@author 李昀哲

步骤及分析：

- 生成两个频谱S（原音频）和clip\_S（截取片段）。例子中：S(257\*3987), clip\_S(257\*311)

其中：clip\_S矩阵大小固定

- 进行最大池化处理（自定义代码如下）

1. 对矩阵resize，能保证被3\*3池化

|  |
| --- |
| **function [dst\_img] = resize(img, s) %s为池化窗口大小**  **is\_divide = 0;**  **a = zeros(1, size(img, 2));**  **while is\_divide == 0**  **if mod(size(img,1), s) == 0**  **is\_divide = 1;**  **break;**  **end**  **%disp(img);**  **%disp(a);**  **img = [img;a];**  **end**  **b = zeros(size(img,1), 1);**  **is\_divide = 0;**  **while is\_divide == 0**  **if mod(size(img,2), s) == 0**  **is\_divide = 1;**  **break;**  **end**  **img = [img b];**  **end**  **dst\_img = img;**  **end** |

1. 找出3\*3中最大的作为结果保存，最大池化函数如下：

|  |
| --- |
| **function [dst\_img] = max\_pooling(img, s)**  **num\_3\_row = size(img, 1)/s;**  **num\_3\_column = size(img, 2)/s;**    **result = [];**    **for i = 1:num\_3\_row**    **for j = 1:num\_3\_column**  **if i == 1**  **if j == 1**  **a = img(1:s, 1:s);**  **else**  **a = img(1:s, 1\*s\*(j-1):s\*j);**  **end**  **b = max(a);**  **c = max(b);**  **result = [result c];**  **continue**  **end**    **if j == 1**  **a = img(1\*s\*(i-1):s\*i, 1:s);**  **else**  **a = img(1\*s\*(i-1):s\*i, 1\*s\*(j-1):s\*j);**  **end**  **b = max(a);**  **c = max(b);**  **result = [result c];**  **end**    **end**    **dst\_img = reshape(result, num\_3\_row, num\_3\_column);**    **end** |

- 得到最大池化后的S\_resize(86\*104)和clip\_S\_resize(86\*1579)

- 计算相关性

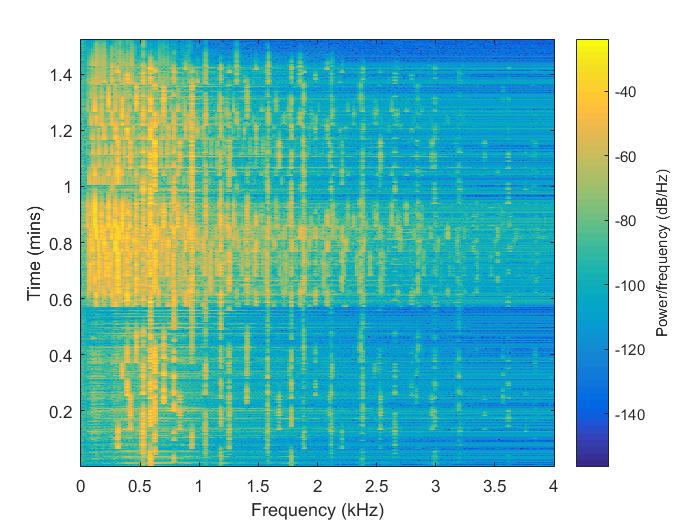
|  |
| --- |
| **function relate = calcu\_relate(S, clip\_S, step)**  **%S = [86 xxx]**  **%clip\_S = [86 104]**  **relate\_X = [];**  **for i = 1:step:size(S, 2)**  **if i + 103 > size(S, 2)**  **S\_part = S(:, end-103:end);**  **r = corrcoef(S\_part, clip\_S);**  **%取结果的一行二列的元素**  **r = r(1, 2);**  **relate\_X = [relate\_X r];**  **break**  **end**  **S\_part = S(:, i:i+103);**  **r = corrcoef(S\_part, clip\_S);**  **r = r(1, 2);**  **relate\_X = [relate\_X r];**  **end**  **relate = max(relate\_X);**  **end** |

* 1. 池化窗口的选择暂且选择为3\*3。
  2. 池化的作用：结合网上查阅资料及前阶段阅读的Dive-into DL中的关于过拟合和欠拟合的问题，池化主要用于对于过拟合现象导致的错误，但对于池化导致的原数据的特征降维，可能会导致一些重要信息的丢失。

1. 滑动距离不宜太短，也不宜和矩阵列数一致，经过测试，步长设置为110较为合适。
2. 实现思路如上所示，代码如下：

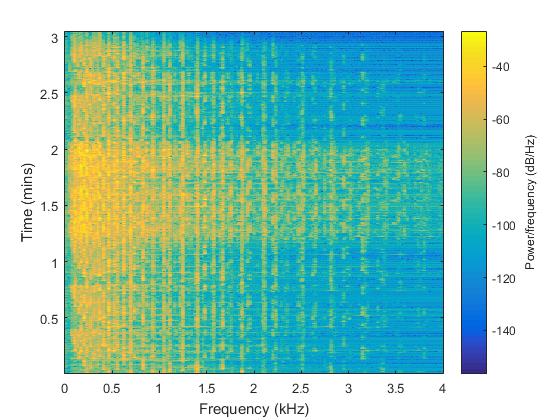
|  |
| --- |
| **%片段音频**  **[clip, Fs\_clip] = get\_tune\_noise('u1234567','sines.wav');**  **%[clip, Fs\_clip] = get\_tune\_noise('u9876543','hfnoise.wav');**  **clip = resample(clip, 8000, Fs\_clip);**    **window = 64;**  **noverlap = 32;**    **clip\_S = spectrogram(clip, 8000/1000\*window, 8000/1000\*noverlap, 8000/1000\*window, 8000);**  **clip\_S = real(clip\_S);**    **max\_relate = 0;**  **file = 0;**  **%获取原音频**  **for i = 1:39**  **disp(i)**  **if i < 10**  **[y, Fs\_y] = audioread(['rudenko\_0', num2str(i), '.mp4']);**  **else**  **[y, Fs\_y] = audioread(['rudenko\_', num2str(i), '.mp4']);**  **end**  **%得到原音频**  **y = resample(y/max(y)\*max(clip), 8000, Fs);**  **window = 64;**  **noverlap = 32;**  **S = spectrogram(y, 8000/1000\*window, 8000/1000\*noverlap, 8000/1000\*window, 8000);**  **S = real(S);**    **%进行匹配**  **%resize**  **clip\_S\_resize = resize(clip\_S,3);**  **S\_resize = resize(S,3);**  **%max\_pooling**  **clip\_S\_final = max\_pooling(clip\_S\_resize, 3);**  **S\_final = max\_pooling(S\_resize, 3);**  **%calculate rate**  **relate = calcu\_relate(S\_final, clip\_S\_final, 110);**  **disp(relate);**  **if relate > max\_relate**  **max\_relate = relate;**  **file = i;**  **end**    **end**    **disp(file)** |

[clip, Fs\_clip] = get\_tune\_noise('u1234567', 'sines.wav')的结果为‘rudenko\_07’



位置在551列-665列区域内。

[clip, Fs\_clip] = get\_tune\_noise('u9876543', 'hfnoise.wav')的结果为‘rudenko\_02’



位置在1列-104列区域内。