# 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_20181234_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.

```
B=imread('grizzlypeakg.png');
[m2,n2] = size( B );
C = B < 10;
B(C) = 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 ×1000 will equal 1000 image computations, as single short tasks on multitasking computers often take variable time.

**A1.2:** Your answer here.

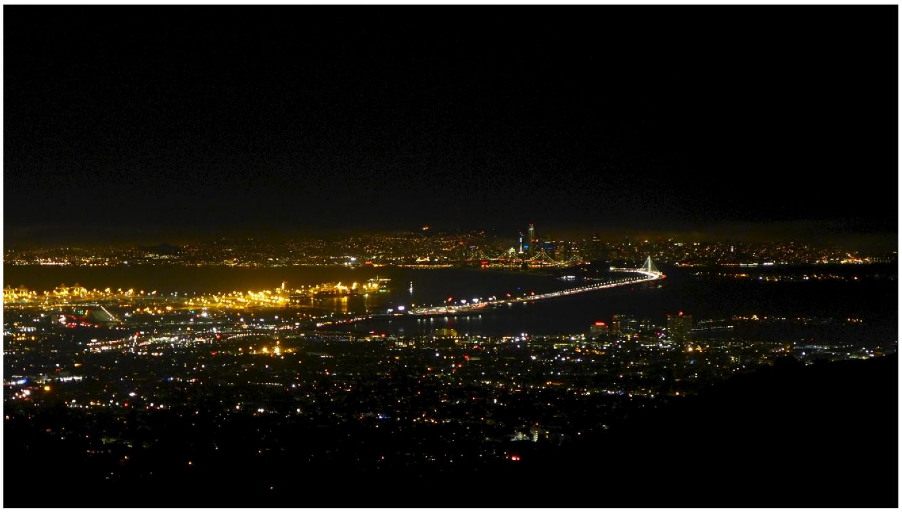
```
tic
for k=0:999
    A=imread('grizzlypeakg.png');
    [m1,n1] = size( A );
    for i=1:m1
        for j=1:n1
            if A(i,j)<=10
                A(i,j) = 0;
            end
        end
    end
    A=zeros(m1,n1);
end
fprintf('method 1:');
toc

tic
for k=0:999
    B=imread('grizzlypeakg.png');
    [m2,n2] = size( B );
    C = B <= 10;
    B(C) = 0;
    B=zeros(m2,n2);
end
fprintf('method 2:');
toc
```

```
>> CV1_2
method 1:Elapsed time is 38.371711 seconds.
method 2:Elapsed time is 29.715092 seconds.
```

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

*Image:* grizzlypeak.jpg

**A1.3:** Your answer here.



```
>> CV1_3
1 try, method 1:Elapsed time is 6.063604 seconds.
1 try, method 2:Elapsed time is 0.034056 seconds.
2 try, method 1:Elapsed time is 6.077723 seconds.
2 try, method 2:Elapsed time is 0.035354 seconds.
3 try, method 1:Elapsed time is 6.229049 seconds.
3 try, method 2:Elapsed time is 0.036830 seconds.
4 try, method 1:Elapsed time is 6.051567 seconds.
4 try, method 2:Elapsed time is 0.037339 seconds.
5 try, method 1:Elapsed time is 5.995840 seconds.
5 try, method 2:Elapsed time is 0.035910 seconds.
6 try, method 1:Elapsed time is 6.090698 seconds.
6 try, method 2:Elapsed time is 0.034144 seconds.
7 try, method 1:Elapsed time is 5.853549 seconds.
7 try, method 2:Elapsed time is 0.034807 seconds.
8 try, method 1:Elapsed time is 5.905430 seconds.
8 try, method 2:Elapsed time is 0.033173 seconds.
9 try, method 1:Elapsed time is 6.270601 seconds.
9 try, method 2:Elapsed time is 0.034299 seconds.
10 try, method 1:Elapsed time is 5.843067 seconds.
10 try, method 2:Elapsed time is 0.034772 seconds.
```

**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.

I should use im2double instead of double function.

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

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

**A2.2:** Your answer here.

```
I1 = im2double(imread('gigi.jpg'));
I1 = I1 - 20;

I2 = imread('gigi.jpg');
I2 = I2 - 20;
doubleI2 = double(I2);
```

In code, I1 and doubleI2 are both problematic.

If you use ”im2double” function, the value of (uint8) 0-255 changes to (double) 0-1. I1 used an ”im2double” function, but after using the function, it did -20. So all black out. I2 did -20 first, but not im2double function. So It makes ”corruption”.