



int aHashCodeFeature(vector<Mat> srcVec, Mat &pHashCode):

1. 转换数据Mat的数据类型为CV_32F ----src.convertTo(src, CV_32F);
2. DCT系数矩阵 (离散余弦变换) ----- dct(src, srcDCT);
3. 求32*32图像的左上8*8方框的均值
4. 左上8*8与3中所求均值比较，> 则设置1；<则设置0，得到8*8的(0, 1)矩阵

cv::Mat MaxMinDisFun(cv::Mat data, float Theta, vector<int> cerIndex) :

1. int start = 0; //初始选一个中心点
int index = start; //相当于指针指示新中心点的位置
int k = 0; //中心点计数，也即是类别
int dataNum = data.rows; //输入的样本数
2.
 - 2.1 首先把第一帧作为第一类中心点C1，计算所有点与它的距离记录为矩阵maxdistance，并将所有点的类别矩阵计为C1类，更新classes向量，并选出最大距离d的点为第二个样本中心点C2，也是当前类中心；
 - 2.2 计算所有点与当前类中心的距离计为矩阵distance并与maxdistance比较，用较小的距离更新maxdistance,且将更新过距离的点分为当前类更新classes向量，并将更新后的maxdistance的最大值计为当前类距离D，且最大值点作为当前类中心
 - 2.3 比较D与d×theat (阈值，可选0-1之间，一般0.5)，若 > ，则重复2.2 和 2.3 否则结束；否则循环结束。