Tongyang Liu

Q1

First, I resembled the 8 bit-planes which each has one bit masked.



import numpy as np

import matplotlib.pyplot as plt

%matplotlib

# bitplanes with just one bit masked

d = io.imread("dollar.tif")

# create dd matrix which has the same size as the original one

dd = np.ndarray(shape = d.shape).astype("uint8")

# transform 8 bitplanes to the original one.

for i in range(8):

dd = dd | (d&(~(1<<i))).astype("uint8")

# create a boolean indicator to see if every element in matrix is equal to the original one.

logical = (dd == d)

# count all the true value

c = 0

for i in range(d.shape[0]):

for j in range(d.shape[1]):

if logical[i][j] == True:

c += 1

# test if the number of true value equals matrix size and it is true.

c == d.shape[0]\* d.shape[1]

plt.imshow(dd,vmin = 0, vmax =255, cmap= plt.cm.gray)

Secondly, I resembled the 8 bit-planes which each has one bit unmasked. I got the same result and same picture as above.

# bitplanes with just one bit unmasked, and all the other bits are masked.

ddd = np.ndarray(shape = d.shape).astype("uint8")

# transform 8 bitplanes to the original one.

for i in range(8):

ddd = ddd | (d&((1<<i))).astype("uint8")

# set the difference

test = ddd - d

# the max of diff is 0

test.max()

plt.imshow(ddd,vmin = 0, vmax =255, cmap= plt.cm.gray)

Q2

a) add a constant



d\_add = d + 100

plt.imshow(d\_add,vmin = 0, vmax =255, cmap= plt.cm.gray)

if after we add a constant, all the number in matrix is still under 255, then , the image is just whiter than before. If after the addition , some number is overflow, then the number will become (num mod 255), which will be a smaller number, and the image will tend to be darker in the region where overflow occurs. Just as shown in the above image.

b) substract a constant

d\_substract = d \* 10

plt.imshow(d\_substract,vmin = 0, vmax =255, cmap= plt.cm.gray)



if after we substract a constant, all the number in matrix is still greater than 0, then , the image is just darker than before. If after the substraction, some number is underflow, then the number will become (num mod 255), which may be a bigger number, and the image will tend to be whiter in the region where underflow occurs. Just as shown in the above image.

c) multiply a constant

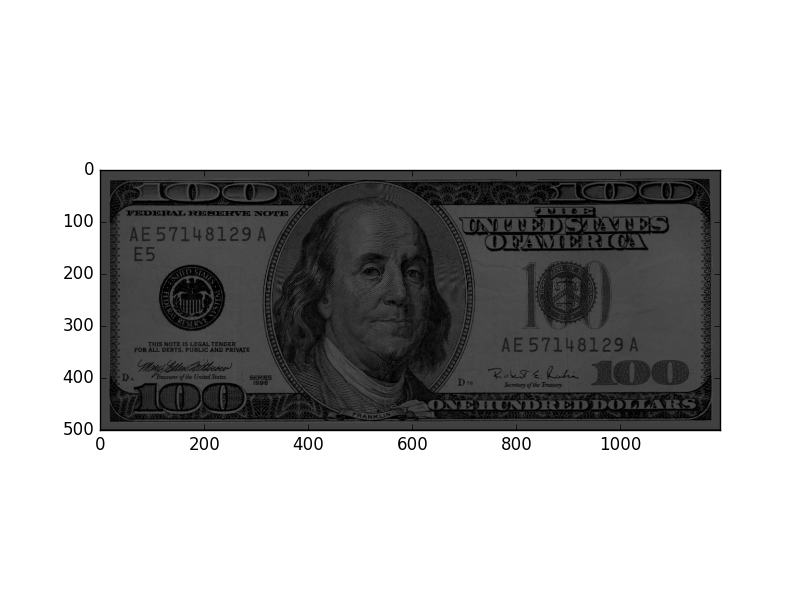


d\_multip = d \*1.5

plt.imshow(d\_multip,vmin = 0, vmax =255, cmap= plt.cm.gray)

if after we multiply a reasonable constant, all the number in matrix is still less than 255, then , the image is just whiter than before. If after the multiplication, some number is overflow, then the number will become (num mod 255), which may turn to be a random number(in this case, I am not very sure what the value would be after modulo), and there will be some random points distributed on the image. The whiter image shown above with multiplication of 1.5

d) divide a constant



d\_div = d / 3

plt.imshow(d\_div,vmin = 0, vmax =255, cmap= plt.cm.gray)

if after we divide a reasonable constant, the image is just darker than before. If we divide the image by a really big number, the results can be very small so the image will simply become a black image with black shades we can not really tell from 0.

If we want to get an expected image after these arithmetic operation, we need to do it carefully and we may write our own program to deal with underflow or overflow in order to get a desired image after these operations.