**ASSIGNMENT-2 REPORT**

Ques1:

1. ASSUMPTIONS:: n is odd(helps in symmetry)

Read the image.

Create gaussian filters(both inbuilt and self implemented).

Apply filters to the images and show them.

Gauss function:

Create a vector from -n/2 to n/2(eg::-2,-1,0,1,2).

Create a meshgrid from x(meshgrid avoids for loops for creating matrix)

Create gaussian matrix and then divide every term by sum to normalize it.

(Increasing sigma decreases blur, larger n destroys some details)

1. ASSUMPTIONS::n is odd

Read image.

Create filters and apply to the images and show them.

Median function:

Pad the image.

Take median of each block of nxn dimensions.

Replace each entry with the median.

Use of im2col and col2im for vectorizing the code.

(Larger n reduces sharpness as take median of larger area)

1. Third part done along with first and second part of the question.

Used imfilter for applying filters.

1. Since the noise is scattered, only peaks of noise would be present.

For such cases we use median filter as it puts spikes at extreme corner and thus avoids it.

1. Take fft of the image and get the frequency domain.

Apply the filter and take ifft to gt back to the time domain.

Filter:

Since we get noise(dots) in the shape of a rectangle we create a rectangular filter which

Removes everything in the frequency domain except for the centre region.

Ques2:

1. We have to apply N filters and convolution is done with step size of S.

Let the input be zero padded with padding of Z every time.

New dimensions will be [W+2Z,H+2Z,C].

Output width:W-F+2Z)/S+1

Output height:H-F+2Z)/S+1

We will do this recursively.

Let the output after (i-1)th convolution be W(i-1) and H(i-1).

Then W(i)=(W(i-1)-F+2Z)/S+1

Then H(i)=(H(i-1)-F+2Z)/S+1

1. For one convolution(in one block):

Additions=F\*F-1

Multiplications=F\*F

Total=2\*F\*F-1

For whole image total operations=(2\*F\*F-1)\*W\*H\*C

Total operations=∑(I from 0 to N-1) (Wi\*Hi)\*(2\*F\*F-1)\*C­­

Ques3:

1. ASSUMPTIONS(continuous signal without any pause)

Read the signals and the file.

Preprocess the audios and concatenate them into a single vector.

Take a sample of data and do dot product with the audio files and select the highest value digit.

Display the result.

Ques4:

1. In frequency domain we remove the noise by applying certain filters.

So by hit and trial, I noticed that noise had low value so if I removed all values which are less than 2\*mean, the signal got cleared of noise.

Ques5: If image is not a power of 2 then we have to just change the terminal condition and add a few

lines of code more to handle it.

1. ASSUMPTIONS(n is power of 2)

Read the image.

Apply self fft.

Apply inbuilt fft.

Note their time and plot them.

Algorithm:

Tdfft takes the image and performs two odff to get the desired output.

Next we implement recursive odfft in which we divide the signal into even and odd part.

Then we calculate X matrix and then get the final signal by concatenating them.

1. ASSUMPTIONS(n is power of 2)

Same as above but calculated W matrix to get the answer in one go

(TIME:inbuilt fft>self fft>self dft)

Ques6:

1. The image obtained after two fft’s is flipped in both x and y and is lighter than the original

Image.

To fix it frequency domain:

Flip in x and y dimensions.

Make it a liitle dark.

Explanations::

So when we take FFT twice the signal gets circularly flipped.

So it gets flipped in both x and y axis.

Also when we do FFT some constants are multiplied which change the intensity of the image and

make it look dark or bright from the original image.

Ques7:

1. Store starting points of each window and then take stft of each window

Plot the stft as a spectrogram.

1. Form spectrogram of the image
2. Create two vectors corresponding to two frequencies.

For each digit find corresponding frequencies and create signals

Add them and add pause.

Keep on concatenating the signal to get desired tone.