7 - 8.2. Cropping and Concatenating Images

October 6, 2024

Jarrian Vince G. Gojar
Instructor I
College of Information and Communications Technology, Sorsogon State University, Philippines

1 Introduction

In image processing, `cropping` refers to extracting a segment of an image. This is useful when you want to focus on a specific part of an image.

`Concatenating` images refers to combining two or more images into a single image.

Read More:

- Cropping
- Concatenation

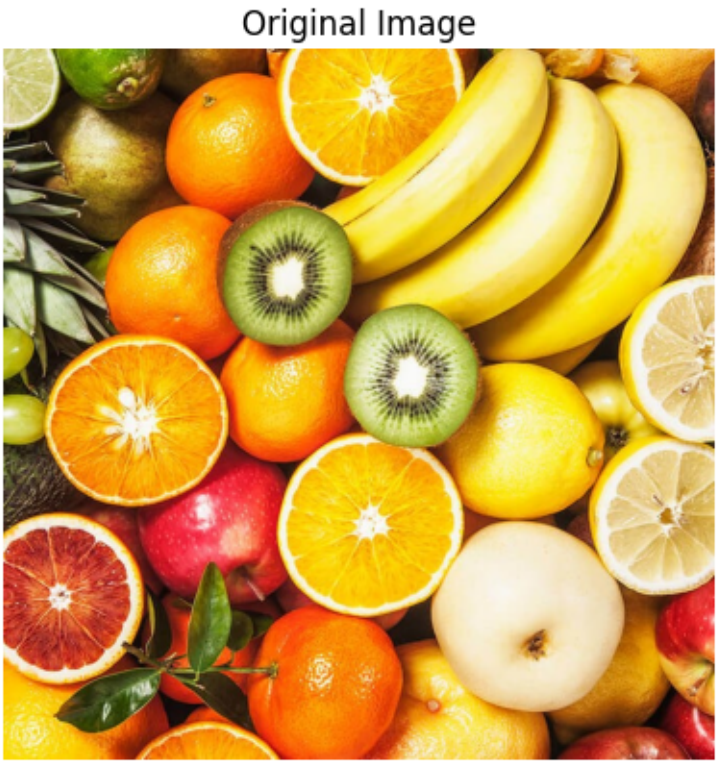
2 Setup

[ ]:
```
%pip install opencv-python opencv-contrib-python numpy matplotlib
```

3 Initial Setup

[25]:
```
# Import Libraries
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Asset Root
asset_root = os.path.join(os.getcwd(), '../../assets')

# Image Path
image_path = os.path.join(asset_root, 'images', 'fruits.jpg')

# Read Image and convert to RGB
input_image = cv2.cvtColor(cv2.imread(image_path), cv2.COLOR_BGR2RGB)

# Display Both Image
```

```
plt.figure("Fruits")

plt.imshow(input_image, cmap='gray')
plt.title("Original Image")
plt.axis('off')

plt.show()
```



4 Cropping Images

To `crop` an image, you need to specify the region of interest (ROI) that you want to extract. The ROI is defined by the top-left corner and the bottom-right corner of the region.

[65]:
```
# Define the region of interest
x1, y1 = 200, 150
x2, y2 = 350, 300

# Crop the image
cropped_image = input_image[y1:y2, x1:x2]

# Display Both Image
```

```
plt.figure("Fruits")

plt.subplot(1, 2, 1)
plt.imshow(input_image)
plt.title(f"Original Image: {input_image.shape[0]}x{input_image.shape[1]}")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cropped_image)
plt.title(f"Cropped Image: {cropped_image.shape[0]}x{cropped_image.shape[1]}")
plt.axis('off')

plt.show()
```

An image can be cropped by specifying the region of interest (ROI) using the top-left and bottom-right corners of the region. The `input_image[y1:y2, x1:x2]` syntax is used to crop the image. The `y1:y2` and `x1:x2` specify the range of rows and columns to be extracted from the image. `x1` and `y1` are the coordinates of the top-left corner of the region, while `x2` and `y2` are the coordinates of the bottom-right corner of the region.

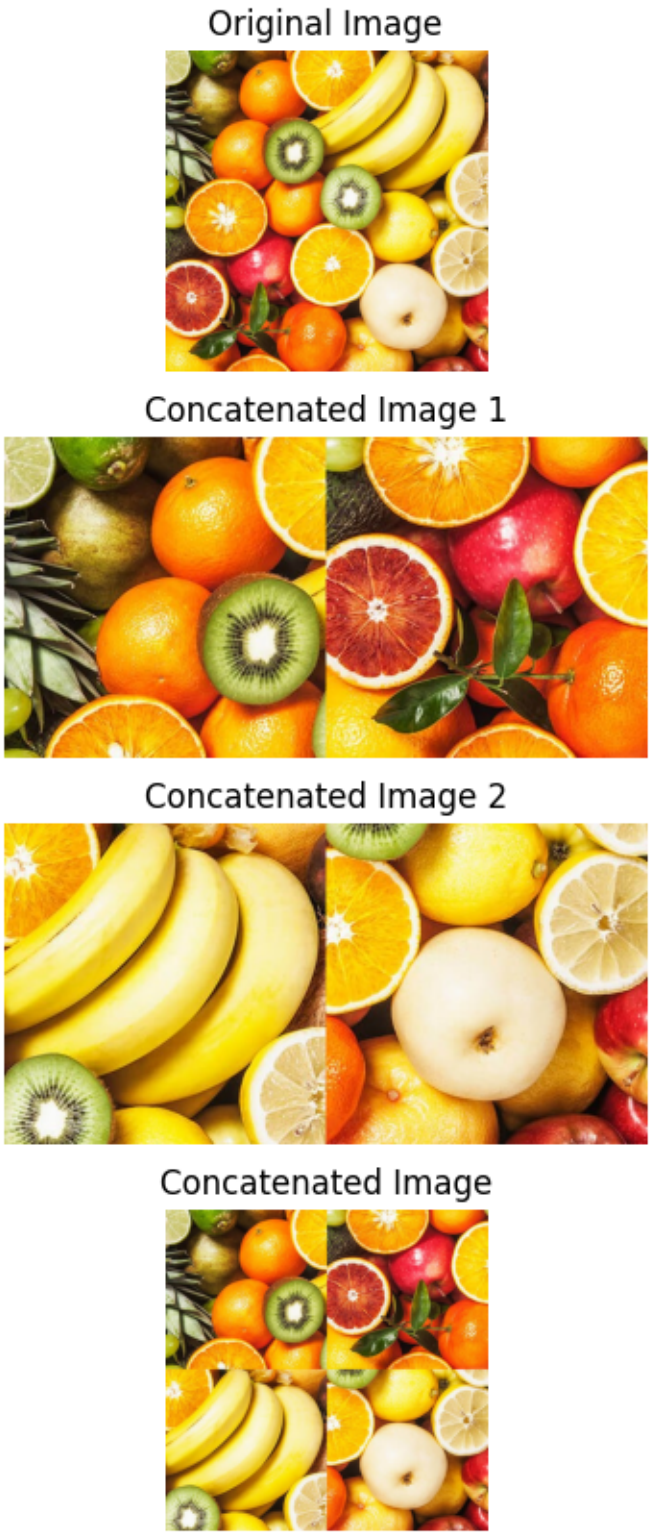
5 Concatenation of Images

In image processing, `concatenation` refers to combining two or more images into a single image.

To concatenate images, you need to ensure that the images have the same dimensions along the axis you want to concatenate them.

[68]:
```
# Set the dimensions of the images
width = input_image.shape[1] // 2
height = input_image.shape[0] // 2
```

```
# Divide the image into four parts
image1 = input_image[:height, :width]
image2 = input_image[:height, width:]
image3 = input_image[height:, :width]
image4 = input_image[height:, width:]

# Concatenate the images
concat1 = np.concatenate((image1, image3), axis=1)
concat2 = np.concatenate((image2, image4), axis=1)
concatenated_image = np.concatenate((concat1, concat2), axis=0)

plt.figure(figsize=(10, 10))

plt.subplot(4, 1, 1)
plt.imshow(input_image)
plt.title(f"Original Image")
plt.axis('off')

plt.subplot(4, 1, 2)
plt.imshow(concat1)
plt.title(f"Concatenated Image 1")
plt.axis('off')

plt.subplot(4, 1, 3)
plt.imshow(concat2)
plt.title(f"Concatenated Image 2")
plt.axis('off')

plt.subplot(4, 1, 4)
plt.imshow(concatenated_image)
plt.title(f"Concatenated Image")
plt.axis('off')

plt.show()
```



In the code snippet above, we divided the original image into four parts and concatenated them

to form a new image. To concatenate the images, the dimensions of the images must be the same along the axis you want to concatenate them. In this case, we concatenated the images along the horizontal and vertical axes. The resulting image is a combination of the four parts of the original image arranged in a 2x2 grid.

The `np.concatenate()` function is used to concatenate the images along the specified axis. The `axis` parameter specifies the axis along which the concatenation should be performed. In this case, we concatenated the images along the horizontal axis (axis=1) and vertical axis (axis=0) to form the final image.

6 Summary

- `Cropping` an image involves selecting a region of interest (ROI) from the original image.
- To crop an image, you need to specify the coordinates of the top-left and bottom-right corners of the region of interest.
- `Concatenating` images involves combining two or more images into a single image.
- To concatenate images, the images must have the same dimensions along the axis you want to concatenate them.
- The `np.concatenate()` function is used to concatenate images along the specified axis.

7 References

- Thomas G. (2022). Graphic Designing: A Step-by-Step Guide (Advanced). Larsen & Keller. ISBN: 978-1-64172-536-1
- Singh M. (2022). Computer Graphics and Multimedia. Random Publications LLP. ISBN: 978-93-93884-95-4
- Singh M. (2022). Computer Graphics Science. Random Publications LLP. ISBN: 978-93-93884-03-9
- Singh M. (2022). Computer Graphics Software. Random Publications LLP. ISBN: 9789393884114
- Tyagi, V. (2021). Understanding Digital Image Processing. CRC Press.
- Ikeuchi, K. (Ed.). (2021). Computer Vision: A Reference Guide (2nd ed.). Springer.
- Bhuyan, M. K. (2020). Computer Vision and Image Processing. CRC Press.
- Howse, J., & Minichino, J. (2020). Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning. Packt Publishing Ltd.
- Kinser, J. M. (2019). Image Operators: Image Processing in Python. CRC Press.