**Eastern Mediterrenean University**

**Software Engineering Department**

**CMSE424 – Image Processing**

**Çağıl Peköz**

**17330172**

**Lab 2 Report**

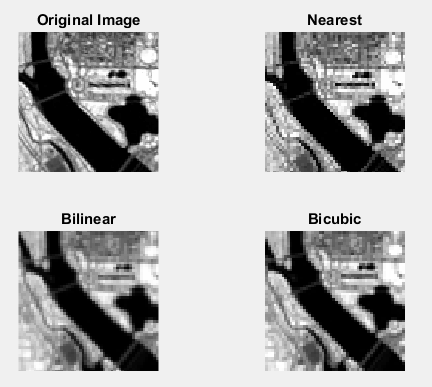
E1) a) When we do any kinds of scaling and/or rotation with images, we can use different interpolation methods. They all vary based on their cost of performance. Nearest only considers the nearest pixels, bilinear and bicubic considers more and more nearby pixels. As you can see on the Figure 1, using the nearest method, the edges of the river is too sharp. But as we use bilinear and bicubic, the edges become smoother and better looking.

Figure Shrinking

b)

Shrinking:

nearest = imresize(B,0.5, 'nearest');

bilinear = imresize(B,0.5, 'bilinear');

bicubic = imresize(B,0.5, 'bicubic');



Rotate:

nearest = imrotate(B,45, 'nearest');

bilinear = imrotate(B,45, 'bilinear');

bicubic = imrotate(B,45, 'bicubic');

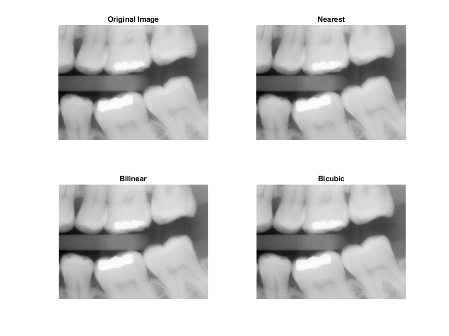


Zoom:

nearest = imresize(B,2, 'nearest');

bilinear = imresize(B,2, 'bilinear');

bicubic = imresize(B,2, 'bicubic');



E2) a)

grayImage2 = B - rem(B,2);

diffImage = double(B) - double(grayImage2);



b) We basically used the square pictures as a mask and multiplied it over the original image to segment the parts that we want. We used mat2gray to be able to do vector calculations with 0-1, and then we turned it into double for the multiplication and when we are done we converted it back to 0-255 intensity so that we can see it.

grayImage = mat2gray(B);

multiplied = double(second\_pic) .\* double(grayImage);

multiplied = uint8(255\*mat2gray(multiplied));



c) The negative image is created by imcomplement command, for the Set Union, I have used the max function with 0.25 threshold, which means that anything below 0.25 whiteness is ignored.

grayImage = mat2gray(B);

whites=max(grayImage,0.25);

negative = imcomplement(grayImage);

