AI导论 Group1 中期进度汇报 2020.11.20

黄钊恒 2018202064 郝文轩 2018200077

Task9 HumanEye

目录

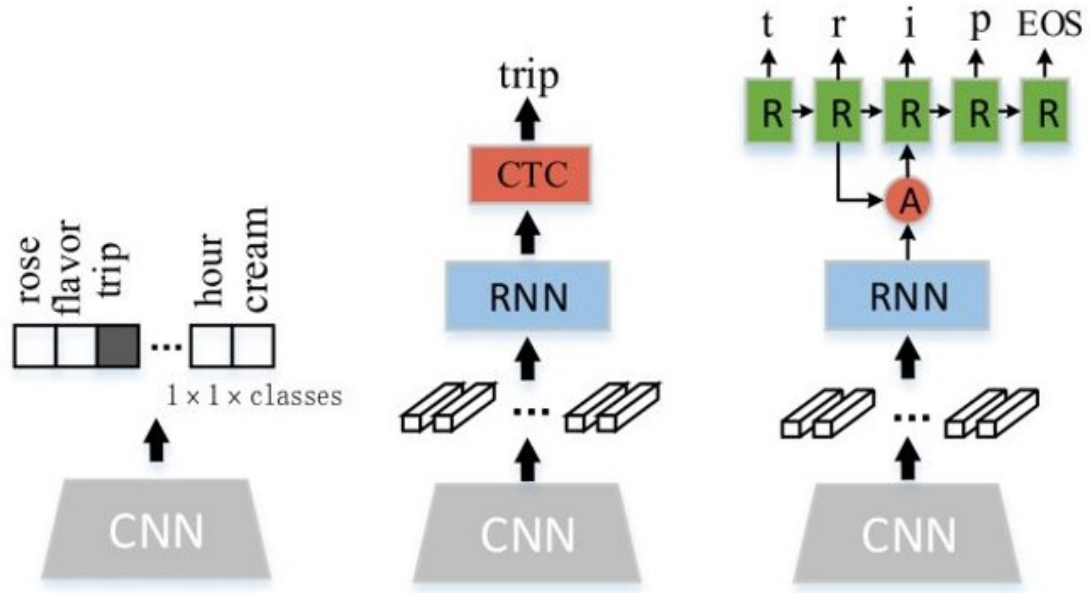
1. OCR

光学字符识别（Optical Character Recognition, OCR）是指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程；亦即将图像中的文字进行识别，并以文本的形式返回。

文字检测即检测文本的所在位置和范围及其布局。通常也包括版面分析和文字行检测等。文字检测主要解决的问题是哪里有文字，文字的范围有多大，目前流行的算法框架有：Faster R-CNN、FCN、RRPN、TextBoxes、DMPNet 、CTPN、SegLink……

文本识别是在文本检测的基础上，对文本内容进行识别，将图像中的文本信息转化为文本信息。文字识别主要解决的问题是每个文字是什么。识别出的文本通常需要再次核对以保证其正确性。文本校正也被认为属于这一环节。而其中当识别的内容是由词库中的词汇组成时，我们称作有词典识别（Lexicon-based），反之称作无词典识别（Lexicon-free）。

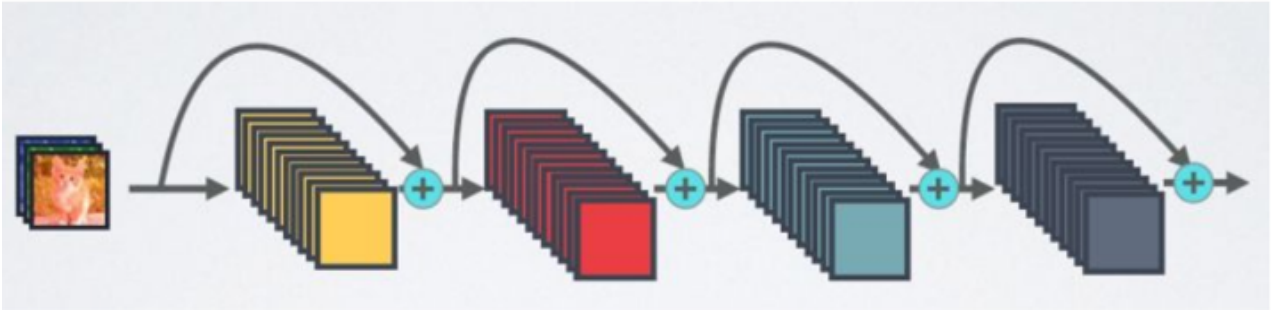
在此步骤之中，采用的算法核心结构主要为：CNN（DenseNet）、RNN（LSTM、GRU）、Attention机制……

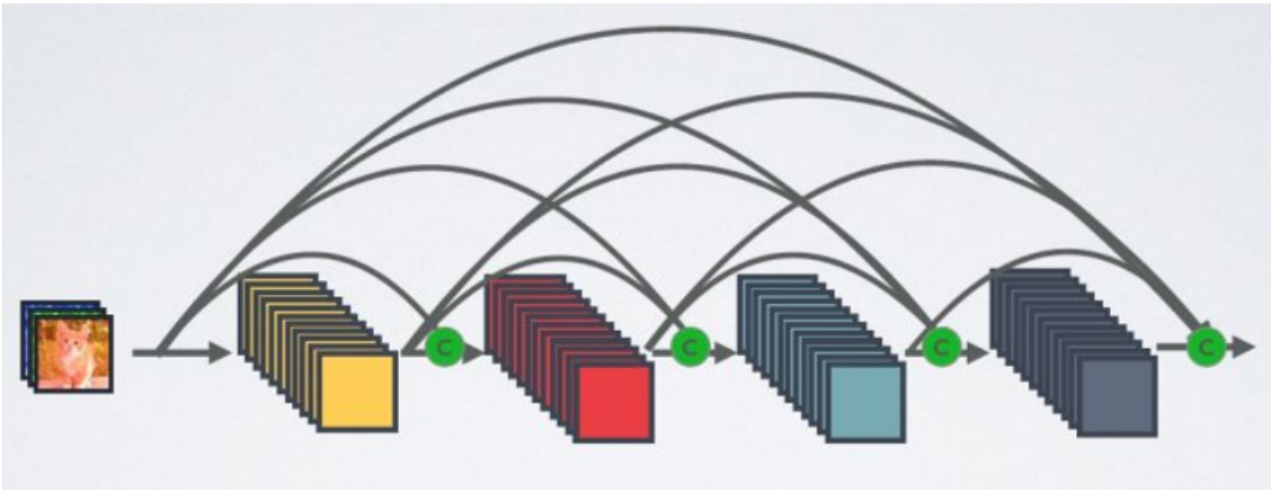


CNN: DenseNet

相比ResNet，DenseNet提出了一个更激进的密集连接机制：即互相连接所有的层，具体来说就是每个层都会接受其前面所有层作为其额外的输入。

结合以下两图我们可以看到，ResNet是每个层与前面的某层（一般是2~3层）短路连接在一起，连接方式是通过元素级相加。而在DenseNet中，每个层都会与前面所有层在channel维度上连接在一起（这里各个层的特征图大小是相同的），并作为下一层的输入。对于一个L层的网络，DenseNet共包含个$L\times(L-1)/2$连接，相比ResNet，这是一种密集连接。而且DenseNet是直接连接来自不同层的特征图，这可以实现特征重用，提升效率，这一特点是DenseNet与ResNet最主要的区别。



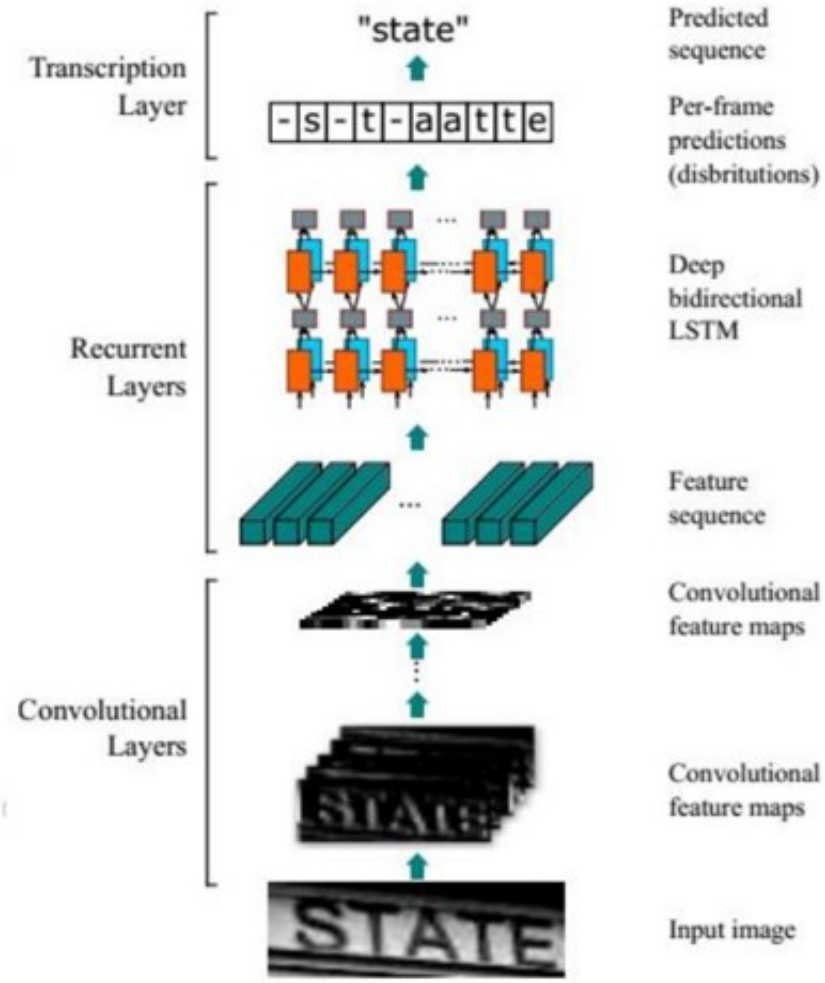


SimpleRNN -> LSTM

LSTM层是SimpleRNN层的一种变体，它增加了一种携带信息跨越多个时间步的方法。假设有一条传送带，其运行方向平行于你所处理的序列。序列中的信息可以在任意位置跳上传送带，然后被传送到更晚的时间步，并在需要时原封不动地跳回来。这实际上就是LSTM的原理：它保存信息以便后面使用，从而防止较早期的信号在处理过程中逐渐消失。

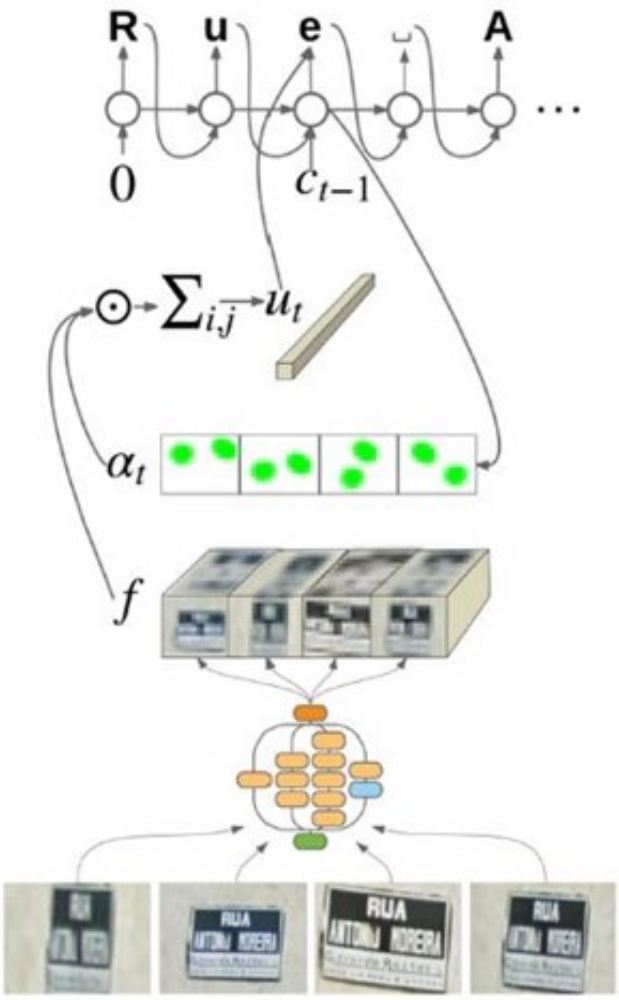
CNN➡双向LSTM➡（分类）➡CTC

利用CRNN模型。以CNN特征作为输入，双向LSTM进行序列处理使得文字识别的效率大幅提升，也提升了模型的泛化能力。先由分类方法得到特征图，之后通过CTC对结果进行翻译得到输出结果。

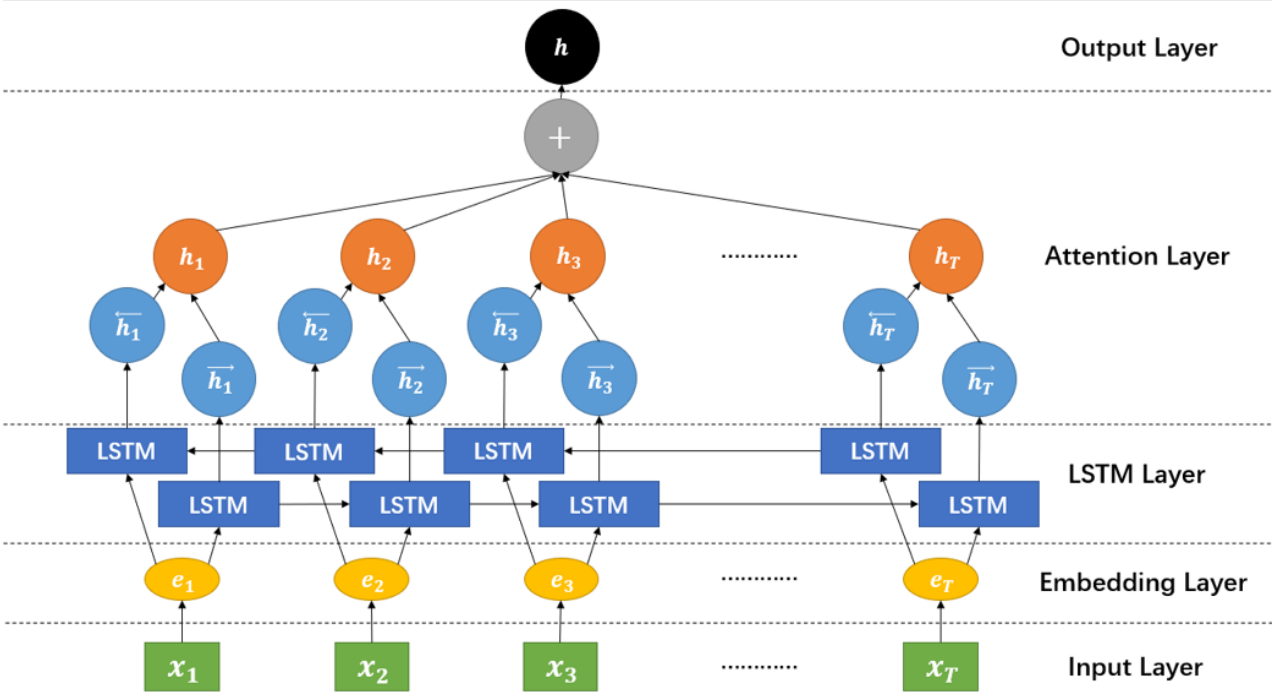


CNN➡Attention➡RNN

在此，我们引入注意力机制（Attention），即以CNN特征作为输入，通过注意力模型对RNN的状态和上一状态的注意力权重计算出新一状态的注意力权重。之后将CNN特征和权重输入RNN，通过编码和解码得到结果。



CNN➡双向LSTM➡Attention➡RNN➡CTC

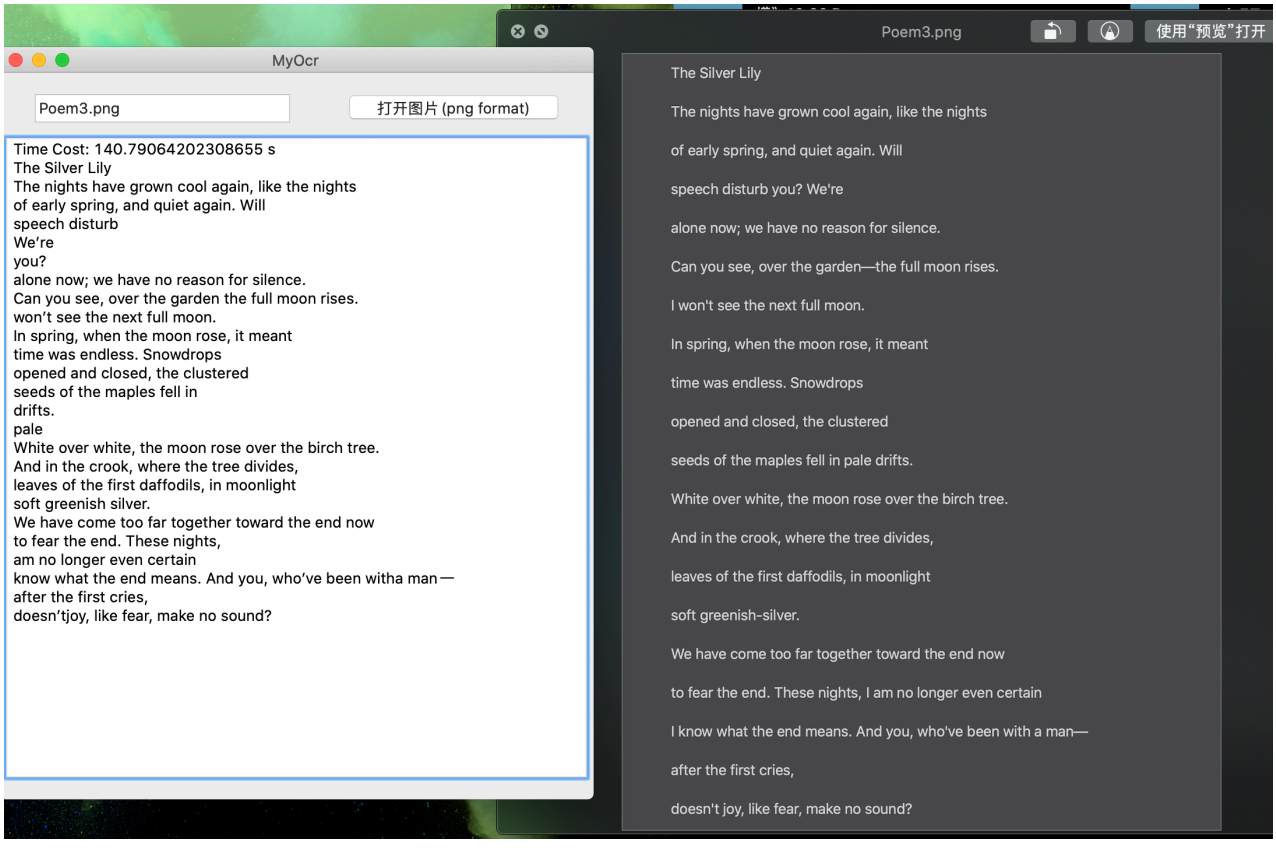


我们的OCR模型

结合上文，我们的逻辑结构为图像输入——预处理——图像检测——ResNet+LSTM+CTC——解码，结合在easyocr库中。

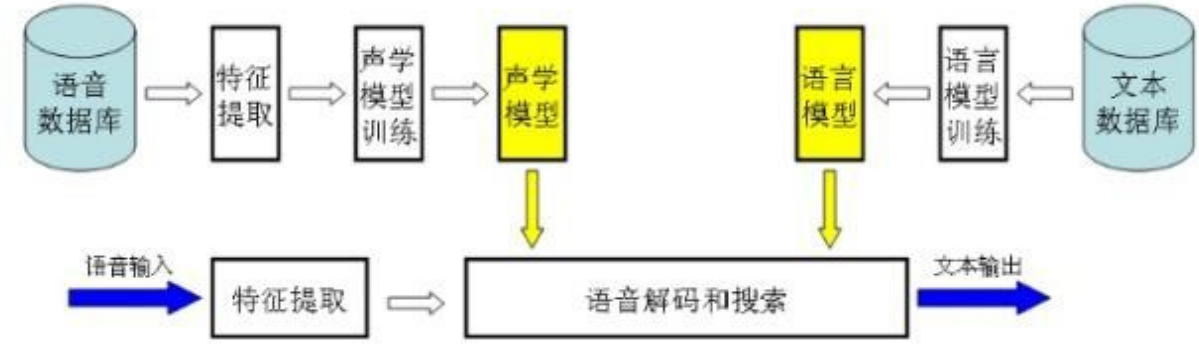
我们的模型演示

我们构建了一个UI界面，展示OCR识别结果，下图可以看出，微信的聊天记录截图，能被我们的OCR模型准确识别。

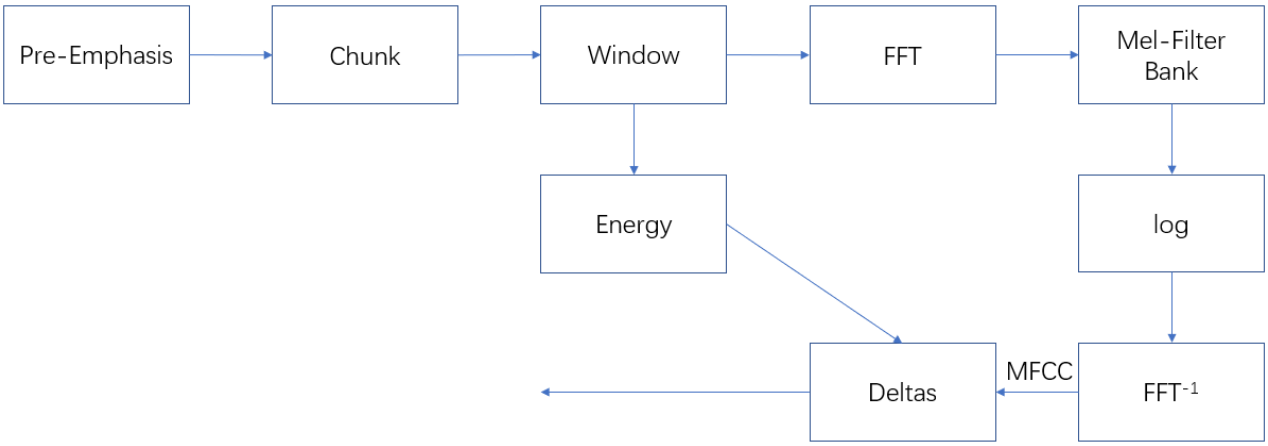


2. ASR模型

ASR（自动语音识别）就是将麦克风采集到的自然声音转化为文字的过程，相当于人的耳朵+大脑（一部分）。TTS技术（语音合成），是将文字转化为声音（朗读出来），类比于人类的嘴巴。大家在$Siri$等各种语音助手中听到的声音，都是由TTS来生成的，并不是真人在说话。TTS的技术实现方法，主要有两种："拼接法"和"参数法"。



流程框架



Pre-Emphasis预加重

预加重，即对语音信号的高频部分应用预加重滤波器进行加重。目的即为，为了去除口唇辐射的影响，放大语音的高频分辨率。其一般采取一阶FIR高通数字滤波器来实现预加重，公式如下：

$$H(z) = 1 - az^{-1}$$

其中a为预加重系数，$0.9<a<1.0$。

我们小组查找资料并总结预加重滤波器的作用有以下几个方面： （1）平衡频谱，因为高频通常比低频具有更小的幅度 （2）避免在傅立叶变换操作期间出现数值问题 （3）还可改善信号噪声比（SNR）

可以使用以下公式中的一阶滤波器将预加重滤波器应用于信号x。设t时刻的语音采样值为$x(n)$，经过预加重处理后的结果为$y(t)=x(t)-ax(t-1)$，其中滤波器系数a的典型值为0.95或0.97。

Chunk+Window+FFT

经过预加重后，我们需要将信号分成短帧。 此步骤的基本原理是：信号中的频率会随时间变化。因此，在大多数情况下，对整个信号进行傅立叶变换是没有意义的，因为我们会随时间丢失信号的频率轮廓。 为避免这种情况，我们可以假设信号的频率在很短的时间内是固定的，其特性可以看作是一个准稳态过程，即具有短时性。

加窗：与一个窗函数相乘，以此进行傅立叶展开（窗函数：一般具有低通特性，矩形窗：主瓣宽度最小，旁瓣高度最高，会导致泄漏现象。汉明窗：主瓣最宽，旁瓣高度最低，可以有效克服泄漏现象，具有更平滑的低通特性）。对语音进行加窗主要有两个目的：（1）使全局更加连续，避免出现吉布斯效应；（2）加窗时，原本没有周期性的语音信号呈现出周期函数的部分特征。

因此，通过在此短帧上进行傅立叶变换，我们可以通过串联相邻帧来获得信号频率轮廓较好的近似。公式如下：

$$F(\omega) = F[f(t)] = \int_{-\infty}^{+\infty} f(t)e^{-iwt}dt$$

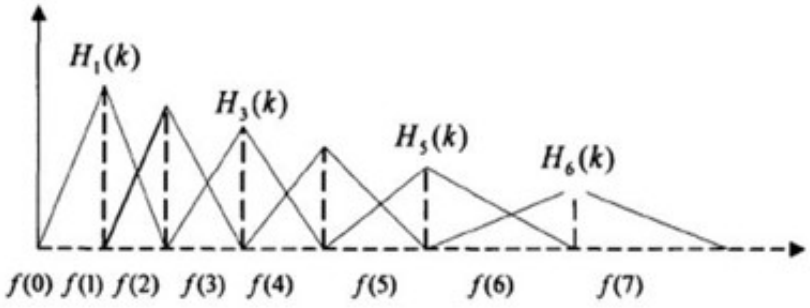
语音处理中典型帧大小为20毫秒至40毫秒，连续帧之间有50%± 10%的重叠。

Mel-Filter Bank

Mel滤波，通过Mel滤波器组进行滤波，以得到符合人耳听觉习惯的声谱，最后通常取对数将单位转换成db。很多文章中使用的Mel滤波器组，其赋值是一样的，都是1，即滤波器并不随宽的增加而改变，这样就导致了三角形的面积变化。因此，还有一种Mel滤波器的设计就是随着宽的增加而改变高度，保证其面积不变，公式如下：

$$H_m(k) = \begin{cases} 0 & ,k < f(m-1) \\ \dfrac{2(k-f(m-1))}{(f(m+1)-f(m-1))(f(m)-f(m-1))} & ,f(m-1) \le k \le f(m) \\ \dfrac{2(f(m+1)-k)}{(f(m+1)-f(m-1))(f(m)-f(m-1))} & ,f(m) \le k \le f(m+1) \\ 0 & ,k \ge f(m+1) \end{cases}$$

式中m代表第m个滤波器；k代表横轴坐标，也就是自变量；$f(m)$代表第m个滤波器的中心点的横坐标值。其效果图如下：

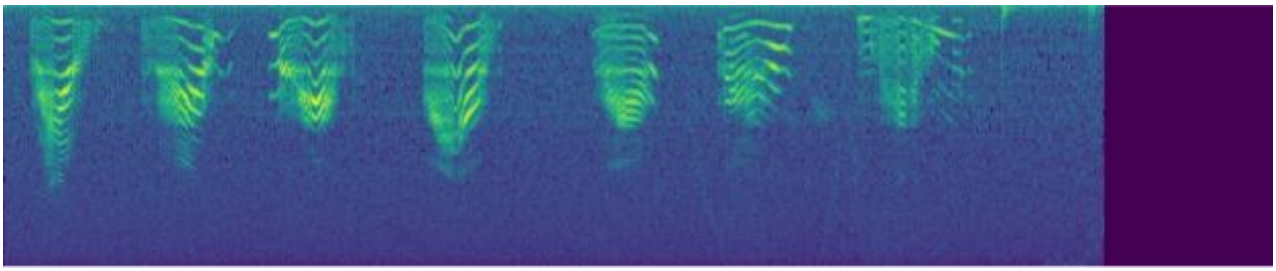


MFCC

*Filter banks*和*MFCC*语音特征提取，整体是相似的，*MFCC*只是多了一步*DCT*（离散余弦变换）罢了。通过*DCT*，得到倒谱系数，也就是*MFCC*，通常保留1~13维，然后可以加上*delta*，*delta-delta*，和每帧能量*energy*。公式如下：

$$\sqrt{\frac{1}{2N}} * 2 * \sum_{n=0}^{N-1} x^{'}[n] cos(\frac{(n+\frac{1}{2})\pi k}{N}) = \sqrt{\frac{2}{N}} * \sum_{n=0}^{N-1} x^{'}[n] cos(\frac{(n+\frac{1}{2})\pi k}{N})$$

*DCT*变换具有非常良好的频域能量聚集度（也就是能够把图像或音频中更重要的信息聚集），那么对于那些不重要的频域区域和系数就能够直接裁剪掉（类似淘金，石头里重要的金子都弄到一起，剩下没用的石子就可以扔掉）。我们小组将该步骤应用于猫叫声的渐进处理效果如下：



Energy

语音的短时能量即每一帧语音包含的能量。我们可以通过短时能量分析去除高频环境噪声的干扰。语音和噪声的区别可以体现在他们的能量上，语音段的能量比噪声段的能量大，如果环境噪声和系统输入的噪声比较小，只要计算输入信号的短时能量就可以把语音段和噪声背景区分开来。同时，基于能量的算法还可以用来检测浊音，通常效果也是比较理想的。

由语音短时能量的定义，就可以得到短时能量的计算公式：

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 = x(n)^2 * h(n)$$

其中，$w(n) = \begin{cases} 1 & 0 <= \mathrm{n} <= \mathrm{N}-1 \\ 0 & \text{其它} \end{cases}$

由于短时能量是语音的时域特征，因此，在不使用傅里叶变换的情况下，这里的窗口是一种方窗，我们推断可得，这里的语音短时能量就相当于每一帧中所有语音信号的平方和。

Deltas

deltas和deltas-deltas，被称为微分系数和加速度系数。使用它们的原因是：MFCC只是描述了一帧语音上的能量谱包络，但是语音信号似乎有一些动态上的信息，也就是MFCC随着时间的改变而改变的轨迹。文章表明，计算MFCC轨迹并把它们加到原始特征中可以提高语音识别的表现。

以下是deltas的一个计算公式，其中t表示第几帧，N通常取2，c指的就是MFCC中的某个系数。deltas-deltas就是在deltas上再计算一次deltas。

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

MFCC中每个系数都做这样的计算，最后会得到12个一阶差分和12个二阶差分，我们通常在论文中看到的“MFCC以及它们的一阶差分和二阶差分”指的就是这个。

值得一提的是，deltas和deltas-deltas也可以用在别的参数上来表述动态特性，有论文中是直接在log Mels上做一阶差分和二阶差分的，在此不一一赘述。

3. 知识图谱

知识图谱（ Knowledge Graph）的概念由谷歌2012年正式提出，旨在实现更智能的搜索引擎，并且于2013年以后开始在学术界和业界普及。目前，随着智能信息服务应用的不断发展，知识图谱已被广泛应用于智能搜索、智能问答、个性化推荐、情报分析、反欺诈等领域。另外，通过知识图谱能够将Web上的信息、数据以及链接关系聚集为知识，使信息资源更易于计算、理解以及评价，并且形成一套Web语义知识库。知识图谱以其强大的语义处理能力与开放互联能力，可为万维网上的知识互联奠定扎实的基础，使Web 3.0提出的“知识之网”愿景成为了可能。

知识图谱定义

知识图谱：是结构化的语义知识库，用于迅速描述物理世界中的概念及其相互关系。

知识图谱通过对错综复杂的文档的数据进行有效的加工、处理、整合，转化为简单、清晰的“实体,关系,实体”的三元组，最后聚合大量知识，从而实现知识的快速响应和推理。

知识图谱有自顶向下和自底向上两种构建方式。所谓自顶向下构建是借助百科类网站等结构化数据源，从高质量数据中提取本体和模式信息，加入到知识库中；所谓自底向上构建，则是借助一定的技术手段，从公开采集的数据中提取出资源模式，选择其中置信度较高的新模式，经人工审核之后，加入到知识库中。

技术架构：信息抽取➡知识融合➡知识加工（更新）

信息抽取

实体抽取（Entity Extraction）：实体抽取又称为命名实体识别（named entity recognition，NER），是指从文本数据集中自动识别出命名实体。

关系抽取（Relation Extraction）：文本语料经过实体抽取，得到的是一系列离散的命名实体，为了得到语义信息，还需要从相关的语料中提取出实体之间的关联关系，通过关联关系将实体（概念）联系起来，才能够形成网状的知识结构，研究关系抽取技术的目的，就是解决如何从文本语料中抽取实体间的关系这一基本问题。

属性抽取（Attribute Extraction）：属性抽取的目标是从不同信息源中采集特定实体的属性信息。例如针对某个公众人物，可以从网络公开信息中得到其昵称、生日、国籍、教育背景等信息。属性抽取技术能够从多种数据来源中汇集这些信息，实现对实体属性的完整勾画。

知识融合

实体链接（entity linking）：是指对于从文本中抽取得到的实体对象，将其链接到知识库中对应的正确实体对象的操作。其基本思想是首先根据给定的实体指称项，从知识库中选出一组候选实体对象，然后通过相似度计算将指称项链接到正确的实体对象。

知识合并：常见的知识合并需求有两个，一个是合并外部知识库，另一个是合并关系数据库。

将外部知识库融合到本地知识库需要处理两个层面的问题：

1. 数据层的融合，包括实体的指称、属性、关系以及所属类别等，主要的问题是如何避免实例以及关系的冲突问题，造成不必要的冗余；通过模式层的融合，将新得到的本体融入已有的本体库中。
2. 然后是合并关系数据库，在知识图谱构建过程中，一个重要的高质量知识来源是企业或者机构自己的关系数据库。为了将这些结构化的历史数据融入到知识图谱中，可以采用资源描述框架（RDF）作为数据模型。业界和学术界将这一数据转换过程形象地称为RDB2RDF，其实质就是将关系数据库的数据换成RDF的三元组数据。

知识加工

本体构建：本体（ontology）是指工人的概念集合、概念框架，如“人”、“事”、“物”等。本体可以采用人工编辑的方式手动构建（借助本体编辑软件），也可以以数据驱动的自动化方式构建本体。因为人工方式工作量巨大，且很难找到符合要求的专家，因此当前主流的全局本体库产品，都是从一些面向特定领域的现有本体库出发，采用自动构建技术逐步扩展得到的。

自动化本体构建过程包含三个阶段：

1. 实体并列关系相似度计算
2. 实体上下位关系抽取
3. 本体的生成

知识推理：在我们完成了本体构建这一步之后，一个知识图谱的雏形便已经搭建好了。但可能在这个时候，知识图谱之间大多数关系都是残缺的，缺失值非常严重，那么这个时候，我们就可以使用知识推理技术，去完成进一步的知识发现。我们可以发现：如果A是B的配偶，B是C的主席，C坐落于D，那么我们就可以认为，A生活在D这个城市。

质量评估：可以对知识的可信度进行量化，通过舍弃置信度较低的知识来保障知识库的质量。

知识更新

从逻辑上看，知识库的更新包括概念层的更新和数据层的更新。

概念层的更新是指新增数据后获得了新的概念，需要自动将新的概念添加到知识库的概念层中。 数据层的更新主要是新增或更新实体、关系、属性值，对数据层进行更新需要考虑数据源的可靠性、数据的一致性（是否存在矛盾或冗杂等问题）等可靠数据源，并选择在各数据源中出现频率高的事实和属性加入知识库。

知识图谱的内容更新有两种方式：

1. 全面更新：指以更新后的全部数据为输入，从零开始构建知识图谱。这种方法比较简单，但资源消耗大，而且需要耗费大量人力资源进行系统维护。
2. 增量更新：以当前新增数据为输入，向现有知识图谱中添加新增知识。这种方式资源消耗小，但目前仍需要大量人工干预（定义规则等），因此实施起来十分困难。

我们的知识图谱模型

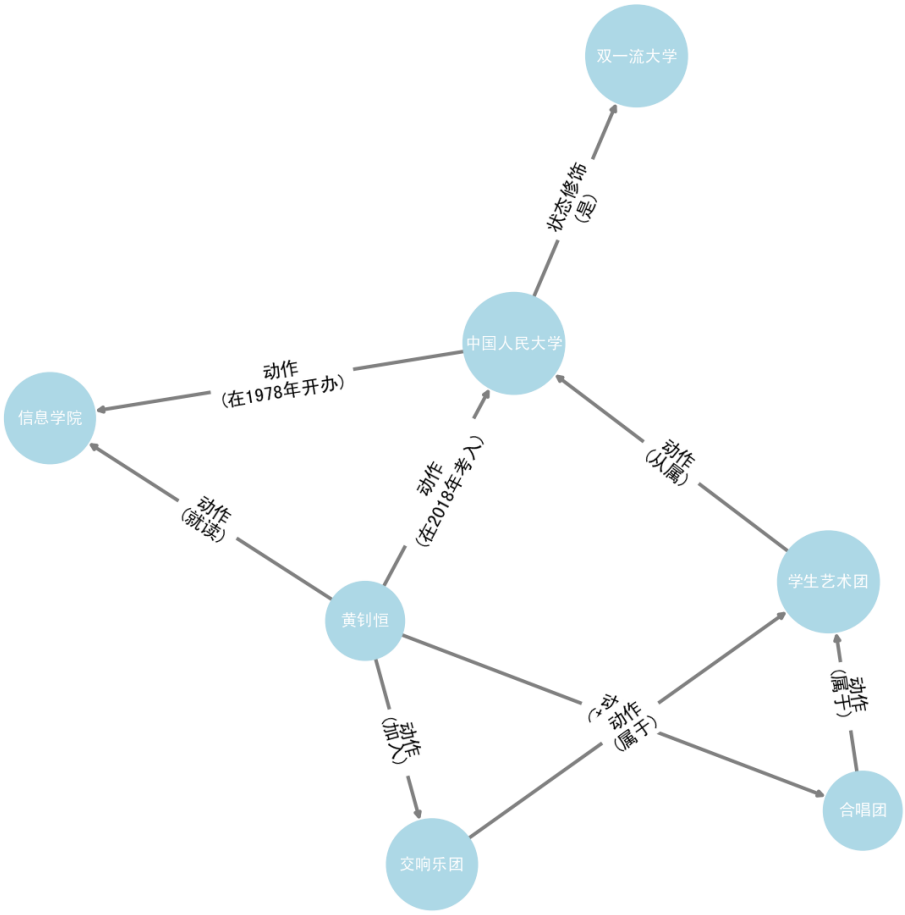
我们运用知识图谱技术获得三元组的信息与可信度，并使用networkx第三方库可视化。

如图所示，你可以看到，如果两个节点之间存在关系，他们就会被一条无向边连接在一起，那么这个节点，我们就称为实体（Entity），它们之间的这条边，我们就称为关系（Relationship）。

知识图谱的基本单位，便是“实体（Entity）-关系（Relationship）-实体（Entity）”构成的三元组，这也是知识图谱的核心。

实体: 指的是具有可区别性且独立存在的某种事物。实体是知识图谱中的最基本元素，不同的实体间存在不同的关系。如图中的【中国人民大学】等。

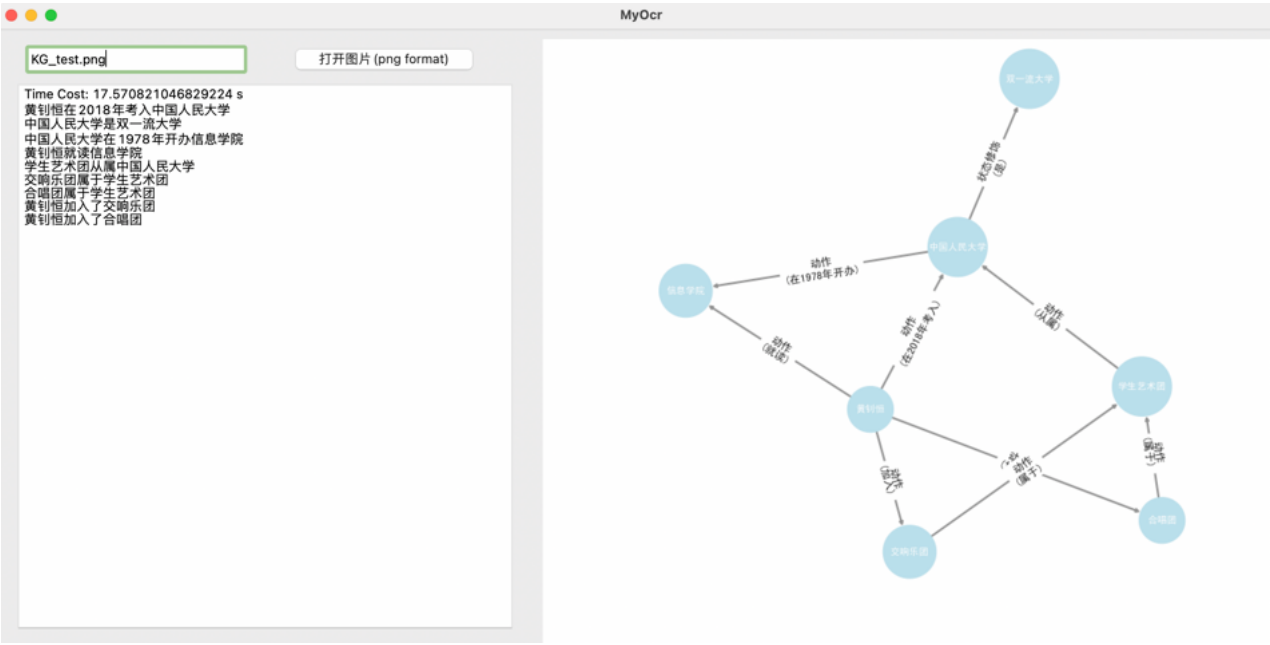
关系: 关系是连接不同的实体，指代实体之间的联系。通过关系节点把知识图谱中的节点连接起来，形成一张大图。如图中的【中国人民大学】与【双一流大学】之间存在关系等。



4. 当前项目进度

在UI界面输入图像相对路径，其中，图像的大小、存储空间不限，但需为PNG格式。点击【打开图片(png format)】，则后台将会运行深度学习模型，产生文字识别的结果。在UI的下左侧的文字框中，将显示深度学习模型的运行时间，以及文本识别结果。此外，我们将文本识别结果代入知识图谱模型，并以可视化的形式将知识图谱展示在右侧。

以下为我们的App。输入KG_test.png，点击【打开图片(png format)】，在17秒后，显示OCR模型识别的文字信息，同时在右侧展示该文本数据所组成的知识图谱。



任务目标

1. 根据三元组、置信度等知识图谱数据，设计一个聊天机器人，并尝试通过图灵测试。
2. 找到更复杂的关系语句集，测试我们的知识图谱鲁棒性。
3. 探索更好的语音识别模型。

项目优势

我们使用了wxpython库开发UI，这是一个面向工程的、功能全面的第三方库，因此在项目结束时，我们的成果将具有更高的用户友好度。

同时，networkx是我们在【数据科学导论】课程上接触的较好地展示元素变量之间关系的第三方库，也是我校公众号【RUC新闻坊】采用的新闻数据可视化库，因此我们在使用该库后，对于用户而言，他们能更加敏锐地觉察出该知识图谱的特点，我们的工程完成度也更高。

不仅如此，我们采用了基于ResNet+LSTM+CTC与ASR-MFCC模型（见上文论述部分），在论证原理正确性的基础上，我们使用了easyocr与smoothnlp库，取得了符合预期的结果。