**实践作业1-基于聚类的图像分割**

1、使用Python实现K-Means聚类算法

实现说明：

①首先初始化k个cluster；

②计算每个数据点到cluster的欧氏距离（就是相减再平方）；

③把每个数据点归到某个cluster的类里面；

④用该类别里所有数据点的均值代替该cluster的值。

⑤重复上面的②，③，④步骤直至cluster的值收敛

2、使用Python实现HAC聚类算法

实现说明：

①初始化聚类树节点个数为像素个数

②从所有节点中找到距离最近的两个节点，合并成一个父节点

③重复②步骤，直至父节点个数等于聚类数k

注：在实际运行中，我发现HAC的时间复杂度非常高，对大图片处理起来非常耗时，所以我将图片像素进行压缩，这个操作导致了一定的失真，但是计算速度有明显提升。

3、基于两种算法提取图像特征序列进行图像分割

运行说明：

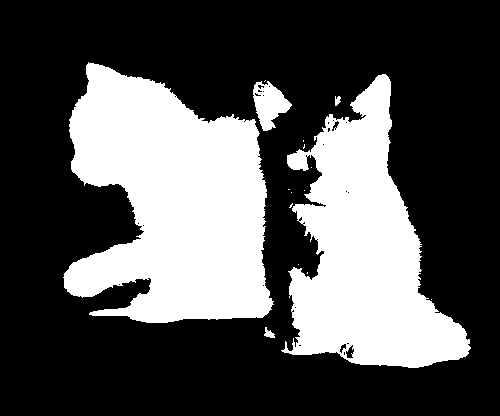
①读取图片和相应Groundtruth

②图片处理成二维数据，并输入聚类算法，返回每个像素的预测分类

③将预测分类和Groundtruth输入评估函数，得到评估结果

分割结果截图：从左至右依次为原图、kmeans结果、HAC结果

第一组：

cat1_result_HAC

第二组：

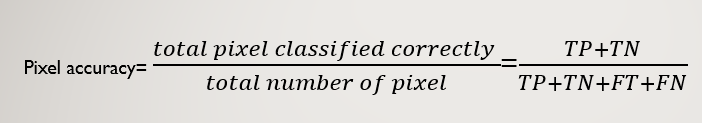
cat2_result_HAC

4、评估

实现说明：

①根据像素预测的正确与否，计算TP、TN、FP、FN值

②根据下图公式，计算像素正确率

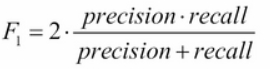


③计算像素真实值和预测值的交集overlap与并集union

④根据下图公式，计算IoU

IMG_256

⑤根据下图公式，依次计算P值、R值、F1值

IMG_256 IMG_256 

评估结果截图：从左至右依次为像素准确率、IoU、F1分数

第一组：





第二组：





**实践作业2-图像拼接**

实现说明：

①用SIFT提取图像中的特征点，并对每个关键点周围的区域计算特征向量。

②在分别提取好了两张图片的关键点和特征向量以后，利用它们进行两张图片的匹配。

③匹配完之后可以获得透视变换H矩阵，用这个的逆矩阵来对第二幅图片进行透视变换，将其转到和第一张图一样的视角。

④透视变化完后就可以直接拼接图片，即将图片通过numpy直接加到透视变化完成的图像的左边，覆盖掉重合的部分，得到拼接图片，但是这样拼接得图片中间会有一条很明显的缝隙。采用加权平均法来处理这个问题，同样把第一张图叠在左边，但是对第一张图和它的重叠区做一些加权处理，重叠部分，离左边图近的，左边图的权重就高一些，离右边近的，右边旋转图的权重就高一些，然后两者相加，使得两张图片之间的过渡是相对平滑地。

结果截图：依次是拼图1、拼图2、拼接后的图





运行说明：

①读取需要拼接的两张图片

②执行拼接算法

③保存拼接完成的结果