**一种利用Python和Opencv进行车牌识别的方法**

1. **模板库的制作和使用**

模板制作所使用的模板如图





模板制作代码如下：

import pickle  
import cv2 as cv  
import matplotlib.pyplot as plt  
import numpy as np  
kernel = np.ones((3,3),np.uint8)  
  
moban = cv.imread()  
moban\_gray = cv.cvtColor(moban,cv.COLOR\_BGR2GRAY)  
\_,moban\_erzhi = cv.threshold(moban\_gray,127,255,cv.THRESH\_BINARY)  
  
contours= cv.findContours(moban\_erzhi,cv.RETR\_EXTERNAL,cv.CHAIN\_APPROX\_SIMPLE)[0]  
list = []  
for i in range(len(contours)):  
 x,y,w,h = cv.boundingRect(contours[i])  
 img\_cut = moban\_erzhi[y:y+h,x:x+w]  
 resize\_cut = cv.resize(img\_cut,(18,36))  
 list.insert(0,resize\_cut)  
 print(x,y,w,h)  
  
moban\_list = open('D:\pycharm\plate\_recognize\moban\_list','wb')  
pickle.dump(list,moban\_list)  
moban\_list.close()

1. **车牌区域的获取**

代码如下：

#读入图片

img = cv.imread('图片')  
#BGR转HSV  
hsvimg = cv.cvtColor(img,cv.COLOR\_BGR2HSV)  
H,S,V = cv.split(hsvimg)  
#设置阈值，此处为蓝色  
lowblue = np.array([100,43,100])  
highblue = np.array([124,255,255])  
#掩膜操作，将阈值内的区域置255，阈值外的置0  
mask = cv.inRange(hsvimg, lowblue, highblue)  
#用掩膜将蓝色区域截出  
BlueThings = cv.bitwise\_and(img, img, mask=mask)  
#蓝色区域转灰度图  
gray\_blue = cv.cvtColor(BlueThings,cv.COLOR\_BGR2GRAY)  
erzhi = cv.threshold(gray\_blue,1,255,cv.THRESH\_BINARY)[1]  
imgclose = cv.morphologyEx(erzhi,cv.MORPH\_CLOSE,kernel,iterations=1)  
imgopen = cv.morphologyEx(imgclose,cv.MORPH\_OPEN,kernel,iterations=1)  
#取轮廓  
contours,hierarchy = cv.findContours(imgopen,cv.RETR\_EXTERNAL,cv.CHAIN\_APPROX\_SIMPLE)  
#画出轮廓  
imgcopy = img.copy()  
drawcontours = cv.drawContours(imgcopy,contours,-1,(0,255,0),2)  
#得到轮廓外接矩形并筛出车牌部分  
area = []  
rect = []  
for i in range(len(contours)):  
 x,y,w,h = cv.boundingRect(contours[i])  
 if 2 < w/h < 4:  
 area.append(w\*h)  
 rect.append(contours[i])  
areamax\_index = area.index(max(area))  
x\_f,y\_f,w\_f,h\_f = cv.boundingRect(rect[areamax\_index])  
imgrect = cv.rectangle(imgcopy, (x\_f, y\_f), (x\_f + w\_f, y\_f + h\_f), (0, 255, 0), 2)  
#通过矩形的参数截出车牌  
final\_plate = img[y\_f:y\_f+h\_f,x\_f:x\_f+w\_f]

1. **通过与模板进行匹配得到车牌**

代码如下：

plate\_final = []  
  
  
#初始化模板  
moban\_file = open('moban\_list','rb')  
moban\_list = pickle.load(moban\_file)  
name\_file = open('name\_list','rb')  
name\_list = pickle.load(name\_file)  
  
  
#读入图片  
plate = cv.imread('test.png')  
gray = cv.cvtColor(plate,cv.COLOR\_BGR2GRAY)  
  
#汉字部分寻找  
\_,erzhi1 = cv.threshold(gray,127,255,cv.THRESH\_BINARY)  
imgopen = cv.morphologyEx(erzhi1,cv.MORPH\_OPEN,kernel,iterations=1)  
imgclose = cv.morphologyEx(imgopen,cv.MORPH\_CLOSE,kernel,iterations=3)  
cv.imshow('imgclose',imgclose)#############################################  
contours\_target2,\_2 = cv.findContours(imgclose,cv.RETR\_TREE,cv.CHAIN\_APPROX\_SIMPLE)  
  
#筛选外接矩形，得到汉字部分  
for i in range(len(contours\_target2)):  
 x1,y1,w1,h1 = cv.boundingRect(contours\_target2[i])  
 if h1/w1 >1.5 and x1 < erzhi1.shape[1]/8:  
 # imgrect = cv.rectangle(plate, (x1, y1), (x1 + w1, y1 + h1), (0, 255, 0), 2)  
 provin\_name = erzhi1[y1-int(0.06\*h1):y1+h1+int(0.06\*h1),x1-int(0.06\*w1):x1+w1+int(0.06\*w1)]  
 provin\_name = cv.resize(provin\_name,(76,76))  
cv.imshow('zi',provin\_name)  
  
#汉字模板匹配  
similar\_list1 = []  
for i in range(len(name\_list)):  
 (score1, diff) = compare\_ssim(provin\_name, name\_list[i], full=True)  
 similar\_list1.append(score1)  
index\_max1 = similar\_list1.index(max(similar\_list1))  
if index\_max1 == 0:  
 plate\_final.append('台')  
if index\_max1 == 1:  
 plate\_final.append('澳')  
if index\_max1 == 2:  
 plate\_final.append('港')  
if index\_max1 == 3:  
 plate\_final.append('新')  
if index\_max1 == 4:  
 plate\_final.append('青')  
if index\_max1 == 5:  
 plate\_final.append('宁')  
if index\_max1 == 6:  
 plate\_final.append('甘')  
if index\_max1 == 7:  
 plate\_final.append('陕')  
if index\_max1 == 8:  
 plate\_final.append('藏')  
if index\_max1 == 9:  
 plate\_final.append('渝')  
if index\_max1 == 10:  
 plate\_final.append('川')  
if index\_max1 == 11:  
 plate\_final.append('贵')  
if index\_max1 == 12:  
 plate\_final.append('云')  
if index\_max1 == 13:  
 plate\_final.append('桂')  
if index\_max1 == 14:  
 plate\_final.append('粤')  
if index\_max1 == 15:  
 plate\_final.append('湘')  
if index\_max1 == 16:  
 plate\_final.append('鄂')  
if index\_max1 == 17:  
 plate\_final.append('豫')  
if index\_max1 == 18:  
 plate\_final.append('鲁')  
if index\_max1 == 19:  
 plate\_final.append('闽')  
if index\_max1 == 20:  
 plate\_final.append('赣')  
if index\_max1 == 21:  
 plate\_final.append('皖')  
if index\_max1 == 22:  
 plate\_final.append('浙')  
if index\_max1 == 23:  
 plate\_final.append('苏')  
if index\_max1 == 24:  
 plate\_final.append('沪')  
if index\_max1 == 25:  
 plate\_final.append('辽')  
if index\_max1 == 26:  
 plate\_final.append('京')  
if index\_max1 == 27:  
 plate\_final.append('吉')  
if index\_max1 == 28:  
 plate\_final.append('黑')  
if index\_max1 == 29:  
 plate\_final.append('蒙')  
if index\_max1 == 30:  
 plate\_final.append('晋')  
if index\_max1 == 31:  
 plate\_final.append('冀')  
if index\_max1 == 32:  
 plate\_final.append('津')  
  
  
  
  
# o = cv.drawContours(plate.copy(),contours\_target2,-1,(0,0,255),1)  
# cv.imshow('2',provin\_name)  
  
#数字字母部分寻找  
\_,erzhi2 = cv.threshold(gray,127,255,cv.THRESH\_BINARY)  
erode2 = cv.erode(erzhi2,kernel,iterations=2)  
dilate2 = cv.dilate(erode2,kernel,iterations=2)  
contours\_target,\_ = cv.findContours(dilate2,cv.RETR\_EXTERNAL,cv.CHAIN\_APPROX\_SIMPLE)  
# o = cv.drawContours(plate.copy(),contours\_target,-1,(0,0,255),1)  
# cv.imshow('2',o)  
  
#选出数字部分并对各个字母数字排序  
all\_x = []  
for n in range(len(contours\_target)):  
 x, y, w, h = cv.boundingRect(contours\_target[n])  
 if h/w > 1.5:  
 all\_x.append(x)  
all\_x\_reset = sorted(all\_x)  
all\_x\_reset.insert(0,0)  
#得到车牌样本  
all\_target = []  
for i in range(len(all\_x)):  
 for m in range(len(contours\_target)):  
 x,y,w,h = cv.boundingRect(contours\_target[m])  
 if h/w > 1.5:  
 a = max(all\_x\_reset)  
 if a-1 < x < a + 1:  
  
 target = gray[y:y+h,x:x+w]  
 \_, target\_erzhi = cv.threshold(target, 127, 255, cv.THRESH\_BINARY)  
 target\_resize = cv.resize(target\_erzhi,(18,36))  
 all\_target.insert(0,target\_resize)  
 all\_x\_reset.pop()  
  
#对样本各个字母进行模板匹配，得到最终结果  
  
for i in range(len(all\_target)):  
 similar\_list = []  
 for m in range(len(moban\_list)):  
 (score, diff) = compare\_ssim(all\_target[i], moban\_list[m], full=True)  
 similar\_list.append(score)  
 index\_max = similar\_list.index(max(similar\_list))  
 if index\_max == 0:  
 plate\_final.append('A')  
 if index\_max == 1:  
 plate\_final.append('B')  
 if index\_max == 2:  
 plate\_final.append('C')  
 if index\_max == 3:  
 plate\_final.append('D')  
 if index\_max == 4:  
 plate\_final.append('E')  
 if index\_max == 5:  
 plate\_final.append('F')  
 if index\_max == 6:  
 plate\_final.append('G')  
 if index\_max == 7:  
 plate\_final.append('H')  
 if index\_max == 8:  
 plate\_final.append(1)  
 if index\_max == 9:  
 plate\_final.append('J')  
 if index\_max == 10:  
 plate\_final.append('K')  
 if index\_max == 11:  
 plate\_final.append('L')  
 if index\_max == 12:  
 plate\_final.append('M')  
 if index\_max == 13:  
 plate\_final.append('N')  
 if index\_max == 14:  
 plate\_final.append(0)  
 if index\_max == 15:  
 plate\_final.append('P')  
 if index\_max == 16:  
 plate\_final.append('Q')  
 if index\_max == 17:  
 plate\_final.append('R')  
 if index\_max == 18:  
 plate\_final.append('S')  
 if index\_max == 20:  
 plate\_final.append('T')  
 if index\_max == 21:  
 plate\_final.append('U')  
 if index\_max == 22:  
 plate\_final.append('V')  
 if index\_max == 23:  
 plate\_final.append('W')  
 if index\_max == 24:  
 plate\_final.append('X')  
 if index\_max == 25:  
 plate\_final.append('Y')  
 if index\_max == 26:  
 plate\_final.append('Z')  
 if index\_max == 27:  
 plate\_final.append(0)  
 if index\_max == 28:  
 plate\_final.append(1)  
 if index\_max == 29:  
 plate\_final.append(2)  
 if index\_max == 30:  
 plate\_final.append(3)  
 if index\_max == 31:  
 plate\_final.append(4)  
 if index\_max == 32:  
 plate\_final.append(5)  
 if index\_max == 19:  
 plate\_final.append(6)  
 if index\_max == 33:  
 plate\_final.append(7)  
 if index\_max == 34:  
 plate\_final.append(8)  
 if index\_max == 35:  
 plate\_final.append(9)  
  
#输出结果  
print('目标车牌为：',plate\_final)