## 实验步骤：

本次实验为通过基于SML算法的GMM高斯混合模型对图像进行标注的实验。

**实验步骤一：**收集数据集。

本次实验收集了四个语义类：w1文件夹内为建筑，w2文件夹内为树木，w3文件夹内为天空，w4文件夹内为公路。每个语义类中含有10张300\*450像素的训练图像。图像来源：百度图片。

**实验步骤二：**编写程序。

本实验共编写了10个独立的函数，1个综合性的函数。各个函数的具体功能已在README.md文件中进行说明。

详细步骤：

1. 对每一个语义类，加载数据集为（图片个数，行像素数，列像素数，通道数）的python numpy array形式的矩阵。
2. 对每一个已加载的数据集矩阵，针对每一个图片，通过一个8×8的像素窗口，每次滑动2个像素，将该图片切分为n个区域，区域的个数为：(图像宽度-6)/2 × (图像高度)-6)/2。从而，矩阵的维度变为：（图片个数，n，8，8，通道数）
3. 对每一个数据集矩阵，针对每一张图片的每一个区域进行DCT变换，并将DCT变换后的结果，对每一个区域只取左上角的数据进行压缩，然后将每一个区域的YBR三通道进行交错排列，从而，数据集矩阵的维度变为：（图片个数，区域个数，通道数×压缩后的行像素数×压缩后的列像素数）
4. 对每一个数据集矩阵，针对每一张图片，对N个区域执行K-means算法。该算法中，首先对N个区域等分为4个部分，然后求出每一部分的均值，并将各个部分的概率值初始化为1/4，以此作为k-means算法的初始化均值和初始化各聚类概率值，然后执行k-means算法的详细步骤，更新均值和各个聚类的概率值，最后得出EM算法所需的初始化均值和各个聚类的概率值。
5. 对每一个数据集矩阵，针对每一张图片，对N个区域执行EM算法。EM算法初始化的均值和各个聚类的概率值即K-means算法求出的均值和概率值。EM算法初始化的协方差矩阵是使用每一张图片的N个区域的向量和K-means算法求出的均值向量通过协方差矩阵的计算公式求出来的协方差矩阵。然后执行EM算法的详细步骤，更新均值、概率和协方差矩阵。
6. 对每一个语义类数据集，上一步已经得出了每一张图片的4个高斯混合分布的详细参数，于是，本步就针对每一张图片及每一个高斯混合模型的均值进行k-means算法，以此来初始化下一步的EM算法的初始值。该算法中，首先对每一张图片和每一个高斯混合模型等分为16个部分，然后求出各个部分的均值，并将各个部分的概率值初始化为1/16，以此作为k-means算法的初始化均值和初始化各聚类概率值。然后执行K-means算法的详细步骤，更新均值和各个聚类的概率值。
7. 对每一个语义类数据集，针对每一个图片的每一个高斯混合分布模型，由上一步所得出的类模型的初始化均值以及每一张图片的每一个高斯混合分布模型的均值，通过协方差矩阵的计算公式，计算出类模型的初始化协方差矩阵。现在，类模型的初始化均值、初始化概率、初始化协方差矩阵有了，并且有了每一张图片的每一个高斯混合分布模型的参数，然后就可以执行EM算法步骤，更新类模型的均值、概率、协方差矩阵。
8. 最后将类模型的高斯混合分布模型的各个参数分别保存在各自的.csv格式的文件中。
9. 读取每一个类模型的高斯混合分布模型的参数，对单张图片进行标注。

**实验步骤三：**测试程序。

本实验在测试程序阶段出现了不少问题。

第一个问题：在dct()函数后面起初多写了两个函数：rearrange()用于将dct变换后的图像进行YBR三通道交错排列；compress\_dct()用于将交错排列后的图像进行压缩。但后来发现，如果在dct变换阶段就使用python numpy中的矩阵运算进行压缩的话，效率会更高。因此，就将之前写的那两个函数注释掉了，并将其全部综合到dct()一个函数中了。

第二个问题：关于k-means算法。起初以为其均值可以随机初始化，但是采用随机初始化的方法弱点在于不知道各个向量的取值在什么范围，从而会导致程序出现奇怪的bug。研究之后，改为等分N个向量，然后对每个等分出来的n个向量求均值，将此均值作为初始化的均值之后，程序运行成功，k-means可以正常进行了。

第三个问题：关于多元变量的高斯分布函数的计算。其协方差矩阵有可能会出现不满秩的情况。假设变量是D维的，如果D维变量的高斯分布函数只在D-1维的空间中，那么其协方差矩阵就不会满秩。在数据量比较大的时候，一般不会出现这种问题，但是在数据量比较小的时候，就很容易会出现这种问题。但是增大数据集的同时，k-means和EM算法又会运行更久，考虑到时间的问题，只好采用小数据集解决时间效率的问题，采用迭代少次数来解决协方差矩阵变为不满秩的问题。

另外，在此问题中，起初我编写了另一个多元变量的高斯分布函数gaussian\_pinv()，企图采用伪逆矩阵的方式来解决掉协方差矩阵不满秩的问题，但是失败了。

第四个问题：关于EM算法的问题。起初，我采用了随机初始化协方差矩阵的方法。实际上，不仅是均值和协方差，刚开始时，几乎所有的变量我都采用了随机初始化的方法，但是后来程序报错后才发现，不能简单地将这些变量初始化。如果在K-means算法中已经计算出了均值，就已经可以计算出协方差矩阵的值了。在EM算法中，还有一个问题，是numpy内置库的问题。如果直接采用np.dot(x[n, :] - mu[k, :], (x[n, :] - mu[k, :]).T) / (x.shape[0] - 1)的方法来计算协方差矩阵的话，得到的结果会是对每一个k而言，协方差矩阵中的每一个值都是完全相同的。因为x-mu[k, :]是一个一维的向量，而这个向量采用.T的方式将其转置将仍然是个一维向量，也就是说，这个时候再做点积，那么得到的将是一个标量，因此，计算出的协方差矩阵中的每一个值都会完全相同。因此需要将其先转变为二维向量之后再计算协方差矩阵，才能得到正确的协方差矩阵。

**实验步骤四：**标注图像。

标注图像的函数为label()函数。该函数首先读取了各个类模型的高斯混合分布模型的参数，然后，对单张图片进行了切分区域以及DCT变换抽取出了图像的特征，然后，使用该特征代入到类模型的高斯混合分布模型的计算公式中，计算出各个类模型的预测概率，然后选出计算出的概率值最大的那个类作为该图像属于的类，并输出。