```
#!pip install ipympl
```

Setup

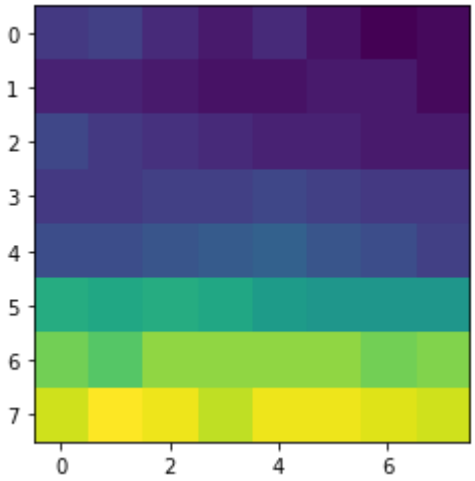
```
import numpy as np
import scipy.fftpack as fftpack
import matplotlib.pyplot as plt
import os
from google.colab import output
output.enable_custom_widget_manager()
```

```
#%matplotlib widget
```

```
def dct2(block):
    return fftpack.dct(fftpack.dct(block.T, norm="ortho").T, norm="ortho")

# implement 2D IDCT
def idct2(a):
    return fftpack.idct(fftpack.idct(a.T, norm="ortho").T, norm="ortho")

def compute_psnr(original: np.ndarray, compressed: np.ndarray):
    mse = np.mean((original.astype(np.float64) - compressed.astype(np.float64)) ** 2)
    if mse == 0:  # MSE is zero means no noise is present in the signal .
        # Therefore PSNR have no importance.
        return 100
    max_pixel = 255.0
    psnr = 20 * np.log10(max_pixel) - 10 * np.log10(mse)
    return psnr
```

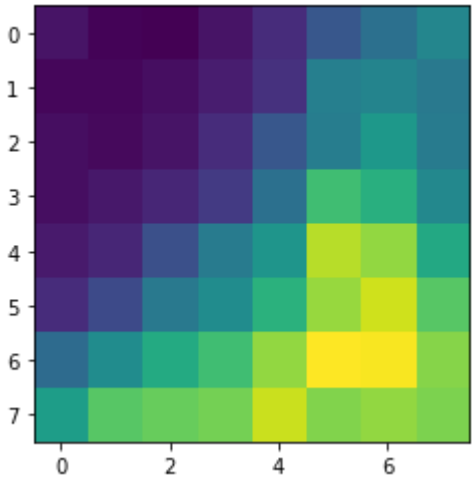
```
def JPEG_BPP(width, height, bits):
  return bits/(width* height)

```

```
data_block = np.array([
    [124,121,126,124,127,143,150,156],
    [125,121,124,124,127,142,148,159],
    [122,120,123,125,128,143,152,158],
    [120,119,122,125,129,142,152,155],
    [122,119,121,126,130,140,152,158],
    [119,120,121,125,128,139,152,158],
    [117,120,120,124,127,139,150,157],
    [118,118,120,124,125,139,151,156]
    ]).T

plt.imshow(data_block)
plt.show()
```



Quantization tabel

```
q_table = np.array([
        [16,12,14,14,18,24,49,72],
        [11,12,13,17,22,35,64,92],
        [10,14,16,22,37,55,78,95],
        [16,19,24,29,56,64,87,98],
        [24,26,40,51,68,81,103,112],
        [40,58,57,87,109,104,121,100],
        [51,60,69,80,103,113,120,103],
        [61,55,56,62,77,92,101,99]
]).T

plt.imshow(q_table)
plt.show()
```



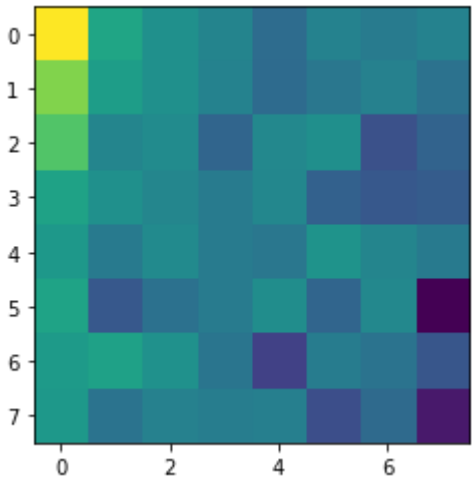
1 Still image compression

1a)

```
data_block_DCT2 = dct2(data_block)
```

```
2
3  print(data_block_DCT2)
4  np.savetxt(
5      "data_block_DCT2.csv", (data_block_DCT2), fmt="%d"
6  )  # used for tabular in latex with generator
7
8  plt.imshow(np.log(np.abs(data_block_DCT2)))
9  plt.show()
10
```

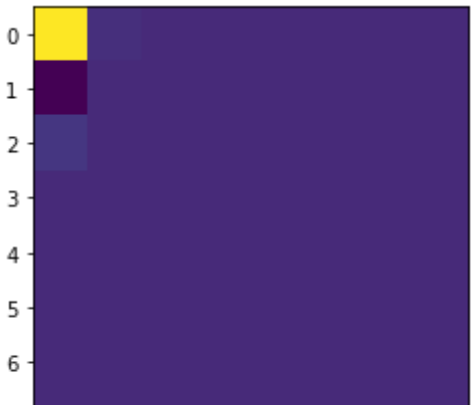
```
[[ 1.06387500e+03  6.56530796e+00 -2.24204932e+00  1.22031790e+00
  -3.75000000e-01 -1.08739329e+00  7.93388210e-01  1.13473851e+00]
 [-1.02438801e+02  4.56749149e+00  2.26372588e+00  1.12061110e+00
   3.58148693e-01 -6.33633466e-01 -1.05302724e+00 -4.80185917e-01]
 [ 3.77706166e+01  1.31440734e+00  1.77404852e+00  2.58329695e-01
  -1.50951150e+00 -2.21817548e+00 -1.00951481e-01  2.32881560e-01]
 [-5.67404043e+00  2.24214694e+00 -1.32600744e+00 -8.13211172e-01
   1.41728756e+00  2.21194688e-01 -1.39308310e-01  1.70278151e-01]
 [-3.37500000e+00 -7.45091955e-01 -1.75689591e+00  7.76371149e-01
  -6.25000000e-01 -2.65967417e+00 -1.30175526e+00  7.62049287e-01]
 [ 5.98943278e+00 -1.39948066e-01 -4.59461377e-01 -7.78805312e-01
   1.99935482e+00 -2.65215953e-01  1.46434767e+00  4.71007726e-03]
 [ 3.97325697e+00  5.52797698e+00  2.39904852e+00 -5.58772548e-01
  -5.12349883e-02 -8.47568736e-01 -5.24048519e-01 -1.30130006e-01]
 [-3.43314493e+00  5.19814083e-01 -1.07206540e+00  8.71070333e-01
   9.63382468e-01  9.02810104e-02  3.30504406e-01  1.09356359e-02]]
```



1b)

```
1 DCT2_quantized = np.floor(data_block_DCT2 / q_table + 0.5)
2 print(DCT2_quantized)
3 np.savetxt(
4     "DCT2_quantized.csv", (DCT2_quantized), fmt="%d"
5 )  # used for tabular in latex with generator
6
7 plt.imshow(DCT2_quantized)
8 plt.show()
```

```
[[66.  1.  0.  0.  0.  0.  0.  0.]
 [-9.  0.  0.  0.  0.  0.  0.  0.]
 [ 3.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.]]
```



1c)

```
dequantized = q_table * DCT2_quantized

data_block_IDCT2 = idct2(dequantized)

print(data_block_IDCT2)
plt.imshow(data_block_IDCT2)
plt.show()

np.savetxt(
    "data_block_IDCT2.csv", (data_block_IDCT2), fmt="%d"
)  # used for tabular in latex with generator

PSNR = compute_psnr(data_block, data_block_IDCT2)
print("psnr: ", PSNR)

```

```
[[122.04159744 121.75124661 121.21474822 120.5137793  119.75505601
  119.05408709 118.5175887  118.22723787]
 [120.87413868 120.58378785 120.04728946 119.34632054 118.58759725
  117.88662833 117.35012994 117.05977911]
 [120.45901832 120.16866749 119.6321691  118.93120018 118.17247689
  117.47150797 116.93500958 116.64465875]
 [123.3230826  123.03273177 122.49623338 121.79526446 121.03654117
  120.33557225 119.79907386 119.50872303]
 [130.77236584 130.48201501 129.94551662 129.2445477  128.48582441
  127.78485549 127.2483571  126.95800627]
 [141.6727822  141.38243137 140.84593298 140.14496406 139.38624077
  138.68527184 138.14877346 137.85842263]
 [152.62277995 152.33242911 151.79593073 151.0949618  150.33623851
  149.63526959 149.09877121 148.80842037]
 [159.49167326 159.20132243 158.66482405 157.96385512 157.20513183
```

4 JPEG and JPEG2000

4d)

```
im_list_bike_pgm_jp2 = ['bike_jp2_005bpp.pgm','bike_jp2_02bpp.pgm','bike_jp2_04bpp.pg
im_list_bike_pgm_jp2_decoded_bbp = ['bike_jp2_0.1976bpp.pgm','bike_jp2_0.4543bpp.pgm'

im_list_cafe_pgm_jp2 = ['cafe_jp2_005bpp.pgm','cafe_jp2_02bpp.pgm','cafe_jp2_04bpp.pg
im_list_cafe_pgm_jp2_decoded_bbp = ['cafe_jp2_0296bpp.pgm','cafe_jp2_073bpp.pgm','caf

im_list_woman_pgm_jp2_decoded_bbp = ['woman_jp2_02bpp.pgm','woman_jp2_04bpp.pgm','wom

im_list_bike_pgm_jpeg_decoded = ['bike_jpeg_decoded_N5.pgm','bike_jpeg_decoded_N20.pg
im_list_cafe_pgm_jpeg_decoded = ['cafe_jpeg_decoded_N5.pgm','cafe_jpeg_decoded_N20.pg
im_list_woman_pgm_jpeg_decoded = ['woman_jpeg_decoded_N5.pgm','woman_jpeg_decoded_N20

im_list_bike_jp2 = ['bike_005bpp.jp2','bike_02bpp.jp2','bike_04bpp.jp2','bike_06bpp.j
im_list_cafe_jp2 = ['cafe_005bpp.jp2','cafe_02bpp.jp2','cafe_04bpp.jp2','cafe_06bpp.j

im_list_bike_jpg = ['bike_N5.jpg','bike_N20.jpg','bike_N40.jpg','bike_N60.jpg','bike_
im_list_cafe_jpg = ['cafe_N5.jpg','cafe_N20.jpg','cafe_N40.jpg','cafe_N60.jpg','cafe_
im_list_woman_jpg = ['woman_N5.jpg','woman_N20.jpg','woman_N40.jpg','woman_N60.jpg','
```

```
def showImage(im_list):
  for im in im_list:
    title = im
    f = open(im,'rb')
    im = plt.imread(f)
    plt.title(title)
    plt.imshow(im, cmap='gray')
    plt.show()
```

```
if False:
  showImage(im_list_bike_pgm_jp2)
  showImage(im_list_cafe_pgm_jp2)
```

```
  showImage(im_list_bike_pgm_jpeg_decoded)
  showImage(im_list_cafe_pgm_jpeg_decoded)
  showImage(im_list_bike_jp2)
  showImage(im_list_cafe_jp2)
  showImage(im_list_bike_jpg)
  showImage(im_list_cafe_jpg)
```

Find BPP

```
def get_list_of_BPP(paths):
  BPPs = []
  for path in paths:
    size_in_bytes = os.path.getsize(path)
    print(path, size_in_bytes*0.0009765625, "KB")
    size_in_bits = size_in_bytes*8
    BPPs.append(JPEG_BPP(2048, 2560, size_in_bits))

  return BPPs
```

```
cafe_BPPs = get_list_of_BPP(im_list_cafe_jpg)
bike_BPPs = get_list_of_BPP(im_list_bike_jpg)
woman_BPPs = get_list_of_BPP(im_list_woman_jpg)
```

```
cafe_N5.jpg 189.21875 KB
cafe_N20.jpg 466.6064453125 KB
cafe_N40.jpg 701.7587890625 KB
cafe_N60.jpg 908.095703125 KB
cafe_N80.jpg 1327.3720703125 KB
bike_N5.jpg 126.466796875 KB
bike_N20.jpg 290.7587890625 KB
bike_N40.jpg 452.517578125 KB
bike_N60.jpg 603.1982421875 KB
bike_N80.jpg 919.1376953125 KB
woman_N5.jpg 106.578125 KB
woman_N20.jpg 263.9462890625 KB
woman_N40.jpg 416.9189453125 KB
woman_N60.jpg 570.2421875 KB
woman_N80.jpg 896.3173828125 KB
```

```
print(cafe_BPPs)
print(bike_BPPs)
print(woman_BPPs)
```

```
[0.295654296875, 0.7290725708007812, 1.0964981079101563, 1.4188995361328125, 2.07401
[0.1976043701171875, 0.45431060791015626, 0.7070587158203125, 0.9424972534179688, 1.
[0.1665283203125, 0.41241607666015623, 0.6514358520507812, 0.89100341796875, 1.40049
```

```
def path_to_imlist(paths):
  image_collection = []
  for path in paths:
    f = open(path,'rb')
    im = plt.imread(f)
```

```
    image_collection.append(im)

  return image_collection
```

```
jpeg_bike_imgs = path_to_imlist(im_list_bike_pgm_jpeg_decoded)
jp2_bike_imgs = path_to_imlist(im_list_bike_pgm_jp2_decoded_bbp)

jpeg_cafe_imgs = path_to_imlist(im_list_cafe_pgm_jpeg_decoded)
jp2_cafe_imgs = path_to_imlist(im_list_cafe_pgm_jp2_decoded_bbp)

jpeg_woman_imgs = path_to_imlist(im_list_woman_pgm_jpeg_decoded)
jp2_woman_imgs = path_to_imlist(im_list_woman_pgm_jp2_decoded_bbp)

im_cafe_original = path_to_imlist(["cafe.pgm"])[0]
im_bike_original = path_to_imlist(["bike.pgm"])[0]
im_woman_original = path_to_imlist(["woman.pgm"])[0]
```

```
jpeg_quality = ["5", "20", "40", "60", "80"]

def compare_psnr(original, images):
  for i, quality in enumerate(jpeg_quality):
    psnr = compute_psnr(original, images[i])
    print('psnr for jpeg quality level', quality, psnr)
```

Bike

```
print("JPEG")
compare_psnr(im_bike_original, jpeg_bike_imgs)
print("JP2")
compare_psnr(im_bike_original, jp2_bike_imgs)
```

```
JPEG
psnr for jpeg quality level 5 24.6708302665451
psnr for jpeg quality level 20 29.637224110797867
psnr for jpeg quality level 40 32.17728153025021
psnr for jpeg quality level 60 33.95135473067994
psnr for jpeg quality level 80 36.925046349052984
JP2
psnr for jpeg quality level 5 28.446817305964487
psnr for jpeg quality level 20 32.980091981958395
psnr for jpeg quality level 40 35.621720410663656
psnr for jpeg quality level 60 37.71226822502957
psnr for jpeg quality level 80 40.95358376205617
```

Cafe

```
print("JPEG")
compare_psnr(im_cafe_original, jpeg_cafe_imgs)
print("JP2")
compare_psnr(im_cafe_original, jp2_cafe_imgs)
```

```
JPEG
psnr for jpeg quality level 5 21.691378202078237
psnr for jpeg quality level 20 26.503611241802368
psnr for jpeg quality level 40 29.213705946527384
psnr for jpeg quality level 60 31.199640170698554
psnr for jpeg quality level 80 34.672451157529295
JP2
psnr for jpeg quality level 5 23.983529928713452
psnr for jpeg quality level 20 29.36882671042407
psnr for jpeg quality level 40 32.88008973232255
psnr for jpeg quality level 60 35.12816362089012
psnr for jpeg quality level 80 39.68858463340644
```

Woman

```
print("JPEG")
compare_psnr(im_woman_original, jpeg_woman_imgs)
print("JP2")
compare_psnr(im_woman_original,·jp2_woman_imgs)
```

```
JPEG
psnr for jpeg quality level 5 24.906864454111698
psnr for jpeg quality level 20 29.629634033350957
psnr for jpeg quality level 40 32.08678947082698
psnr for jpeg quality level 60 33.89549193425248
psnr for jpeg quality level 80 37.007761356958454
JP2
psnr for jpeg quality level 5 28.335104539134253
psnr for jpeg quality level 20 32.48822793480885
psnr for jpeg quality level 40 35.25514518827637
psnr for jpeg quality level 60 37.6386427595168
psnr for jpeg quality level 80 41.06731818707129
```

✓ 0s completed at 3:52 PM