4 - 6.7. Image Sharpening

October 9, 2024

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

1 Introduction

`Image Sharpening` is a technique to enhance the fine details and highlight the edges in an image. It is a post-processing technique that can be used to improve the quality of an image. The sharpening process increases the contrast between the pixels in the image, making the edges appear more defined and the details more pronounced.

In image processing, **`image sharpening`** is achieved by applying a **`sharpening filter`** to the image. Quieter often, the sharpening filter is the `Unsharp Mask` filter.

Read More:

- Image Sharpening
- Unsharp Masking

2 Setup

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

3 Initial Setup

[1]:
```
# Import Libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Asset Root
asset_root = '../../assets/'

# Image Path
image_path = asset_root + '/images/cat_close-up.jpg'

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

```
# Display Both Image
plt.figure("Cat Close-up", figsize=(12, 12))

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

plt.show()
```

4 Unsharp Mask Filter

The `Unsharp Mask` filter is a sharpening filter that works by blurring the image and then subtracting the blurred image from the original image. This process enhances the edges in the image and makes the details more pronounced.

[4]:
```
# Unsharp Mask Filter
def unsharp_mask(image, kernel_size=(5, 5), sigma=1.0, amount=1.0, threshold=0):
    blurred = cv2.GaussianBlur(image, kernel_size, sigma)
    sharpened = cv2.addWeighted(image, 1.0 + amount, blurred, -amount,␣
 ↪threshold)
    return blurred, sharpened

# Apply Unsharp Mask Filter
blurred_image, sharpened_image = unsharp_mask(input_image, kernel_size=(5, 5),␣
 ↪sigma=1.0, amount=5.0, threshold=0)

# Display Both Image
plt.figure("Cat Close-up")

plt.subplots(3, 2, figsize=(12, 18))

plt.subplot(3, 2, 1)
plt.imshow(input_image)
plt.title("Original Image")
plt.axis('off')

plt.subplot(3, 2, 2)
plt.imshow(input_image[100:400, 100:400])
plt.title("Original Image - Crop")
plt.axis('off')

plt.subplot(3, 2, 3)
plt.imshow(blurred_image)
plt.title("Blurred Image")
plt.axis('off')

plt.subplot(3, 2, 4)
plt.imshow(blurred_image[100:400, 100:400])
plt.title("Blurred Image - Crop")
plt.axis('off')

plt.subplot(3, 2, 5)
plt.imshow(sharpened_image)
plt.title("Sharpened Image")
plt.axis('off')
```

```
plt.subplot(3, 2, 6)
plt.imshow(sharpened_image[100:400, 100:400])
plt.title("Sharpened Image - Crop")
plt.axis('off')

plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```

Original Image

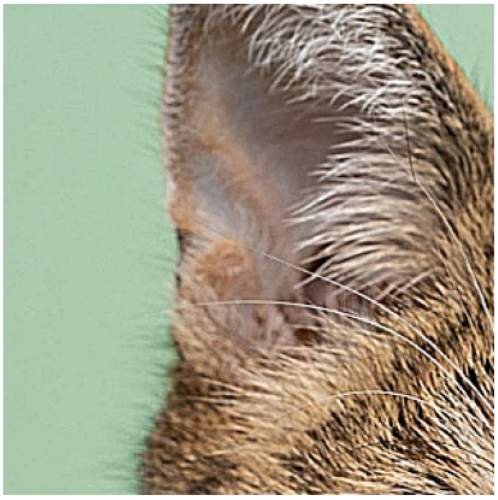
Original Image - Crop

Blurred Image

Blurred Image - Crop

Sharpened Image

Sharpened Image - Crop



The code above applies the `Unsharp Mask` filter to the input image and displays the original image, the blurred image, and the sharpened image. The `Unsharp Mask` filter enhances the edges in the image and makes the details more pronounced. It first blurs the image using a blurring filter (in this case, a `Gaussian blur`) and then subtracts the blurred image from the original image to obtain the sharpened image.

Read More:

- Image Sharpening
- Unsharp Masking

5 Summary

- `Image sharpening` is a technique used to enhance the fine details and highlight the edges in an image.
- The `Unsharp Mask` filter is a sharpening filter that works by blurring the image and then subtracting the blurred image from the original image.
- The `Unsharp Mask` filter enhances the edges in the image and makes the details more pronounced.
- The `Unsharp Mask` filter can be applied to images to improve their quality and make the details more visible.

6 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.