Assignment 1 (40 points)

Sebastian Zdarowski CSC 481

September 29, 2019

From the text:

2.2 (*1/2*)

If you use a sheet of white paper…

Ans: Visual processing that is responsible for this is called Simultaneous contrast

481 Students: 2.4 (*1/0*)

When you enter a dark theater…

Ans: Visual process responsible for this would be called brightness adaptation.

481 Students: 2.11 (*4/0*) (note that this is a vast simplification of the complex networking tasks that might be involved, and 56K baud modems haven't been used in decades, but it's an interesting comparison, nonetheless.)

A common measure of transmission for digital data is the baud rate….

a. How many seconds would it take to transmit a sequence of 500 images of size 1024x1024 pixels with 256 intensity using a 3 M-Baud modem (10^6).

   Ans: Bits Calculation: (1024^2 * (1 +8 + 1))

   Time calc: 10485760 / (3*10^6) = 3.49 Seconds x 500 images = 1745 Total Seconds for 500 images

b. What would the time be using a 30 G baud modem (10^9)

   Ans: Bits calculation: (1024^2 * (1 + 8 + 1))

   Time Calc 10485760 / (30*10^9) = 3.4953e-04 seconds x 500 images = 0.1748 seconds for 500 images

2.16 (*3/4*)

These two subsets are M-Adjacent and 8 adjacent, 4 neighbors isn't satisfied by this vector due to the 1s being diagonal, M- Adjacent and 8 Adjacent do qualify for these vectors.

2.20 (*4/6*)

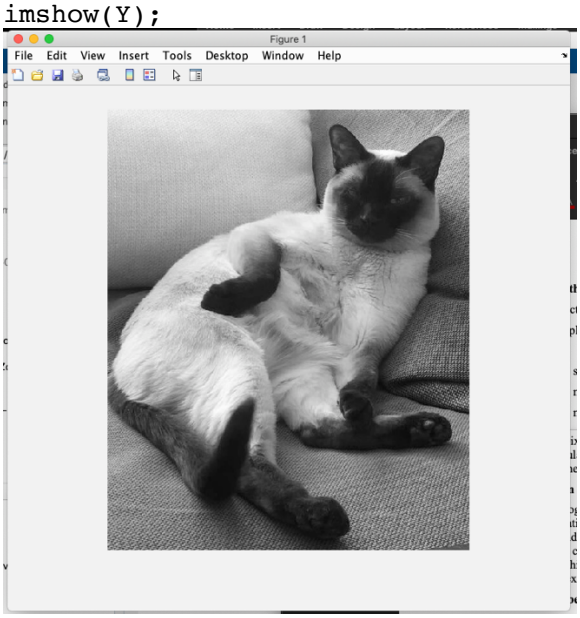
Programming:

Getting familiar with image manipulation in Matlab (*10*/*10*)

Write a program/function that will:

(a) Read and display an image

```
% Homework 1 Sebastian Zdarowski
X = imread('collins.jpg');
Y = rgb2gray(X);
imshow(Y);
```



(b) Calculate the size of the image

```
%Calculate Size
size(Y)
ans =684   563
```

(c) Calculate the maximum pixel value

```
%Calculate max pixel
max(Y(:))
ans =uint8 255
```

(d) Calculate the mean pixel value

```
%Calculate mean value
mean(Y(:))
ans =121.3527
```

(e) Change the pixel values of the image in the following way: all pixels' values less than the average calculated at (d) will be equal to 0 and all the others will be equal to 1. What type of image is the new generated image?

```
%E
Z = zeros(size(Y));
Z = (Y>mean(Y(:))) == 1;
imshow(Z)
```

Binary image is created when setting all pixel values that are less than mean to 0 and setting all others to 1.

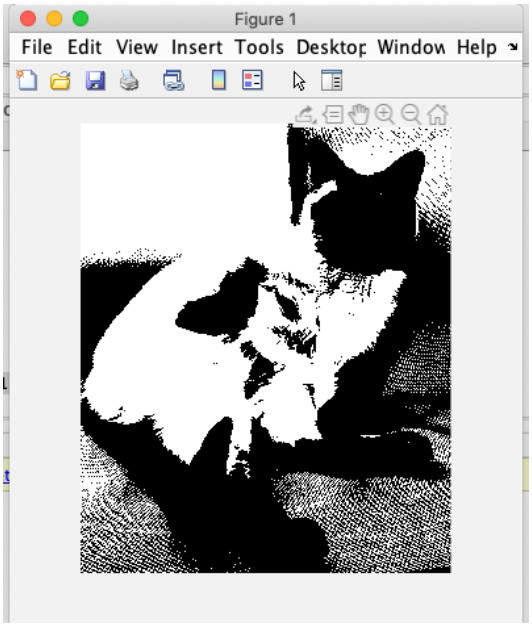
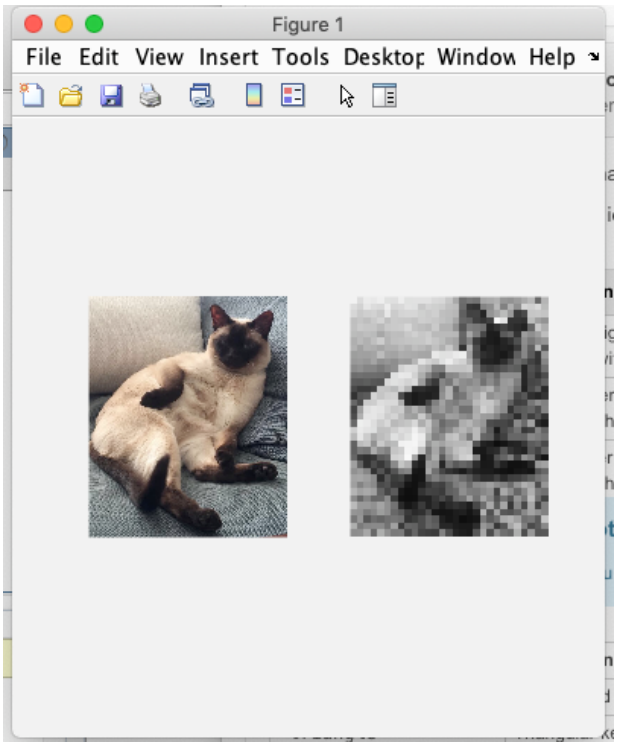


Image Interpolation (*5/7*)

Write a computer program to produce results comparable to Figure 2.27. That is, given an input image, reduce its spatial resolution, and then return it to its original resolution. Use all of nearest neighbor, bilinear and bicubic interpolation to do this. **481 Students (*2/0*):** Design your program such that the desired change in spatial resolution (e.g. 0.5, which will halve the image in each dimension, or 2.0, which will double the image in each dimension) is a variable input to your program. Show an example run of your code.

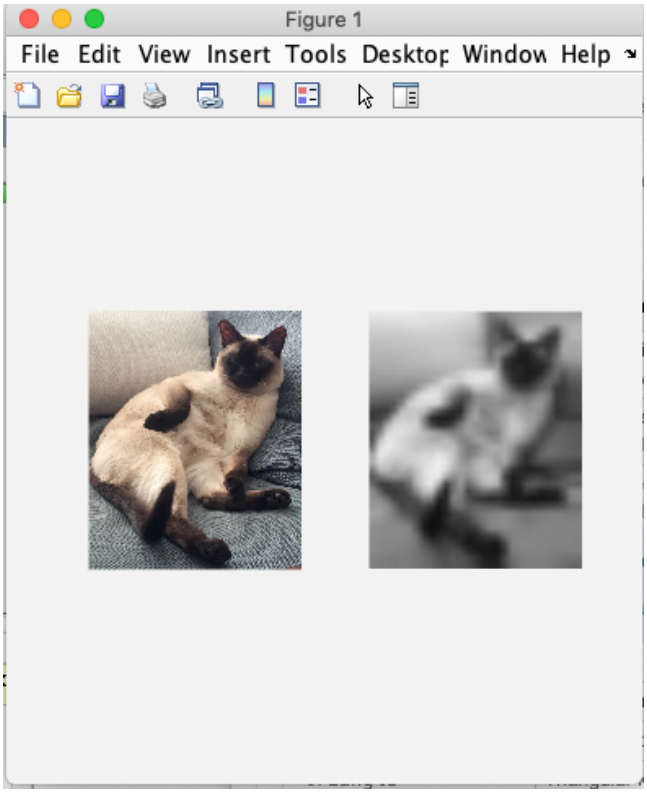
Neighbor Output: change = .05

```
%imreizeCust(X, 'nearest', 0.05);
```



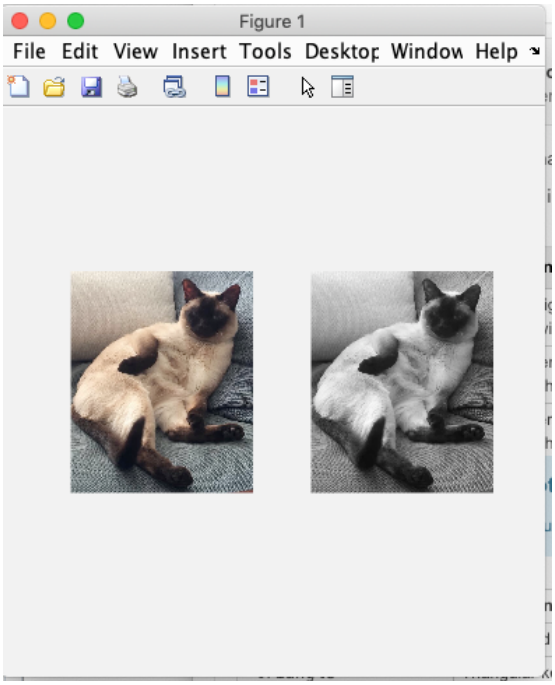
Bilinear output change = .05

```
%imreizeCust(X, 'bilinear', 0.05);
```



Bicubic change = 2.0

```
imreizeCust(X, 'bicubic', 2.0);
```



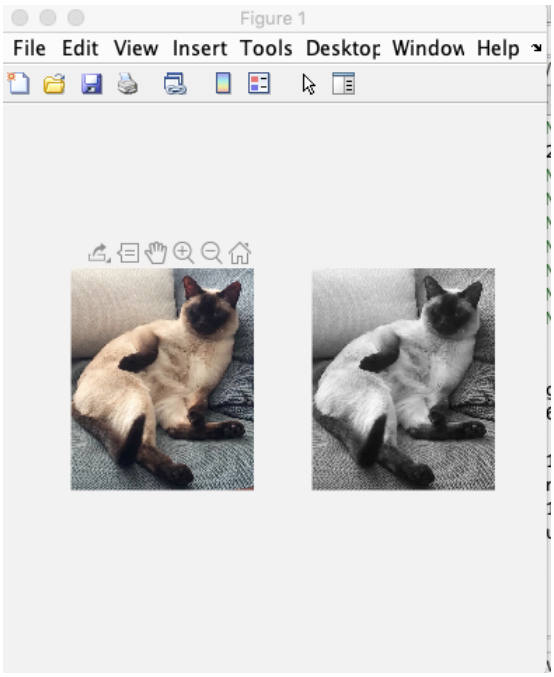
Code:
```
%imreizeCust(X, 'nearest', 0.05);
%imreizeCust(X, 'bilinear', 0.05);
imreizeCust(X, 'bicubic', 2.0);

function [ output ] = imreizeCust(input, method, change)
x = rgb2gray(input);
%interpolation method function imresize
reduce = imresize(x, double(change), method);
output = imresize(reduce, 1.0/double(change), method);
subplot(1, 2, 1) %show side by side
imshow(input) %output
subplot(1, 2, 2) %Show side by side
imshow(output)
end
```

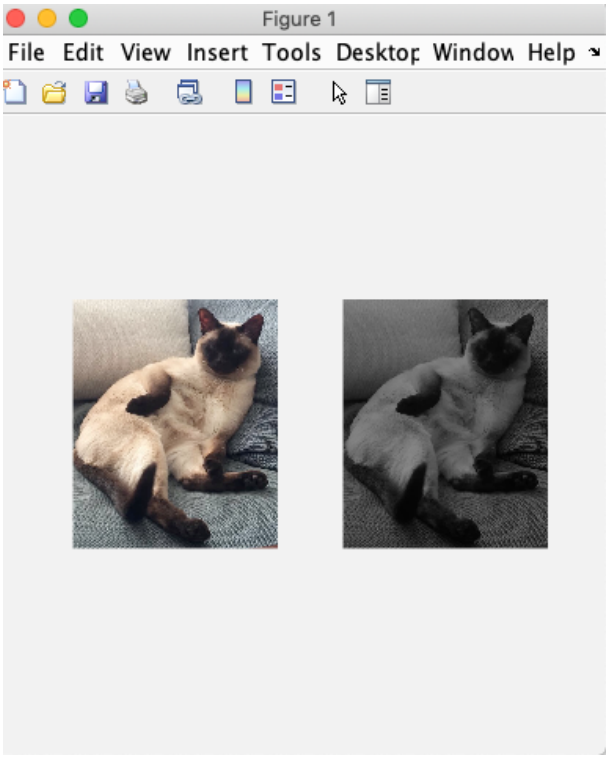
Reducing the Number of Gray Levels in an Image (*8/11*)

Write a computer program capable of reducing the number of gray levels in an image from 256 to 2, in integer powers of 2. **481 Students (*2/0*):** Design your program such that the desired number of gray levels (256, 128, 64, 32, 16, 8, 4, 2) is a variable input to your program. Show an example run of your code.

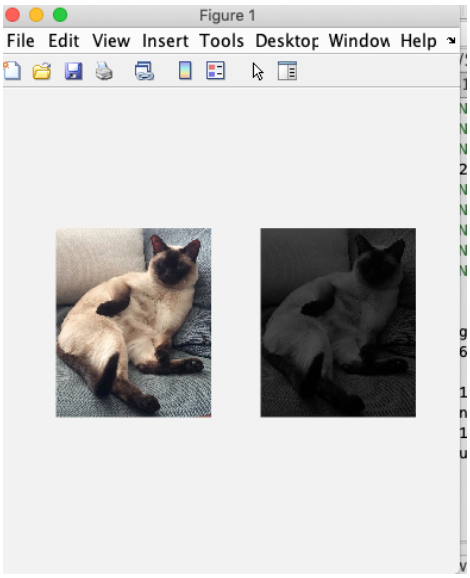
Level = 256



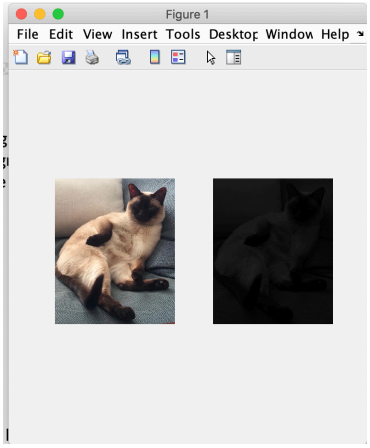
Level = 128



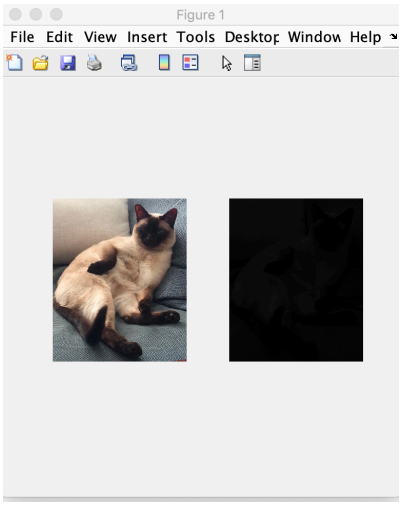
Level = 64



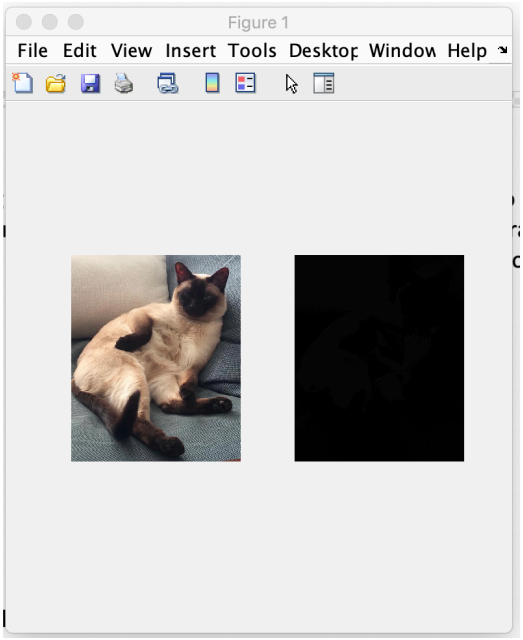
Level = 32



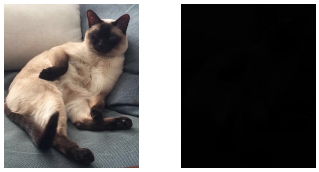
Level = 16



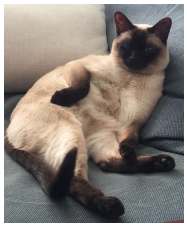
Level = 8



Level = 4



Level = 2



Code:

```
%customFN2 Question 3 reduce # of gray levels in an image from 256 to 2
%customFN2(X, 256);
%customFN2(X, 128);
%customFN2(X, 64);
%customFN2(X, 32);
%customFN2(X, 16);
%customFN2(X, 8);
%customFN2(X, 4);
customFN2(X, 2);


function [ output ] = customFN2(input, grayL)
x = rgb2gray(input);
int = 256.0/double(grayL);
output = x/double(int) % divide image by levels
subplot(1, 2, 1) %show side by side
imshow(input) %output
subplot(1, 2, 2) %Show side by side
imshow(output)
end
```

General submission instructions:

(a) Be kind to your aging, over-worked professor and submit only a single document. This can be pdf, MS Word, OpenOffice, etc. Do not submit a zip file.

(b) Your single document should include the input image for your problem, if required, and answers to each of the sub-problems (text, image or both, as appropriate). For example, 1(e) will require you to show an output image and a text answer, 1(a) only an image, and 1(b) only text. Your document should also include code that you wrote to generate your answers.

(c) You may use any images you like for the programming; I encourage you to use images that might be useful/interesting for your final project.

(d) Feel free to use whatever functions MatLab supplies. Also feel free to write your own, if you are so inclined; it will take more time, but you will gain a deeper understanding of the material. It is one thing, for example, to implement bicubic interpolation using the matlab `resize`, quite another to write a bicubic interpolator yourself.

(e) Feel free to use whatever programming language you like. Much of the class will use MatLab, but python, Java, C++ (and probably others) all have good support for image processing.

(f) Point values for each question are indicated as ***x/y*** in which ***x*** is the point value for 481 students and ***y*** is the point value for 381 students.