

Image Background Removal

Build a tool that automatically removes the background from an image, isolating the main subject (e.g., a person or object) and producing a transparent or solid-colored background.

The project should include the following features:

- Identify and segment the foreground from the background.
- Generate a clean output image with the background removed or replaced.
- Allow the user to replace the background with another image.

Hint:

OpenCV includes a semi-automated method that requires a rough bounding box names GrabCut algorithm. You can use it to iteratively refine a mask to separate foreground from background.

For simple backgrounds with high contrast you may use binary thresholding and find the largest contour as the subject.

For consistency please use Python and OpenCV

Your program should allow the user to load an input image, automatically or manually select the subject (optional: allow bounding box input), remove or replace the background, and save the output with a new background image.

Deliverables

- 1- Python code
- 2- A report describing the method you used, some screenshots, and a discussion about the challenges and possible (future) solutions.

=====

Color Detection and Replacement

Create a tool that can detect specific colors in an image and replace them with another color. This can be used for simple color-based editing, or visual effects. Replacing a color with another color should preserve the brightness and shading of the original color. For instance, if you replace green area having multiple shades with red, the areas should have shades of red.

You may follow these steps:

Convert image from **RGB to HSV** (Hue, Saturation, Value) for easier color segmentation.

Apply **color thresholding** to detect the target color range.

Create a **mask** of the detected region.

Replace pixels in the mask with the new color. (only replace hue component)

Example:

You want to replace all blue areas in an image (e.g., a blue sky or blue shirt) with green.

Deliverables

- 1- Python code
- 2- A report describing the method you used, some screenshots, and a discussion about the challenges and possible (future) solutions.

=====

Image Inpainting

Remove unwanted objects (e.g., wires, text, or even people) from an image and fill in the missing parts using surrounding pixels so the result looks natural.

You can use classic inpainting techniques such as Telea's algorithm (cv2.INPAINT_TELEA) or Navier-Stokes-based method (cv2.INPAINT_NS). Both methods are supported by OpenCV and can fill in the missing region using surrounding pixels intelligently.

Steps:

Read the image and create a binary mask (same size as image) where the area to be removed is white (255), and the rest is black (0). You can draw this manually or detect it via color/shape.

Then apply inpainting and display the image.

Deliverables

- 1- Python code
- 2- A report describing the method you used, some screenshots, and a discussion about the challenges and possible (future) solutions.

