

# 《计算机视觉》实验报告

姓名：戴枫源 学号：19120199

## 实验二

### 一. 任务 1

#### a) 核心代码：

```
import numpy as np
import cv2 as cv

img = cv.imread('./chandler.PNG',-1)
rows,cols,way = img.shape

# 平移：x 轴平移 100 像素，y 轴平移 150 像素
M = np.float32([[1,0,100],[0,1,150]])
dst1 = cv.warpAffine(img,M,(cols,rows))

# 缩放：缩放到 1024*768；按比例缩小（60%）
dst2_1 = cv.resize(dst1,(1024,768))
dst2 = cv.resize(dst2_1,None,fx=0.6,fy=0.6)

# 翻转：水平翻转，垂直翻转，水平+垂直翻转
dst3 = cv.flip(dst2,-1)

# 旋转：给出旋转中心，旋转角度，对图片旋转
# 将图片相对中心旋转 90 度
```

```

rows4,cols4,way = dst3.shape
M4 = cv.getRotationMatrix2D(((cols-1)/2.0,(rows-1)/2.0),90,1)
dst4 = cv.warpAffine(dst3,M,(cols4,rows4))

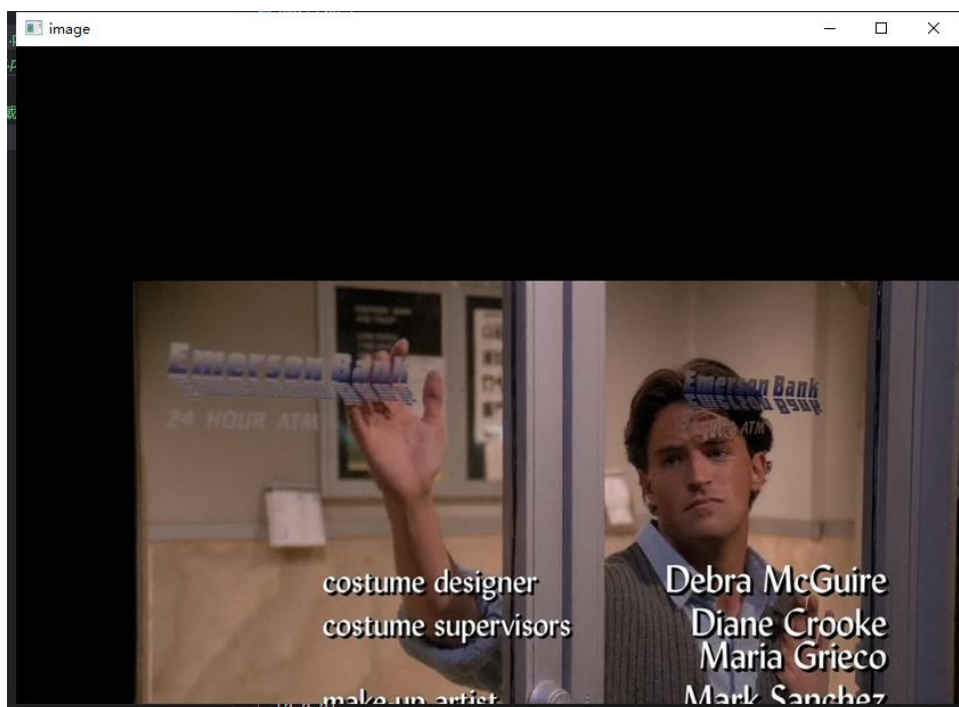
# 缩略：将图片缩小 0.5 倍，放到原图的左上角
dst5_1 = cv.resize(dst4,None,fx=0.5,fy=0.5)
rows5,cols5 = dst5_1.shape[:2]
dst5 = img[:]
dst5[:rows5,:cols5] = dst5_1

cv.imshow('image',dst5)
cv.waitKey(0)
cv.destroyAllWindows()

```

b) 实验结果截图

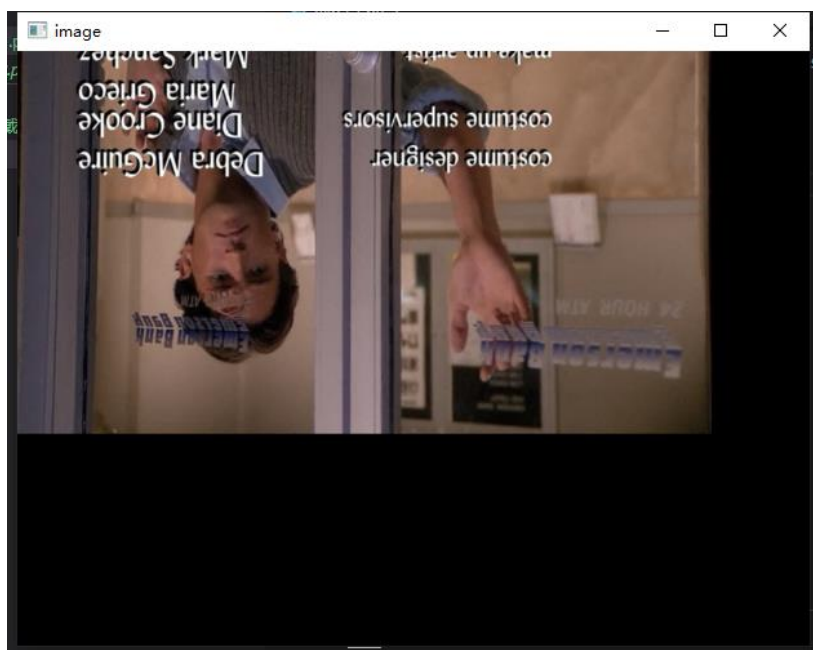
第一步操作：



第二步操作：



第三步操作:



第四步操作:



第五步操作:



### c) 实验小结

关于任务 1，我觉得比较困难的地方在于第五步如何将图片放置于原图的左上角。我是以矩阵形式读入图片，所以直接对原图的矩阵的左上角部分进行赋值操作，赋值为之前进行过操作的图片，然后再显示，就能达到目标效果了。

## 二. 任务 2

### a) 核心代码：

```
import numpy as np
import cv2 as cv
import math

# 读入为灰度图像
img = cv.imread('./chandler.PNG',0)

rows,cols = img.shape

# 调整灰度图片为正方形（边长不小于 500 像素）
img2 = cv.resize(img,(min(rows,cols),min(rows,cols)))

# 用圆形掩膜对图片进行切片，并保存切片后的图像
def distance(a,b,x,y):
    return math.sqrt(math.pow((a-x),2)+math.pow((b-y),2))

d = min(rows,cols)
o_i,o_j = d/2,d/2
for i in range(d):
    for j in range(d):
        if distance(i,j,o_i,o_j)>d/2:
            img2[i,j] = 255
```

```
cv.imshow('image',img2)
k = cv.waitKey(0)
if k == 27:
    cv.destroyAllWindows()
elif k == ord('s'):
    cv.imwrite('exp2.2.PNG',img2)
    cv.destroyAllWindows()
```

b) 实验结果截图



### c) 实验小结

我的思路是用两个 `for` 循环进行遍历。如果循环到的点离圆心的距离超过半径的话，就让这个点的颜色置为白色，就得到上图所示的效果了。