**人脸图像的面部区域裁剪与几何校正作业报告**

针对给定的人脸图像，采用人机交互方式获取两只眼睛的中心坐标、嘴巴 中心坐标；基于这三个位置，实现关于面部区域的裁剪与几何校正，使得输出 的面部区域大小为 101 行\*101 列，并且两只眼睛中心、以及嘴巴中心坐标(行号, 列号)分别为 (25，24)、(25，76)、(75，50)。 最终，得到两种不同灰度插值方法下的输出图像。

**作业目的：** 1. 会使用 python 进行图像文件的读取、可视化、文件保存；

2. 掌握图像几何变换的实现流程；

3. 掌握采用最小二乘法进行仿射变换模型参数估计的方法；

4. 理解并实现三种典型的灰度插值方法。

（1）输入图像文件的读取、显示

# 1-1.图像数据的读取

img = Image.open('人脸图像.jpg')

img.save('new\_人脸图像.jpg')

img.show()



（2）人机交互方式获取输入图像的三个中心

img = cv2.imread(img)

cv2.namedWindow('image')

data = list()

loc = cv2.setMouseCallback('image', on\_EVENT\_LBUTTONDOWN)

cv2.imshow('image', img)

if cv2.waitKey(0) == ord('s'):

cv2.destroyAllWindows()

**(119,225) ,(114,286) ,(175,260)**

（3）输出图像到输入图像的几何变换模型的估计

**引入齐次坐标，采用一个通用的变换模型（仿射变换模型）表示图像的各种几何变换。**

**#获得仿射变换矩阵**

**getPos = [**

**[119,225],**

**[114,286],**

**[175,260]**

**]**

**targetPos = [**

**[25,24],**

**[25,76],**

**[75,50]**

**]**

**def toHomogeneous(data, axis=1):**

**data = np.insert(data, len(data[0]), 1, axis=axis) #默认插入一列1**

**return data**

**def getTrans(oriPos, targetPos):**

**oriPos = np.mat(oriPos)**

**targetPos = np.mat(targetPos)**

**trans = np.linalg.inv(oriPos.T\*oriPos)\*oriPos.T\*targetPos**

**return trans**

def get\_xycorner(img):

xlen, ylen = img.shape[0]-1, img.shape[1]-1

xycorner = np.array([

[0,0,1],

[0,ylen,1],

[xlen,0,1],

[xlen,ylen,1]

]) #四个顶点的齐次坐标

return xycorner

xycorner = get\_xycorner(img)

（4）灰度插值方法的函数实现

**最近邻插值**，简单、快速。将放大后未知的像素点位置换算到原始图像上，并与原始图像上邻近的4个像素点做比较，最靠近的【如欧式距离等方式计算】邻近点的像素值即为该像素点的像素值。

#获得输入灰度图片的灰度值矩阵及形状

img\_arr, img\_shape = get\_ordPicture\_mess(ord\_picture)

#获得近邻法所得灰度图片矩阵

gray1\_img\_arr = create\_gray\_img\_by\_neighbor(img\_arr, W)

**双线性插值**，较慢。将放大后未知的像素点位置换算到原始图像上，并将原始图像上邻近的4个像素点的灰度级别加权求和得到（权值可以采用距离进行度量，距离越近，权值越大）。

#双线性灰度插值法得到的灰度图片矩阵

gray2\_img\_arr = create\_gray\_img\_by\_bilinear(img\_arr, W)

**双三线性插值**

（5）输出图像的显示、以及图像文件的保存

**最近邻插值法**

Img1 = Image.fromarray(gray1\_img\_arr)

cv2.imwrite('近邻法.jpg',gray1\_img\_arr)

img1.show()



**双线性插值法**

Img2 = Image.fromarray(gray2\_img\_arr)

cv2.imwrite('双线性.jpg',gray2\_img\_arr)

img2.show()



（7）实验结果与感想

 

通过此次试验，算是真正体会到了人脸识别的技术初步原理，虽然在获取坐标，更改坐标以及灰度插值函数实现的过程中遇到了些挫折，不过还是带着期待的心情，慢慢地解决问题。期待以后的精彩展现。