

Write a program that reads an image, and converts its color space to HSV, applies color-based filtering to identify brown areas, and then changes the color of those areas to red. The resulting image highlights the regions that were detected as brown

Code

```
Write a program that reads an image, and converts its color space to HSV, applies color-based filtering to identify brown areas, and then changes the color of those areas to red. The resulting image highlights the regions that were detected as brown
```

```
import cv2
import numpy as np

def highlight_brown_areas(image_path, output_path):
    # Read the image
    image = cv2.imread(image_path)

    # Convert the image to HSV color space
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # Define the lower and upper bounds for brown color in HSV
    lower_brown = np.array([10, 50, 50])
    upper_brown = np.array([30, 255, 255])

    # Create a mask based on the brown color range
    mask = cv2.inRange(hsv_image, lower_brown, upper_brown)

    # Create a red color image
    red_color = np.array([0, 0, 255], dtype=np.uint8)
    red_image = np.zeros_like(image)
    red_image[:, :] = red_color

    # Use the mask to highlight the brown areas in the original image
    highlighted_image = np.where(mask[:, :, np.newaxis], red_image, image)

    # Save the resulting image
    cv2.imwrite(output_path, highlighted_image)

if __name__ == "__main__":
    # Replace 'input_image.jpg' with the path to your input image
    input_image_path = '/content/input_image.jpg'

    # Replace 'highlighted_image.jpg' with the desired output file path
    output_image_path = '/content/highlighted_image.jpg'

    highlight_brown_areas(input_image_path, output_image_path)
```

Output

Before



After

