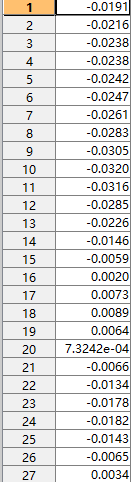
1. 已有数据视频特征feature.mat下video\_feature 15\*20\*31\*1514, 1514表示每一帧，也就是每帧对应15\*20\*31矩阵
2. 音频提取：

主要思路：音频其实是一串连续数据集合，那么只要截取每帧所对应的数字集合然后对其melcepst，这样就能保证音频和视频每帧都是对应的。

|  |
| --- |
| mel\_zlei.m |
| clear;  clc;  close all;  addpath('H:\dev\small\_programs\movie\_music\voicebox\voicebox');  [S,FS,Nbits] = wavread('H:\dev\small\_programs\movie\_music\ffmpeg-0.5\ffmpeg-0.5\234.wav'); %S表示音频数据，1541944\*1矩阵，现将1541944大致均分成1514个部分，也就是每帧对应1018个序列    C=melcepst(S(1:1018,1),FS,12,20);  for i=1:1514%循环是对每帧的音频数据循环，然后对每帧音频序列提取特征  voice\_feature(i,:,:) = melcepst(S(1018\*(i-1)+1:1018\*i,1),FS,12,20); %12是默认的，20表示滤波器个数，也就是得出的特征里面的2\*20中的20  end |

音频序列：



得到的音频特征在feature.mat下，voice\_feature矩阵，是一个1514\*2\*20矩阵，也就是每帧对应2\*20一个矩阵。

1. Cca特征融合函数

|  |
| --- |
| ccaFuse.m |
| function trainZ = ccaFuse(trainX, trainY, mode)    [n,p] = size(trainX);  if size(trainY,1) ~= n  error('trainX and trainY must have the same number of samples.'); % 如果特征数量不一样的话就报错  elseif n == 1  error('trainX and trainY must have more than one sample.');  end  q = size(trainY,2);      if ~exist('mode', 'var')  mode = 'sum'; % 默认融合模式是加和  end      %% 变量取中心，规整数据，为了下一步计算协方差矩阵<http://stackoverflow.com/questions/13703063/calculating-covariance-matrix-in-matlab>可以参考    meanX = mean(trainX);  meanY = mean(trainY); % 列取平均  trainX = bsxfun(@minus, trainX, meanX);  trainY = bsxfun(@minus, trainY, meanY); % 每个值减去平均值      %% 用PCA对数据X降维    % 计算协方差矩阵  if n >= p  C = trainX' \* trainX; % pxp  else  C = trainX \* trainX'; % nxn  end    % 特征值分解  [eigVecs, eigVals] = eig(C);  eigVals = abs(diag(eigVals));    % 忽略为0特征值  maxEigVal = max(eigVals);  zeroEigIdx = find((eigVals/maxEigVal)<1e-6);  eigVals(zeroEigIdx) = [];  eigVecs(:,zeroEigIdx) = [];    % 降序排列  [~,index] = sort(eigVals,'descend');  eigVals = eigVals(index);  eigVecs = eigVecs(:,index);    % 得到投影矩阵  if n >= p  Wxpca = eigVecs;  else  Wxpca = trainX' \* eigVecs \* diag(1 ./ sqrt(eigVals));  end  clear C eigVecs eigVals maxEigVal zeroEigIndex    % 更新矩阵  trainX = trainX \* Wxpca;      %% 对数据Y进行PCA分解    % 同上  if n >= q  C = trainY' \* trainY; % qxq  else  C = trainY \* trainY'; % nxn  end    % 同上  [eigVecs, eigVals] = eig(C);  eigVals = abs(diag(eigVals));    % 同上  maxEigVal = max(eigVals);  zeroEigIndex = find((eigVals/maxEigVal)<1e-6);  eigVals(zeroEigIndex) = [];  eigVecs(:,zeroEigIndex) = [];    % 同上  [~,index] = sort(eigVals,'descend');  eigVals = eigVals(index);  eigVecs = eigVecs(:,index);    % 同上  if n >= q  Wypca = eigVecs;  else  Wypca = trainY' \* eigVecs \* diag(1 ./ sqrt(eigVals));  end  clear C eigVecs eigVals maxEigVal zeroEigIndex    % 取得降维矩阵  trainY = trainY \* Wypca;      %% Fusion using Canonical Correlation Analysis (CCA)用CCA融合    [Wxcca,Wycca] = canoncorr(trainX,trainY); % cca函数    trainXcca = trainX \* Wxcca;  trainYcca = trainY \* Wycca;      if strcmp(mode, 'concat') % Fusion by concatenation (Z1)用连接来进行融合  trainZ = [trainXcca, trainYcca];  else % Fusion by summation (Z2)用加和来进行融合  trainZ = trainXcca + trainYcca;  end |

以上是cca特征融合函数

顺便写了个测试函数

|  |
| --- |
| Test.m |
| x = round(8\*rand(20,15))  y = round(8\*rand(20,2))  z = ccaFuse(x,y) |

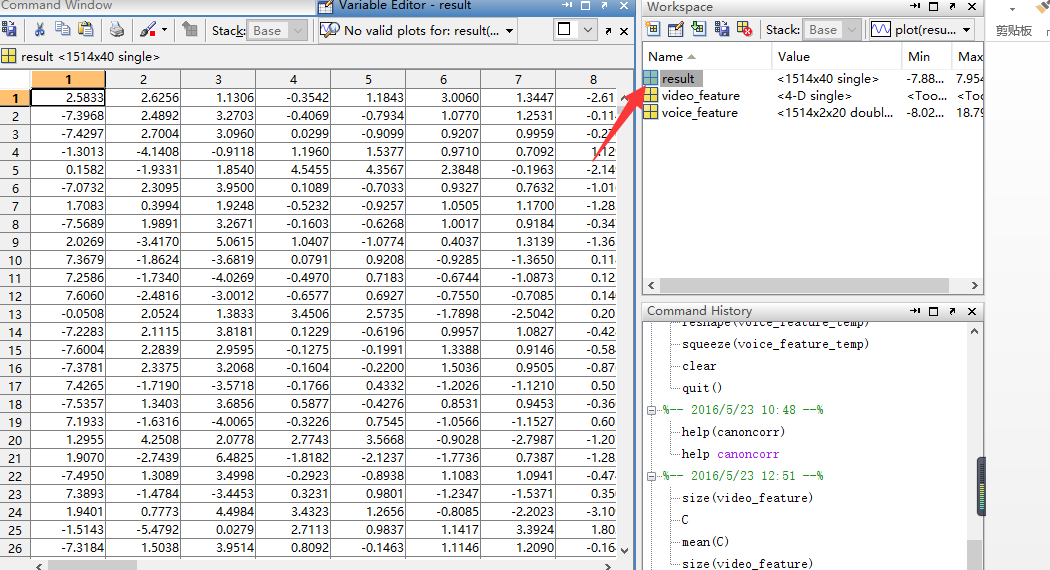
可以随机一个x是20\*15矩阵，随机一个y是20\*2，只要第一个数字是一样的就行，不然不能融合，!!因为是特征融合必须特征数量是一样的！！我们可以得到一个Z融合后矩阵，也就20\*2的矩阵

1. 将视频和音频数据融合

|  |
| --- |
| main.m |
| for i=1:1514%对每一帧进行循环  video\_feature\_temp = video\_feature(:,:,:,i); %回顾一下，video\_feature也就是视频特征数据也就是你给我的数据是15\*20\*31\*1514，这里提取第i帧数据也就是15\*20\*31  voice\_feature\_temp = voice\_feature(i,:,:); %同样，voice\_feature是第2步中1514\*2\*20矩阵，这边提取每帧所对应音频特征2\*20矩阵  %combine video feature temp  temp = video\_feature\_temp(:,:,1);  for j=2:31  temp = [temp;video\_feature\_temp(:,:,j)];  end  video\_feature\_temp = temp';%20\*465result这边上面几行将视频特征15\*20\*31变成20\*456矩阵，为了下一步特征融合，因为知道融合条件是特征数量是一致的  voice\_feature\_temp = squeeze(voice\_feature\_temp);  voice\_feature\_temp = voice\_feature\_temp';%20\*2result将音频数据变成20\*2的矩阵，本来是2\*20矩阵  result\_temp = ccaFuse(video\_feature\_temp,voice\_feature\_temp);%cca融合得到结果是20\*2矩阵  result\_temp = result\_temp';  result\_temp = [result\_temp(1,:),result\_temp(2,:)];%result将20\*2矩阵转换成40\*1矩阵为了保存进行下一步SVM  result(i,:) = result\_temp; %全部都保存在result矩阵中  end |

1. 结果

result\_cca.mat中的result矩阵



是一个1514\*40的矩阵，1514表示每一帧，40表示融合特征，之后能进行SVM等运算