

# 第十二次作业讲评

郭明非

2022-04-25

## 1.1 对图像做直方图均衡，然后用otsu阈值二值化

```
# TODO TODO TODO
# 张钊 1900012979
def histogram_equalization(grey):
    new_grey = grey.copy()
    temp_list, _ = np.histogram(new_grey.flatten(), 256, [0,1])
    prob = np.cumsum(temp_list)/np.float32(new_grey.size)
    img_map = (np.clip(prob, 0, 1))
    for i in range(new_grey.shape[0]):
        for j in range(new_grey.shape[1]):
            new_grey[i][j] = img_map[int(255 * new_grey[i][j])]
    return new_grey
equalized_gray = histogram_equalization(gray)
show_images([gray, equalized_gray], ['Gray', 'Equalized Gray'])

bin_img = (255 * equalized_gray).copy().astype(np.uint16)

ret, bin_img = cv.threshold(bin_img, 0, 255, cv.THRESH_BINARY + cv.THRESH_OTSU)

show_images([equalized_gray, bin_img], ['Equalized Gray', 'Binary'])
```

需要多传递一个参数cv2.THRESH\_OTSU。此时函数cv2.threshold()会自动寻找最优阈值。

## 1.2 去掉谱线

```
# 谢子辰 1900010620
from scipy import signal

l=np.shape(bin_img)[1]

comp=np.vstack([np.ones((3,l)),np.zeros((29,l))])
kernel=np.vstack([comp]*5)

new_bin_img=bin_img.copy()
arr=signal.convolve2d(bin_img,kernel,'valid')
idx=list(arr).index(arr.min())
for r in range(1,6):
    new_bin_img[idx+r*32-4:idx+r*32][:]=255
plt.imshow(new_bin_img)

kernel2=np.ones((4,4),dtype=np.uint8)
dilate=cv.erode(new_bin_img,kernel2,1)
plt.imshow(dilate)
```

直接用五条谱线的滑动窗口去对齐原图。

由于图像是白底，所以使用erode完成膨胀。膨胀和腐蚀都是形容白色部分的，所以为了膨胀黑色部分，需要使用erode腐蚀白色部分。

## 1.2 去掉谱线

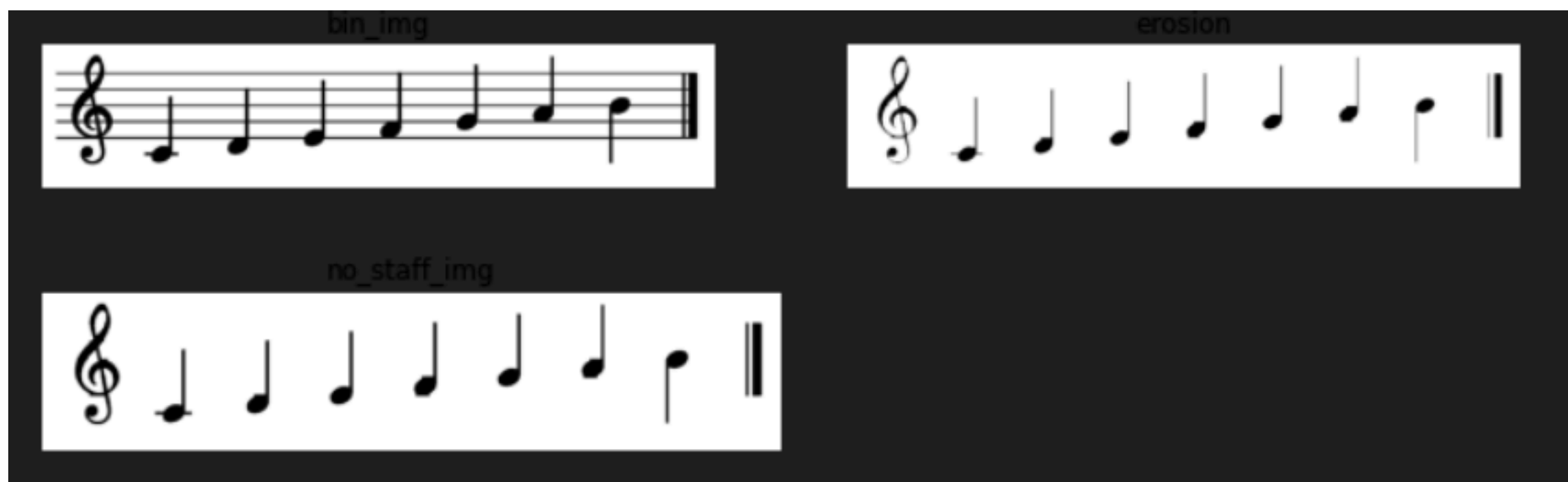
```
# 刘华秋 2000012917
from sklearn.cluster import KMeans
kernel=np.ones((1,bin_img.shape[1]))
conv=np.array([int(row@kernel.T) for row in bin_img]).reshape(-1,1)
km = KMeans(n_clusters=2)
km.fit(conv)
label=km.labels_
linedrop_img=np.array([list(bin_img[i]) if label[i]==0 else [255]*bin_img.shape[1] for i
show_images([linedrop_img])
```

直接用一条线去对齐原图，使用KMeans选出谱线所在的位置，避免调整阈值。

## 1.2 去掉谱线

```
# 李政 1900012146
kernel = np.ones((5, 5), dtype=np.uint8)
erosion = cv.dilate(bin_img, kernel, iterations=1)
show_images([bin_img, erosion], ['bin_img', 'erosion'])
kernel = np.ones((4, 4), dtype=np.uint8)
no_staff_img = cv.erode(erosion, kernel, iterations=1)
show_images([no_staff_img], ['no_staff_img'])
```

不匹配谱线直接膨胀  
+腐蚀。



## 1.3 用hog特征分类

```
# 取出音符
from skimage.measure import label, regionprops
bw = 1-no_staff_img
label_img = label(bw)
show_images([label_img])

boundary = []
for region in regionprops(label_img):
    if region.area >= 100:
        boundary.append(region.bbox)
```

label用来做连通区域标记，输入参数包括input，background（默认为0），以及connectivity。题目默认no\_staff\_img是用0/1表示的，所以用1-no\_staff\_img，对应的background是0，所以可以正确标记。但是如果no\_staff\_img是用0/255表示的，应该用255-no\_staff\_img&background是0，或者1-no\_staff\_img&background是2。

## 1.3 用hog特征分类

```
# 刘珈征 1900012924
# 认为一条竖线就是终止符
import random
from sklearn.naive_bayes import MultinomialNB

lab = np.array(["谱号", "音符", "音符", "音符", "音符", "音符", "音符", "音符", "终止符", "终止符"])
for i in range(10):
    components[i] = cv.resize(components[i], (64, 64)) # 图片resize统一大小, 方便生成等长的hog
lab = np.expand_dims(lab, 0).repeat(100, axis=0)
lab = lab.reshape((-1))
X = components.copy()
for k in range(99): #随机旋转, 随机缩放, 数据增强
    for i in range(10):
        ang = random.random() * 360
        size = random.random() * 0.4 + 0.4
        M = cv.getRotationMatrix2D((32, 32), ang, size)
        X.append(cv.warpAffine(components[i], M, (64, 64), borderValue=255))
```

数据增强可以随机旋转缩放/添加高斯噪声。为了使hog特征长度一样方便训练，需要resize保证图片大小一样。

## 1.3 用hog特征分类

```
hog=cv.HOGDescriptor(winSize,blockSize,blockStride,cellSize,nbins)
X1 = []
for i in X:
    X1.append(hog.compute(i,blockSize,cellSize).reshape((-1)))
X1 = np.array(X1)
X1.reshape(10*100, -1)

# 由于空间问题, 这里没有办法shuffle, 但是由于旋转缩放是随机的, 可以视为shuffle过了
X_train, X_test = X1[:900,:], X1[900:,:] #9: 1的训练测试集
y_train, y_test = lab[:900], lab[900:]
clsm = MultinomialNB()
clsm.fit(X_train,y_train)
print('MultinomialNB Training Score: %.2f' % clsm.score(X_train,y_train))
print('MultinomialNB Testing Score: %.2f' % clsm.score(X_test, y_test))
By_pred = clsm.predict(X_test)
show_images(X[900:910])
print(By_pred[0:10])
```