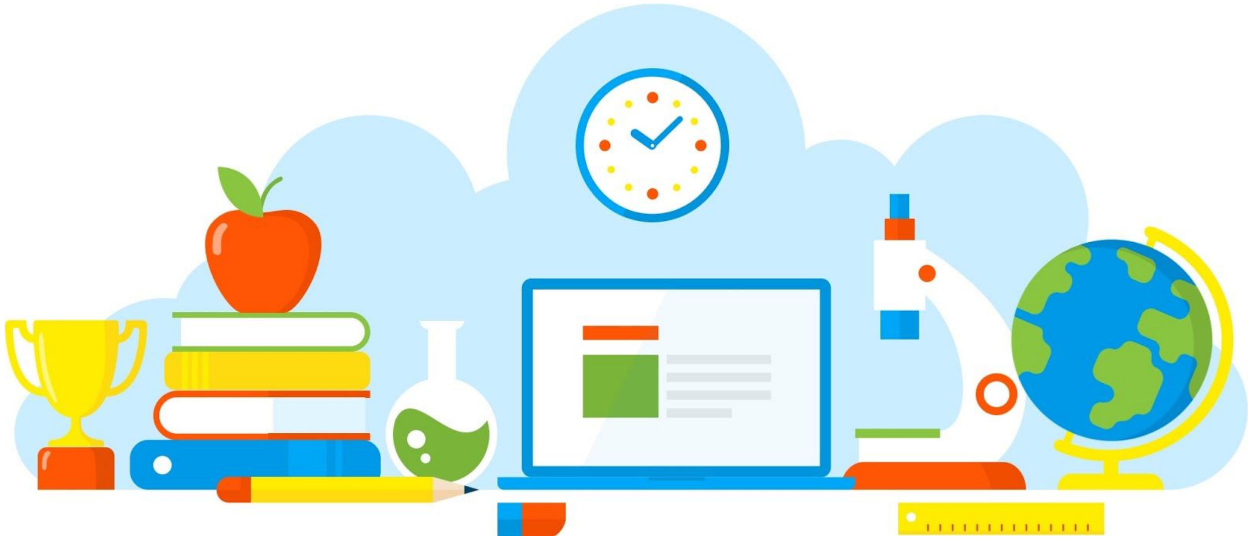


DSAA

Assignment - 1 Report
Matlab

Pradeep Yarlagadda
20161164

Question 1

The aim of this problem is to resize a given image according to the given scale factor which is a positive integer. This is done using Interpolation. Nearest Neighbor Interpolation and Bilinear Interpolation are used to solve this problem.

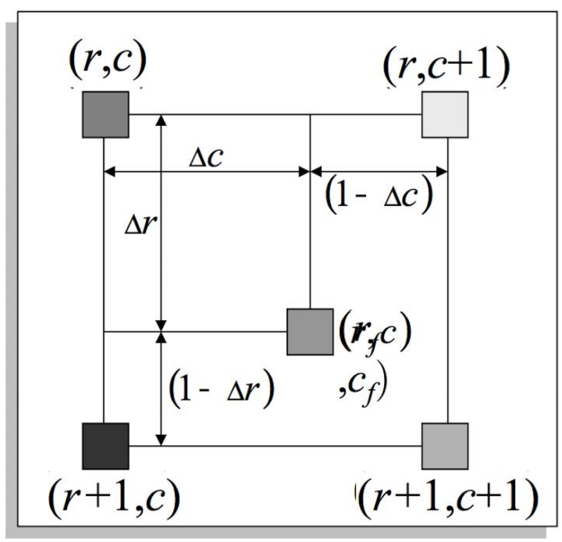
Solution

A) Nearest Neighbor Interpolation

In this method, we assign the value of the nearest pixel to the pixel in the output visualization. This is the fastest interpolation method but the resulting image may contain jagged edges.

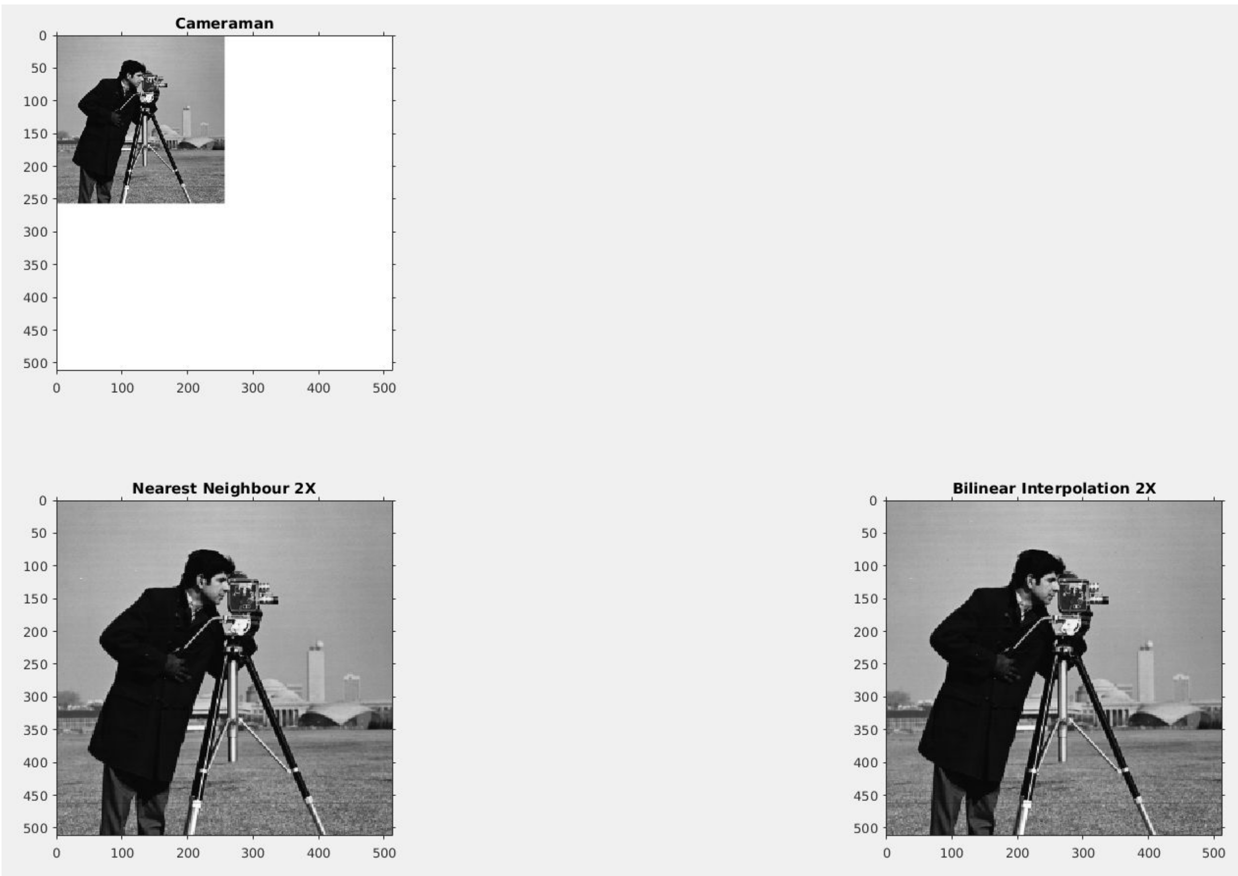
B) Bilinear Interpolation

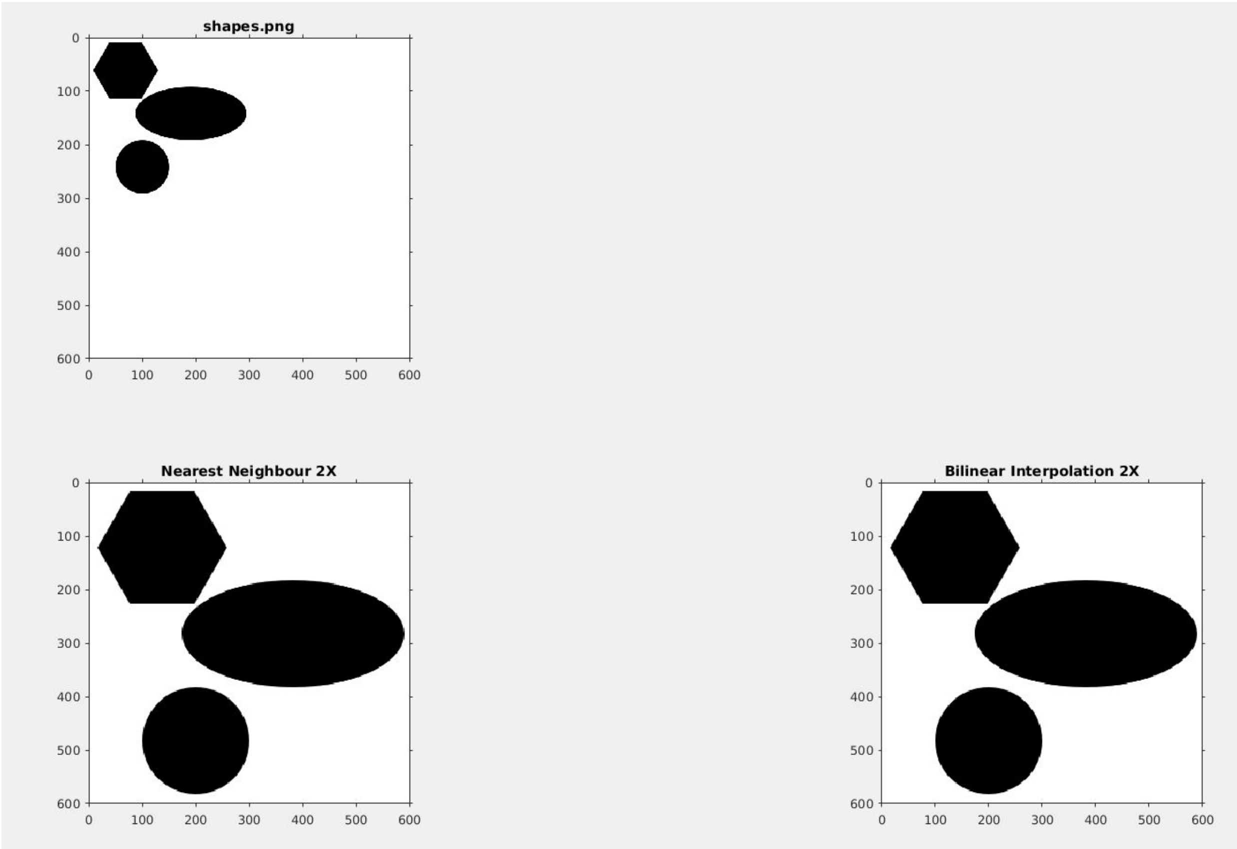
In this method, we take the weighted mean of the nearest four pixels and use it to calculate the value of output pixel.

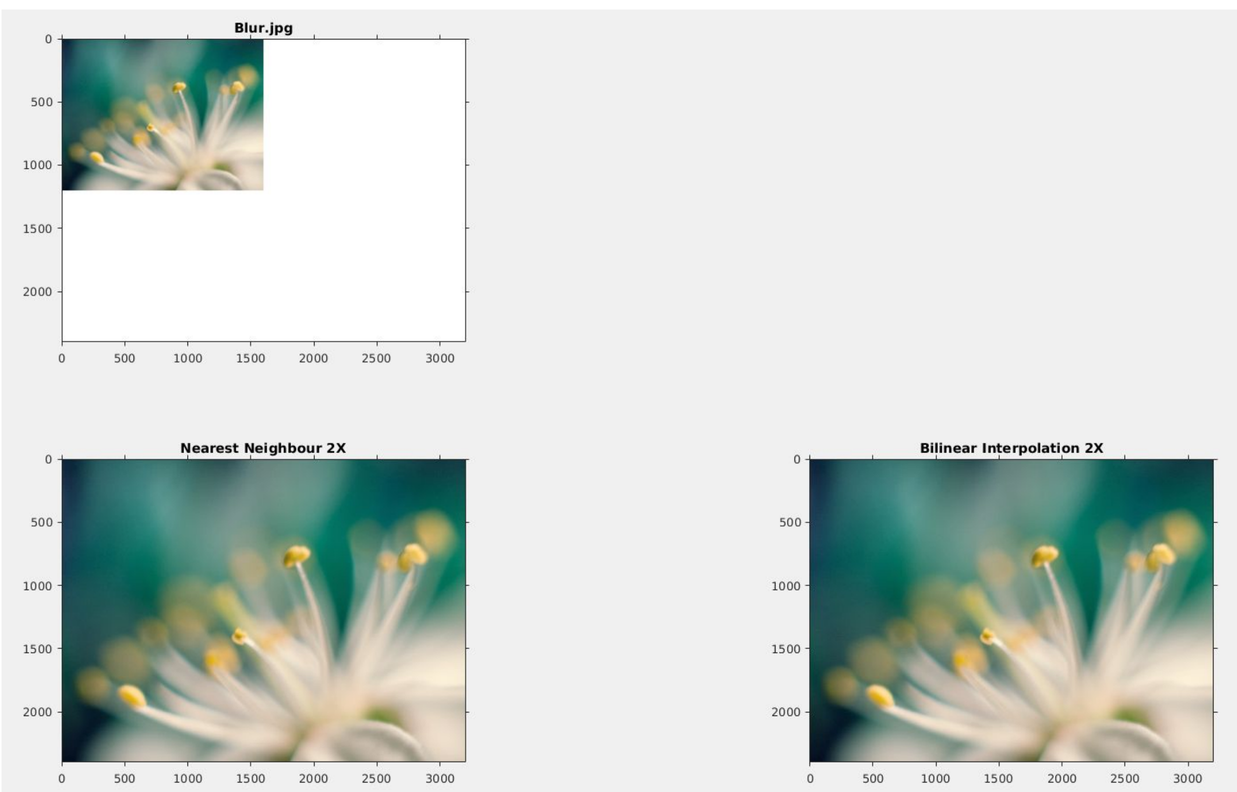


As the above image shows, for each pixel value J(r', c') the weighted mean corresponding to I(r, c), I(r, c+1) , I(r+1, c) and I(r+1, c+1) are calculated. The weighted mean is calculated first by taking the values corresponding to I(r, c), I(r+1, c) ; I(r, c+1), I(r+1, c+1) and calculating the weighted mean by using the vertical distances from required pixel. Let those values be x and y. Now the weighted mean of x and y is calculated by using the horizontal distances of x and y from required pixel and filled into it.

--------------------------images-------------------




shapes.png
Nearest Neighbour 2X
Bilinear Interpolation 2X
0
100
200
300
400
500
600


Blur.jpg
Nearest Neighbour 2X
Bilinear Interpolation 2X
0
500
1000
1500
2000
2500
3000

From the above images, we can observe that Bilinear interpolation gives images with finer quality. As the Scaling factor increases, the amount of pixelation in both the images increases. This can be observed more in the images interpolated with Nearest Neighbor approach.

The 2 different resampling methods vary on how the cell values of an output raster are determined after a geometric operation is done.

Nearest Neighbor Interpolation:

It is best used for categorical data. The values that go into the grid stay exactly the same. The value of the output cell is determined by the nearest cell center on the input grid . It preserves original values in the unaltered scene.

Disadvantages:

Nearest Neighbor when used on continuous data, produces blocky results. Noticeable position errors, especially along linear features where the realignment of pixels is obvious.

Bilinear Interpolation:

It should be used for continuous data like elevation. Unlike Nearest-neighbor, it only uses the values of nearest four pixels.

Disadvantages:

The output value could be different than the nearest input, but is always within the same range of values as the input. Since the values can change, bilinear is not recommended for categorical data. It takes more time and is more complex than nearest neighbor.

Bicubic Interpolation can be used for interpolation as it uses more values for computation of the output pixel. MATLAB uses this approach, along with other image optimization methods, when image is resized using ***imresize()*** method. Hence, it can be used.

Question 2

A brief overview of the question: We just need to convolute the given matrix with the image which is in built into MATLAB. This can easily be done using the conv2 function.

Solution

a) The given matrix is called horizontal mask of sobel operator.

When this matrix is convolved with an image, it highlights the horizontal edges of the image.As the central row is zero, the original value of edge is not retained but rather the value of edge will be the difference between upper and lower pixel values around that particular edge.This gives more weightage to the pixel values around the edge region ,thus increasing the sudden change of intensities and making the edge more visible. Also, the horizontal edge is more visible than vertical because rows have more weight than columns for a particular edge.

M = [-1 -2 -1 ; 0 0 0 ; 1 2 1]

(b) When this matrix is convolved with an image, it highlights the vertical edges of the image.As the central row is zero, the original value of edge is not retained but rather the value of edge will be the difference between right and left pixel values around that particular edge.This gives more weightage to the pixel values around the edge region, thus increasing the sudden change of intensities and making the edge more visible. Also, the vertical edge is more visible than horizontal because columns have more weight than rows for a particular edge.

It simply works like the first order derivative and calculates the difference of pixel intensities in a edge region.

M’ = [-1 0 1 ; -2 0 2 ; -1 0 1]

Question 3

An image of size (Width, Height, Channels) is convolved with N filters of size (F,F,Channels).The convolution is done with a step size of S units, and the input is also padded with a zero padding of Z. The convolution always happens in such a way that the filter is always contained in the image.

(a) Predict the dimensions of the output of this convolution.

(b) How many additions and multiplications are involved in this convolution?

Solution

Given,

Input-size	=	W x H x channels
Filter-size	=	F x F x channels
Step-size	=	S
Zero-Padding=	Z	
No.of Filters	=	N

a)After Zero padding the size of the input becomes

$$(W + 2 * Z) \times (H + 2 * Z) \times \text{channels}$$

let k1,k2 be +ve positive numbers

$$k1 * S + F = W + 2 * Z$$

$$k1 = \text{floor}((W+2*Z-F)/S)$$

Similarly

$$k2*S + F = H + 2*Z$$

$$k2 = \text{floor}((H+2*Z-F)/S)$$

Therefore,After applying one filter the dimensions become

$$(k1+1)x(k2+1)x\text{channels}$$

for width

$$W' = \text{floor}((W+2*Z)/S-F/S+1)$$

$$W' = \text{floor}((W+2*Z)/S + 1-(F/S))$$

let

$$A = 1-(F/S)$$

$W' = floor((W+2*Z)/S)+A$

Now for the same 2nd filter

$W'' = floor(W'/S)+A$

$W'' = ((W+2*Z)/S + A)/S + A$

$W'' = (W+2*Z)/S^2 + A/S + A$

Similarly then After the n^{th} filter applied

$W^n = (W+2*Z)/S^n + A/S^{n-1} + A/S^{n-2} + ... + A$

Which is in GP sequence

$W^n = (W+2*Z)/S^n + (A/S^{n-1})*((S^n-1)/(S-1))$

$W^n = floor((W+2*Z)/S^n + (A/S^{n-1})*((S^n-1)/(S-1)))$ where $A = 1-(F/S)$

Similarly for the Height also

$H^n =$ $floor((W+2*Z)/S^n + (A/S^{n-1})*((S^n-1)/(S-1)))$ where $A = 1-(F/S)$

The no.of channels remain same through the process therefore the final output dimensions are

W^n x H^n x channels

B) Number of Multiplications and Additions:

inp-size (W+2*Z) x (H+2*Z) x channels {After padding}

⇒ {W'*H'*(F*F)*channels} multiplications

After 1st Fil W 'x H' x channels

⇒ {W''*H''*(F*F)*channels} multiplications

After 2nd Fil W''x H''x channels

⇒ {W^n*H^{n*}(F*F)*channels} multiplications

After nth Fil W^n x H^n x channels

Therefore Total No.of multiplications are:

(W'*H' + W''*H'' + ... + W^n*H^n)*(F*F)*channels

Similarly, Total No.of Additions are

(W'*H' + W''*H'' + ... + W^n*H^n)*(F*F-1)*channels

Question 4

A brief overview of the question: (a) Reading a wav file and recording our own voice. (b) Subsampling to certain frequencies (c) Simulating it in different environments.

Solution

(a) The first part, which is to first sample our own audio file was implemented using the MATLAB' s audio recorder object. Next we used the getaudiodata with which we can get the get the signal as a matrix with dimension one.

(b) Now for the subsampling part, let us say we need to subsample it to 8kHz. Now the input was of 44.1 KHz frequency. So we need to take 8000 data points from the array of 44100 data points which were already obtained in part 'a'. Implementation was done by sampling out data points at equal intervals starting from index one. The interval would be Input Freq divided by the Output freq.

(c) For the third part, we just convolute the external wav files which are read by audio read function with the initial sampled matrix. Also, the frequencies were made same by resampling.

Question 5

Create a matrix of size 3x3 which when convolved with input.png results in a white line where the white meets the black (sample out.png). Now do the following and give report on what you

See:

(a) Convolve blur.jpg with the matrix you designed previously.

(b) Convolve blur.jpg with the transpose of matrix you designed previously.

(c) Add the images obtained from (a) and (b) after convolution with blur.jpg.

(d) Take any other 3 images instead of blur.jpg and follow steps (a),(b) and (c).

Solution

The filter should detect a horizontal edge in the image.

For this purpose we can use:

M = [1 2 1 ; 0 0 0 ; -1 -2 -1]

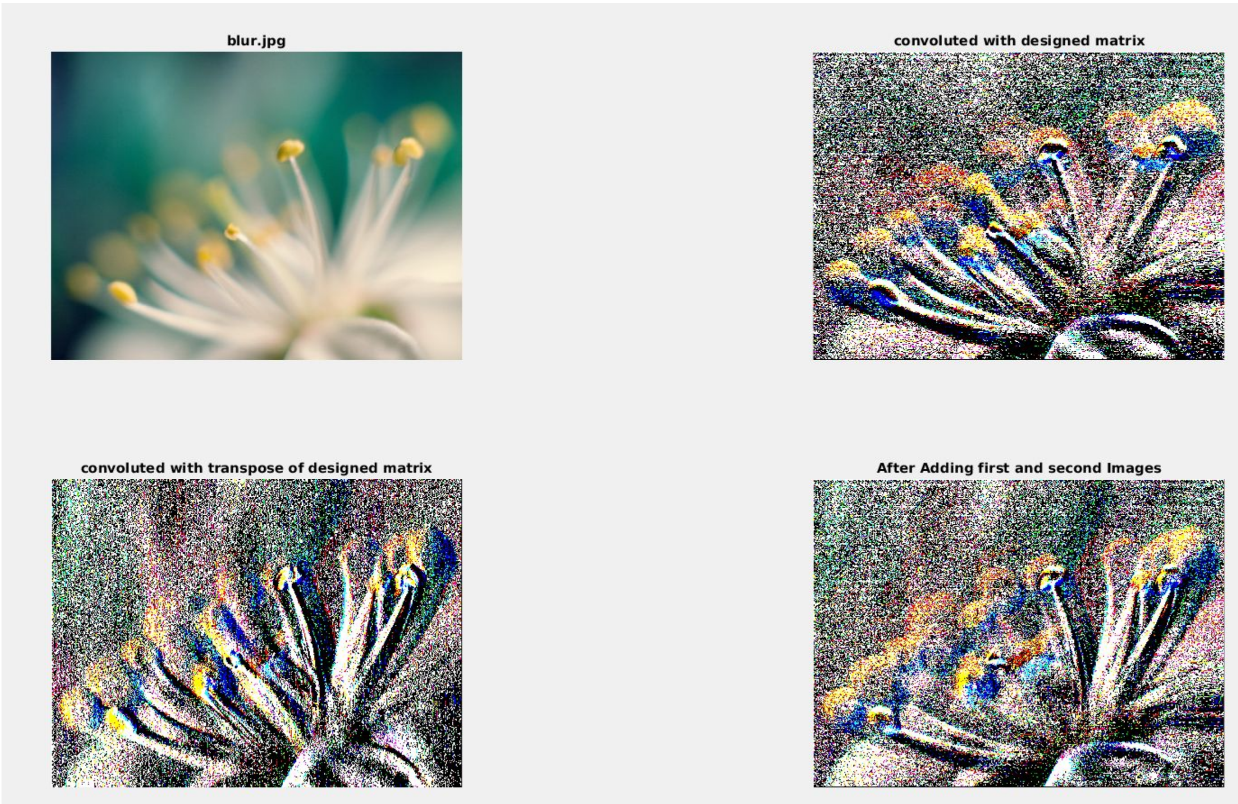
It is similar to the **Sobel edge detecting filters**.

If we do convolution on “sample_inp.png” with the above filter we get the following image.

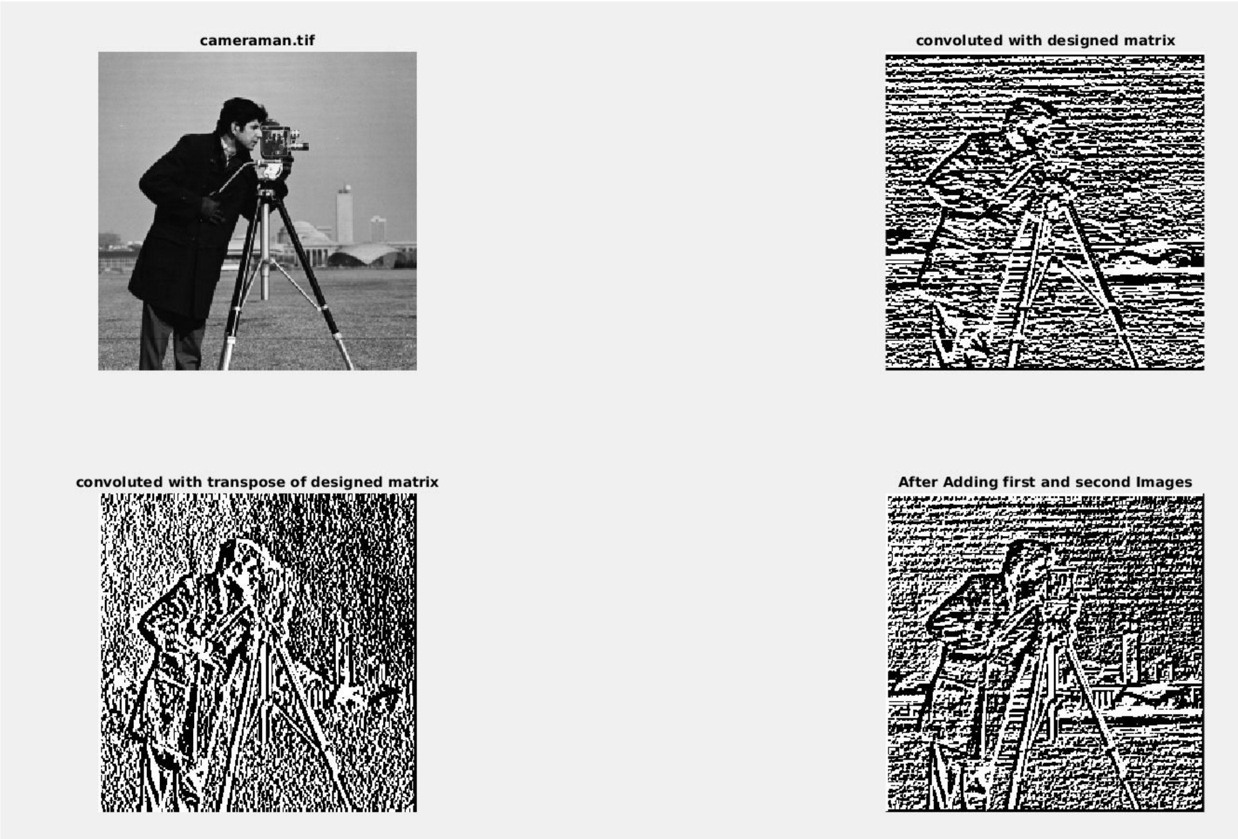
←--img---->

It **detects edges in image in Horizontal direction**. So, if we apply the same filter on blur.jpg Here are the following results:

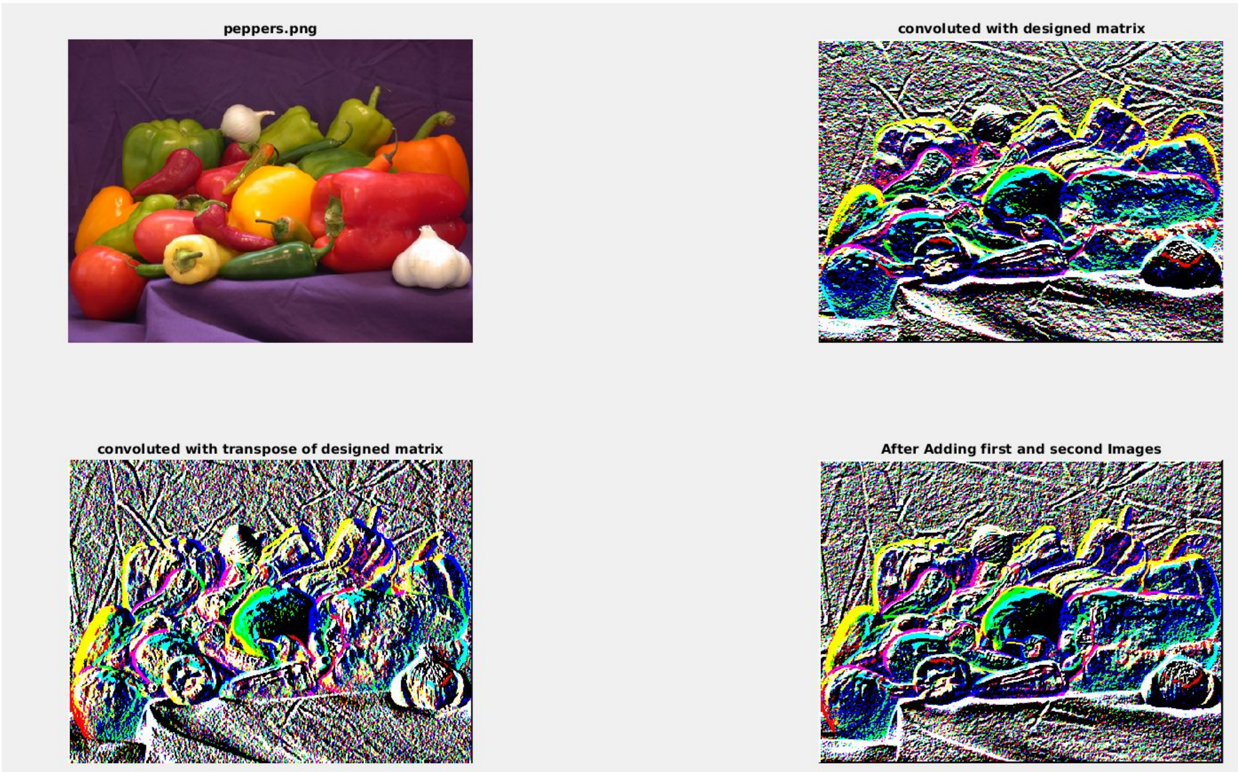
Applying filter's on blur.jpg



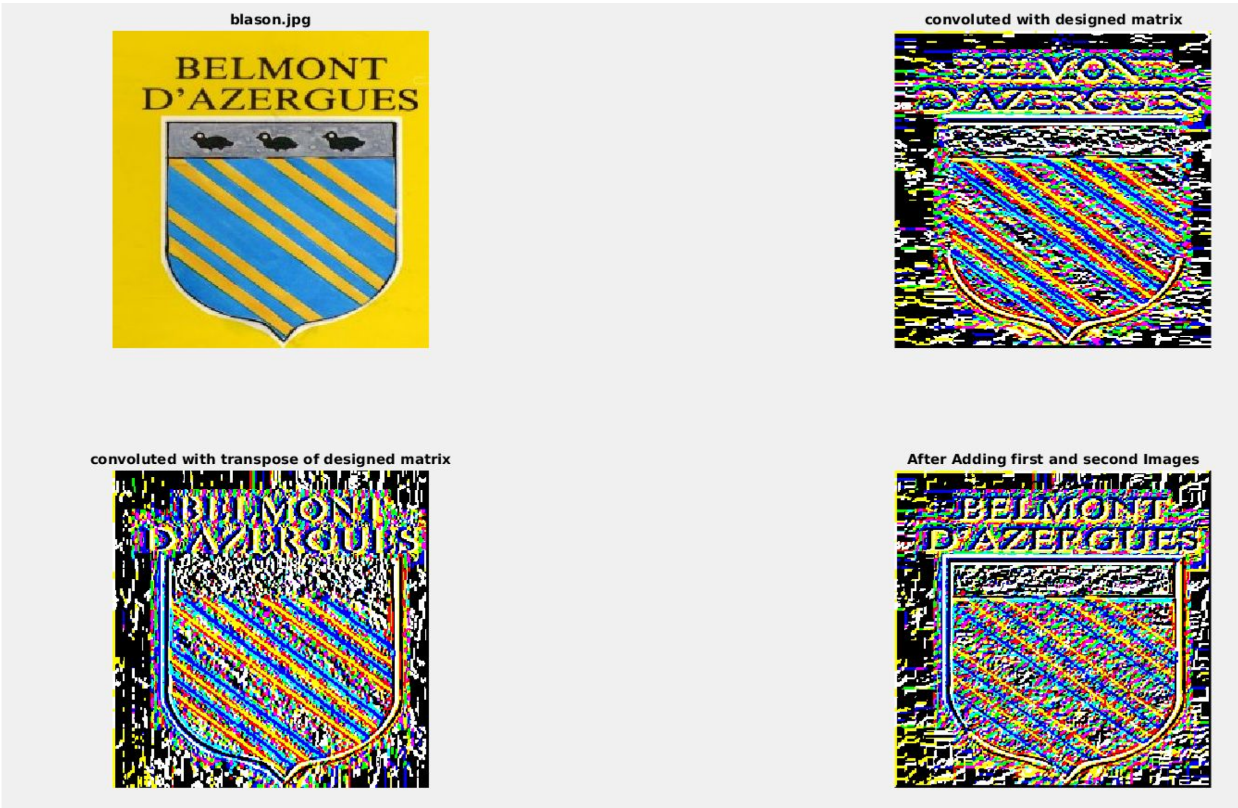
Applying filter's on cameraman.tif



Applying filter's on pepper.png



Applying filter's on blason.png



Question 6

A brief overview of the Question: We need to detect an image within another image both of which are present in the same directory.

Solution

Approach Followed: Consider Normalized correlation of the two matrices. Let the output be denoted by O.

In this, we are trying to find that point in O such that the value is 1. Now this point corresponds to a particular sub matrix in the original matrix as the

value obtained is by sliding the filter across it. Now the claim is that this sub matrix is the answer. Let this sub matrix be M'.

Proof: Let us say M' is the right answer, that is it is the image to be detected. In that case the value would be:

sum(Element-wise dot products) / sqrt(sumsqr(M')*sumsqr(Filter))

Sumsqr denotes the sum of squares of all the elements of the corresponding matrix.

That values boils down to 1 after simple algebraic manipulations. All other values are less than one as it is normalized. So we then use MATLAB's find function to find for that value and then we draw a rectangle around it using the MATLAB' s imrect function.

Question 7

Convolution filters on 1-D audio signals are used to achieve certain sound effects in sound processors for orchestras and musical performances. Figuring out the impulse response of a particular instrument is an experimental problem, since these impulse responses are often infinite in nature and cannot be calculated using traditional deconvolutional

methods.

Consider the following example problem where you are given an input signal and an output signal. The output has been generated by convolving the given input signal with a finite impulse response filter. Assuming the filter to be causal in nature, find it. The convolution applied is valid. Only values calculated without zero padding are included in the result. Your code should output the filter coefficients rounded to the nearest integer. (Nearest integer, neither rounded up nor down).

Input = [12, 20, 3, 10, 22, 19, 23, 16, 0, 21, 23, 16, 18];

Output = [75, 52, 33, 97, 251, 211, 63, 65];

Solution

Given the filter is casual in nature. So, it uses only the previous values for computing future values. Also there is not zero padding to the input and convolution is valid.

Size of A(input) = 13

Size of B(output) = 8

Size of Filter = sizeof(A) - sizeof(B) + 1

Therefore size of filter is 6.

Let flipped version of the filter be [a , b , c , d , e , f]

Since the output signal contains 8 values. We can get 8 equations out of which 6 equations can be used to solve 6 equations.

The 6 equations are as follows:

a*12+b*20+c*3+d*10+22*e+f*19 = 75

a*20+b*3+c*10+d*22+e*19+f*23 = 52

a*3 +b*10+c*22+d*19+e*23+f*16 = 33

a*10+b*22+c*19+d*23+e*16+f*0 = 97