Exercise 1 Basis Function

March 7, 2018

Xiangyi Cheng (xxc283)

1 Concept and Background

A discrete consine transform (DCT) shows a finite sequence of data points in terms of a sum of cosine functions osillating at different frequency. Specifically, DCT transforms the information of images in spatial space to frequential space in computer vision which is quite similar to Fourier transform, while DCT is only using for real numbers.

For the basis functions of a MxM dimension DCT, MxM functions should be involved. Each of them is one basis function whose dimensions are still MxM. If we are explaning the values of each unit with u,v,i,j where u and v are the location of the basis function while i and j are the position of the unit in the basis function, the mathematical expression should be

$$B_{u,v}(i,j) = C(u)C(v)cos\frac{\Pi(2i+1)u}{2M}cos\frac{\Pi(2j+1)v}{2N}$$

2 Implement

Of course, necessary libraries and packages should be added to start coding.

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import math
```

To avoid multiple loops in the main function, a function for plotting each basis function was defined in advanced. Each value should follow the formula showed above.

```
In [2]: def dctbasis(u,v,m,n):
                # set the constant value of cu and cv
                if u==0:
                        cu=1.0/math.sqrt(m)
                else:
                        cu=math.sqrt(2.0/m)

                if v==0:
                        cv=1.0/math.sqrt(n)
                else:
                        cv=math.sqrt(2.0/n)
```

```
    # initialize the matrix of each basis
    dctb=np.zeros(shape=(m,n))

    # fill values in each basis
    for i in range(0,m): # m times iteration
            for j in range(0,n): # n times iteration
                    dctb[i,j]=cu*cv*math.cos(math.pi*(2*i+1)*u/(2*m))*math.cos(math.

    return dctb
```

After having the defined function for plotting each basis function, our main function is ready to be come up with. Although the exercise pointed out that the dimension is 16x16, I made a generally used function for all dimenisons. Therefore, it requires the users to input the dimensions first. We can plot the 8x8 basis functions for testing.

```
In [18]: m=input("what is the value of dimension m?")
         n=input("what is the value of dimension n?")
```

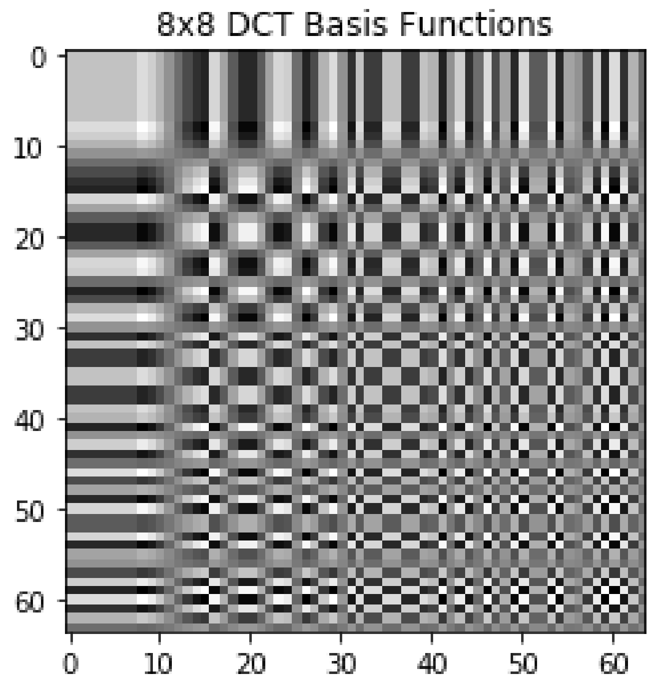
```
what is the value of dimension m?8
what is the value of dimension n?8
```

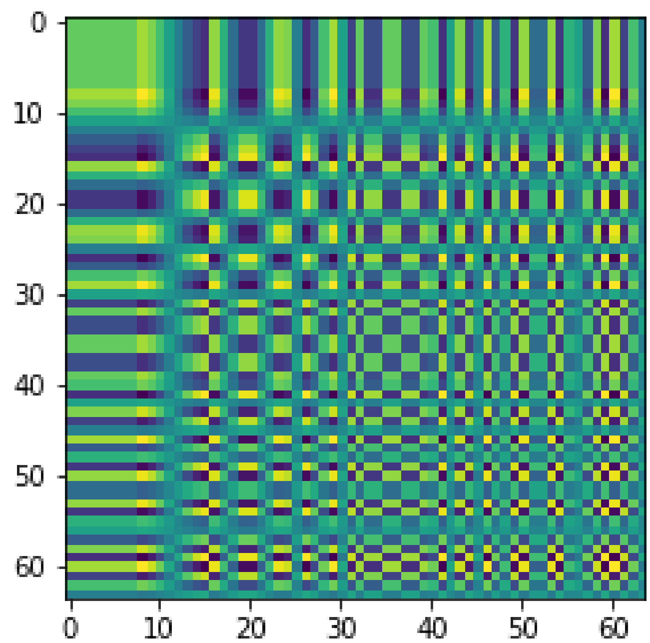
Then the aim is to have a loop to calculate the values for each basis and fill all the values into one matrix. The dimension of this matrix should be the square value of the dimensions for each basis. To achieve the goal, I generated a zero matrix first and then fill each basis into the big matrix by running the defined functions in each position in 16x16 times.

```
In [19]: u=m
         v=n

         basis=np.zeros(shape=(m*n,m*n))

         for u in range(0,m):
                 for v in range(0,n):
                         onebasis=dctbasis(u,v,m,n)
                         basis[u*m:(u+1)*m,v*n:(v+1)*n]=onebasis[:,:]
```

At this point, the matrix called basis in the code is the basis functions concluded all the basis. So it is time to plot it out with black and white, full color, respectively.

```
In [20]: plt.imshow(basis,cmap='gist_gray')
         plt.title ('8x8 DCT Basis Functions')
         plt.show()
         plt.imshow(basis)
         plt.show()
```



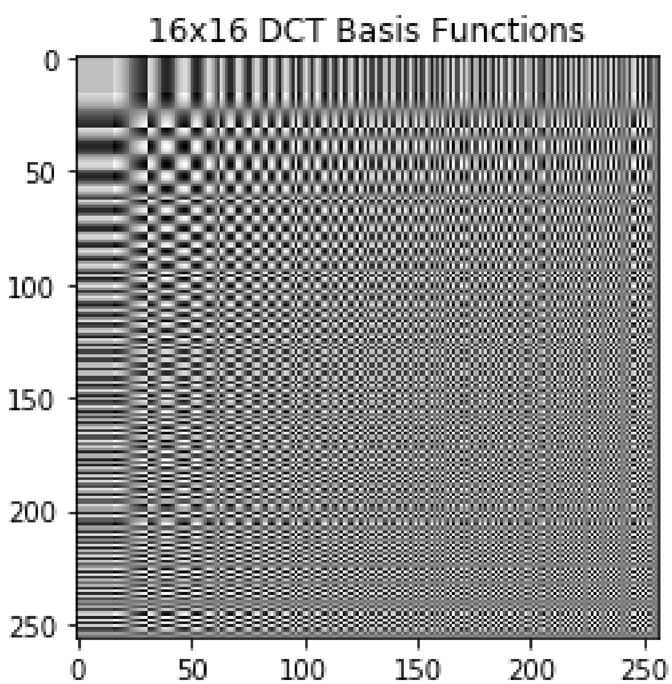


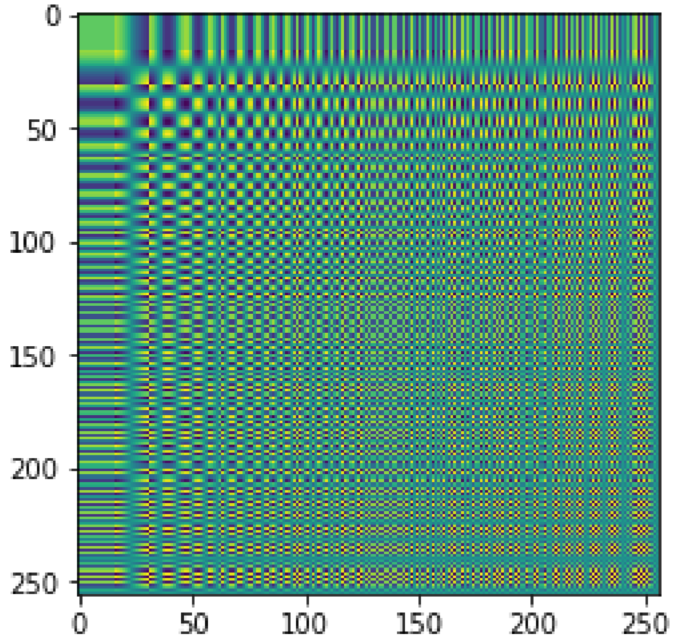
Once passing the test, the function is ready to plot 16x16 DCT basis functions. The only modification is changing the inputted dimension values.

```
In [21]: m=input("what is the value of dimension m?")
         n=input("what is the value of dimension n?")
```

```
what is the value of dimension m?16
what is the value of dimension n?16
```

```
In [22]: u=m
         v=n

         basis=np.zeros(shape=(m*n,m*n))

         for u in range(0,m):
                 for v in range(0,n):
                         onebasis=dctbasis(u,v,m,n)
                         basis[u*m:(u+1)*m,v*n:(v+1)*n]=onebasis[:,:]


         #basis=ListedColormap(colormap.colors[::-1])
         plt.imshow(basis,cmap='gist_gray')
         plt.title('16x16 DCT Basis Functions')
         plt.show()
         plt.imshow(basis)
         plt.show()
```





3 Conclusion and Discussion

This exercise aims to plot the basis functions of a 16x16 discrete cosine transform (DCT). To have a better understanding of the concept and the knowledge behind DCT, a generally used function was developed. The function consists of a self-defined function for calculating every unit inside each basis and a main function to fill every basis in and plot it. Then input 8x8 as the dimensions to plot the basis functions for testing. In the end, required 16x16 was applied to show the final results.

From the results, we can tell that the frequency is changing with the basis. The frequency is increasing by comparing each row and each column seperately. One of the most obvious differences of frequency is obtained by comparing the first basis with the last one. Lowest frequency is shown in the first basis while higher frequency exists in both horizontal and vertical directions. Once getting the basis functions, it is easy to apply low pass and high pass filter to the images.