Reyhane Shahrokhian 99521361

HomeWork1 of Computer Vision Course

Dr. Mohammadi

Q1:

When we quantize a grayscale image whose brightness intensity is between 0 and 255 to 64 levels, the new range of pixel brightness values will be between 0 and 63. This reduces the quality of the image. By increasing the number of quantization levels, image details are reduced and may cause unwanted artifacts in the image. This process causes the loss of detailed information in the image.

Q2:

A slower shutter speed can lead to better image quality in low-light situations, while a faster shutter speed can be used to freeze motion and create a sharp image of a moving subject.
So, for capturing a bird flying, it is better to use a fast shutter speed to freeze the motion of the bird.

Q3:

- Shutter speed:

  A slow shutter speed means that your camera sensor is exposed to light for a longer time, resulting in more motion blur in your video. This can be useful for creating a sense of movement, speed, or action, especially when combined with a fast-moving subject or a camera pan. So the photographer should use a slow shutter speed.

- Depth of field:

  Depth of Field refers to the distance between the nearest and farthest objects in a photograph that appear acceptably sharp and in focus. To have a clear subject and a blurred background, the depth of field should be reduced.

- Field of view:

  Field of View (FoV) is the observable area that can be seen through the eyes or an optical device like a camera lens or telescope. A wider field of view allows for capturing more of a scene, while a narrower field of view is useful for focusing on specific details or subjects.

Sources:

https://write.corbpie.com/all-about-shutter-speed/

https://photonify.com/3-steps-for-adjusting-the-depth-of-field-on-your-camera/

Q5:

5-A:

matplotlib reads a color image in RGB manner only but OpenCV reads the color image in BGR format so when reading an image using OpenCV and showing using matplotlib we need to change the channels.

matplotlib یک تصویر رنگی را فقط به روش RGB می خواند اما OpenCV تصویر رنگی را با فرمت BGR می‌خواند بنابراین هنگام خواندن تصویر با استفاده از OpenCV و نشان دادن با استفاده از matplotlib باید کانال ها را تغییر دهیم.

Now try to make Matplotlib output like OpenCV and show it.

```
1 ########################Start Code#################################
2 plt.imshow(I[: , : , ::-1])
3 plt.show()
4 ##########################End Code#################################
```

Source.

5-B:

B-1:

First the image should be reversed vertically, then our two images are concatenated with each other.

Mode 1 : The main image should be joined vertically with its vertical invert.

```
1 ########################Start Code#################################
2 #reverse image
3 I_reverse = cv2.flip(I, 0)
4 #concatenating them
5 result = cv2.vconcat([I, I_reverse])
6 #display
7 cv2_imshow(result)
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
10 ##########################End Code#################################
```

B-2:

The only difference with the first mode is the order in concatenating them.

Mode 2 : The vertical invert of the original image should be joined vertically with the original image.

```
########################Start Code#################################
#concatenating them
result = cv2.vconcat([I_reverse, I])
#display
cv2_imshow(result)
cv2.waitKey(0)
cv2.destroyAllWindows()
#########################End Code#################################
```

B-3:

The steps are the same as mod1 except that the image is flipped and concatenated horizontally.

Mode 3 : The main image should be joined horizontally with its horizontal invert.

```
########################Start Code#################################
#reverse image
I_reverse = cv2.flip(I, 1)
#concatenating them
result = cv2.hconcat([I, I_reverse])
#display
cv2_imshow(result)
cv2.waitKey(0)
cv2.destroyAllWindows()
#########################End Code#################################
```

B-4:

The only difference with the previous mode is the order in concatenating them.

Mode 4 : The horizontal invert of the original image should be joined horizontally with the original image.

```
########################Start Code#################################
#concatenating them
result = cv2.hconcat([I_reverse, I])
#display
cv2_imshow(result)
cv2.waitKey(0)
cv2.destroyAllWindows()
#########################End Code#################################
```

B-5:

The chosen area is colored with yellow in BGR format.

Change the color of one of the windows of the computer faculty building as desired. Try to do this very carefully.

```
1 ########################Start Code###################################
2 #cloring an area with yellow
3 I[298:350,362:387,] = (0, 255, 255)
4 #display
5 cv2_imshow(I)
6 cv2.waitKey(5000)
7 cv2.destroyAllWindows()
8 ##########################End Code#####################################
```

B-6:

First, the image is red and displayed.

```
cv2.imwrite('Colored_Window.jpg', I)
```

```
#reading
logo = cv2.imread('logo.jpg')
#displaying
cv2_imshow(logo)
cv2.waitKey(5000)
cv2.destroyAllWindows()
```

Then, each color channel can be extracted by selecting it(blue:0, green:1, red:2).

```
blue_channel_image = np.zeros_like(logo)
blue_channel_image[:, :, 0] = logo[:, :, 0]
```

```
green_channel_image = np.zeros_like(logo)
green_channel_image[:, :, 1] = logo[:, :, 1]
```

```
red_channel_image = np.zeros_like(logo)
red_channel_image[:, :, 2] = logo[:, :, 2]
```

Blue channel:

As we can see, the white parts of the images are shown as blue. The reason is that white is the combination of blue, green, and red. And we also know that the 2 other channels are zero, so white parts are seen as blue.

The blue parts of the logo are still blue(they were a little of each green and red is the main image but now they are zero and are not count anymore)

Other parts(like the red parts) are seen black due the fact that there is almost no blue color there and other channels are zero.

green channel:

As we can see, the white parts of the images are shown as green. The reason is that white is the combination of blue, green, and red. And we also know that the 2 other channels are zero, so white parts are seen as green.

Other parts(like the red parts) are seen black due the fact that there is almost no green color there and other channels are zero.

red channel:

As we can see, the white parts of the images are shown as red. The reason is that white is the combination of blue, green, and red. And we also know that the 2 other channels are zero, so white parts are seen as red.

The red parts of the logo are still red(they were a little of each green and blue is the main image but now they are zero and are not count anymore)

Other parts(like the blue parts) are seen black due the fact that there is almost no red color there and other channels are zero.