Homework 1 Questions

Instructions

- Compile and read through the included MATLAB 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

```
A = imread('grizzlypeakg.png');
[m1,n1] = size( A );
for i=1:m1
    for j=1:n1
        if A(i,j) <= 10
            A(i,j) = 0;
        end
    end
end
```

Q1.1: How could we speed it up?

A1.1: Your answer here.

```
A = imread('grizzlypeakg.png');
B = A <= 10;
A(B) = 0;
```

We could speed it up by using logical indexing.

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: Your answer here.

```
time1 = 0;
time2 = 0;
for k = 1:1000
    A = imread('grizzlypeakg.png');
    tic;
    B = A <= 10;
    A(B) = 0;
    time1 = time1 + toc;

    A = imread('grizzlypeakg.png');
    tic;
    [m1,n1] = size( A );
    for i=1:m1
        for j=1:n1
            if A(i,j) <= 10
                A(i,j) = 0;
            end
        end
    end
    time2 = time2 + toc;
end
```

We can use the faster algorithm in Answer 1-1). By using 'tic and toc', I measure the time taken by 2 different algorithm. The result is below.

Trial1:
time 1 = 8.7166, time 2 = 15.6610

Trial2:
time 1 = 9.1121, time 2 = 13.2609

Trial3:
time 1 = 8.2952, time 2 = 14.4312

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

Image: grizzlypeak.jpg

A1.3: Your answer here.

```
time3 = 0;

for i = 1:1000
    A = imread('grizzlypeak.jpg');
    tic;
    B = A <= 10;
    A(B) = 0;
    time3 = time3 + toc;
end
disp(time3);
```

There is no change except code loading .jpg file. However, the time has increased by about three times compared to the original speed-up version. The result is below.

Trial1:
time 3 = 28.8884

Trial2:
time 3 = 26.7124

Trial3:
time 3 = 27.4020

Q2: We wish to reduce the brightness of an image but, when trying to visualize the result, all we sees is white with some weird "corruption" of color patches.

Image: gigi.jpg

```
I = double( imread('gigi.jpg') );
I = I - 20;
imshow( I );
```

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

A2.1: Your answer here.

Although we convert an image to floating-point format, the image isn't normalized so there are some entries which are not between 0 and 1. Below code is corrected one.

```
I = imread('gigi.jpg');
I = I - 20;
I = im2double(I);
imshow( I );
```

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

A2.2: Your answer here.

There are some entries that have value of less than or equal to 0.0 after reducing brightness (less than or equal to 20.0 before reducing). They were on the screen with weird color patches, and the other entries whose 3(R, G, B) values are bigger than 0 are shown as white color.