Homework 1 Questions

Instructions

- Compile and read through the included Python tutorial.
- 2 questions.
- Include code.
- Feel free to include images or equations.
- Please make this document anonymous.
- **Please use only the space provided and keep the page breaks.** Please do not make new pages, nor remove pages. The document is a template to help grading.
- If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

Submission

- Please zip your folder with **hw1_student id_name.zip** (ex: hw1_20201234_Peter.zip)
- Submit your homework to KLMS.
- An assignment after its original due date will be degraded from the marked credit per day: e.g., A will be downgraded to B for one-day delayed submission.

Questions

Q1: We wish to set all pixels that have a brightness of 10 or less to 0, to remove sensor noise. However, our code is slow when run on a database with 1000 grayscale images.

Image: grizzlypeakg.png

```
import cv2
import numpy as np
A = cv2.imread('grizzlypeakg.png',0)
m1, n1 = A.shape
for i in range(m1):
   for j in range(n1):
      if A[i,j] <= 10:
         A[i,j] = 0
```

Q1.1: How could we speed it up?

A1.1: Use logical indexing.

```
import cv2
import numpy as np
A = cv2.imread('grizzlypeakg.png',0)
B = A <= 10
A[B] = 0
```

Q1.2: What factor speedup would we receive over 1000 images? Please measure it.

Ignore file loading; assume all images are equal resolution; don't assume that the time taken for one image $\times 1000$ will equal 1000 image computations, as single short tasks on multitasking computers often take variable time.

A1.2: Factor speedup: 882.9766897090233

Q1.3: How might a speeded-up version change for color images? Please measure it.

Image: grizzlypeak.jpg

A1.3: Factor speedup: 292.8242565864542

Q2: We wish to reduce the brightness of an image but, when trying to visualize the result, we see a brightness-reduced scene with some weird "corruption" of color patches.

Image: gigi.jpg

```
import cv2
import numpy as np
I = cv2.imread('gigi.jpg').astype(np.uint8)
I = I - 40
cv2.imwrite('result.png', I)
```

Q2.1: What is incorrect with this approach? How can it be fixed while maintaining the same amount of brightness reduction?

A2.1: Substituting the brightness by 40 for all pixels is incorrect as underflow corrupted the result. It can be fixed by setting pixels with original brightness less than 40 as 0, as code in the below.

```
import cv2
import numpy as np
I = cv2.imread('gigi.jpg').astype(np.uint8)
B = I < 40
I = I - 40
I[B] = 0
cv2.imwrite('result.png', I)
```

Q2.2: Where did the original corruption come from? Which specific values in the original image did it represent?

A2.2: The original corruption came from pixels with original brightness less than 40, as they cause underflow if substituting by 40.