PPT 图像识别与翻译

王 柯 郭佳淼 邓志雄

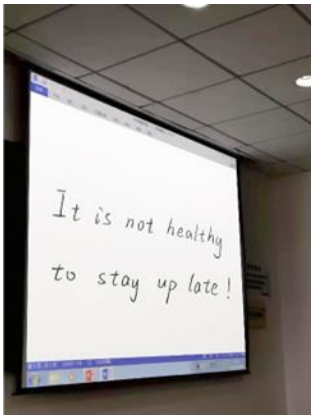
（学号：3200103911 3200103360 3200106068）

摘 要：为了识别并提取含PPT图片中文字内容，以及对全英文PPT进行文字提取并翻译，现基于matlab通过以下步骤和方法来实现这个目的。首先是对图片中PPT范围的标定，先利用OSTU（最大类间分差法）对图片进行二值化处理，然后利用canny算子提取图中所有轮廓，并选择其中最大轮廓（提取PPT区域），再次利用canny算子得到最大轮廓（轮廓检测），再然后将提取出来的最大轮廓边缘转变成规则的四边形（凸多边形覆盖轮廓），最后选择四边形中四个顶点（标定顶点），为下一步图像校正做准备。第二步是图像校正，利用仿射变换对第一步中提取的四个顶点坐标（倾斜）进行转化，将原来倾斜的PPT图像校正为正视的PPT图像。第三步是OCR文字识别，通过调用matlab内置OCR函数提取识别校正后的PPT图像中的文字。第四步是翻译（仅针对英文PPT），利用基于seq2seq2模型的神经网络进行翻译工作。

Abstract： In order to identify and extract the text content in PPT pictures, as well as to extract and translate the full English PPT, the following steps and methods are used to achieve this purpose based on matlab. The first is to calibrate the PPT range in the picture, first use OSTU (maximum inter-class difference method) to binarize the picture, and then use the canny operator to extract all the contours in the picture, and select the largest contour (extract the PPT area) , use the canny operator again to get the largest contour (contour detection), then convert the extracted largest contour edge into a regular quadrilateral (convex polygon covering the contour), and finally select the four vertices in the quadrilateral (calibration vertices), as the next step Prepare for image correction. The second step is image correction, which uses affine transformation to transform the coordinates (oblique) of the four vertices extracted in the first step, and corrects the original oblique PPT image into a front-facing PPT image. The third step is OCR text recognition, by calling the built-in OCR function of matlab to extract and recognize the text in the corrected PPT image. The fourth step is translation (only for English PPT), using the neural network based on the seq2seq2 model for translation work.

关键词：二值化；canny 算子；仿射变换；OCR；神经网络

在当今大学课堂上，有些课程上教师使用纯英文课件来授课，这就导致了学生在做笔记时有诸多不便,现在利用 Matlab 可以实现 PPT 图像识别与翻译，将上课拍照记录的 PPT 图片作为输入对象，输出原始英文文本和翻译后的结果。这样，学生在课后整理笔记时，可以很方便的将课上拍下的含有重点知识的 PPT 转换为文本文档，从而提高了学习效率。这一转换主要是通过以下三个过程来实现：



图表 1：拍摄 PPT 图

1 图像预处理

1.1 问题概述

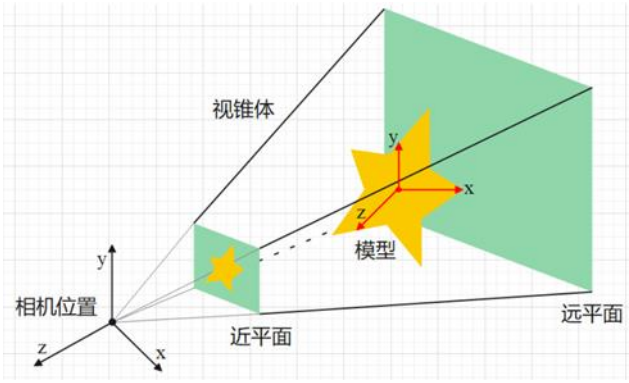
要进行 PPT 图像文字识别和翻译的第一步就是需要对 PPT 图片进行图像预处理。本次预处理主要解决的问题是图像的透视畸变问题。

可以看到图片中的 PPT 区域沿着 x 轴、y 轴、z 轴都有倾斜，以此时的照片进行 OCR 识别和翻译势必会

造成准确率下降，出现错字、漏字的现象。因此在第一步将并非正视视角拍摄的照片进行图像倾斜的矫正就显得尤为重要。

1.2 透视畸变的原理

要解决透视畸变的问题就需要先理解透视畸变的规律。如图所示，相机以一个角度将在近平面和远平面的视锥体之间的图像通过变换投影到近平面上，近平面上显示的图像就是发生了透视畸变后的图形。



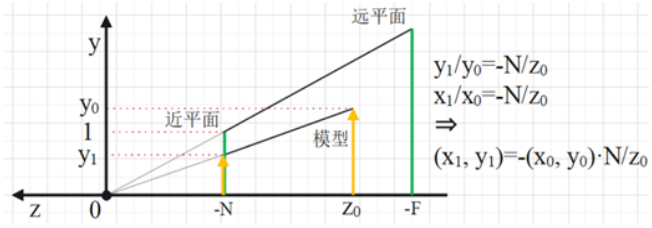
图表 2：透视畸变示意图

视锥体内任意一点的做表面(x0, y0)，与经透视畸变后面的做表面(x1, y1)都符合以下的关系表达式：

$$x_1 = x_0 \cdot \frac{N}{r \cdot (-z_0)} \qquad (1)$$

$$y_1 = y_0 \cdot \frac{N}{-z_0} \qquad (2)$$

但因为其中 z0 并非是常量，因此经过进一步的推导变换，可以得到如下的透视变换矩阵表达式：



图表 3：透视变换

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ w_p \end{bmatrix} = \begin{bmatrix} \frac{N}{r} & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & \frac{N+F}{N-F} & \frac{2N \cdot F}{N-F} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \qquad (3)$$

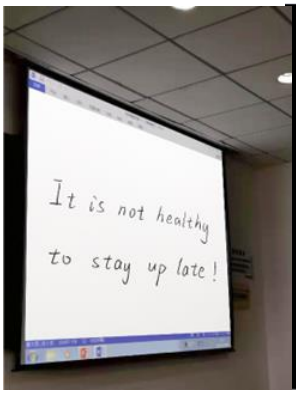
而图像预处理解决透视畸变问题就是该过程的逆过程，通过透视变化矩阵、图片中 PPT 区域的位置信息反解出正视时的 PPT 图像并将其作为输入下一步 OCR 环节的输入图像。

1.3 图像处理的思路

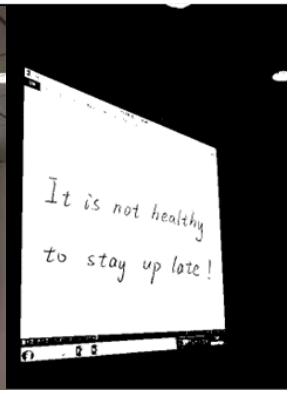
我们将图像处理分成六个环节，分别为图片二值化处理、提取 PPT 区域，轮廓检测、凸多边形覆盖、顶点标定、透视矫正，其中前五个环节主要是为了标定出特定的 PPT 区域，为下一步透视矫正提供准确的输入。

1.3.1 二值化处理

第一步二值化处理，我们利用了最大类间分差法（OSTU），将目标和背景之间进行最初步的区分，并将原图变为二值图像。



图表 4：原图



图表 5：二值化处理后的图像

1.3.2 降噪处理与提取边缘

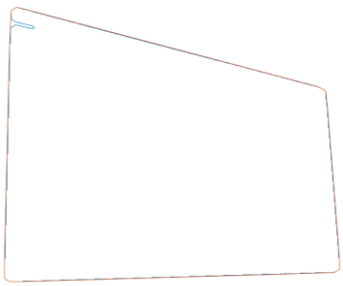
第二步，我们对已经进行过初步的主体提取后的图像进一步进行降噪，去除如灯光、窗户透光等干扰光源造成的影响，提取出 PPT 主体部分。实现这一步的原理是利用 Canny 算子提取所有高亮区域的边缘，并依据干扰光源大多为点光源、边缘较小的特点，寻找边缘区域最大的光源区域作为 PPT 区

域。Canny 算子是一种较为成熟的边缘检测算法，其原理是利用图像灰度沿着水平方向 Gx 和垂直方向 Gy 的偏导数，求出梯度的幅值和方位，找到局部梯度最大值，最后用双阈值算法检测和链接边缘。同时在提取边缘之后我们用凸多边形对 PPT 区域进行了覆盖，为下一步框选四边形轮廓进行了铺垫处理。

图表 6：二值化处理后的图像

图表 7：提取边缘后图片

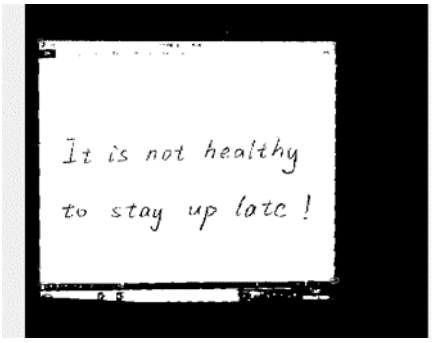
第三步，四边形轮廓确定和角点提取。在这步中，我们将 PPT 区域用一个四边形形状框选，并提取区域的四个角点，作为参数输入到透视矫正中。提取四边形形状我们用了 MATLAB 中的 bwboundaries 函数进行，并对边缘进行了高斯滤波，去掉波动。bwboundaries 函数适用于二值化图像，对于前几步中经过二值化处理和提取处理的图片，用此方法可以最高效的提取边缘。在这之后我们利用 convexhull 函数提取了四边形的凸点作为四个角点输入透视矫正中。



图表 8：提取后的 PPT 边缘

1.3.4 图像矫正

第四步，图像透视矫正第一步需要将四个角点的值带入，利用前文所述的原理求出畸变转换矩阵，再将此矩阵带入值反解出正向时的数组，将位置信息的二值化数组转化为图片输出，进行下一步的 OCR 过程。不过在这一步中我们进行了一定的妥协，因为若 PPT 图像的边缘为深色的话，上一步提取角点的过程有时会有偏离，为了保证图像矫正过程的准确高效，我们在这一步中保留了手动选点的过程，需要使用者按选取四个角点，作为矫正矩阵的补充。

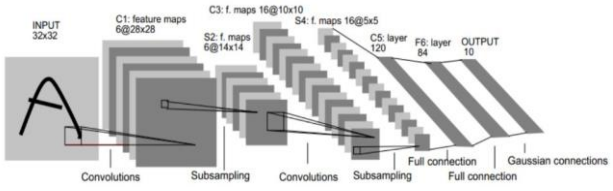


图表 9：矫正后的图像

2 文字识别与处理（OCR）

2.1 图像预处理

传统 OCR 基于数字图像处理和传统机器学习等方法对图像进行处理和特征提取。常用的二值化处理有利于增强简单场景的文本信息，但对于复杂背景二值化的收效甚微。传统方法上采用 HoG 对图像进行特征提取，然而 HoG 对于图像模糊、扭曲等问题鲁棒性很差，对于复杂场景泛化能力不佳。现在使用基于 CNN 的神经网络作为特征提取手段。基于 CNN 的文字识别原理图如下：



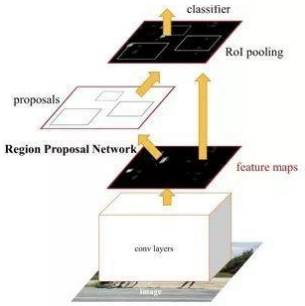
图表 10：CNN 原理图

2.2 文字检测

对于文字检测任务，套用图像检测的方法来框

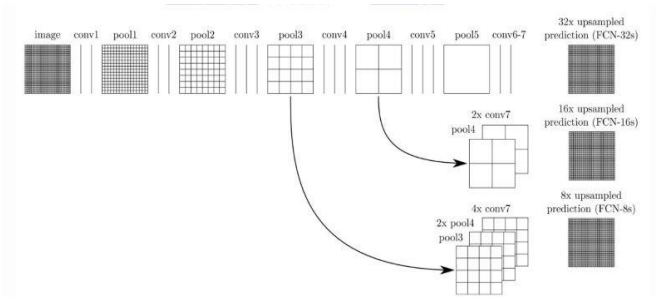
选出图像中的文本区域，常见物体检测方法有：Faster R-CNN（Region Proposal Networks），FCN，RRPN （Rotation Region Proposal Networks），TextBoxes ， DMPNet （Deep Matching Prior Network），CTPN （Connectionist Text Proposal Network），SegLink 等。下面简单介绍与本项目相关的两种方法：

Faster R-CNN 采用助生成样本的 RPN（Region Proposal Networks）网络，将算法结构分为两个部分，先由 RPN 网络判断候选框是否为目标，再经分类定位的多任务损失判断目标类型，整个网络流程都能共享卷积神经网络提取的的特征信息，节约计算成本，且解决 Fast R-CNN 算法生成正负样本候选框速度慢的问题，同时避免候选框提取过多导致算法准确率下降。对于受限场景的文字检测，Faster R-CNN 的表现较为出色。可以通过多次检测确定不同粒度的文本区域。



图表 11：Faster R-CNN

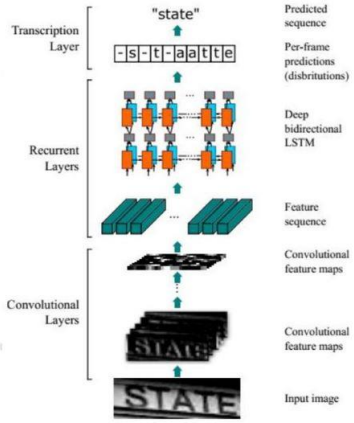
FCN 相较于 Faster R-CNN 算法只能计算 ROI pooling 层之前的卷积网络特征参数，R-FCN 算法提出一种位置敏感分布的卷积网络代替 ROI pooling 层之后的全连接网络，解决了 Faster R-CNN 由于 ROI Pooling 层后面的结构需要对每一个样本区域跑一次而耗时比较大的问题，使得特征共享在整个网络内得以实现，解决物体分类要求有平移不变性和物体检测要求有平移变化的矛盾，但是没有考虑到候选区域的全局信息和语义信息。所以当面对自然场景的通用 OCR，适于多尺度检测的 FCN 较之 Faster R-CNN 有着更好的表现。当采用 FCN 时，输出的掩膜可以作为前景文字的二值图像进行输出。所以在这里采用的正是 FCN 算法。



图表 12:FCN

2.3 文本识别

文本识别在传统技术中采用模板匹配的方式进行分类。但是对于文字行，只能通过识别出每一个字符来确定最终文字行从内容。因此可以对文字行进行字符切分，以得到单个文字。但是通过识别每个单字符以实现全文的识别，这一过程导致了上下文信息的丢失，对于单个字符有较高的识别正确率，其条目识别正确率也难以保证。为了引入上下文这样的序列信息，这里可以选择借用 RNN 和 LSTM 等依赖于时序关系的神经网络。最常见的一种方法是利用 CRNN 模型，这也是这里 OCR 中使用的算法。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。



图表 13:CRNN 网络结构

3 翻译

3.1 问题概述

得到识别结果后，我们需要对其进行中英文翻

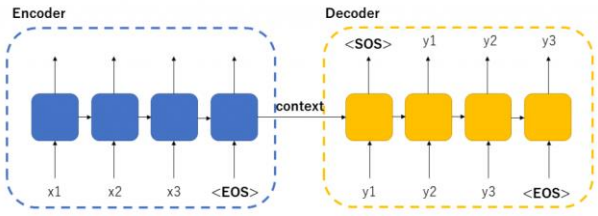
译。机器翻译是 NLP(Natural Language Processing)的一个分支。早期的机器翻译基于统计模型，而如今大多基于神经网络模型。因此，我们采用机器翻译中最流行最成熟的神经网络”seq2seq(sequence to sequence）”模型进行翻译。由于我们的识别基于英文 PPT，因此我们主要实现英文-中文的翻译过程。

3.2 实现原理

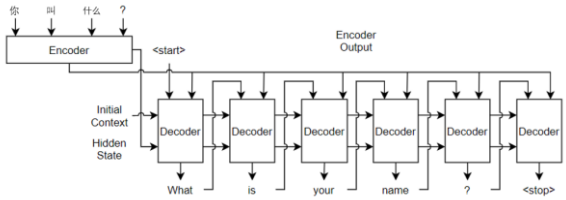
seq2seq 是一个 Encoder－Decoder 结构的网络，它的输入是一个序列，输出也是一个序列，Encoder 中将一个可变长度的信号序列变为固定长度的向量表达，Decoder 将这个固定长度的向量变成可变长度的目标的信号序列。这个结构最重要的地方在于输入序列和输出序列的长度是可变的，可以用于翻译，聊天机器人，句法分析，文本摘要等。

图表 14：Encoder 示意图

图表 15：Decoder 示意图



图表 16：Encoder-Decoder 基本结构图



图表 17：seq2seq 基本结构图

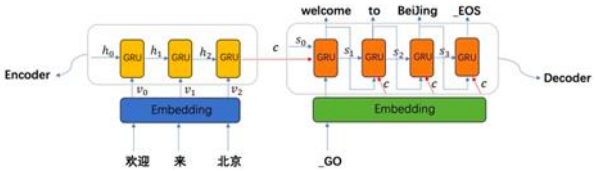
如图 3 所示，Encoder 和 Decoder 一般都是 RNN，通常为 LSTM 或者 GRU，图中每一个方格都为一个 RNN 单元。在 Encoder 中，“欢迎/来/北京”这些词转换成词向量，也就是 Embedding，我们用v_i表示，与上一时刻的隐状态h_{i-1}按照时间顺序进行输入，每一个时刻输出一个隐状态h_i，使用用函数f表达 RNN 隐藏层的变换：

$$h_i = f(v_{i,h_{i-1}}) \tag{4}$$

假设有 t 个词，最终通过 Encoder 自定义函数 q 将各时刻的隐状态变换为向量 c：

$$c = q(h_0,...,h_t) \tag{5}$$

这个 c 就相当于从“欢迎/来/北京”这几个单词中提炼出来的大概意思，包含了这句话的含义。



图表 18：seq2seq 示例

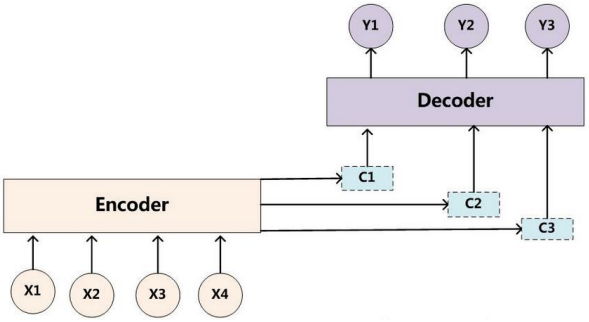
Decoder 的每一时刻的输入为 Encoder 输出的 c 和 Decoder 前一时刻解码的输出 s_(i-1)，还有前一时刻预测的词的向量 E_(i-1)（如果是预测第一个词的话，此时输入的词向量为“_GO”的词向量，标志着解码的开始)，使用用函数 g 表达解码器隐藏层变换：

$$s_i = g(c, s_{i-1}, E_{i-1}) \tag{6}$$

直到解码解出“_EOS”，标志着解码的结束。

为了提高翻译的准确性，解决信息过长，信息丢失的问题，在该模型中，我们在 seq2seq 模型基

础上引入注意力机制（Attention），如下图所示



图表 19：Attention 机制

c 向量跟 Decoder 解码过程中的每一个输出进行加权运算，在解码的每一个过程中调整权重取到不一样的 c 向量，假设编码器每个隐藏状态为 h_j，序列长度为 T，那么在第 i 个时刻 c 向量的计算方式如下：

$$c_i = \sum_{j=1}^{T} a_{ij} h_j \qquad (7)$$

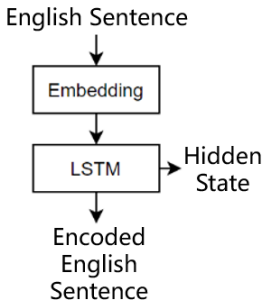
其中，对每个给定的 i：

$$a_{ij} = soft\max(e(s_{i-1}, h_j)) \qquad (8)$$

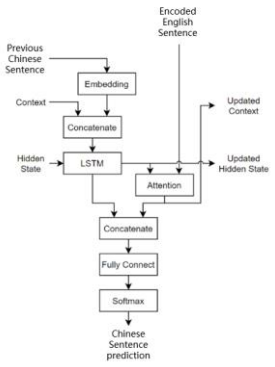
其中，s_{i-1} 是解码器（Decoder）在 $i-1$ 时刻的隐状态，函数 e 为距离计算函数，最简单的形式为内积。

3.3 实现步骤

使用 LSTM 作为 Encoder 和 Decoder 的隐藏层，使用 Matlab 的 DeepLearning toolbox，搭建 LSTM（Long short-term memory，LSTM），根据 seq2seq 模型结构，搭建神经网络，具体结构见下图：

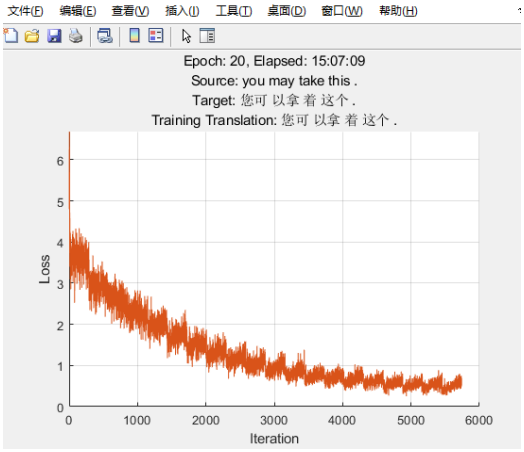


图表 20：Decoder



图表 21：Encoder

从 http://www.manythings.org/anki 获取中文-英文数据集，将数据集划分为 Source（原文-英文）与 Target（目标译文-中文），由于数据量过大，所以每次迭代训练采用其中的 70%数据以加快训练速度。设置 Batchsize=64，Epoch=20，将原始数据集划分为 70%的训练集与 30%的验证集，对网络进行训练。绘制损失函数值图像如下：



图表 22：损失函数值

可以发现，迭代 20 个 Epoch 后 loss 函数值明显较初始降低，收敛于 0 附近，对数据集中的文本有较高的识别准确率。

3.4 模型局限

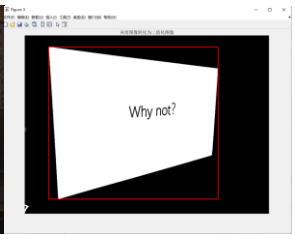
但是，由于数据集不完全，训练电脑设备限制，无法对该模型进行大数据集下的训练（训练耗时过长），故该模型扩展性较差，对数据集之外的文本识别精度较低。且由于数据集为繁体中文，对简体中文的翻译效果较差。

4 成果展示

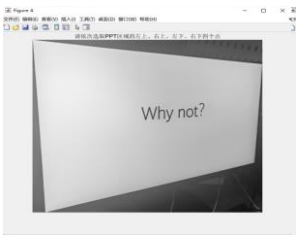
4.1 图像预处理示例

如下示例图片，我们输入原图后，首先是对其进行二值化处理以及降噪处理，然后对 PPT 区域进行框定（如图表 24），然后对框内 PPT 图片进行提取（如图表 25），再然后就是对提取出来的 PPT 图像进行矫正（如图表 26），输出矫正后的结果。

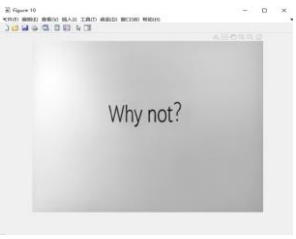
图表 23：原图



图表 24：PPT 区域框定



图表 25：PPT 提取



图表 26：图形矫正

4.2 OCR 识别示例

在这一步中，我们调用 OCR 函数对预处理后的结果（即图表 26）进行文字识别并提取文本，结果如下。

```
text =

    'Why not?

     '
```

图表 27：OCR 识别结果

4.3 翻译示例

最后一步，我们调用先前训练好的神经网络翻译模型，对上一步中 OCR 识别的文本“Why not？”进行翻译，输出结果：“为什么不？”，如下图。

```
strTranslatedNew =

    "为什 么不 ？< stop >

>>
```

图表 28：翻译结果

4.3 代码部署说明

本代码在 Intel-CORE I7 10th GEN 平台中运行，使用 MATLAB Deep Learning Toolbox™与 Text Analytics Toolbox™，MATLAB 版本为 R2021b（先前的版本可能无法运行神经网络相关函数）。

Main.m 为程序主函数，Translate.m 文件为 seq2seq 模型训练代码，net_best.mat 为训练得到模型。其余文件为神经网络搭建中使用的函数。

请运行根目录下 main.m 文件，测试样例置于根目录下 IMG 文件夹中。可通过修改程序中导入图片部分更改测试样例。部分测试结果已在报告中列出。

请注意：由于 MATLAB 内置 OCR 函数对图像质量要求较高，对倾斜文本无法识别，因此在透视矫正时需要正确取点，得到矫正文本，否则可能无法进行文本识别（在根目录 video 下是测试样例视频）

5 结语

通过上面的展示，可以看到，我们已经较好地实现了英文 PPT 图片识别与翻译工作。在预处理这一块内容，我们已经可以将外在因素干扰很多的图片处理的很干净完美，并且对倾斜的 PPT 图片矫正效果也很好；在 OCR 文字这方面，也可以以较高的精度识别并导出英文文本；在翻译这一块，我们根据 seq2seq 模型搭建了相应的神经网络也有不错的表现。

但是，整个项目做完，我们发现，仍有一些地方可以加以改进。比如：在图像预处理的过程中，提取角点时，我们采用的是手动提取，这个可以改进为更方便的自动提取，减少人工操作的时间；在 OCR 识别这一块，目前是只能识别英文，这里可以像翻译这一块，同样借助神经网络，来拓展 OCR 可识别的语言种类，从而使整个项目应用范围更广。最后，在翻译这一块，由于时间以及硬件条件限制，无法进行大数据集的训练，这就导致了翻译精度只是达到了能用的程度，并非特别高，另外，在 OCR 可以识别多种语言的基础上，如果再进一步改进翻译模型，将会极大程度地提高整个项目的应用范围。

参考文献(References):

[1] Lin T Y, Dollar P, Girshick R, et al. Feature Pyramid

Networks for Object Detection[J]. 2016:936-944.

[2] Ma J, Shao W, Ye H, et al. Arbitrary-Oriented Scene Text Detection via Rotation Proposals[J]. IEEE Transactions on Multimedia, 2017,PP(99):1.

[3] Liao M, Shi B, Bai X, et al. TextBoxes: A Fast Text Detector with a Single Deep Neural Network[J]. 2016.

[4]https://ww2.mathworks.cn/help/coder/ug/code-generation-from-lstm-network-that-uses-intel-mkl-dnn.html?requestedDomain=cn

[5] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[6] https://blog.csdn.net/mansir123/article/details/118603653

[7] Automatically Detect and Recognize Text Using MSER and OCR - MATLAB & Simulink - MathWorks 中国