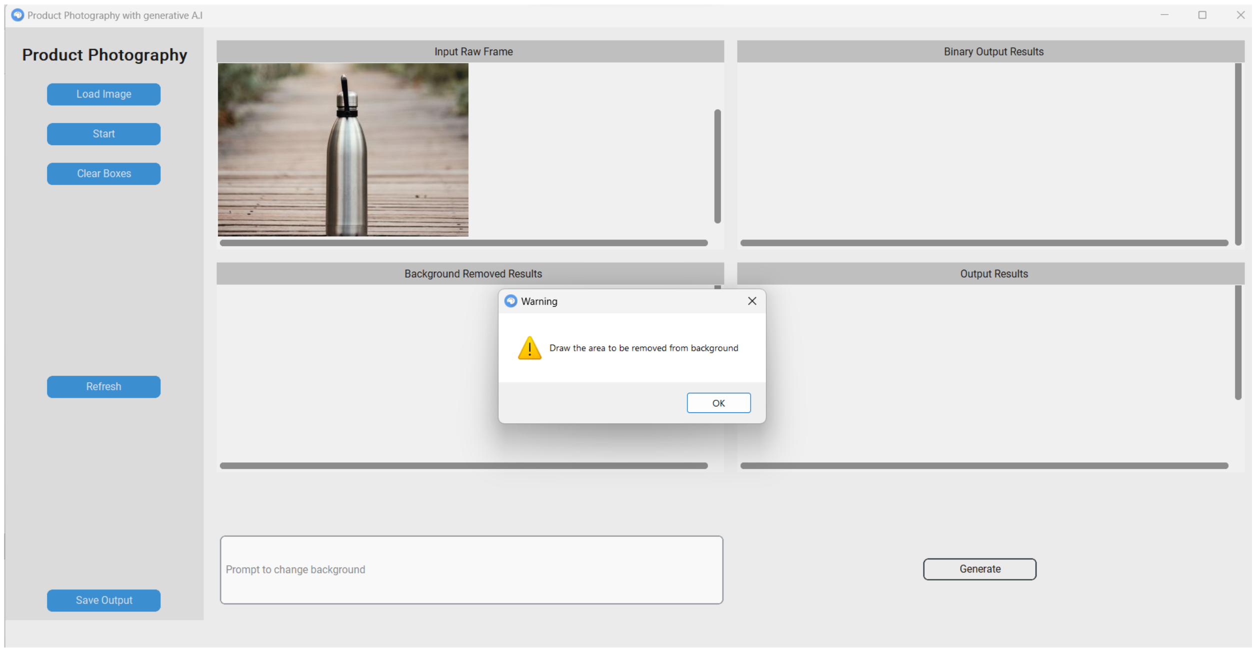
Name	Date modified	Type	Size
Yesterday			
stable_diffusion_output.png	10/29/2023 5:48 PM	PNG File	3
download (2).png	10/29/2023 4:12 PM	PNG File	
mask.png	10/29/2023 11:19 AM	PNG File	
image.png	10/29/2023 11:19 AM	PNG File	1,5
download (1).png	10/29/2023 11:19 AM	PNG File	
overture-creations-5sI6fQgYIuo_mask.png	10/29/2023 10:01 AM	PNG File	



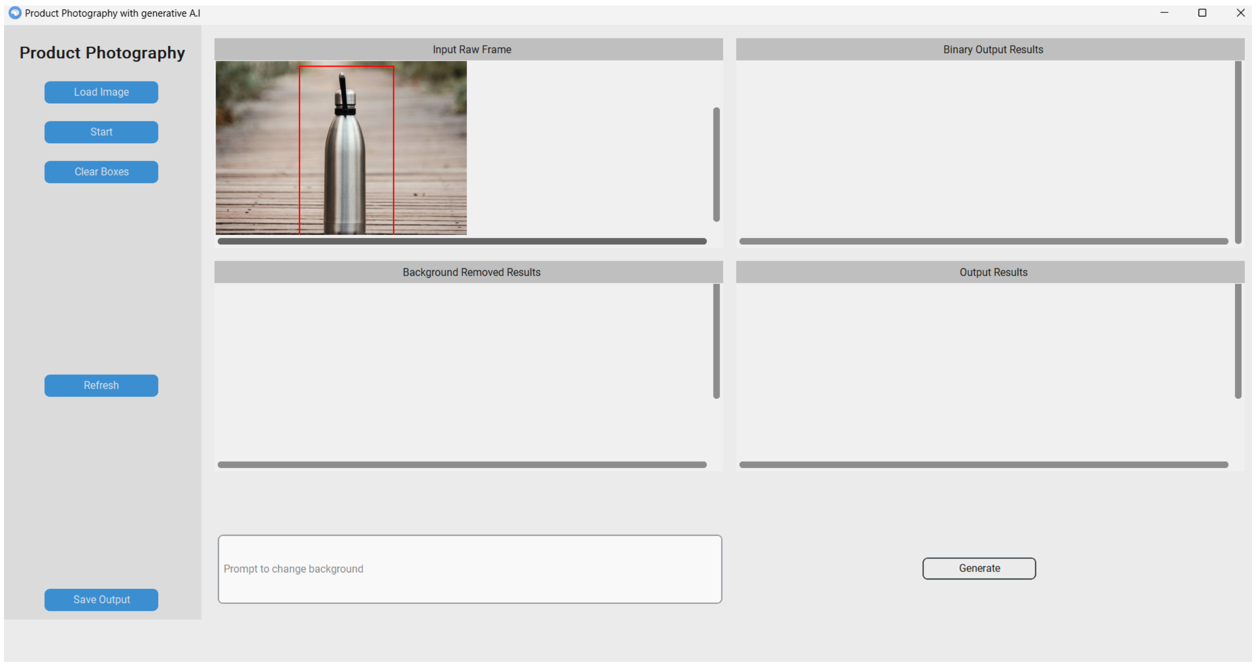


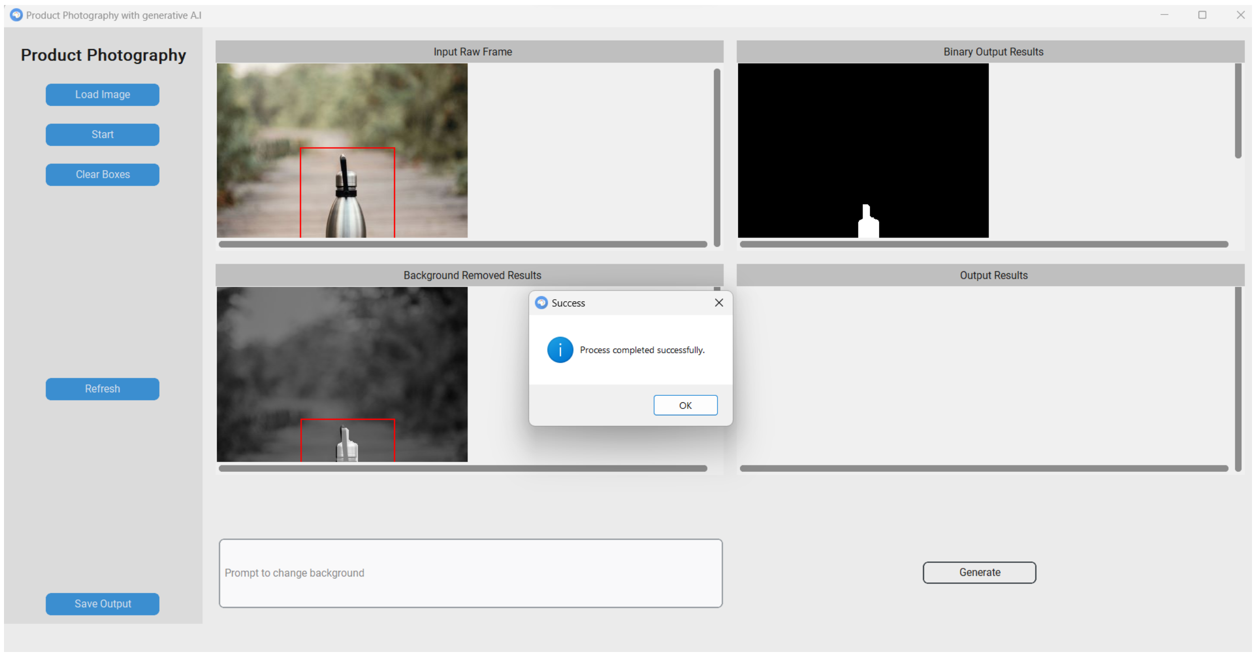
Please start the process by removing background

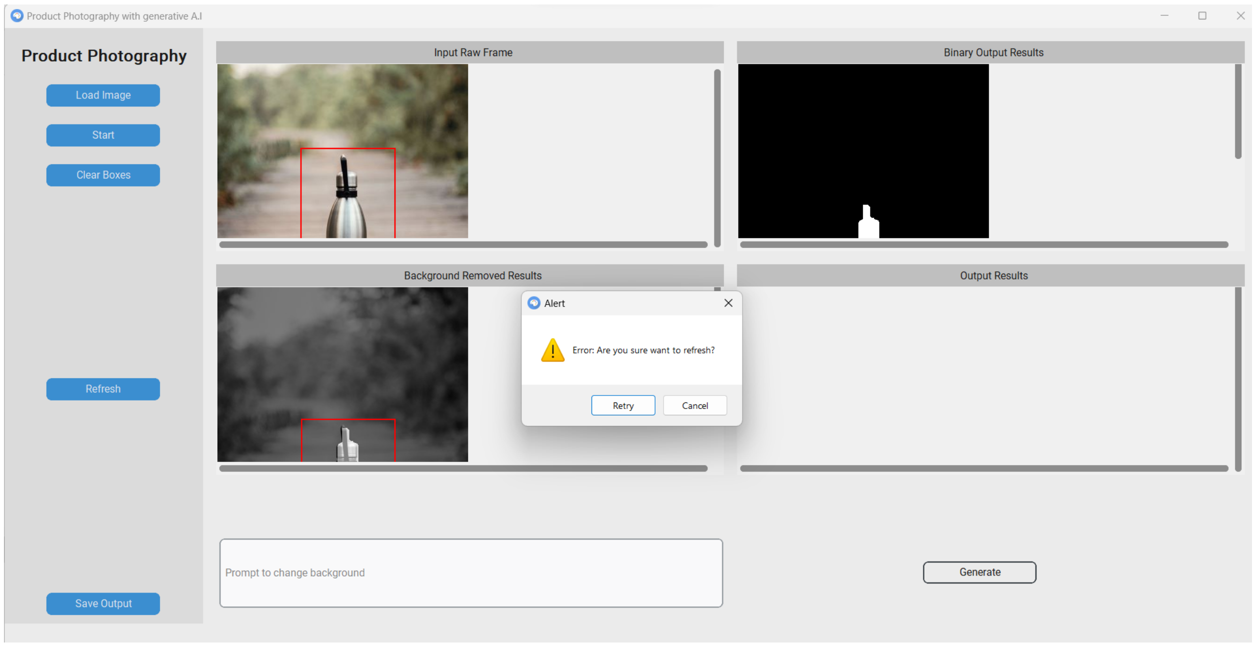




Draw the area to be removed from background


Product Photography with generative A.I
Product Photography
Load Image
Start
Clear Boxes
Refresh
Save Output
Input Raw Frame
Binary Output Results
Background Removed Results
Output Results
Prompt to change background
Generate


Product Photography with generative A.I
Product Photography
Load Image
Start
Clear Boxes
Refresh
Save Output
Input Raw Frame
Binary Output Results
Background Removed Results
Output Results
Success
Process completed successfully.
OK
Prompt to change background
Generate



Error: Are you sure want to refresh?