Homework6 Texture

胡成成 2101210578

Question

四选一：

1. 用根据纹理分割图中不同的区域
2. 比较三纹理图篇，哪两张更相似
3. 纹理迁移
4. 自选图和网络模型，实现风格迁移

Answer

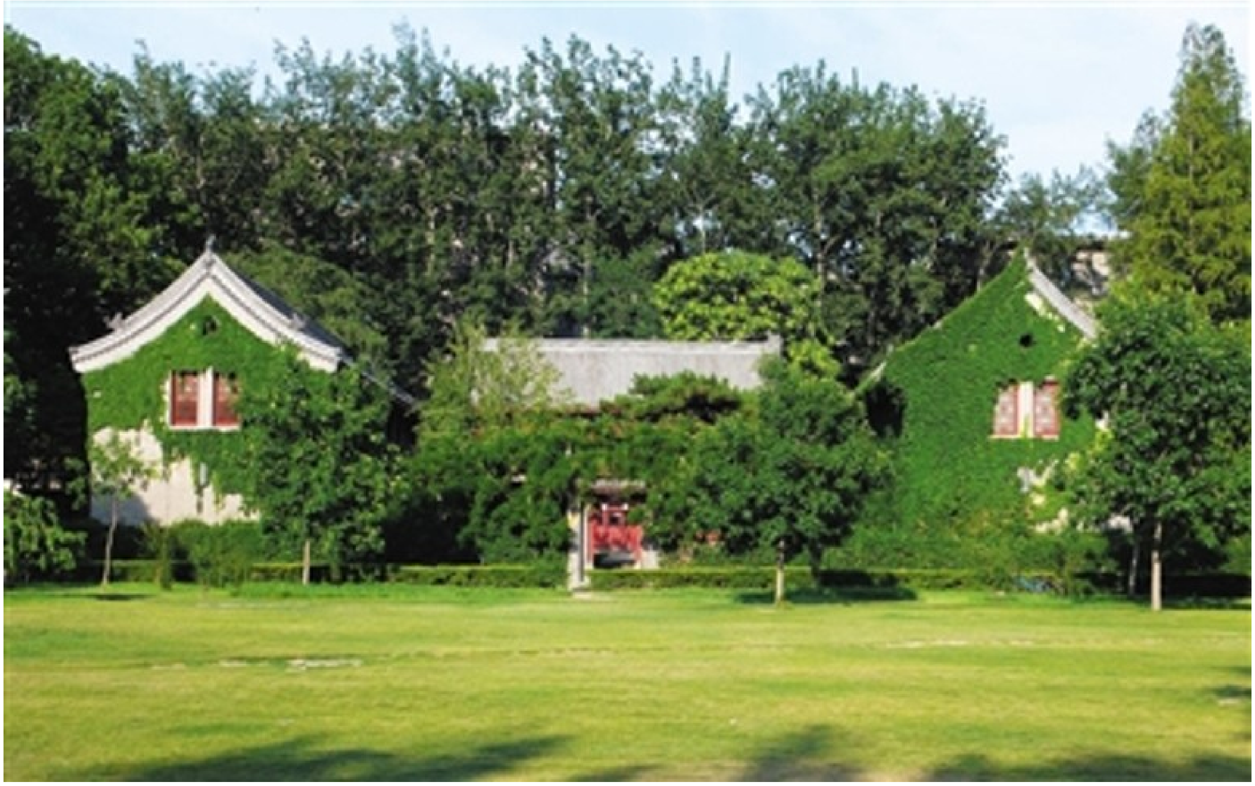
选择第一题：

- 代码：

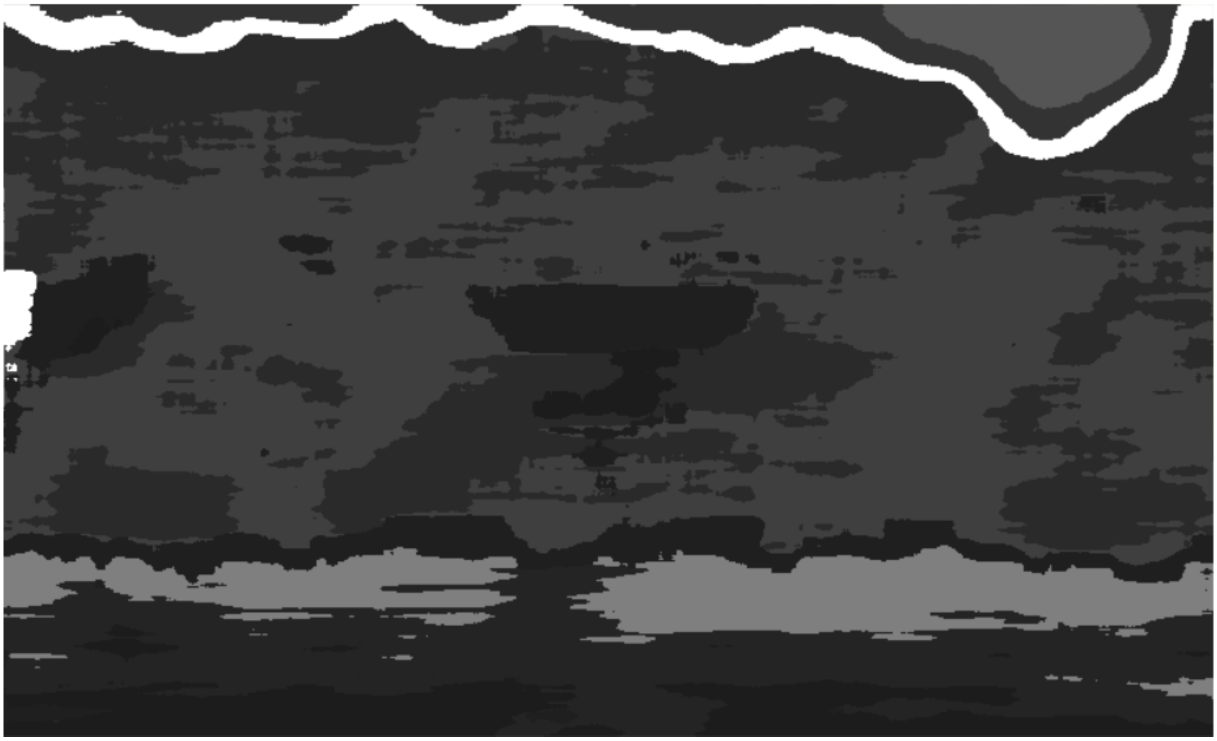
```
import cv2
import numpy as np
from sklearn.cluster import KMeans
```

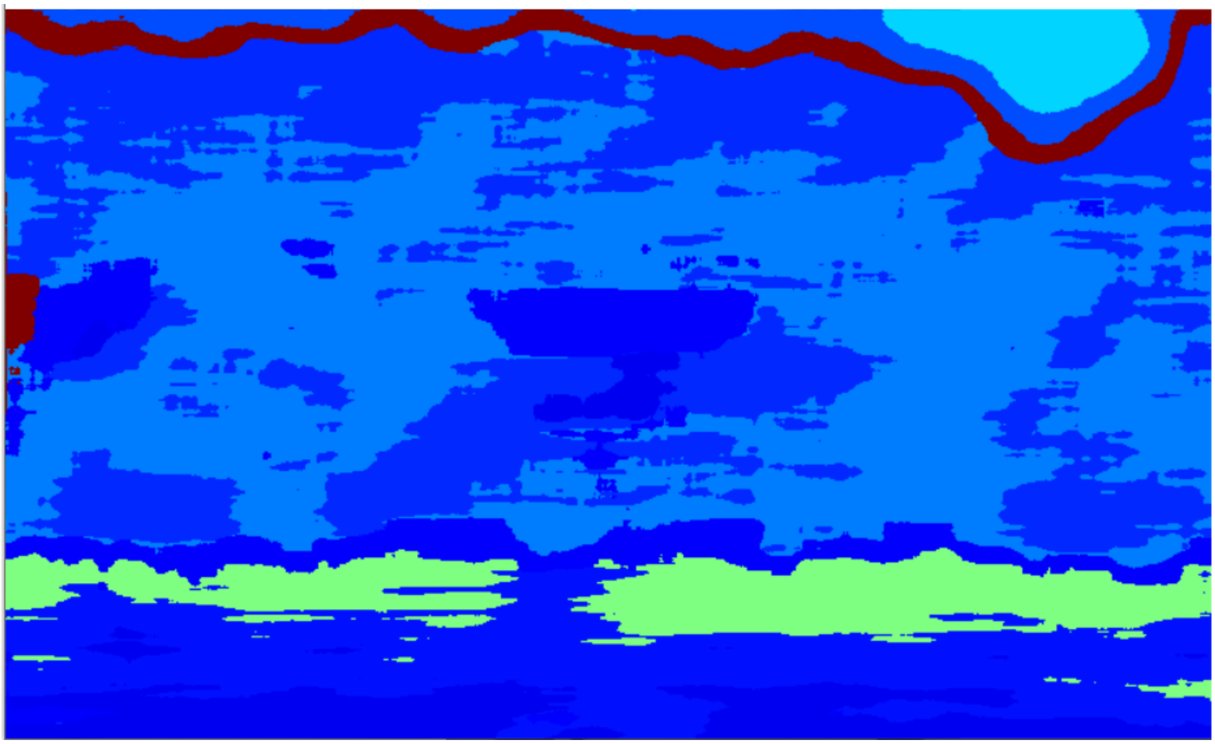
```
def load_image(file):
    img1 = cv2.imread(file, 1)
    cv2.imshow("test.jpg", img1)
    img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    return img1


# LBP图
def get_lbp(grayimg):
    rows, cols = grayimg.shape
    lbpmem = np.zeros((rows, cols), np.uint8)
    for i in range(1, rows - 1):
        for j in range(1, cols - 1):
            sum = 0;
            if (grayimg[i, j] < grayimg[i - 1, j - 1]):
                sum = 128 + sum
            if (grayimg[i, j] < grayimg[i - 1, j]):
                sum = sum + 64
            if (grayimg[i, j] < grayimg[i - 1, j + 1]):
                sum = sum + 32
            if (grayimg[i, j] < grayimg[i, j + 1]):
                sum = sum + 16
            if (grayimg[i, j] < grayimg[i + 1, j + 1]):
                sum = sum + 8
            if (grayimg[i, j] < grayimg[i + 1, j]):
                sum = sum + 4
            if (grayimg[i, j] < grayimg[i + 1, j - 1]):
                sum = sum + 2
            if (grayimg[i, j] < grayimg[i, j - 1]):
                sum = sum + 1
            lbpmem[i, j] = sum
    return lbpmem


# LBP直方图分布
def lbph_map(lbp_img):
    row, col = lbp_img.shape
    half_size = 21  # 图像区域大小
    strip_num = 64  # 直方图等级
    lbphv = []
    for i in range(half_size, row - half_size):
        for j in range(half_size, col - half_size):
            roi_img = lbp_img[(i - half_size):(i + half_size), (j - half_size):(j + half_size)]
            hist = np.zeros(strip_num, np.float32)
            hist = cv2.calcHist([roi_img], [0], None, [strip_num], [0, 255])
            histf = hist.flatten()
            histf = (histf / ((2 * half_size) * (2 * half_size)))
            lbphv.append(histf)
    return lbphv, row - 2 * half_size, col - 2 * half_size


# 分割
def lbp_segment(data, row, col):
    label = KMeans(n_clusters=9).fit_predict(data)
    label = label.reshape([row, col])
    newimg = np.zeros((row, col), np.uint8)
    for i in range(row):
        for j in range(col):
            newimg[i, j] = 255 / (label[i][j] + 1)
    return newimg


def main():
    img = load_image("test.jpg")
    row, col = img.shape
    lbphimg = get_lbp(img)
    lbph_img, m_row, m_col = lbph_map(lbphimg)
    print(m_row, m_col)
    labelimg = lbp_segment(lbph_img, m_row, m_col)
    colorlab = cv2.applyColorMap(labelimg, cv2.COLORMAP_JET)
    cv2.imshow("segment", labelimg)
    cv2.imshow("color", colorlab)
    cv2.waitKey(0)


if __name__ == "__main__":
    main()
```

- 输入图片

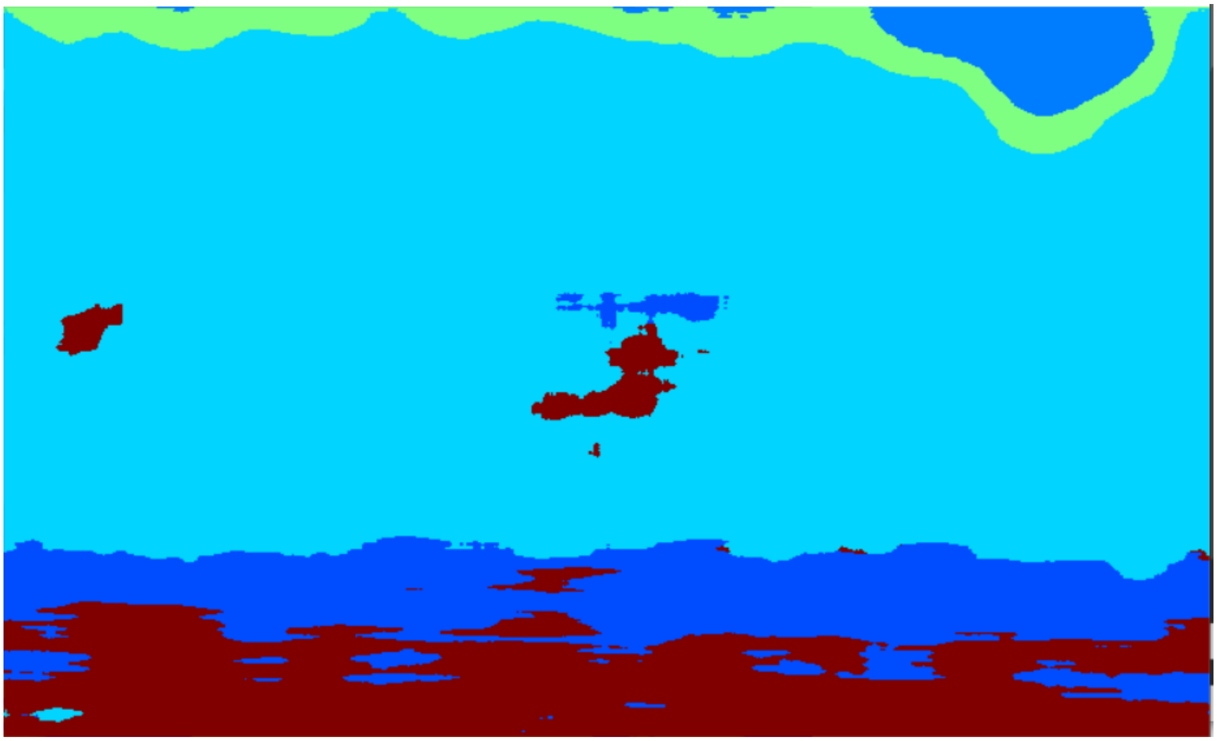


- `n_clusters=9` 结果：

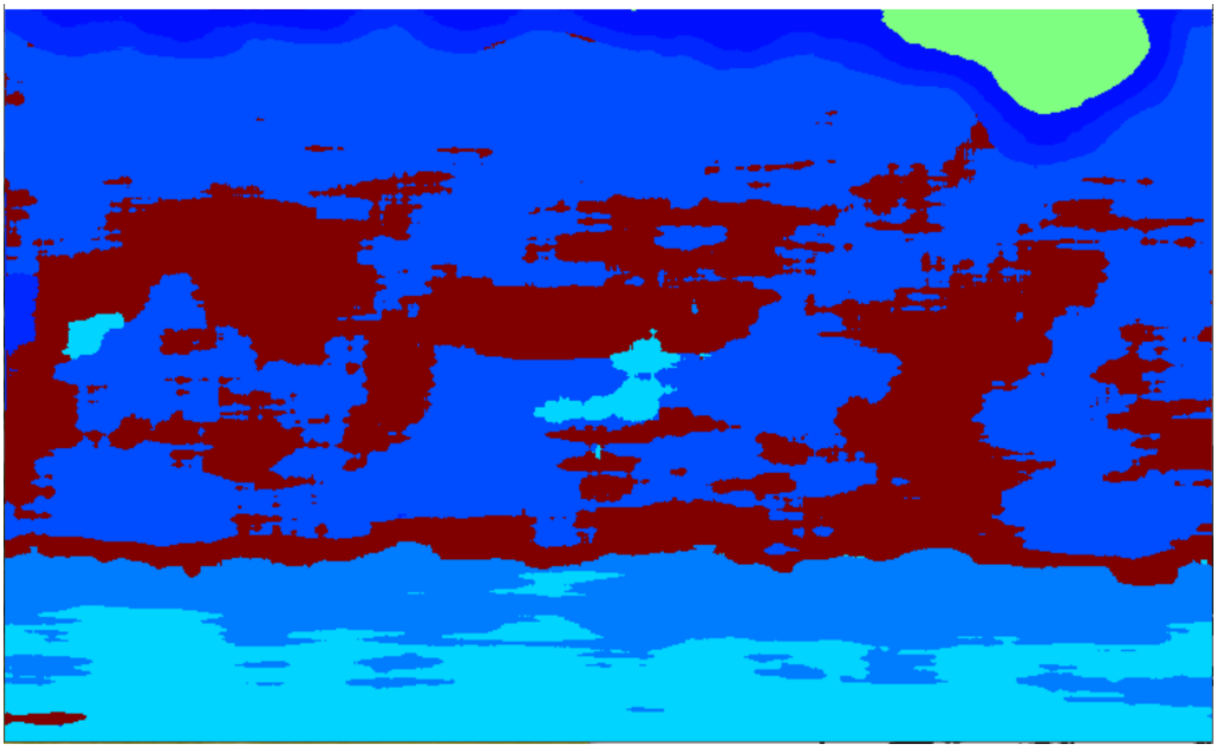




- `n_clusters=5` 结果：



- `n_clusters=7` 结果：



- 结论：可以看出，kmeans聚类类簇个数对分割结果影响还是很大的