

## Source Code

### (A) Dithering

```
import argparse
import numpy as np
import cv2

# parse the command line arguments
parser = argparse.ArgumentParser()
parser.add_argument('-i', '--image', type=str,
                    default='img/default.jpg', help='path to the image')
parser.add_argument('-o', '--output', type=str,
                    default='', help='name of the output image')
parser.add_argument('-t', '--type', type=str, default='',
                    help='type of the output image')
args = parser.parse_args()

if(args.output == ''): args.output = args.image.split('/')[ -1 ].split('.')[ 0 ]
if(args.type == ''): args.type = args.image.split('.')[ -1 ]

# Load grayscale image and save original
img = cv2.imread( args.image, cv2.IMREAD_GRAYSCALE )
cv2.imwrite(args.output + '_original.' + args.type, img)

# Define dithering matrix D2
D2 = np.array([[0, 128, 32, 160],
               [192, 64, 224, 96],
               [48, 176, 16, 144],
               [240, 112, 208, 80]])

# Repeat D2 to create array D of image size
rows, cols = img.shape[:2]
D = np.tile(D2, (rows // 4 + 1, cols // 4 + 1))[:rows, :cols]

# Threshold image using D
I_thresh = np.where(img > D, 255, 0).astype(np.uint8)

# Save thresholded image
cv2.imwrite(args.output + '_dithering.' + args.type, I_thresh)
```