

# 復旦大學



## 本科生课程项目报告

课程名称: 数字信号处理

课程代码: COMP130139.01

姓 名: 陈杨栋

学 号: 15307130072

学 院: 计算机学院

专 业: 计算机科学与技术

# 数字信号处理课程项目——语音识别项目报告

15307130072 陈杨栋

## 一、问题介绍

本次课程项目，在收集全班同学说 20 个指定中英文单词的语音信号后，通过数字信号处理和模式识别的技术，对老师实时说出的单词进行预测。这个问题属于语音识别领域的一个极小的子问题。如今随着计算机技术的飞速发展，人们对于人机交互的愿望也越来越强烈，而要让机器明白人在说什么，进行语音识别就是一个首先要解决的关键问题。目前，语音识别主要面临下面这些难题：首先，自然语言的识别需要将连续的讲话分为词、音素等单位，在本次项目中，由于只有单个词，这一问题还不算突出；然后是人的说话方式不同，不同的人说同样的词，其频率受不同人的身体条件影响，而同一个人说同一个词，在状态不同的时候也有不小的差距；接着是语音的模糊性，比如在本次实验中，“语音”和“余音”，“背景”和“北京”这两组词，从发音上来讲就极为相似，很难有效区分同组两个词之间的区别，但是这些词的意思又相去甚远，识别错误会让后面可能要继续进行的工作更加困难，但是同样，如果有上下文的帮助，对具体用词的区分又有一定的帮助，脱离语境，这两组词较难区分；最后，环境噪声也是不可避免的，在本次数据的录制过程中，有的同学就处在相对嘈杂的环境中，如果有效剔除环境噪声对语音识别的干扰也是需要解决的一个问题。下面介绍一些比较常用的应对这种语音识别任务的方案。

## 二、相关方法介绍

通过本学期的学习，我们首先了解到的是一套相对传统的方案，首先进行端点检测，然后对实际有效的语音区域提取特征，主要用到快速傅里叶变换、Mel

滤波器组、差分方程提取 MFCC 特征等技术，然后对提取到的语音特征建立参考模板或者进行模式匹配，可用的技术有 VQ 码本、隐马尔科夫模型等方案，或者也可以采用更加先进的 SVM 乃至神经网络等技术对特征进行分类。这些方案的基础，都是基于对语音信号的时频分析，然后通过数字信号处理领域多年积累的经验得到的一些能用于语音识别的特征，这些特征高度浓缩了语音中的关键信息，但同样会损失掉一些细微的差别，而且隐马尔科夫模型更是面临训练时间较长，但分类效果不够理想的问题。通过查阅资料，我发现有一种更为快速，对特征保留也更加完整的方案：语谱图+卷积神经网络。这一方案的核心思想使用卷积神经网络对经过短时傅里叶变换的语音时频特征图进行分类，由于卷积神经网络良好的平移、尺度、形变不变性，可以较好地适应语音信号多变的特点，同时，也可以免去端点检测等一些前期处理工作，由于短时傅里叶变换可以较好的同时保留语音的时域和频域两方面的特征，其不会丢失太多关键信息，可能会对识别效率的提升起到积极作用。

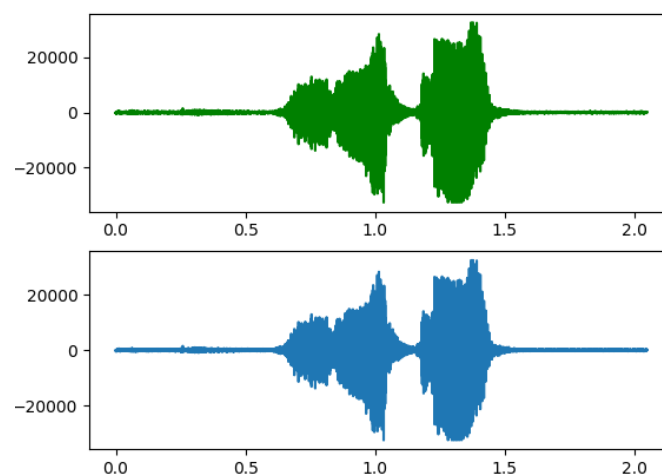
### 三、算法介绍

首先介绍算法的总体框架和目标，算法主要分为两个部分：从语言信号的.WAV 文件生成二维图片格式的语谱图，然后构建合适的卷积神经网络，使用生成的语谱图进行训练，由于个体的差异性，并且最终测试的老师语音并没有相关训练集，要注意搭建网络需要有较好的泛化能力，同时测试时要保证测试集和训练集不能同时存在同一个人的声音，这样可以得出相对可靠的准确度信息。此外，由于算力仅仅局限于自己的电脑，网络规模必须足够小，要在能够接受的时间内训练出需要的模型，深层的神经网络在这次项目中并不合适。在整个研究过程中，全部采用 Python 作为编程语言，一方面它是机器学习算法的主力语

言，方便前后对接，另一方面，它丰富的第三方库可以简化代码的编写，也可以对自己编写的代码互相检验，保证结果的合理性。本次实验主要需要实现的功能包括：(1) 对 WAV 格式文件的读取和处理，获得离散时间信号；(2) 对离散时间信号的快速傅里叶变换和短时傅里叶变换，并进行适当的加窗，并根据信号强度绘制语谱图；(3) 构建卷积神经网络，对语谱图进行分类，并保存好模型，便于后面的预测；(4) 整合前后功能并做好录音系统，便于接受测试。下面分别介绍这几部分内容：

### (1) 处理 WAV 格式的音频文件

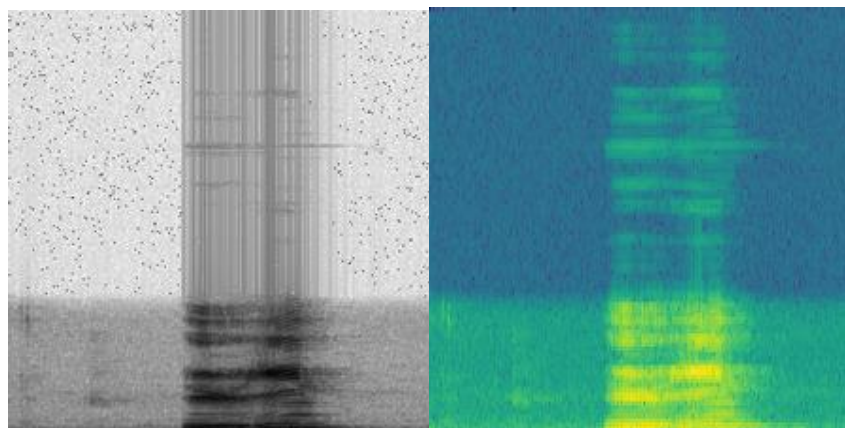
WAV 格式的文件主要由文件元数据和声音数据组成，Python 的原生库 wave 对此提供了支持，可以读取到一个 wav 文件的元数据（即基本信息，如采样率、帧数等）和具体的声音数据，读取到本次收集的数据文件的采样率为 48k 赫兹，双声道。两个声道分别存储在数据向量的奇数位置和偶数位置。我抽了一些样本，分别绘制了奇数和偶数位置的波形图，如下图所示



可见，在本次实验的采样中，两个声道信息基本一致，故我在绘制语谱图的时候只需取其中一个声道即可得到完整的语音信息。直接拷贝通过 wave 库读取到的 frame 的奇数位置作为离散时间信号，即可进行下一步操作。

## (2) 短时傅里叶变换

得到离散时间信号后，我们需要做的是对这个离散信号作短时傅里叶变换，并根据变换结果绘制语谱图。首先需要确定一个窗长度和步长，通过对频率范围和 wav 文件帧数的判断，当窗长为帧率除以 80 的时候，绘制出的语谱图比较接近于方形，这样在对语谱图进行规整化的时候，也不容易丢失相关信息，而且从我肉眼的观察来看，也可以较好的区分不同语音的特征。确定好窗长之后，按照一般经验，将步长设置为窗长的一半，然后对数据加窗，这里我采用了 numpy 提供的汉明窗，可以直接得到离散的窗的幅值，乘上对应位置即可。然后就可以进行快速傅里叶变换，变换后对得到的频域信息求模值后取对数，这样就得到了当前窗的各个频率的信号强度，根据信号强弱确定当前图像的灰度，信号越强，颜色越深，将得到的强度归一化到 0-255 之间，即可得到图像的灰度矩阵，最后调整一下行列的坐标即得到了相应的语谱图。同样的，我根据上课学习到的快速傅里叶变换的知识，自己尝试写了一个快速傅里叶变换的程序，将不足 2 的幂次的数据补 0 扩展到 2 的幂次，然后可以拆分奇偶之后递归调用快速傅里叶变换，程序整体非常简单。最后，为了验证我自己写的语谱图的正确性，我调用了 matplotlib 库提供的函数 `specgram` 对同一段语音绘制的语谱图进行比对，如下图所示

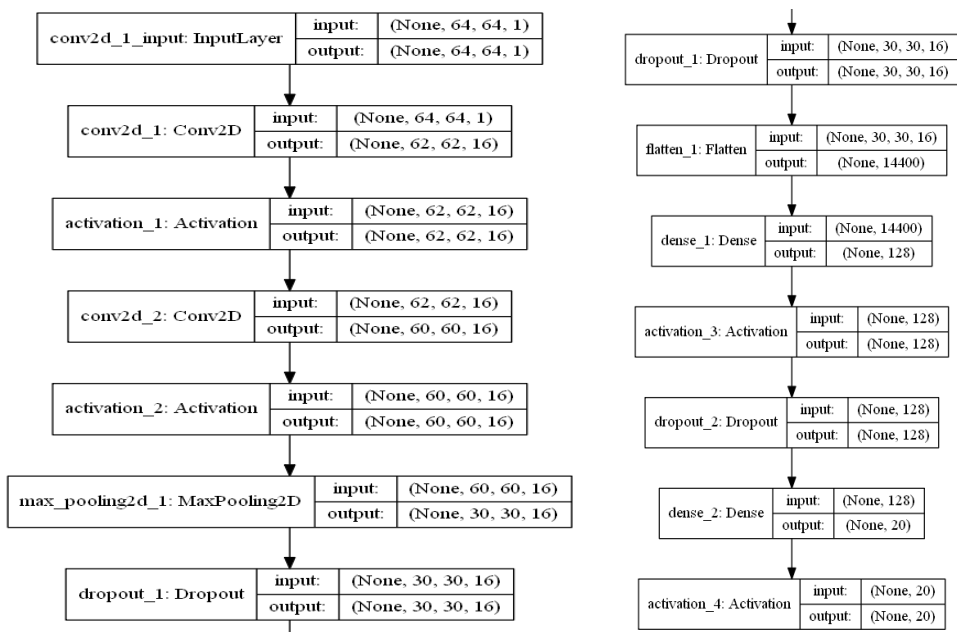


左图为我自己的程序跑出来的灰度图结果，右图为第三方库提供的结果，可以看出在低频区具有较高的相似性，高频区也大致相同，而影响人声音的位置主要集中在低频部分，基本可以认为，我的语谱图生成程序较为正确。而语谱图的颜色信息主要蕴含强度信息，灰度图和彩色区别不是太大，同时灰度图只有单通道，一定程度上精简了冗余信息，可以降低卷积神经网络的规模，使训练更加快速、高效。

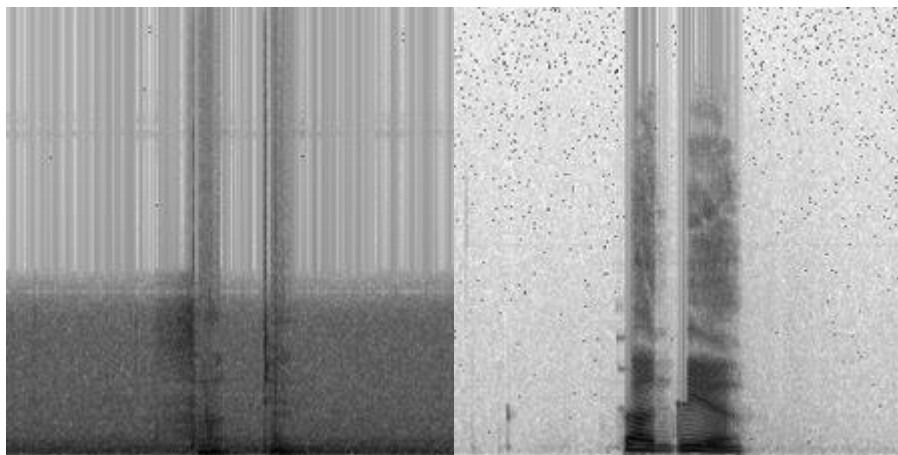
(3) 搭建并训练卷积神经网络

在本次实验中，由于使用到的网络较为直观，我采用了 TensorFlow 作为后端，keras 作为上层接口，可以十分迅速的构建出预想的网络。

在编写好生成语谱图的算法并对每个声音文件都生成对应的语谱图之后，就可以着手构建卷积神经网络了。受限于算力，我无法构建较深的网络，也来不及尝试多种多样的网络架构，首先，我搭建了一个最为简单的 CNN 进行了一次尝试，采用两个卷积层，一个池化层和一个全连接层，激活函数使用 relu，最后使用 softmax 激活函数输出最后分类。网络可视化架构如下图：



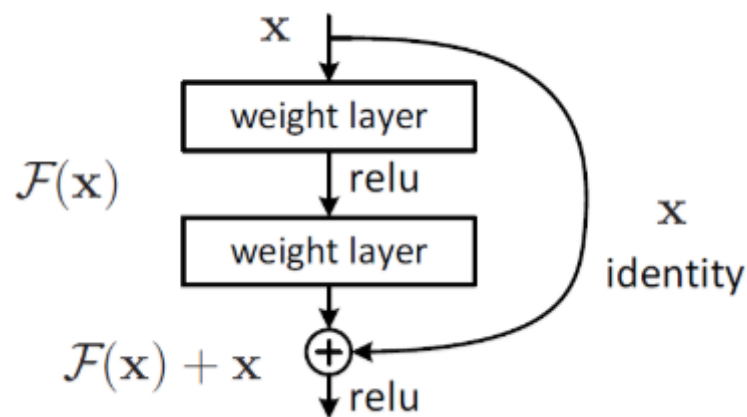
然后就可以对网络进行训练，并保存相应模型。由于这只是一个实验性的模型，我缩小了语谱图的规模，缩小到  $64 \times 64$  的规模，这样以 64 的 batch\_size 进行训练，训练一遍 12000 个样本只需要 8 秒钟，在训练至 30-40 遍的时候 loss 趋于收敛，在交叉验证集上的精度在 62% 左右。查看了一下预测错误的样本，发现主要错误如刚开始预测那样集中在两个地方，一个是“语音”“余音”、“背景”“北京”这两组词的互相混淆，一个是一位在测试集中的同学的样本过于嘈杂，噪声和人实际声音的区别太小，观察其语谱图和正常较为干净的同类语谱图对比如下，



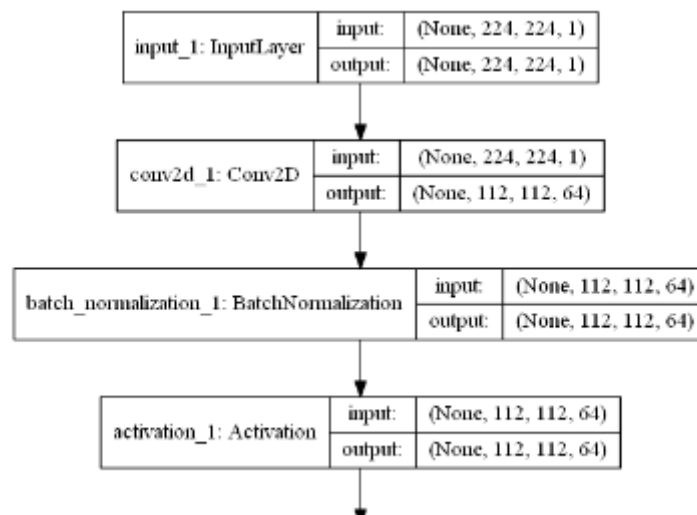
上图左侧为噪音严重的图，右侧为正常的图，可以看出，在低频部分，噪声点铺满了整个下半区，而根据对人发音的了解我们知道，人的声音也主要集中在低频部分，这意味着 CNN 在池化后提取特征的主要依据就是下半区的黑白对比度，而噪点太多，会使神经网络提出的特征的突出程度大大降低，考虑这种数据会让网络的识别能力起到反效果，我暂时删去这些数据，进行了第二次实验，发现准确率获得了一定的提升，达到了 75% 左右。

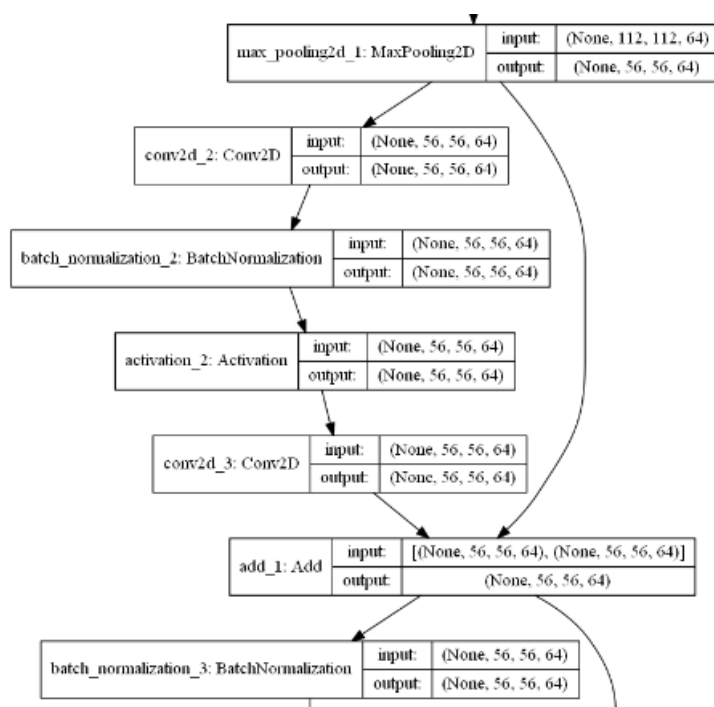
上述的实验验证了使用语谱图加上卷积神经网络进行语音分类的可行性，下面尝试使用更加先进的深度残差网络，这种网络的基本残差模块有两种，由于算力限制，我采用了 resnet18 和 34 所使用的基本残差块，并相应缩减了中间层的

规模，保证最终可以成功训练完成一个网络。基本残差块的模型如下图所示



关于残差网络为什么效果会更好，引用知乎上一位答主的回答，假设  $F$  是求和前的网络映射， $H$  是从输入到求和后的网络映射， $H(x)=F(x)+x$ 。比如把 5 映射到 5.1，那么引入残差前是  $F'(5)=5.1$ ，引入残差后是  $H(5)=5.1$ ， $H(5)=F(5)+5$ ， $F(5)=0.1$ 。这里的  $F'$  和  $F$  分别是普通网络和残差网络的映射，引入残差后的映射对输出的变化更敏感。比如原来是从 5.1 到 5.2，映射  $F'$  的输出增加了  $1/51=2\%$ ，而对于残差结构从 5.1 到 5.2，映射  $F$  是从 0.1 到 0.2，增加了 100%。明显后者输出变化对权重的调整作用更大，所以效果更好。这位答主认为残差网络起到了一个差分放大的作用，非常贴切。下图是使用 keras 提供的画图工具画出的构建好的 resnet 的局部模型



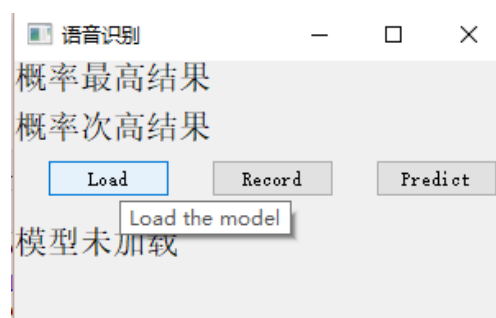


上图便是网络中一个典型的基本模块，残差网络由若干个这样的模块拼接而成，最后做一个平均池化并加上全连接层，便是最后的输出。Resnet 的具体构建方式我借鉴了一位博主的代码，自行搭建了一个类似 resnet 架构的网络，使之符合我自己的需求。

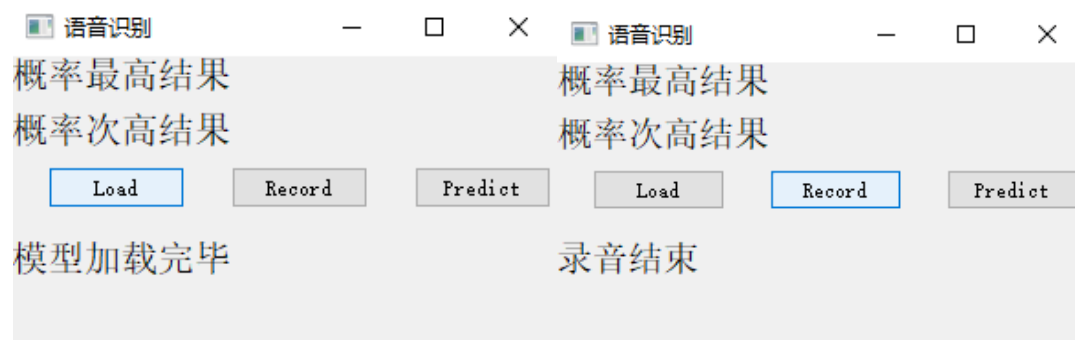
#### (4) 图形界面构建

在完成了模式匹配模型的建立之后，剩余的工作就是编写一个声音采集程序，并制作一个 UI 界面，值得注意的点就是要保证采集的 wav 文件格式和用于训练的文件格式一致，这样得出的语谱图才更加可靠，然后可以依据生成训练集的相同步骤，进行相应的训练。录音使用了 pyaudio 库。下面是我制作的界面截图和基本使用方法。

首先打开程序 python main.py，会进入第一个页面



上方两行显示预测结果，然后三个按钮对应加载预训练模型，录音和预测，鼠标停留会有按钮功能提示。最后一行是当前状态。加载完毕模型和录音完毕分别会有变化



如图，加载模型需要若干秒，可能会造成界面未响应，点击 predict 后会有两个结果显示，后面跟随的数字是模型对该结果的信任度。



#### 四、实验设置

首先，我使用上述的简单 CNN、64\*64 的灰度语谱图进行训练和测试，考虑到语音识别中的个体差异性和数据的重复方式，我划分出了几位同学的全部语音

作为测试集，首先使用了 4 位同学，包括一位环境噪声极大的样本，准确度如下

```
Epoch 100/100  
11194/11194 [=====] - 8s 724us/step - loss: 0.0768 - acc: 0.9746 - val_loss: 2.8751 - val_acc:  
0.6256  
Test score: 2.8750571166351437  
Test accuracy: 0.625625
```

其中，score 是在验证集上的 loss，accuracy 是对验证集进行测试得到的准确度。

随后我剔除了环境噪声较大的所有 400 个样本，用剩余的 1200 个人进行重新测试，结果如下

```
Epoch 100/100  
11194/11194 [=====] - 15s 1ms/step - loss: 0.0795 - acc: 0.9730 - val_loss: 1.0183 - val_acc: 0.  
.7592  
Test score: 1.0187633393208186  
Test accuracy: 0.7591666666666667
```

准确度和 loss 都有了极大的提高，经过计算可知在那位同学的 400 个样本中，仅能预测正确 20% 的情况，而随机就有 5% 的准确率，由此可见，在嘈杂环境中，对语音预测准确度的影响非常大。

在小规模的网络和数据上测试完毕后，我开始在 resnet 上进行训练和测试，这里的测试集仍然使用上面的 3 个人的准确度，由于算力不足，我训练了约 70000 个大小为 16 的 batch，此时训练集的 loss 每 1000 个 batch 下降以及不到 0.01，测试集的 loss 开始跳动，我停止了训练并保存好模型，最终，在测试集上的 loss 和准确度达到了如下水平，考虑那 0.05 正好等于两组易混词随机后的概率，理论上其他词以及可以较好的区分出来，而易混词和其他词也一样可以区分，只是互相之间无法区分。

```
Test score: 0.4835046601295471  
Test accuracy: 0.9508333333333333
```

#### 四、结论

通过本次实验，我自己构建了一个语音识别系统从前到后的一套过程，从录音到数据处理到模型构建再到最后的预测，最后的精度表明使用 CNN 识别语谱

图在有限词的判别这一任务中有不错的效果，resnet 是 2015 年先进的网络，到现在又出现了更多更强的网络，以及针对语谱图识别的 DFCNN 等效果更加强悍的网络，可以展望，在不远的未来，CNN 在语音识别中会展现出更为优秀的能力。

## 参考文献

- [1] <http://lanbing510.info/2017/08/21/ResNet-Keras.html>
- [2] <https://github.com/stensaethf/Spectrogram>
- [3] keras 官方文档
- [4] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2016:770-778.
- [4] <https://www.zhihu.com/question/53224378/answer/159102095>