Bolin He

A53316428

Academic Integrity Policy: Integrity of scholarship is essential for an academic commu- nity. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.

By including this in my report, I agree to abide by the Academic Integrity Policy men- tioned above.

Bolin He, PID: A53316428

Table of Contents

Oct 18,2019

Question 1

```
clear all;
clc;
A = [3 9 5 1;4 25 4 3;63 13 23 9;6 32 77 0;12 8 6 1];
B = [0 1 0 1;0 1 1 0;0 0 0 1;1 1 0 1;0 1 0 0];

fprintf('\nAnswer\n')
C = A.*B

C24 = dot(C(2,:),C(4,:));
fprintf('The inner product of the 2nd and 4th row of C is 800.\n')

Cmax = max(max(C));
[x1,y1] = find(C==Cmax);
[x1,y1];
fprintf('The maximum value is 32, it locates at[4,2]\n')

Cmin = min(min(C));
[x2,y2] = find(C==Cmin);

fprintf('The minimum value is 0, it locates at:\n')
[x2,y2]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

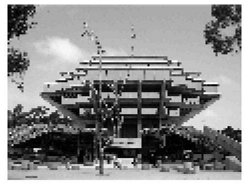
```
Answer

C =

     0     9     0     1
     0    25     4     0
     0     0     0     9
     6    32     0     0
     0     8     0     0

The inner product of the 2nd and 4th row of C is 800.
The maximum value is 32, it locates at[4,2]
```

```
The minimum value is 0, it locates at:

ans =

     1     1
     2     1
     3     1
     5     1
     3     2
     1     3
     3     3
     4     3
     5     3
     2     4
     4     4
     5     4
```

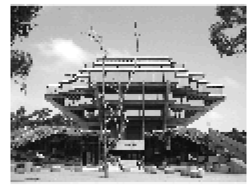
Question 2

```
clear all;
clc;
A = imread('geisel.jpg');
subplot(2,3,1)
imshow(A)
xlabel('A')

B = rgb2gray(A);
subplot(2,3,2)
imshow(B)
xlabel('B')

C = B+15;
% Any pixel values greater than 255 will automatically set to 255
subplot(2,3,3)
imshow(C)
xlabel('C')

D = flipud(B);  %To flip image across horizontal axes
D = fliplr(D);  %To flip image across vertical axes
subplot(2,3,4)
imshow(D)
xlabel('D')

E = B <= median(median(B));
subplot(2,3,5)
imshow(E)
xlabel('E')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

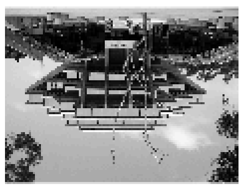
A



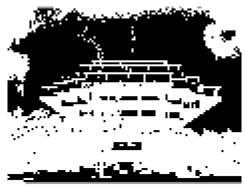
B



C



D

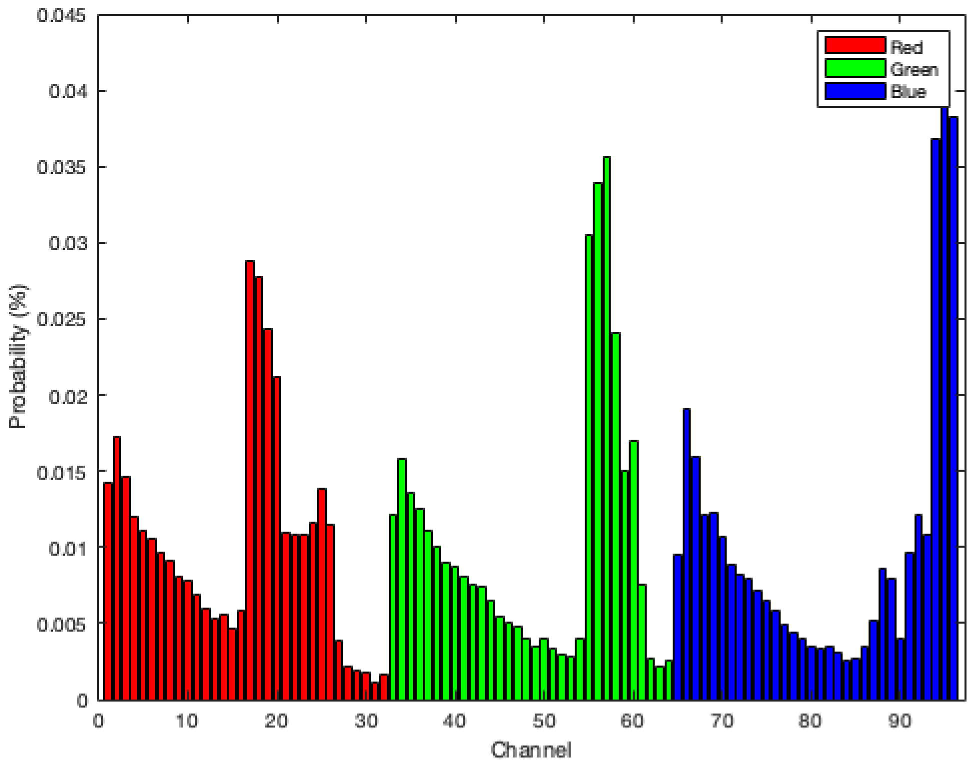


E

Question 3

```
clear all;
clc;
close all;
T = compute_norm_rgb_histogram('geisel.jpg');

% function [h] = compute_norm_rgb_histogram(input)
% % input (RGB/color image) and one output (1 x 96 vector)
% I = imread(input);
% R = I(:,:,1);
% G = I(:,:,2);
% B = I(:,:,3);
%
% % Initialize
% [x,y] = size(R);
% n = 255/32;
% h = zeros(1,3*32);
%
% %  Count the numbers
% for bins = 1:32
%     for i = 1:x
%         for j = 1:y
%             if R(i,j)>n*(bins-1) && R(i,j)<=n*bins
%                 h(bins) = h(bins)+1;
```

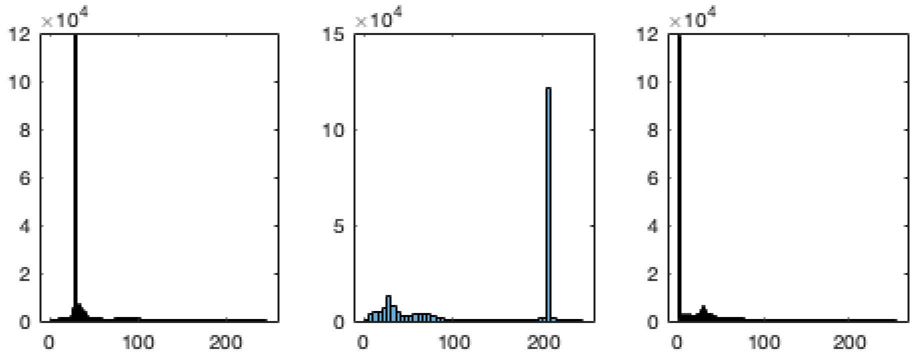
```
%             end
%         end
%     end
% end
%
% for bins = 33:64
%     for i = 1:x
%         for j = 1:y
%             if G(i,j)>n*(bins-33) && G(i,j)<=n*(bins-32)
%                 h(bins) = h(bins)+1;
%             end
%         end
%     end
% end
%
%
% for bins = 65:96
%     for i = 1:x
%         for j = 1:y
%             if B(i,j)>n*(bins-65) && B(i,j)<=n*(bins-64)
%                 h(bins) = h(bins)+1;
%             end
%         end
%     end
% end
%
% % Plot the histogram
% h = h./sum(h);
% bk = zeros(1,32); % blank matrix
% bar([h(1,1:32),bk,bk],'r')
% hold on
% bar([bk,h(1,33:64),bk],'g')
% hold on
% bar([bk,bk,h(1,65:96)],'b')
% legend('Red','Green','Blue')
% xlabel('Channel')
% ylabel('Probability (%)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

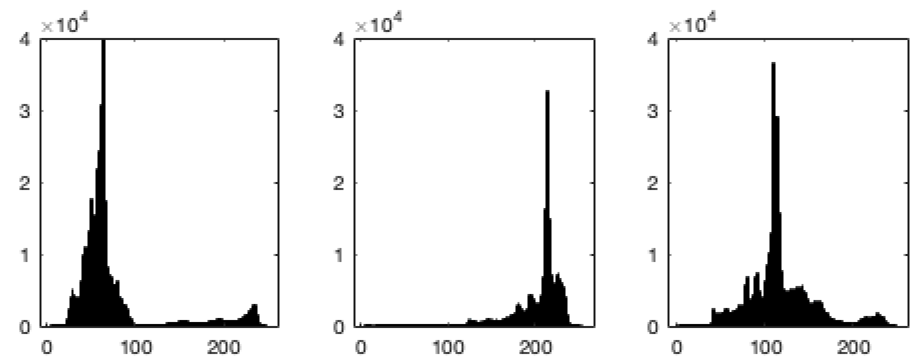


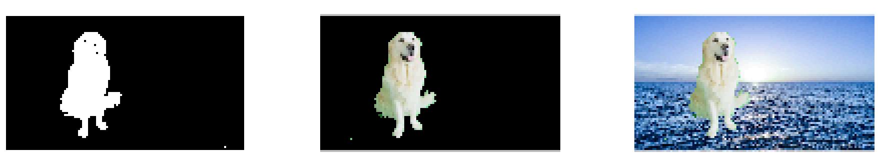
Question 4

```
clear all;
clc;
% Load data
% travolta
BG1 = imread('sea.jpeg'); % background
tra = imread('travolta.jpg');
traR = tra(:,:,1);
traG = tra(:,:,2);
traB = tra(:,:,3);

% Plot RGB histogram
subplot(2,3,1)
histogram(traR)
subplot(2,3,2)
histogram(traG)
subplot(2,3,3)
histogram(traB)

% Process image
traG(traR<120 & traG >100 & traB <90 )=0;
traR(traG == 0) = 0;
traB(traG == 0) = 0;
tra(:,:,1) = traR;
```

```
tra(:,:,2) = traG;
tra(:,:,3) = traB;
BG1(:,:,1) = tra(:,:,1) + BG1(:,:,1).*(uint8(~logical(traG)));
BG1(:,:,2) = tra(:,:,2) + BG1(:,:,2).*(uint8(~logical(traG)));
BG1(:,:,3) = tra(:,:,3) + BG1(:,:,3).*(uint8(~logical(traG)));

% Show image
subplot(2,3,4)
imshow(logical(traG))
subplot(2,3,5)
imshow(tra)
subplot(2,3,6)
imshow(BG1)


% dog
BG2 = imread('sea2.jpg');
dog = imread('dog.jpg');
dogR = dog(:,:,1);
dogG = dog(:,:,2);
dogB = dog(:,:,3);

% Plot RGB histogram
figure
subplot(2,3,1)
histogram(dogR)
subplot(2,3,2)
histogram(dogG)
subplot(2,3,3)
histogram(dogB)

% Process image
dogG(dogR<120 & dogG>110 & dogB<180)=0;
dogR(dogG == 0) = 0;
dogB(dogG == 0) = 0;
dog(:,:,1) = dogR;
dog(:,:,2) = dogG;
dog(:,:,3) = dogB;
BG2(:,:,1) = dog(:,:,1) + BG2(:,:,1).*(uint8(~logical(dogG)));
BG2(:,:,2) = dog(:,:,2) + BG2(:,:,2).*(uint8(~logical(dogG)));
BG2(:,:,3) = dog(:,:,3) + BG2(:,:,3).*(uint8(~logical(dogG)));

% Show image
subplot(2,3,4)
imshow(logical(dogG))
subplot(2,3,5)
imshow(dog)
subplot(2,3,6)
imshow(BG2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


×10^4
12
10
8
6
4
2
0
0
100
200
×10^4
15
10
5
0
0
100
200
×10^4
12
10
8
6
4
2
0
0
100
200


×10^4
4
3
2
1
0
0
100
200
×10^4
4
3
2
1
0
0
100
200
×10^4
4
3
2
1
0
0
100
200



Question 5

```
clear all;
clc;
close all;
% (i)
fprintf('Answer(i)\n')
fprintf('The interpolation methods for image sampling include:\n')
fprintf('Nearest neighbor interpolation, Linear Interpolation, Cubic
 Interpolation.\n')

% (ii)
fprintf('\nAnswer(ii)\n')
fprintf('Nearest neighbor interpolation: We simply select the sample
 point that is nearest to x, which coule be the largest integer values
 that is less than or equal to x.\n')
fprintf('Linear Interpolation: Between to adjacent sample points k and
 k+1 we assume the function is a linear function.\n')
fprintf('Cubic Interpolation: We look at two pixels on the left and
 two on the right, interpolate it as cubic function.\n')

% (iii)
fprintf('\nAnswer(iii)\n')
a = imread('_LHY4390.jpg');
b = imread('Hua.jpg');
c = imread('IMG_1528.jpg');

% Image a downsampling
figure
adown1 = imresize(a,0.3,'nearest');
adown2 = imresize(a,0.3,'bilinear');
adown3 = imresize(a,0.3,'cubic');
subplot(3,3,1), imshow(adown1)
ylabel('0.3')
subplot(3,3,2), imshow(adown2)
subplot(3,3,3), imshow(adown3)

adown4 = imresize(a,0.5,'nearest');
adown5 = imresize(a,0.5,'bilinear');
adown6 = imresize(a,0.5,'cubic');
subplot(3,3,4), imshow(adown4)
ylabel('0.5')
subplot(3,3,5), imshow(adown5)
subplot(3,3,6), imshow(adown6)


adown7 = imresize(a,0.7,'nearest');
adown8 = imresize(a,0.7,'bilinear');
adown9 = imresize(a,0.7,'cubic');
subplot(3,3,7), imshow(adown7)
xlabel('Nearest')
ylabel('0.7')
subplot(3,3,8), imshow(adown8)
```

```
xlabel('Linear')
subplot(3,3,9), imshow(adown9)
xlabel('Cubic')

% Image b downsampling
figure
bdown1 = imresize(b,0.3,'nearest');
bdown2 = imresize(b,0.3,'bilinear');
bdown3 = imresize(b,0.3,'cubic');
subplot(3,3,1), imshow(bdown1)
ylabel('0.3')
subplot(3,3,2), imshow(bdown2)
subplot(3,3,3), imshow(bdown3)

bdown4 = imresize(b,0.5,'nearest');
bdown5 = imresize(b,0.5,'bilinear');
bdown6 = imresize(b,0.5,'cubic');
subplot(3,3,4), imshow(bdown4)
ylabel('0.5')
subplot(3,3,5), imshow(bdown5)
subplot(3,3,6), imshow(bdown6)

bdown7 = imresize(b,0.7,'nearest');
bdown8 = imresize(b,0.7,'bilinear');
bdown9 = imresize(b,0.7,'cubic');
subplot(3,3,7), imshow(bdown7)
xlabel('Nearest')
ylabel('0.7')
subplot(3,3,8), imshow(bdown8)
xlabel('Linear')
subplot(3,3,9), imshow(bdown9)
xlabel('Cubic')

% Image c downsampling
figure
cdown1 = imresize(c,0.3,'nearest');
cdown2 = imresize(c,0.3,'bilinear');
cdown3 = imresize(c,0.3,'cubic');
subplot(3,3,1), imshow(cdown1)
ylabel('0.3')
subplot(3,3,2), imshow(cdown2)
subplot(3,3,3), imshow(cdown3)

cdown4 = imresize(c,0.5,'nearest');
cdown5 = imresize(c,0.5,'bilinear');
cdown6 = imresize(c,0.5,'cubic');
subplot(3,3,4), imshow(cdown4)
ylabel('0.5')
subplot(3,3,5), imshow(cdown5)
subplot(3,3,6), imshow(cdown6)

cdown7 = imresize(c,0.7,'nearest');
cdown8 = imresize(c,0.7,'bilinear');
cdown9 = imresize(c,0.7,'cubic');
```

```
subplot(3,3,7), imshow(cdown7)
ylabel('0.7')
xlabel('Nearest')
subplot(3,3,8), imshow(cdown8)
xlabel('Linear')
subplot(3,3,9), imshow(cdown9)
xlabel('Cubic')

fprintf('We can observe that the downsampling quality is different
 from three methods. Obvious pixels can be noticed by using Nearest
 Neighbor Interpolation.\n')
fprintf('The Cubic Interpolation is better than Linear Interpolation,
 follows by Nearest Neighbot Interpolation.\n')

% (iv)
fprintf('\nAnswer(iv)\n')
% Image a upsampling
figure
aup1 = imresize(a,1.5,'nearest');
aup2 = imresize(a,1.5,'bilinear');
aup3 = imresize(a,1.5,'cubic');
subplot(3,3,1), imshow(aup1)
ylabel('1.5')
subplot(3,3,2), imshow(aup2)
subplot(3,3,3), imshow(aup3)


aup4 = imresize(a,1.7,'nearest');
aup5 = imresize(a,1.7,'bilinear');
aup6 = imresize(a,1.7,'cubic');
subplot(3,3,4), imshow(aup4)
ylabel('1.7')
subplot(3,3,5), imshow(aup5)
subplot(3,3,6), imshow(aup6)

aup7 = imresize(a,2,'nearest');
aup8 = imresize(a,2,'bilinear');
aup9 = imresize(a,2,'cubic');
subplot(3,3,7), imshow(aup7)
ylabel('2')
xlabel('Nearest')
subplot(3,3,8), imshow(aup8)
xlabel('Linear')
subplot(3,3,9), imshow(aup9)
xlabel('Cubic')

% Image b upsampling
figure
bup1 = imresize(b,1.5,'nearest');
bup2 = imresize(b,1.5,'bilinear');
bup3 = imresize(b,1.5,'cubic');
subplot(3,3,1), imshow(bup1)
ylabel('1.5')
subplot(3,3,2), imshow(bup2)
```

```
subplot(3,3,3), imshow(bup3)

aup4 = imresize(b,1.7,'nearest');
aup5 = imresize(b,1.7,'bilinear');
aup6 = imresize(b,1.7,'cubic');
subplot(3,3,4), imshow(aup4)
ylabel('1.7')
subplot(3,3,5), imshow(aup5)
subplot(3,3,6), imshow(aup6)

aup7 = imresize(b,2,'nearest');
aup8 = imresize(b,2,'bilinear');
aup9 = imresize(b,2,'cubic');
subplot(3,3,7), imshow(aup7)
ylabel('2')
xlabel('Nearest')
subplot(3,3,8), imshow(aup8)
xlabel('Linear')
subplot(3,3,9), imshow(aup9)
xlabel('Cubic')

% Image c upsampling
figure
cup1 = imresize(c,1.5,'nearest');
cup2 = imresize(c,1.5,'bilinear');
cup3 = imresize(c,1.5,'cubic');
subplot(3,3,1), imshow(cup1)
ylabel('1.5')
subplot(3,3,2), imshow(cup2)
subplot(3,3,3), imshow(cup3)

cup4 = imresize(c,1.7,'nearest');
cup5 = imresize(c,1.7,'bilinear');
cup6 = imresize(c,1.7,'cubic');
subplot(3,3,4), imshow(cup4)
ylabel('1.7')
subplot(3,3,5), imshow(cup5)
subplot(3,3,6), imshow(cup6)

cup7 = imresize(c,2,'nearest');
cup8 = imresize(c,2,'bilinear');
cup9 = imresize(c,2,'cubic');
subplot(3,3,7), imshow(cup7)
ylabel('2')
xlabel('Nearest')
subplot(3,3,8), imshow(cup8)
xlabel('Linear')
subplot(3,3,9), imshow(cup9)
xlabel('Cubic')

fprintf('The result is quite close to answer(iii). By using Nearest
 Neighbor Interpolation, pixels are clear and rough.')
fprintf('Linear Interpolation show much better result, which is
 smoother at curves and boundaies.')
```

```
fprintf('Cubic Interpolation is the best one among these three, a
 little better than the Linear method.\n')

% (v)
fprintf('\nAnswer(v)\n')
% Image a
figure
aa1 = imresize(a, 0.1,'Nearest');
aa1 = imresize(aa1, 10, 'Nearest');
subplot(3,3,1)
imshow(aa1)
ylabel('Downsampling:Nearest')

aa2 = imresize(a, 0.1,'Nearest');
aa2 = imresize(aa2, 10, 'Bilinear');
subplot(3,3,2)
imshow(aa2)

aa3 = imresize(a, 0.1,'Nearest');
aa3 = imresize(aa3, 10, 'Cubic');
subplot(3,3,3)
imshow(aa3)

aa4 = imresize(a, 0.1,'Bilinear');
aa4 = imresize(aa4, 10, 'Nearest');
subplot(3,3,4)
imshow(aa4)
ylabel('Downsampling:Linear')

aa5 = imresize(a, 0.1,'Bilinear');
aa5 = imresize(aa5, 10, 'Bilinear');
subplot(3,3,5)
imshow(aa5)

aa6 = imresize(a, 0.1,'Bilinear');
aa6 = imresize(aa6, 10, 'Cubic');
subplot(3,3,6)
imshow(aa6)

aa7 = imresize(a, 0.1,'Cubic');
aa7 = imresize(aa7, 10, 'Nearest');
subplot(3,3,7)
imshow(aa7)
ylabel('Downsampling:Cubic')
xlabel('Upsampling:Nearest')

aa8 = imresize(a, 0.1,'Cubic');
aa8 = imresize(aa8, 10, 'Bilinear');
subplot(3,3,8)
imshow(aa8)
xlabel('Upsampling:Linear')

aa9 = imresize(a, 0.1,'Cubic');
aa9 = imresize(aa9, 10, 'Cubic');
```

```
subplot(3,3,9)
imshow(aa9)
xlabel('Upsampling:Cubic')

% Image b
figure
bb1 = imresize(b, 0.1,'Nearest');
bb1 = imresize(bb1, 10, 'Nearest');
subplot(3,3,1)
imshow(bb1)
ylabel('Downsampling:Nearest')

bb2 = imresize(b, 0.1,'Nearest');
bb2 = imresize(bb2, 10, 'Bilinear');
subplot(3,3,2)
imshow(bb2)

bb3 = imresize(b, 0.1,'Nearest');
bb3 = imresize(bb3, 10, 'Cubic');
subplot(3,3,3)
imshow(bb3)

bb4 = imresize(b, 0.1,'Bilinear');
bb4 = imresize(bb4, 10, 'Nearest');
subplot(3,3,4)
imshow(bb4)
ylabel('Downsampling:Linear')

bb5 = imresize(b, 0.1,'Bilinear');
bb5 = imresize(bb5, 10, 'Bilinear');
subplot(3,3,5)
imshow(bb5)

bb6 = imresize(b, 0.1,'Bilinear');
bb6 = imresize(bb6, 10, 'Cubic');
subplot(3,3,6)
imshow(bb6)

bb7 = imresize(b, 0.1,'Cubic');
bb7 = imresize(bb7, 10, 'Nearest');
subplot(3,3,7)
imshow(bb7)
ylabel('Downsampling:Cubic')
xlabel('Upsampling:Nearest')

bb8 = imresize(b, 0.1,'Cubic');
bb8 = imresize(bb8, 10, 'Bilinear');
subplot(3,3,8)
imshow(bb8)
xlabel('Upsampling:Linear')


bb9 = imresize(b, 0.1,'Cubic');
bb9 = imresize(bb9, 10, 'Cubic');
```

```
subplot(3,3,9)
imshow(bb9)
xlabel('Upsampling:Cubic')

% Image c
figure
cc1 = imresize(c, 0.1,'Nearest');
cc1 = imresize(cc1, 10, 'Nearest');
subplot(3,3,1)
imshow(cc1)
ylabel('Downsampling:Nearest')

cc2 = imresize(c, 0.1,'Nearest');
cc2 = imresize(cc2, 10, 'Bilinear');
subplot(3,3,2)
imshow(cc2)

cc3 = imresize(c, 0.1,'Nearest');
cc3 = imresize(cc3, 10, 'Cubic');
subplot(3,3,3)
imshow(cc3)

cc4 = imresize(c, 0.1,'Bilinear');
cc4 = imresize(cc4, 10, 'Nearest');
subplot(3,3,4)
imshow(cc4)
ylabel('Downsampling:Linear')

cc5 = imresize(c, 0.1,'Bilinear');
cc5 = imresize(cc5, 10, 'Bilinear');
subplot(3,3,5)
imshow(cc5)

cc6 = imresize(c, 0.1,'Bilinear');
cc6 = imresize(cc6, 10, 'Cubic');
subplot(3,3,6)
imshow(cc6)

cc7 = imresize(c, 0.1,'Cubic');
cc7 = imresize(cc7, 10, 'Nearest');
subplot(3,3,7)
imshow(cc7)
xlabel('Upsampling:Nearest')
ylabel('Downsampling:Cubic')

cc8 = imresize(c, 0.1,'Cubic');
cc8 = imresize(cc8, 10, 'Bilinear');
subplot(3,3,8)
imshow(cc8)
xlabel('Upsampling:Linear')

cc9 = imresize(c, 0.1,'Cubic');
cc9 = imresize(cc9, 10, 'Cubic');
subplot(3,3,9)
```

```
imshow(cc9)
xlabel('Upsampling:Cubic')

%
fprintf('After trying all the combinations, I discover that the
 cubic interpolation downsampling combining with cubic interpolation
 upsampling works best!')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Answer(i)
The interpolation methods for image sampling include:
Nearest neighbor interpolation, Linear Interpolation, Cubic Interpolation.

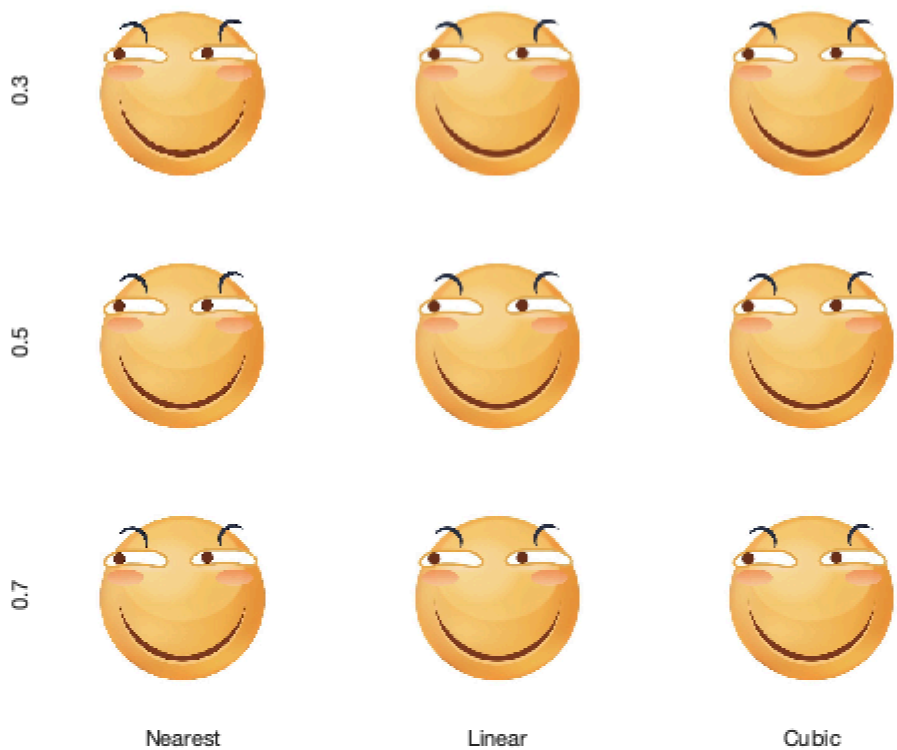
Answer(ii)
Nearest neighbor interpolation: We simply select the sample point that is nearest to x, which coule be the largest integer values that is less than or equal to x.
Linear Interpolation: Between to adjacent sample points k and k+1 we assume the function is a linear function.
Cubic Interpolation: We look at two pixels on the left and two on the right, interpolate it as cubic function.

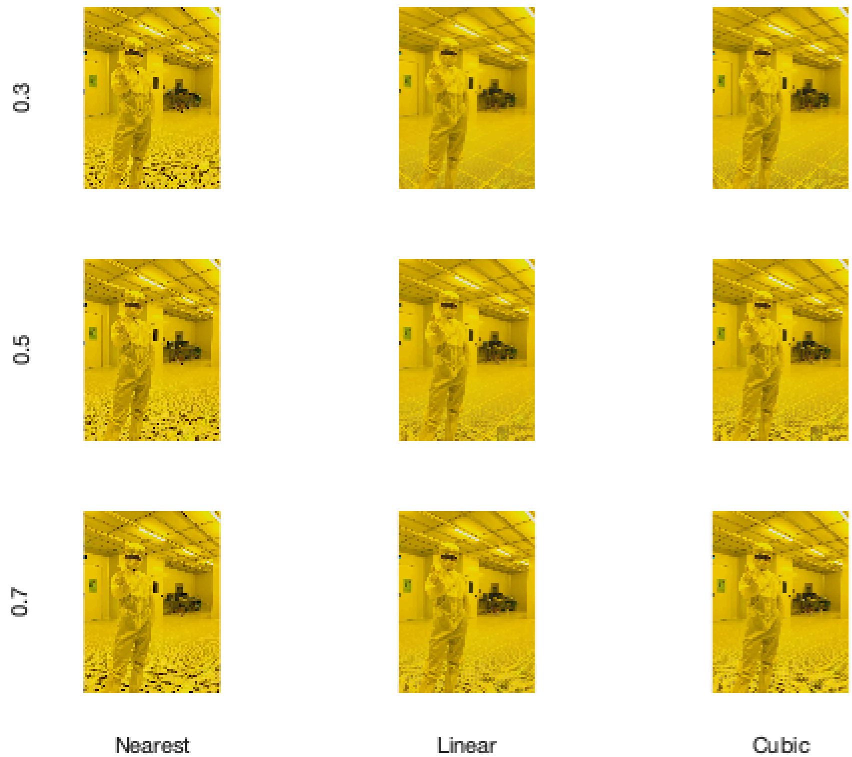
Answer(iii)
We can observe that the downsampling quality is different from three methods. Obvious pixels can be noticed by using Nearest Neighbor Interpolation.
The Cubic Interpolation is better than Linear Interpolation, follows by Nearest Neighbot Interpolation.

Answer(iv)
The result is quite close to answer(iii). By using Nearest Neighbor Interpolation, pixels are clear and rough.Linear Interpolation show much better result, which is smoother at curves and boundaies.Cubic Interpolation is the best one among these three, a little better than the Linear method.

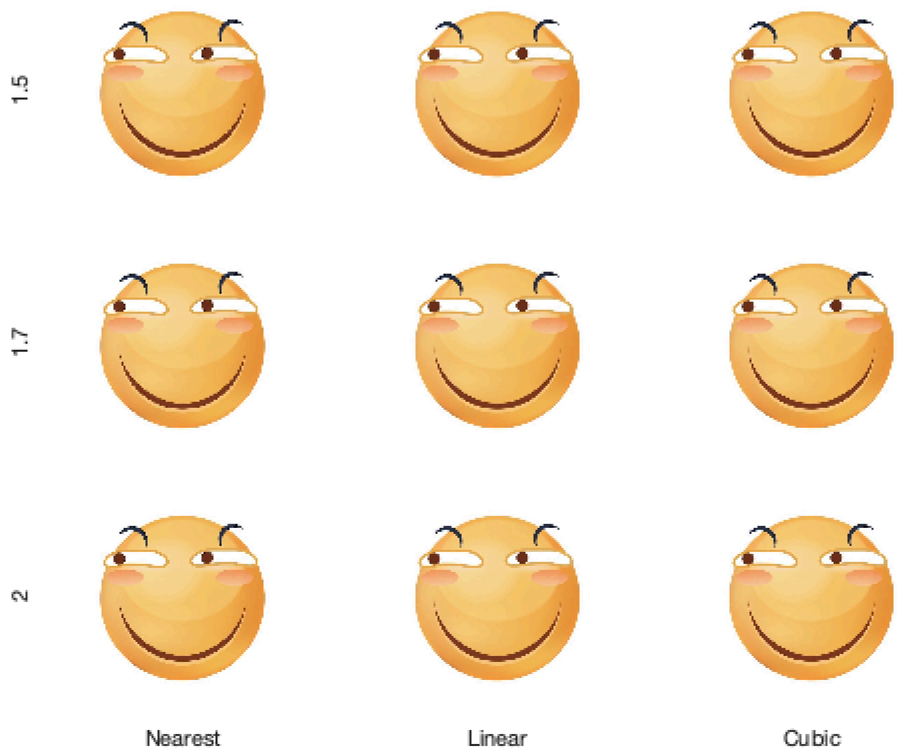
Answer(v)
After trying all the combinations, I discover that the cubic interpolation downsampling combining with cubic interpolation upsampling works best!

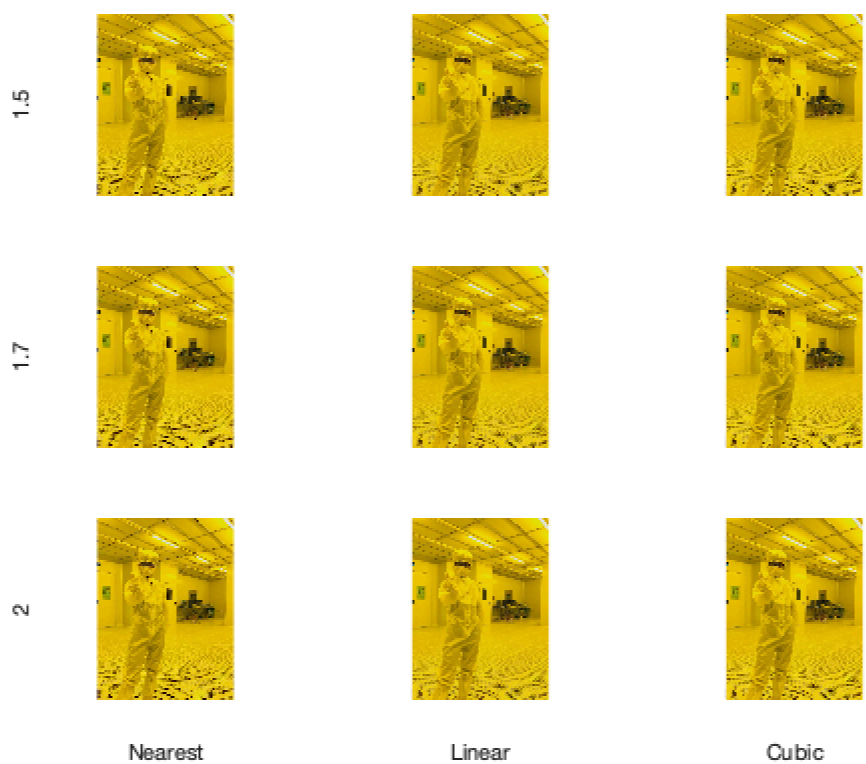

0.3
0.5
0.7
Nearest
Linear
Cubic

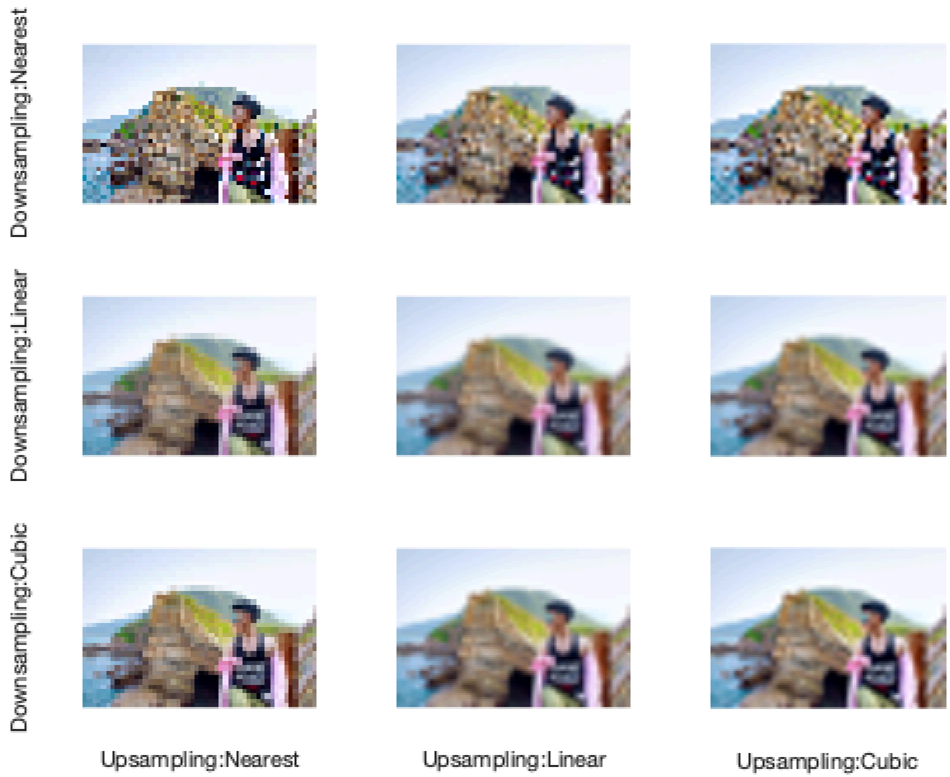

0.3
0.5
0.7
Nearest
Linear
Cubic


0.3
0.5
0.7
Nearest
Linear
Cubic


1.5
1.7
2
Nearest
Linear
Cubic


1.5
1.7
2
Nearest
Linear
Cubic


1.5
1.7
2
Nearest
Linear
Cubic


Downsampling:Nearest
Downsampling:Linear
Downsampling:Cubic
Upsampling:Nearest
Upsampling:Linear
Upsampling:Cubic


Downsampling:Nearest
Downsampling:Linear
Downsampling:Cubic
Upsampling:Nearest
Upsampling:Linear
Upsampling:Cubic

Published with MATLAB® R2018a