DIGITAL IMAGE PROCESSING: CODING TEST

张立远, 12012724

Problem 1

```
import numpy as np
import cv2

img = cv2.imread('./Figure1.tif')
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('img', img)

img_log = np.log(1.0+img)
img_log = np.uint8(255 * (img_log - np.min(img_log))/(np.max(img_log)-
np.min(img_log)))
cv2.imshow('img_log', img_log)

cv2.waitKey(0)
```

The solution is to make a logarithm transform

$$IMG_{log} = \lambda \log\left(1 + IMG\right)$$

And then make a `MIN_MAX` normalization and typecast it into `np.uint8`

Problem 2

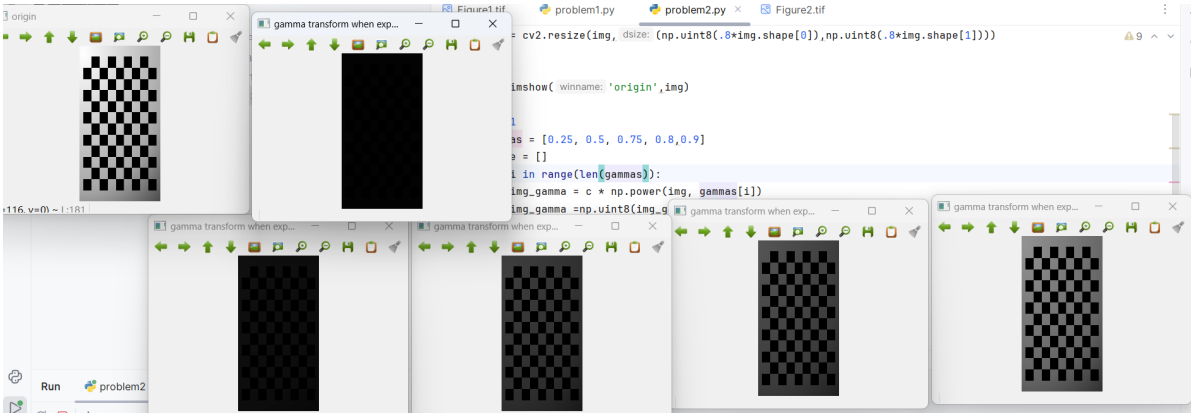
```
import numpy as np
import cv2
```

```
img = cv2.imread('./Figure2.tif')
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

img = cv2.resize(img,(np.uint8(.8*img.shape[0]),np.uint8(.8*img.shape[1])))


cv2.imshow('origin',img)

c = 1
gammas = [0.25, 0.5, 0.75, 0.8,0.9]
title = []
for i in range(len(gammas)):
    img_gamma = c * np.power(img, gammas[i])
    img_gamma =np.uint8(img_gamma)
    img_gamma = cv2.equalizeHist(img_gamma)
    title = "gamma transform when exp = "+ str(gammas[i])
    cv2.imshow(title,img_gamma)

cv2.waitKey(0)
```

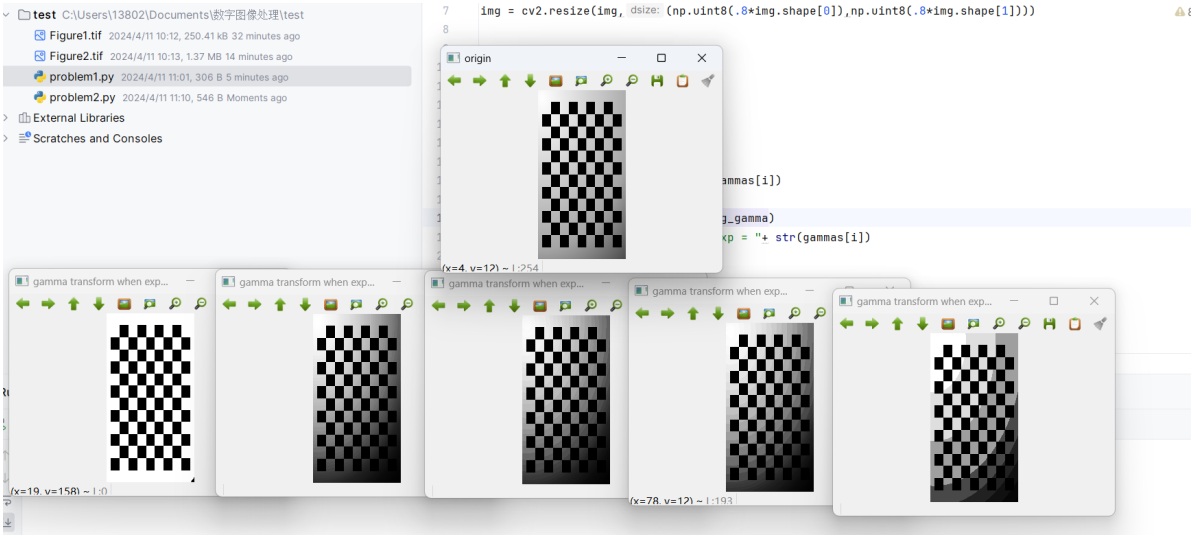
The idea is to first make an power law transform (Gamma transform).

$$IMG_{power} = c \cdot IMG^{\lambda}$$

We choose `c=1` here. We select multiple different `\lambda`, the results are:



All of these are too dark in gray scale, so we make a histogram equalization.



Surprisingly , we find that when `\lambda` is 0.25, together with histogram equalization, the image can be transformed into black-white mosaic pattern.