**DTFV源码说明**

1. **文件简介：**

Script文件夹里面是原作者写的python脚本，直接从视频文件到fv特征输出，后续可以借鉴一下，只需要修改相应的文件路径（DenseTrackStab、compute\_fv、pca因子矩阵、码书）。

Data文件夹是测试数据。

Src文件夹里面是源码（包括采样的select\_pts，训练pca的train\_pca，训练码本的train\_gmm，计算fv的compute\_fv）。

1. **功能描述：**

通过输入iDT数据，对原数据采样，用采样数据训练pca，训练gmm，然后对所有数据训练pca后，计算fv。

1. **输入说明：**
   1. Select\_pts，该文件调用时输入的参数只有一个输出目标文件的地址（精确到文件名）。
   2. Train\_pca，输入参数如下inputFile outputMat projDimension [projectedFile]，即就是输入特征路径，PCA因子矩阵输出路径，通过PCA降到的维数，输入特征经过PCA因子矩阵降维后的特征矩阵。
   3. Train\_gmm，输入参数如下inputData outputCodeBook numClusters，inputData是降维后的特征，outputCodeBook是训练后的码本，numClusters是高斯聚类个数。
   4. Compute\_fv，输入参数源文件中只有输出地址。
2. **处理过程说明：**

需要提醒的是使用Select\_pts时，该函数原文件中使用的是控制台输入，建议修改成使用文件流。作者使用的是从生成随机数，通过判断随机数的大小，采集10000个样本点。然后再等间隔100取样，这会导致样本点太少无法训练gmm。因此可以自行修改源文件中cache和后续采样间隔的大小，使最终获得足够的样本点并输出。

获得输出的样本数据之后，对其使用train\_pca训练pca，获得pca因子矩阵和降维后的数据，对降维后的数据使用train\_gmm训练gmm，获得码本。然后根据自己数据集的pca因子矩阵和码本修改.lst文件，对所有数据compute\_fv。

1. **输出说明：**
   1. Select\_pts输出的是采样目标文件（Sample.traj，Sample.hof，Sample.Hog，Sample.mbhx，Sample.mbhy）。
   2. Train\_pca输出PCA因子矩阵（wenjian.hof/hog/traj/mbhx/mbhy.mat）和降维后的文件(wenjian.pca.hof/hog/traj/mbhx/mbhy)。
   3. Train\_gmm输出训练后的码本(wenjian.hof/hog/traj/mbhx/mbhy.gmm)。
   4. Compute\_fv输出训练后fv特征(5种描述子训练后的特征分别保存到相应的文件中)。
2. **操作环境说明：**

需要安装vl\_feat工具箱，编译好vl\_feat工具箱后，将vlfeats/bin/glnx64中的libvl.so文件拷贝到dtfv/src/vl文件夹中。

其中需要先修改动态链接库的查找目录，

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:dtfv-master/src/vl

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH: dtfv-master/src/alglib

(建议将路径改为绝对路径)。

或者使用vim打开/etc/ld.so.conf文件，将上述库路径加入后，sudo ldconfig。