

清华 大学

# 综合 论文 训 练

题目：基于深度学习的图片自动生成  
网页前端代码方法

系 别：软件学院

专 业：软件工程

姓 名：蒋梦青

指导教师：李春平 副教授

2018 年 6 月 14 日

## 关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名： 蒋梦青 导师签名： 李长虹 日 期： 2018.06.25

## 中文摘要

从网页截图生成对应的网页前端代码是最近一年来提出的人工智能领域问题，它由图片描述任务发展而来。本文进一步扩展这个任务，首次提出从手绘的网页设计草稿生成对应的网页前端代码，这使得问题更复杂多变，对模型的泛化能力和稳健程度提出了更高的要求。

本文参考了在图片描述任务上的最新研究，采用一个端到端 CNN 编码—RNN 解码的深度神经网络模型，将网页布局转换为其对应的 HTML 代码。因此，本文建立了两个数据集，一个是程序生成的大规模网页截图数据集，一个是收集的手写网页草图数据集，本文提出的模型在这两个数据集上都展示出了代码的高质量和准确性，分别获得了 BLEU 分数 66.45 和 49.21，并且无 HTML 语言语法错误。除此以外，本文验证了模型经过在大规模的截图数据集上训练后在手写数据集上进行迁移学习能极大提高模型在手写数据集上的性能。

最后，本文搭建了 "Doodle2Code" 的网页应用程序，用户可以在线绘图并得到模型输出代码的渲染结果，并对代码可以进行修改到满意的程度，本文在这个应用程序的基础上进行了用户研究。

**关键词：**深度学习；计算视觉；代码生成

## ABSTRACT

Generation of front-end web code from given a web screenshot, which is derived from Image Caption task, has been raised as a new task in the field of Artificial Intelligence in the recent year. This paper expands the task, creatively generating front-end web code from hand-written web page doodles, which makes the problem much more complicated and varied, and requires much better generalization and robustness of the model .

This paper uses recent advances in Image Caption and represents an end-to-end deep neural model with a CNN-encoder and an RNN-decoder, translating a web page layout into HTML code that displays as the given image after browser rendering. Accordingly, two datasets are built for this task: one is a large-scale program-generated web screenshot dataset, the other a collected hand-written web doodle dataset. The proposed model performs well on both datasets and outputs code with high quality and high accuracy, achieving 66.45 and 49.21 BLEU score respectively. More impressively, there are no HTML grammar errors. In addition, experiments have proved that transfer learning from the large-scale screenshot dataset strongly enhances the model's performance on the doodle dataset.

Lastly, in this work, a web application named "Doodle2Code" has been developed, allowing users to translate their doodles into HTML code and modify the generated code until seeing the exact web page they wanted online, on which a user study is conducted.

**Keywords:** Deep learning; Computer Vision; Code generation

# 目 录

<b>第 1 章 引言 .....</b>	<b>1</b>
1.1 课题背景与研究意义介绍 .....	1
1.2 问题定义：什么是从图片生成网页前端代码 .....	1
1.3 本文的主要工作与贡献 .....	2
1.4 文章结构 .....	3
<b>第 2 章 相关工作 .....</b>	<b>4</b>
2.1 图片描述 .....	4
2.2 代码生成 .....	6
2.3 本章小结 .....	8
<b>第 3 章 模型架构 .....</b>	<b>9</b>
3.1 视觉编码器：卷积神经网络 .....	9
3.2 序列解码器：循环神经网络 .....	11
3.3 词嵌入 .....	12
3.4 本章小结 .....	13
<b>第 4 章 实现细节 .....</b>	<b>14</b>
4.1 数据集 .....	14
4.1.1 大规模机器生成的网页截图数据集 .....	14
4.1.2 手写数据集 .....	16
4.2 数据预处理及训练过程 .....	17
4.3 推断过程 .....	19
4.4 本章小结 .....	20
<b>第 5 章 实验结果 .....</b>	<b>21</b>
5.1 评价标准 .....	21
5.1.1 语法错误率 .....	21
5.1.2 BLEU .....	21
5.1.3 编辑距离与编辑时间 .....	22

5.2 生成结果及分析.....	23
5.2.1 用户研究 .....	25
5.3 本章小结 .....	25
<b>第 6 章 结论和展望.....</b>	<b>28</b>
6.1 结论 .....	28
6.2 未来工作展望 .....	28
<b>插图索引 .....</b>	<b>30</b>
<b>表格索引 .....</b>	<b>31</b>
<b>公式索引 .....</b>	<b>32</b>
<b>参考文献 .....</b>	<b>33</b>
<b>致 谢 .....</b>	<b>36</b>
<b>声 明 .....</b>	<b>37</b>
<b>附录 A 外文资料的调研阅读报告或书面翻译 .....</b>	<b>38</b>

## 主要符号对照表

CNN	卷积神经网络 (Convolution Neural Network)
RNN	循环神经网络 (Recurrent Neural Network)
LSTM	长短期记忆网络 (Long Short-Term Memory)
BLEU	双语评估研究 (Bilingual Evaluation Understudy)
NIC	神经图像描述 (Neural Image Caption)
HTML	超文本标记语言 (HyperText Markup Language)
Bootstrap	Twitter 公司开源的 CSS/HTML 框架

# 第1章 引言

## 1.1 课题背景与研究意义介绍

在软件工程中，一个重要的挑战是为开发人员提供更高效的开发工具。基于用户界面的设计来实现网页前端的过程通常是程序开发人员的责任，但这项工作往往非常消耗时间，并且程序员的大部分精力并不能集中在实现核心的功能或逻辑，而是网页内容的排版和样式调整上，因为这部分的代码通常非常繁琐、并且为了达到好的视觉效果往往要经过非常多的调整。同时，对设计人员或者产品经理来说，用组件拼出的草图、甚至手绘的设计稿，是交流想法的最高效途径。

人工智能 (Artificial Intelligence) 在近年里迅速发展、受到了广泛关注，其子领域深度学习 (Deep Learning) 在机器视觉 (Computer Vision) 和自然语言处理 (Natural Language Processing) 等任务上已经有了大量的研究和成熟的应用，最近两年来也出现了一项新的任务，图片描述 (Image Caption)，即对一张图片生成用直白的英语描述图片内容的句子，这意味着模型需要在识别图片中物体的语义信息之上，还能检测出物体之间的位置、逻辑关系，并且生成符合自然语言文法的单词序列。这项任务将机器视觉和自然语言处理结合在了一起，并且最先进的研究进展已经取得了合理且优秀的结果。

受此启发，本文致力于应用深度学习的方法搭建深度神经网络模型，从图片（网页截图或手绘草图）自动生成对应的网页前端代码，以实现前端工程的智能化、自动化。除此之外，由于这个任务非常新颖，基本没有在这个任务上的成熟的工作，本文从问题定义，到数据集建立、实验设计、实验效果评估和分析，完善了这个任务的解决方案，对这个任务上的后续工作具有借鉴意义。

## 1.2 问题定义：什么是从图片生成网页前端代码

具体来说，如图1.1所示，模型接受的输入是图片  $I$ ，图片可以是网页截图、或者手绘的网页设计草图，目标的正确输出序列为  $S = \{S_1, S_2, \dots\}$ ，即能正确描述图片  $I$  内容的 HTML 代码，其中每个单词  $S_T$  来自自定义的代码词库，词库的生成过程详见第4.1节。

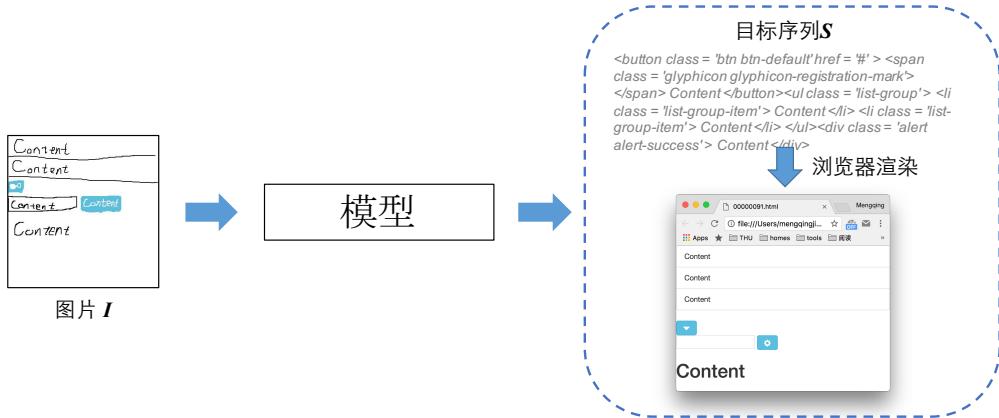


图 1.1 从图片生成网页前端代码流程图

模型训练的过程就是最大化目标序列的  $p(S|I)$ , 以逐词输出接近正确的代码序列。

### 1.3 本文的主要工作与贡献

本文的主要工作与贡献有如下三点:

- 本文首次提出从手写画生成对应网页 HTML 代码的任务, 并提出了一个端到端的生成模型来学习网页的元素、排版和样式, 这个生成模型基于深度学习框架, 是一个结合了用于视觉特征的卷积神经网络 (CNN) 和用于序列生成的循环神经网络 (RNN) 的架构, 并且使用了词嵌入空间, 采用随机梯度下降算法 (SGD) 在自建数据集上训练后达到了输出代码完全无编译错误、BLEU 分数最高 63 的效果。
- 本文为这个任务建立了两个数据库, 一是用程序自动随机生成的大规模标准 Bootstrap 风格的网页截图数据集, 二是收集的手写网页草图, 并且本文研究了模型在这两个数据集上的迁移学习问题, 同时验证了大规模截图数据集对训练手写数据集起到了重要的作用。
- 本文用训练得到的模型和网页前后端搭建了完整的、用户体验良好的 "Doodle2Code" 软件, 并在此基础上进行了用户研究, 创新性地将编辑距离作为代码生成交互体验的评判标准。

## 1.4 文章结构

本文接下来的内容将按如下结构展开：第 2 章将主要介绍在图片描述和代码生成等领域的相关工作；第 3 章将介绍本文采用的模型架构；第 4 章将具体介绍建立数据集、训练模型等实验过程的细节；第 5 章将介绍本文对这个任务提出的评估标准，并展示在自建数据集上的实验结果和结果分析；第 6 章将对本文进行总结，并展望未来潜在的可行的研究工作。

## 第 2 章 相关工作

本文受到图片描述任务的启发，因此将在本章第 2.1 节介绍和梳理与图片描述相关的文献与工作，但与图片描述的不同点是，本文的目标是生成结构化的 HTML 代码序列，所以本章第 2.2 节还会介绍与代码生成相关的文献与工作。

### 2.1 图片描述

用自然语言来描述一张静态图片的内容近年来得到了广泛的关注，早期的图片描述主要有两种方法：基于搜索的方法和基于模版的方法，但这两种方法有一个共同的问题，就是人为设计的部分太多，不能灵活应对多样化的输入。以基于模版的生成为例，[1-3] 使用传统的检测方法来推断场景元素及其关系，用三元组、短语或者更复杂的图模型来表达它们，然后再用模板将这些场景元素转换为文本。虽然它们已经表现出描述图像的能力，但是这种方法定义的文本生成过程大部分仍然是人为设计的。

随着深度神经网络的发展，尤其是 2015 年之后，出现了很多用深度学习方法进行图片描述的工作。

一些工作先用深度学习视觉检测的方法把图片中的物体识别出来（甚至包括物体的位置、类别、属性等），然后生成语言进行描述，例如 [4-7]。其中 [5] 曾经在 2015 年的 MSCOCO 的图片描述挑战赛中获得了第一名，它采用多示例学习的方法，从图片检测结果中提取出似然值在阈值之上的所有单词，并把这些单词对应回图片上的具体区域，然后采用传统的语言模型对这些单词进行建模和排序。而 [7] 是另一种典型的思路，将计算视觉中图片分类、物体检测、特征提取等成熟的模型结合起来，先进行物体检测，然后提取检测框中图片的特征，对其进行分类和属性识别，最后用循环神经网络编码并解码生成精细的描述句子。

近年来开始工作提出了结合机器视觉领域上的卷积神经网络（CNN）与自然语言处理领域上的循环神经网络（RNN）的端到端的模型，例如 [8-12]。这些工作的主要区别是（1）使用了不同的 CNN 模型或者不同的 RNN 模型来完成这个任务，（2）CNN 与 RNN 组合的方式不同。例如，[8] 和 [9] 将提取图片特征的 CNN 与记忆历史语言序列的 RNN 同时放在一起作为编码器，再使用一个 RNN 来解码；[10-12] 则使用的单一模型，只将编码图像，然后使用一个 RNN 进行

解码的工作，并且验证这种方法比多峰模型冗余更少而效果却更好、网络效率更高。而对 CNN 的使用上，有以 [10] 为代表的工作，使用图片分类的网络，直接提取出网络的隐藏层输出向量，作为图片的编码，同时 [11] 也提出了使用物体检测模型，它和循环神经网络来对齐视觉和语义信息，构建相似性度量；类似的，[13] 提出的“Image2Text”框架先用物体检测把一张图片转换成图片中被检物则成的序列，再输入到 RNN 中生成描述。

还有一些工作是在端到端的模型的基础之上，加入了注意力机制，如 [14-18]，其中 [14] 提出在 CNN 中加入空间的视觉注意力机制，这个注意力就是一个以图片特征图像素数为长度的向量，值为 0-1 的实数，解码器在接收到编码器的中间信息时，通过这个注意力向量可以选择对应需要关注区域的图片特征，实际上就是对原图的特征有一个加权作用，并且这个注意力向量是可以在网络训练的过程中自适应地进行学习的。另外，这篇文章中还提到了“软注意力”和“硬注意力”两种应用注意力向量的方法：“硬注意力”通过对全通道特征乘以注意力向量后采样的方法得到编码结果，“软注意力”方法是做平均后得到编码结果。而 [17] 提出另一种注意力机制的思路，即训练注意力活动框的坐标和仿射参数，来得到一个灵活的空间注意区域。[19] 的工作则不需要注意力机制在每个时刻都被执行，因为语言中存在一些出现频率很高但没有实际意义的停止词（Stop Word），比如“and”，“the”，“how”等，它们并不需要对应图片中的某个区域，但语言模型在生成的过程中有能力自动推断并产生它们，因此在生成序列的每个时刻都会进行判断是否需要视觉的注意力信息。另外，还有工作 [20] 用强化学习来研究在图片描述中的自适应注意力机制。另一些工作更加关注生成序列的部分，在每个时刻额外输入语义信息来引导 LSTM 的训练<sup>[21]</sup>。

有的工作旨在生成更细节化的图片描述，因此会先直接提取出图片中感兴趣的区域，再进行生成描述<sup>[7,22-23]</sup>。其中 [22] 中提出的 DenseCap 在 CNN 编码器与 RNN 解码器之间插入了区域推荐网络（RPN<sup>[24]</sup>），与之前要生成全图描述的工作不同，它的思路是先生成对图中许多具体区域的描述，然后再将它们作为子句串联起来作为全图的描述。

这些工作都表现出了先进的描述图片的性能，验证了深度神经网络可以学习到图像中对象的潜在变量，并用一个可变长度的句子来描述它们之间的关系。

参考 RNNs 在机器翻译中编码-解码（Encoder-Decoder）<sup>[25-26]</sup> 的结构，目前在图片描述任务中表现最佳的端到端的生成模型，通常都使用了卷积神经网络（CNN）作为编码器来提取图片的特征，和循环神经网络（RNN）作为解码器来

生成单词序列<sup>[10,12,14]</sup>，以 [10] 提出的 NIC 为例，它已经被证明具有极强的泛化能力。这种架构充分利用了这两种神经网络的优点，并且解决了解决了生成句子长度不同的问题，将机器翻译中的 RNN 编码器替换为机器视觉任务中的 CNN，提取图像  $I$  的“视觉特征”  $x$ ，然后还是使用 RNN 解码器将这个  $x$  解码为输出序列  $S$ 。基于上述原因，本文采用的亦是这种架构。

## 2.2 代码生成

代码生成是自计算机科学开创以来就存在的任务，最常见的应用就是编译器。常见的编译经过词语分析、语法分析、语义分析和优化，将高级语言翻译为低级的、机器可以直接执行的语言。常见的代码生成过程都是强规则的，并且被翻译的代码也都是精确定义的形式语言。

随着自然语言处理中机器翻译的发展，软件工程领域的研究者们开始探索如何将用户意图，或者说自然语言指令，转化成高级程序语言，例如 [27]。类似的，[28] 提出隐含预测网络（Latent Predictor Network, LPN）用于将游戏卡牌上的描述自动生成代码，所生成的代码为 Python 类对象，对象里含有能实现游戏英雄属性的成员变量值和能实现游戏英雄技能的成员函数。LPN 的创新地将代码分为了三个部分，并用三种不同的方法生成它们：

1. 实现业务逻辑的代码段，这一部分由一个 char-RNNs<sup>[29]</sup>（更具体来说是双向 LSTM）来生成。
2. 单属性值区域，比如游戏英雄的攻击量、防守值、稀有度等，这一部分生成器直接粘贴原文的数字到代码中。
3. 文本区域，比如游戏英雄的名称、类型、描述等，这一部分生成器直接粘贴原文的文本到代码中。

因此 LPN 提出了一个隐含的小神经网络来预测当前的生成器是上述三种的哪一种，随后再进行生成。的确，代码与自然语言不同的一点是代码的功能性更强，人为地加入这种规定能减少模型的学习难度。

与此同时，也有工作开始机器学习方法从截图生成前端代码（不限于网页前端，还有手机应用程序等其它平台的前端界面），比如 [30] 提出了一个复杂的流程：先用 canny 算子等边缘检测方法划分出界面中的组成部分，然后训练了一个 CNN 逐一对这些组件进行分类，最后再用模板语言将它们拼接到一起。

而 [31] 提出了一个端到端网络：用一个浅层 CNN 提取图片特征后，直接加

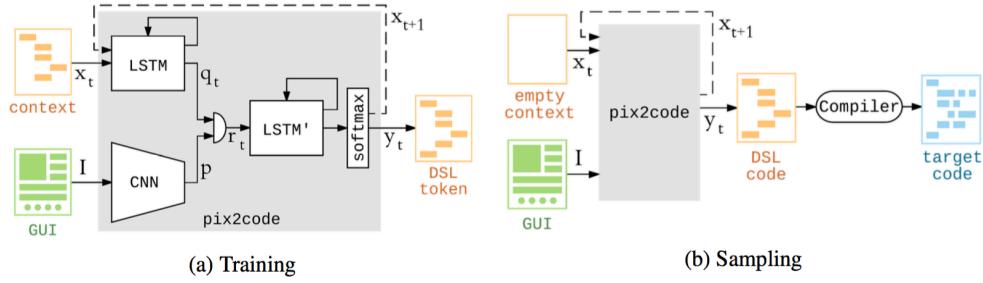


图 2.1 RNN 与 CNN 共同作为编码器的方法<sup>[31]</sup>

入 RNN 编码一解码的结构中，即将 RNN 与 CNN 同时作为编码器，生成领域专用语言（Domain Specific Language，简称 DSL），然后再经由编译器生成不同平台的语言，如图 2.1 所示。[31] 在他们生成的截图数据集上取得了不错的效果，但由于他们的数据集并没有公开，在第 5 章本文复现了他们的方法，并在本文生成的数据集上做了比较。



图 2.2 [32] 模型测试结果

除此之外，还有一项工作<sup>[32]</sup>作者在网络博上贴出了从手写画生成领域专用语言的代码和模型，但是这一工作尚不成熟，笔者用他们提供的模型进行了简单的测试后，发现如图 2.2 展示的那样，无论输入的草图是怎样的，输出的网

页基本都相同（组件的排版和样式均一模一样），说明模型的泛化能力极弱；并且这项工作只接受黑白的手写画输入，但本文在实验中模型接受的输入是彩色的图片，因此在本文中不再复现做对比实验。这也再次说明了从手写网页设计草稿生成可变的、灵活的、合理的网页是相当困难的一项任务。

### 2.3 本章小结

本文总结了近几年在图片描述和代码生成领域的相关文献，尤其是在这两个领域内基于深度学习方法的工作。可以看出，使用卷积神经网络（CNN）处理视觉输入和用循环神经网络（RNN）建立语言序列模型，是现在主流的深度学习解决方案。本文也将借鉴这个思路，在下一章介绍本文采用的方法和模型。

## 第3章 模型架构

本文主要受到 [10] 的启发，提出以概率模型为基础的神经网络来生成图片的描述，这个神经网络是端到端、且可以通过梯度下降的方式进行训练的，训练的过程即最大化正确结果的输出概率来得到最优的结果，当然这一流程同样适用于测试阶段，即

$$\theta^* = \arg \max_{\theta} \sum_{I,S} \log p(S|I; \theta) \quad (3-1)$$

其中  $\theta$  为模型参数， $I$  为输入的图片（或者图片对应的特征向量）， $S$  为图片所对应的正确代码，由于  $S$  是可变长的，因此在  $S_0, \dots, S_N$  上应用链式法来建模联合概率。假设  $N$  为  $S$  的长度，那么有

$$\log p(S|I) = \sum_{t=1}^{N+1} \log p(S_t|I, S_1, \dots, S_t) \quad (3-2)$$

本文简化了这个概率对  $\theta$  的依赖，把  $(S|I)$  视为一对数据进行训练，然后使用随机梯度下降算法（SGD）来最大化公式3-2中的概率对数和（第4.2节会进一步介绍训练过程的实验细节）。

本文提出的模型主要分为三个部分，首先是作为编码器的卷积神经网络（CNN），通过卷积和池化操作从图片中提取固定长度的特征向量，在第 3.1 节进行介绍；第二个部分是词嵌入技术，将词典中的词语映射到一个固定维度的实数向量空间，在第 3.3 节进行介绍；然后第三部分是作为解码器的循环神经网络（RNN），观察到 CNN 传来的特征向量和词嵌入处理后的词语向量后，将其解码为可变长度的代码序列，在第 3.2 节进行介绍。

### 3.1 视觉编码器：卷积神经网络

近年来，CNN 几乎统治了机器视觉领域：从最简单的图片分类任务，到物体检测、语义分割、关键点识别、视频分类，再到与自然语言处理相结合的图片描述、图片问答、视频标注等任务，其中一个主要原因就是 CNN 的拓扑结构便

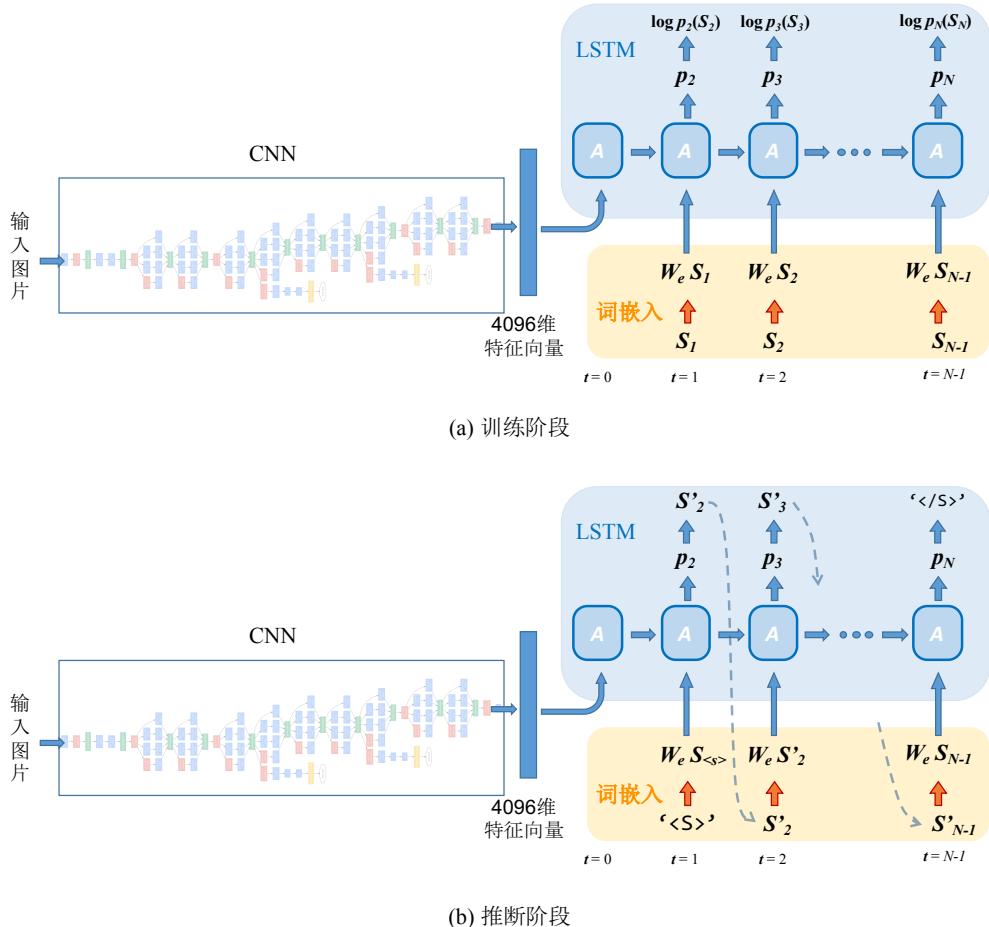


图 3.1 本文提出的网络模型架构图

于学习训练的图像中丰富的潜在表征<sup>[11]</sup>。而在各项任务上的最优性能网络也证明，在一定范围内，网络层数越深性能越好，但对数据集规模和训练过程的要求也就更高。

因此在 CNN 编码器的选择上，与 [10] 的方法类似，本文使用了参数在 ImageNet<sup>[33]</sup> 图片分类数据集和 MSCOCO<sup>[34]</sup> 图片描述数据集上进行预训练的 Inception-v3 模型<sup>[35]</sup>。图 3.2<sup>①</sup>展示了本文采用的 CNN 编码器的网络结构，Inception 系列的卷积神经网络架构采用稀疏的、分解的基础网络单元进行堆叠，增加了网络的宽度，组合了不同感受野大小的神经元，提高对不同图片尺度下的稳定性，并且有效地减少了网络参数、提高网络内部参数资源利用率，加速计算；更重要的是，在本文的开发平台 tensorflow<sup>[36]</sup> 上提供了已预训练好的 inception-v3 模型参数。

<sup>①</sup> 该图摘自<https://hacktilldawn.com/2016/09/25/inception-modules-explained-and-implemented/>

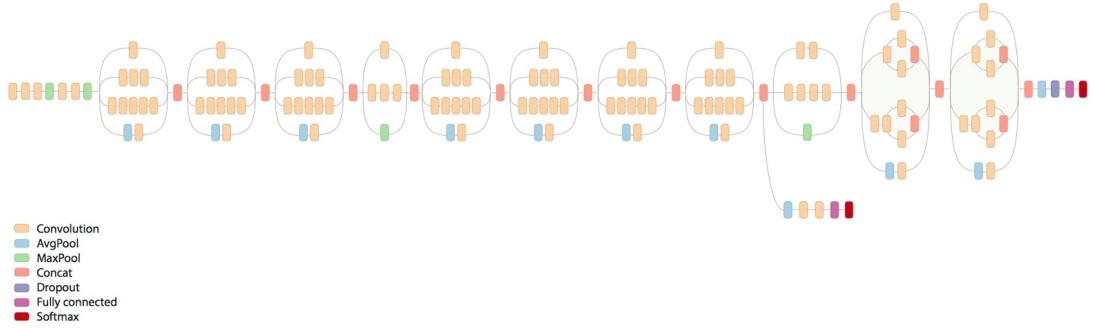


图 3.2 Inception-v3 网络结构图

本文采用 inception-v3 隐藏全联接层输出的 4096 维向量作为图片的特征向量  $x_0$ , 在  $t_0$  时刻传递给 RNN 解码器。

### 3.2 序列解码器：循环神经网络

循环神经网络 (RNN) 常常用来对一段序列中的  $p(S_t|I, S_0, \dots, S_{t-1})$  建模, 它的核心是一个隐含状态, 或者叫记忆量, 记为  $h_t$ , 这个隐含状态在每一个时刻都会根据输入进行更新, 并产生新的输出, 换句话说  $h_t$  包含了  $t-1$  之前的序列信息, 在此基础之上输出  $t$  时刻所有词的似然值分布。这个隐含状态的更新过程为:

$$h_{t+1} = f(h_t, x_t) \quad (3-3)$$

其中  $x_t$  为  $t$  时刻的输入, 除了 CNN 将图片提取为固定长度的向量作为  $x_0$  外, 下一小节将介绍本文使用的词嵌入技术, 它将词语映射到一个嵌入空间中得到一个定长的实数向量; 而  $f$  为一个非线性方程, 由于在拟合数据时, 传统的 RNN 时常出现梯度消失或爆炸, 本文采用了能解决这一问题的长短期记忆网络 (LSTM<sup>[37]</sup>) 的结构作为这个  $f$ , 并且许多研究表明它非常擅长序列对序列的任务, 例如机器翻译、视频标注等。

图 3.3<sup>①</sup>很好地描述了 LSTM 网络展开后的结构与更新过程, 图中的绿色方块是 LSTM 的记忆单元  $c$ , 每个时刻它接收到输入  $x_t$ , 而输出值受到三个“门”

<sup>①</sup> 该图摘自<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

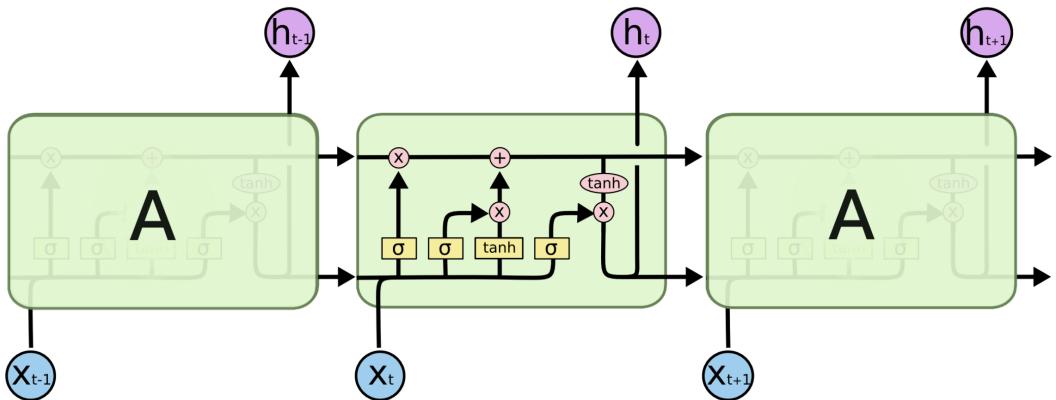


图 3.3 LSTM 网络结构图

控制，即图中的每个乘法元，门的值是一个 0 到 1 的实数，它的大小决定了上层的值留下多少。这三个门的功能分别是决定是否忘记当前单元内的值（遗忘门  $f$ ）、是否它应该读取此刻输入（输入门  $i$ ）和是否输出新的单元值（输出门  $o$ ），这样的乘法门能让 LSTM 有效改善梯度消失和梯度爆炸。三个门的输出结果即更新了单元的状态值（即图 3.3 中横向的两个箭头所传递的值），同时会在每个时刻输出  $h_t$ ，在本文中，图中的  $h_t$  即为模型所需要的  $p_t$ ，即所有单词在嵌入空间上的概率分布。

### 3.3 词嵌入

在许多自然语言处理的工作中，都使用了词嵌入，即将词汇映射到定长的实数向量上，并尽可能使意义相近的词汇所对应的向量之间距离更接近。虽然在我们的任务中生成的是代码而不是自然语言、相比较而言词汇量不大，但是代码中词语之间的功能意义区分度大，在 HTML 语言中，有的词语是作为标签 (tag)，有的词语是作为标签中的属性，有的词语是作为文本内容……而不同功能的词语之间如果出现混乱，会使得代码无法通过编译，因此本文在模型中加入了词嵌入，用于确保语法正确率。同时，词嵌入的参数可以与模型的其余部分同时进行训练，并不会过多增加整个流程的复杂度。

对于词嵌入的方法，本文使用 word2vec<sup>[38]</sup> 来表达词语，记为  $W_e$ ，则词语的独热向量  $S_0$  经过词嵌入后为  $W_e S_0$ 。其中词嵌入的维度选为 32 维。

### 3.4 本章小结

综上所述，如果用  $I$  表示输入图像，用  $S = \{S_1, S_2, \dots\}$  表示输出的代码序列，那么整个生成模型的流程如下：

$$x_0 = \text{CNN}(I) \quad (3-4)$$

$$x_t = W_e S_t, \quad t \in 1 \dots N \quad (3-5)$$

$$p_{t+1} = \text{LSTM}(x_t), \quad t \in 1 \dots N \quad (3-6)$$

这个流程中的每一个部分都是可微的，因此网络中的参数都可以通过随机下降算法同时地进行训练，本文采用了一个参数已经过预训练的 CNN 模型，但 RNN 与词嵌入模型是随机初始化的，所以通过设计每部分不同的学习率，可以控制视觉模型（CNN 部分）和语言模型（RNN 与词嵌入部分）的训练状态。下一章将会详细介绍模型实现中的细节。

## 第 4 章 实现细节

本章将从数据（第 4.1 节）、训练（第 4.2 节）、推断（第 4.3 节）这三个深度学习实验的必须环节出发，展开介绍实验的实现细节。

### 4.1 数据集

由于本文的任务非常新颖，尽管如 2.2 节中所述，最近一年来出现了相关的工作，但没有合适的、大规模的开源数据集，特别地，目前没有开源手写的网页前端草图数据集。因此，在本节中，第 4.1.1 小节将介绍本文提出的随机生成的大规模网页截图训练数据集，随后在第 4.1.2 小节将介绍本文是如何收集手写数据集的。

#### 4.1.1 大规模机器生成的网页截图数据集

收集并标注大规模数据集一直是机器学习中的重要议题，大数据是人工智能研究的燃料，深度学习与机器视觉在这几年的飞速发展很大程度上得益于斯坦福大学开源的 ImageNet 数据集<sup>[33]</sup>。另一方面，建立数据集本身就是在提出问题，即如何训练出模型能够拟合这个数据集的分布。可以说，数据集是任何机器学习任务中至关重要的一个环节，因此如何设计好本文需要的数据集，成为一个重要的问题。

本文提出的模型包括了一个非常深的 CNN 模型和一个有 32 维记忆单元的 LSTM，因此大规模的数据集用于训练是必要的，出于时间精力与经济能力的考虑，进行网络爬虫并标注是不现实的，并且也是难以清理数据和进行学习的，所以本文选择了用程序去自动地、随机地、大规模地生成风格一致的前端静态页面，然后截图并保存代码。同时，这个大规模的机器生成数据集的目的之一是作为在收集的小规模数据集上做迁移学习的基础，所以它应该在元素组成和样式排版上足够多样化，以应对人们手绘草图的变化性。

除此以外，笔者希望数据集侧重的是网页的元素、排版和样式，所以在生成过程中，本文略去了网页中的文本内容，以免增加词嵌入空间的复杂程度（在第 ?? 章中本文会讨论同时生成文本内容的展望），统一用单词“content”来代替文本内容。同理，由于 HTML 中在引用图片时使用的是图片地址，本文将所

有图片统一为一张灰色纯色图片，命名为 img.png。

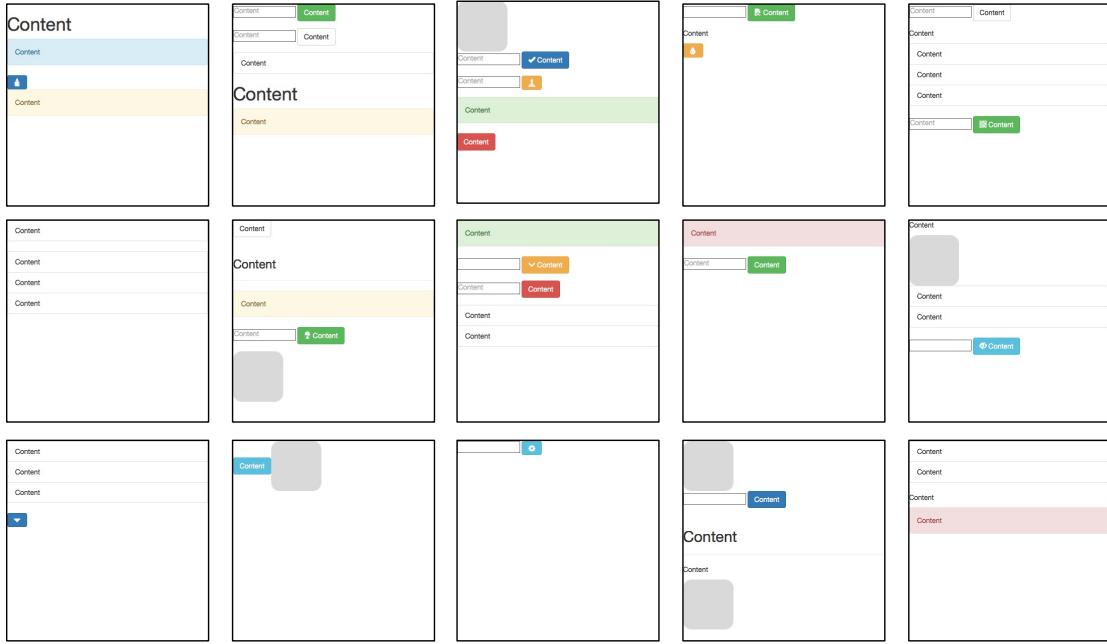


图 4.1 大规模网页截图数据集：选例

与 [31] 类似，本文也使用了 Bootstrap 前端框架以创建更现代化的前端页面以及更简洁的代码，但相比于 [31]，如表4.1所示，本文提出的数据集规模更大，网页元素更多，排版样式更加多样化。

表 4.1 网页截图数据集比较

	训练集 大小	测试集 大小	元素						颜色数
	按钮	列表	表单	标题	段落	图片			
本文提出的大规模 网页截图数据集	1M	1000	✓	✓	✓	✓	✓	✓	6
pix2code 网页截图 数据集 <sup>[31]</sup>	1,500	250	✓			✓	✓		6

图4.2展示了在这个数据集中元素的分布情况，图4.1展示了数据集中的一些样图，可以看到数据集均匀地覆盖了这些网页中的基本元素，并且将它们以各种形式组合了起来。

最后，如何分词并建立词典也是一个值得考虑的问题。在自然语言中，有以

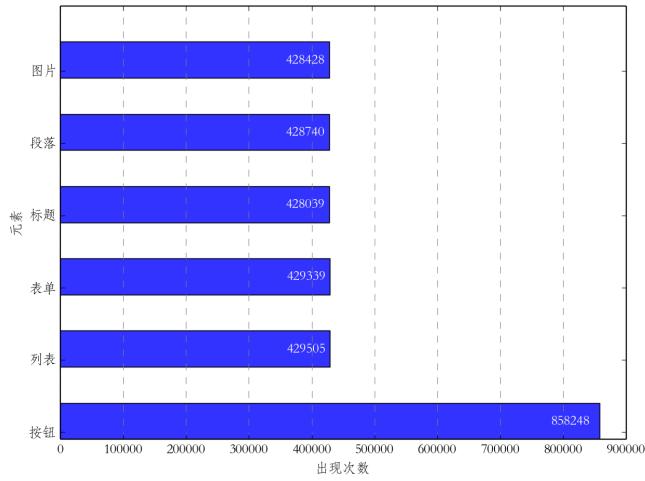


图 4.2 大规模网页截图数据集：元素统计

中文为代表的、词语没有形式上分界符但是可以根据词义进行划分的语言，也有以英文为代表的、单词之间用空格作为自然分界符的语言，而代码则更加复杂，因为它的分界符除了空格，还有各种各样的符号，有些符号需要跟单词分开，以避免造出一些“新词”造成词典的冗余，而有些符号不能跟单词分开，否则无法通过编译，比如在 HTML 中，标签头的尖角符号 < 必须与标签名连在一起，组成一个词语（token）。

最终生成的词典中共有 318 个单词，特别地，其中有 263 个 Bootstrap 小图标的名字，1 个起始符 <s> 和 1 个终止符 </s>。

#### 4.1.2 手写数据集

对数据进行标注一直以来都是一个耗费时间、人力、金钱的工作，相比于将一张手绘的草图进行人工标注其对应的前端代码，本文在采集数据集的时候采取了逆向的思路：即先生成好代码，再让人来模仿它在浏览器里渲染出的前端画面进行绘制。

本文在校内服务器上搭建了一个网站，从 4.1.1 节中的大规模数据集中随机选出代码，交给，然后请了 5 位同学进行绘制（其中 2 位有前端代码编写经验），收集数据的网页工具与收集过程如图 4.3 和图 4.4 所示。一共收集了 200 张手写草图，图 4.5 展示了其中的一些例子。

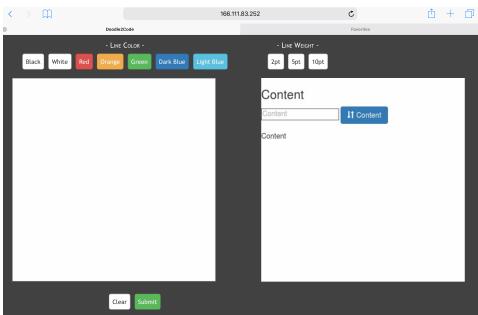


图 4.3 数据收集网站页面

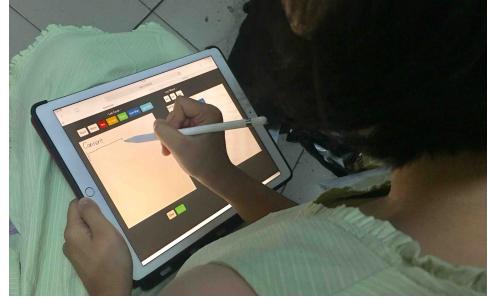


图 4.4 志愿者绘制手写数据

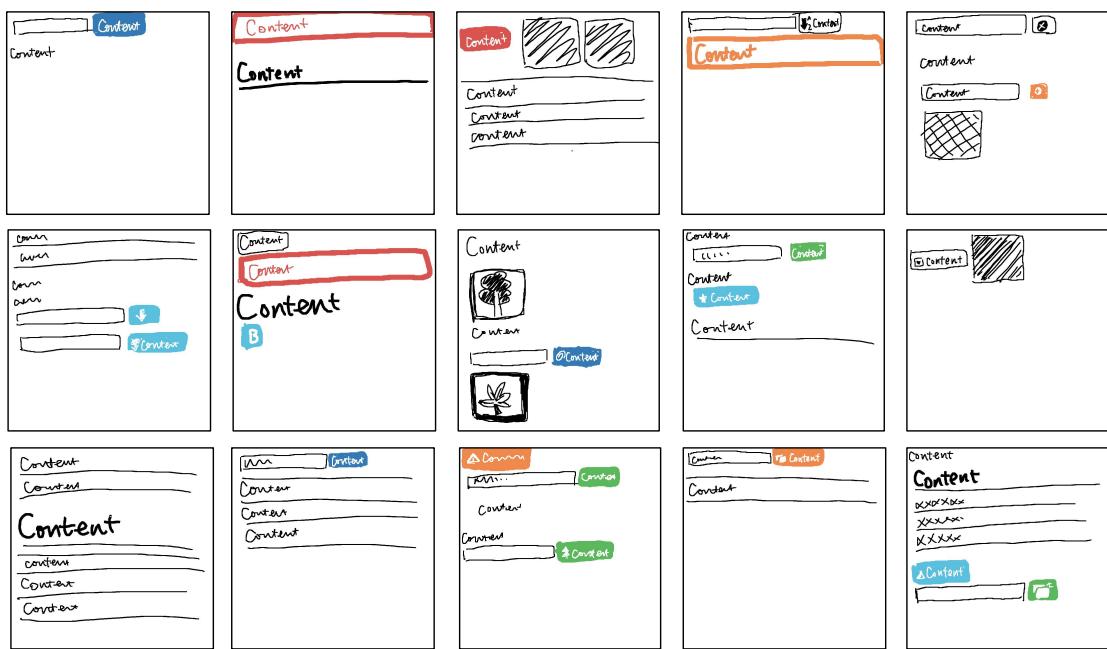


图 4.5 手写数据集选例

## 4.2 数据预处理及训练过程

为了降低训练的难度，参考<sup>[10]</sup>，本文使用了 Inception-v3 模型<sup>[35]</sup>，它的参数在 ImageNet<sup>[33]</sup> 图片分类数据集和 MSCOCO<sup>[34]</sup> 图片描述数据集上进行预训练，而且通过实验证明，经过预训练的模型在泛化能力上明显更强。除此以外，词嵌入和 LSTM 的参数直接进行随机初始化。

网络的输入图像尺寸已经固定，因此在初始化时，输入图像的大小重新拉伸到 256×256 像素大小，并在输入到 CNN 前标准化像素值。与图片描述任务不同，我们不能对图片做随机裁剪、镜面反转等数据集扩大操作，因为生成网页前端代码比用自然语言描述对图片中元素更加位置敏感，所以除了在 RGB 色彩

空间加入一些噪声以外不再做进一步的预处理。

训练的过程以最大化公式3-2中的  $\log p(S|I)$  为目标，因此设计误差函数为：

$$L(I, S) = - \sum_{t=1}^N \log p_t(S_t) \quad (4-1)$$

即将每一时刻 LSTM 输出的概率分布中，取出正确单词对应的的负对数似然值，最后取和。

在大规模网页截图数据集上的训练过程中，本文使用了固定的学习率和每输入完一次全部样本数据（即每个 Epoch）减半的学习率进行了实验，后者效果略有提升。LSTM 和词嵌入的初始学习率设置为 2.0，每个时期（epoch）学习率减半，动量设置为 0，除此之外为了防止 LSTM 过拟合，本文还设置了 LSTM 的 dropout 概率为 0.8。本文还对比了用随机梯度下降算法只训练 LSTM 及词嵌入参数，和除此之外同时小幅调整 CNN 参数（学习率为 0.08）这两个实验，后者的效果更好，但需要更多轮的训练：本文在两块 Nvidia GeForce GTX 1080Ti 显卡上进行训练，一共训练了 100,000 个迭代轮次，每个轮次输入 32 张图片，平均每个轮次训练花费 0.35 秒，一共花费时间大约 10 小时，误差从最初的 0.36 降低到最后稳定的 0.15 左右（见图 4.6）。实验结果详述请见第 5 章。

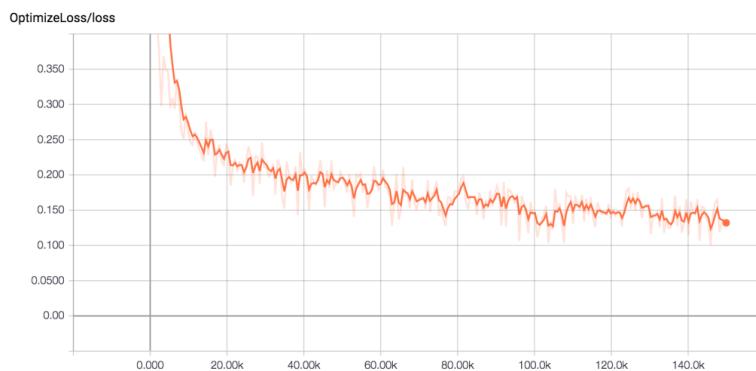


图 4.6 在大规模截图数据集上训练时的误差下降曲线

**迁移学习** 因为手写数据集的数据规模太小，并且图片之间的差异很大（不再像生成的数据集那么规整，多了很多变化的形式，图形中还存在很多抖动和锯齿），而网络参数非常大，非常容易过拟合，于是本文采取了迁移学习的方法在手写数据集上进行训练。具体来说，就是在大规模网页截图数据集上训练好的模型参数基础上做微调（Fine-tuning），将 LSTM 和词嵌入的学习率设置为初

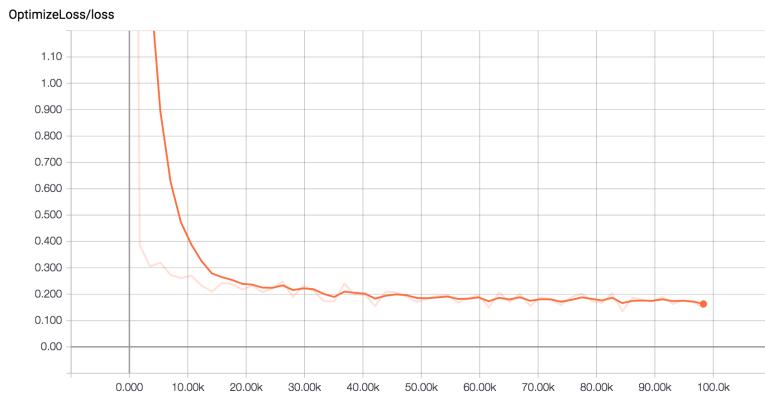


图 4.7 在手写数据集上迁移训练时的误差下降曲线

始为 0.1，每个时期（epoch）学习率减半；CNN 的学习率设置为固定的 0.001，共训练 500 轮，除此之外的参数与在截图数据集上的设置一致。图 4.7 展示了在迁移学习过程中误差的下降情况。

此外，本文还实验了不经过迁移学习和不经过预训练的效果，验证了迁移学习和预训练的必要性。

### 4.3 推断过程

与训练时不同，在对模型进行正向推断的时候，由于没有目标的正确代码序列  $S$ ，在  $t \geq 1$  的每一时刻，模型接受的输入不再是  $S_{t-1}$ ，而应该是输出序列的上一个单词，那么这个单词应该是什么呢？

有很多种方法来确定输出的结果，第一种是直接采样，即根据  $p_1$  采取似然值最大的词作为第一个单词，然后输入这个词的嵌入得到  $p_2$ ，继续如此直到采样到特殊停止符或达到最大长度。

第二种方法是集束搜索（Beam Search），即始终保持最优的  $k$  个候选项作为答案集合，其中  $k$  为束的大小。具体来说，就是迭代地考虑时间  $t$  之前  $k$  个最佳句子的集合，作为候选项分别输入模型生成下一个，即  $t+1$  时刻的单词概率分布，并累计概率值作为分数，得到所有  $k$  个句子推断出下一个单词分布的结果之后，保留其中分数最高的  $k$  个句子，然后继续进行迭代，理论上这能更好地逼近  $S = \arg \max_{S'} p(S'|I)$ 。而在实验中也验证这个方法比直接采样效果更好，因此本文采取了集束搜索方法，并且设置束的大小（即  $k$  的大小）为 20。

在推断的过程中，还有两个参数可以设置，那就是句子的最大长度和长度正则化系数。在 [10] 和 [14] 等图片描述的工作中，将最大长度设为了 40，但显

然 HTML 的代码长度更长，以本文的大规模网页截图数据集为例，平均的代码长度为 39.6 个单词，最长的序列有 135 个单词。因此在推断时设置最长句子限制为 200，实验也表明，如果设置得更短，将会导致代码序列被截断，从而代码不完整，出现语法错误。而长度正则化系数则是一个用于指导输出时更偏向长句或短句的分数系数，大于 1 则偏向于长句，小于 1 则偏向于短句。本文选择保持这个值为 1。

#### 4.4 本章小结

本章介绍了本文建立的两个数据集和在这两个数据集上的实验流程：第一阶段在大规模的截图数据集上进行训练；第二阶段利用上阶段训练好的模型，在小规模的手写数据集上微调以进行迁移学习。下一章将会介绍本文实验的结果。

## 第 5 章 实验结果

本章第 5.1 节首先介绍本文对生成前端代码所采用的评判标准，然后在第 5.2 节中展示实验得到的模型在这些标准下得到的分数和一些测试结果样例的可视化结果，并对测试结果进行分析。

### 5.1 评价标准

相比较于自然语言处理的任务，代码生成任务的实验评价标准的主观性没有那么强，于是能更好的衡量生成结果的质量。相比于 [31] 中使用的分类错误来评价生成的 HTML 代码的质量，本文采用了三种更符合代码特质、语言模型和更接近人类评估水平的评价方法：语法错误率、BLEU 分数和用户编辑距离与时间，共同判断模型的性能。

#### 5.1.1 语法错误率

HTML 语言的编译器就是网页浏览器，但浏览器往往会包容许多 HTML 语言的错误，以尽可能地显示网页内容，而不会报编译错误。所以本文从三个角度去检查 HTML 代码的语法正确性：

1. 标签是否正确闭合。特别是在嵌套多层标签的情况下，能否以正确的顺序结束标签。
2. 属性是否正确排列，属性名是否正确，属性值是否正确。
3. 文本内容是否为单词 `content`。

如果一段代码出现上述三种情况的任意一种，则被判定为错误（Error），所有测试结果统计正确率：

$$A_{gramma} = (1 - n_{error}/n_{samples}) \times 100\% \quad (5-1)$$

#### 5.1.2 BLEU

在自然语言处理中，BLEU 是一种比对两个语言的共现词频率的统计方法，同时考虑了准确率与召回率，测试结果与标注的正确答案越相似，分数就越高，并且在很多文献中被证明与人类评价具有很高的相关性<sup>[39]</sup>。由于在本文的数据

集中每张图片只有一个对应的正确代码序列，因此 BLEU 分数的求解过程为：

$$p_n = \frac{\sum_{n\text{-gram} \in C} Count_{clip}(n\text{-gram})}{\sum_{n\text{-gram} \in C} Count(n\text{-gram})} \quad (5-2)$$

$$BP = \begin{cases} 1 & if c > r \\ e^{1-r/c} & if c \leq r \end{cases} \quad (5-3)$$

$$BLEU = BP \times \exp\left(\sum_{n=1}^N w_n \log p_n\right) \quad (5-4)$$

其中  $C$  是模型输出的一个代码序列， $BP$  是长度惩罚因子， $p_n$  为  $n$  元精度 ( $n$ -gram precision)， $Count_{clip}(n\text{-gram})$  是一个  $n$  元词组在输出序列中的出现次数与在正确序列中的出现次数取较小值， $Count(n\text{-gram})$  则为  $n\text{-gram}$  在  $C$  中的出现次数。而  $w_n$  是一个人为定义的系数， $w_n$  为 1 时得到的是 BLEU-1 分数， $w_n$  为  $\frac{1}{2}$  时得到的是 BLEU-2 分数，以此类推。根据 BLEU 的原理可以看出，通过计算输出结果的 BLEU 分数，能够定性地判断输出代码与正确代码之间内容的相似程度。

本文将报告各项测试的 BLEU-1 和 BLEU-2 分数。

### 5.1.3 编辑距离与编辑时间

本文提出将编辑距离 (Levenshtein distance) 与编辑时间作为代码生成交互体验的评判标准，如图 5.1 所示，本文用训练得到的模型和网页前后端搭建了完整的、用户体验良好的 "Doodle2Code" 软件，用户可以在页面上用鼠标（或者在平板电脑如 iPad 上用电容笔）绘制需要的 Bootstrap 风格前端的彩色草图，提交后实时生成模型输出的代码及对应的渲染结果，用户可以修改代码以修正得到的渲染结果，最后可以导出修改后的 HTML 文件。

本文在此基础上进行了用户研究，用户的修改过程将被记录下来，进而计算编辑距离：每插入、删除、替换一个字符便计算为一个单位距离，另外由于代码比较长的序列相应的编辑距离会较长，因此本文主要参考编辑距离/序列中字符总长度的比例；同时系统还将记录用户的修改时间。

这种判别方法有两个好处：

1. 虽然代码的正确性可以很简便地判断出来，但是代码的正确性不具备唯一性，例如在 HTML 语言中，每个标签的属性可以调换位置，这一般不影响网页渲染出的前端效果，因此我们需要引入一些具有灵活性、主观性的判断标准；
2. 图片生成代码是一个工具性质的应用，所以需要考虑到人机交互的设计，

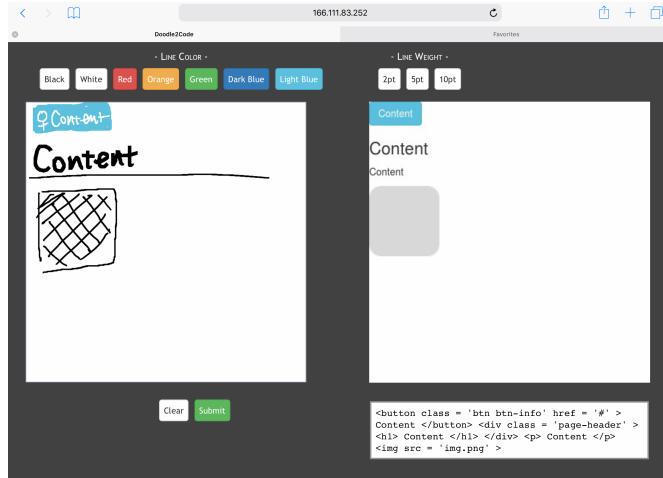


图 5.1 Doodle2Code 用户交互界面

设计人员对最终结果的修改次数与生成的代码质量（例如是否简洁、有序等）和交互方式（例如如何排列输出的代码，是否有简便的定位工具等）相关。

因此考虑用户的编辑次数作为交互体验优化的参考是必要的。

## 5.2 生成结果及分析

表 5.1 实验结果

数据集	模型及训练方法	BLEU-1	BLEU-2	语法正确率
大规模截图数据集	本文模型（没有微调 CNN）	66.21	60.58	100%
	本文模型（同时微调 CNN）	<b>66.45</b>	<b>61.52</b>	100%
	pix2code [31]	50.28	42.61	93.1%
手写数据集	本文模型（没有在大规模数据集预训练）	5.02	1.30	13.2%
	本文模型（没有在手写数据集上迁移训练）	49.12	40.51	100%
	本文模型（经过迁移学习）	<b>49.21</b>	<b>40.98</b>	100%

如表 5.1所示，为在两个数据集用不同的设置训练后的模型进行测试到的 BLEU 分数及语法正确率。

从在大规模的截图数据集上的实验可以看到，在训练阶段经过微调的 CNN

会表现得更好，但令人惊奇的是，即使 CNN 只在 ImageNet 和 MSCOCO 数据集上预训练过，也可以提取出网页截图的特征向量，并且训练出的 LSTM 能输出相当合理的结果，可以看出在大规模数据集上训练后的深度卷积神经网络具有极高的泛化能力。另外，相比于 [31]，本文的模型体现出了更深的卷积神经网络和经过预训练的优势。

在手写数据集上的实验主要验证了在大规模数据集上进行预训练的重要性，直接在小规模的手写数据集上进行训练，会非常容易导致过拟合，笔者调整了许多参数进行实验，效果都非常差。相反的，即使没有在手写数据集上新型微调训练，由于两个数据集的风格仍然是相似的，并且 LSTM 和词嵌入已经学习到了代码的语法结构，模型也能输出相当合理的结果。



图 5.2 迁移学习的作用

经过预训练的模型在手写数据集上进行微调作为迁移学习后，表现有小幅的提升，以图 5.2 为例，手写画相比于标准的网页截图，组件的图形出现扭曲变形，除此之外，由于图形是通过一定宽度的笔触画出来的，会出现涂色不光滑，纹理和边缘更多，而在手写数据集上进行迁移之后能一定程度上增强模型对这种情况的鲁棒性。

表 5.2 推断参数设置对实验结果的影响

最大输出词数	长度正则化系数	BLEU-1	BLEU-2	语法正确率
100	1.0	62.38	57.88	82.9%
200	1.0	<b>66.45</b>	<b>61.52</b>	<b>100%</b>
200	0.5	65.94	61.30	99.9%
200	2.0	50.42	45.75	100%

对于输出代码的语法正确率，从表 5.2 中可以看出，只有在最大单词数较小，或者长度正则化系数小于 1、集束搜索时更偏向更短句时，会出现语法错误的例子，究其原因是因为序列过早结束使输出的代码序列不完整，由此可见模型，尤其是 LSTM 的部分，已经学会了 HTML 代码的语法，能够根据前面代码的记忆信息拟合符合语法的概率分布。

在手写数据集上，表 5.1 对比了两个模型，一个是经过了在大规模截图数据集上训练后、在手写数据集上进行了微调的迁移学习模型，另一个则是直接在手写数据集上进行训练的模型，经过迁移学习的模型有明显的性能提升，第一是因为大规模截图数据集更多样化，能有效地避免过拟合，第二是因为截图数据集和手写数据集的差异在可控的范围内，总体的风格、排版和元素内容是相似的，所以迁移的效果不错。

图 5.3 和图 5.4 分别展示了在两个数据集的测试结果中的一些例子。

### 5.2.1 用户研究

用户研究（User Study）是一种常见于人机交互和交互设计研究中的评估体系，包含多种方法。本文以用户在自搭建的“Doodle2Code”网站上对模型生成代码的编辑次数进行用户研究。本文邀请了五位有至少一年以上前端代码编写经验的同学参与用户研究，每个人进行了 5 次测试，平均修改的编辑距离为 190.31，平均占字符总长度的比例为 43.119%，平均编辑时长为 28.12 秒。

部分结果展示在图 5.4 中。

## 5.3 本章小结

本章对模型在测试集上的推断结果从语法错误率、BLEU 分数和用户研究三个方面进行了评估，首先本文提出的模型相比于以前的相关工作在截图数据集上的效果更好，其次通过迁移学习，本文在手写数据集上得到了不错的结果，并且通过 Doodle2Code 软件进行了用户研究。

输入图像	输出代码	渲染结果
	<pre> &lt;form class = 'form-inline' &gt; &lt;input &gt; &lt;button class = 'btn btn-primary' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-erase' &gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;/ul&gt; &lt;form class = 'form- inline' &gt; &lt;input &gt; &lt;button class = 'btn btn-default' href = '#' &gt; content &lt;/button&gt; &lt;/form&gt; &lt;form class = 'form-inline' &gt; &lt;input &gt; &lt;button class = 'btn btn-primary' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-font' &gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt;</pre>	
	<pre> &lt;div class = 'alert alert-success' &gt; content &lt;/div&gt; &lt;form class = 'form- inline' &gt; &lt;input placeholder = 'content' &gt; &lt;button class = 'btn btn-success' href = '#' &gt; content &lt;/button&gt; &lt;/form&gt;</pre>	
	<pre> &lt;form class = 'form-inline' &gt; &lt;input &gt; &lt;button class = 'btn btn-primary' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-inbox' &gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;form class = 'form-inline' &gt; &lt;input &gt; &lt;button class = 'btn btn- primary' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-circle-arrow-right' &gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; content &lt;/li&gt; &lt;/ul&gt;</pre>	
	<pre> &lt;button class = 'btn btn-warning' href = '#' &gt; content &lt;/button&gt; &lt;div class = &gt;alert alert-warning' &gt; content &lt;/div&gt; &lt;div class = 'alert alert-default' &gt; content &lt;/div&gt;</pre>	

图 5.3 在大规模网页截图数据集上的测试结果选例

输入图像	输出代码	渲染结果
	<pre>&lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;/ul&gt; &lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;/ul&gt; &lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;/ul&gt; &lt;form class = 'form-inline' &gt; &lt;input&gt; &lt;button class = 'btn btn-warning' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-warning-sign'&gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;ul class = 'list-group' &gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;li class = 'list-group-item' &gt; Content &lt;/li&gt; &lt;/ul&gt;</pre>	
	<pre>&lt;img src = 'img.png' &gt; &lt;button class = 'btn btn-primary' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-bookmark'&gt; &lt;/span&gt; &lt;/button&gt; &lt;form class = 'form-inline' &gt; &lt;input placeholder = 'Content' &gt; &lt;button class = 'btn btn-warning' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-ice-lolly'&gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;form class = 'form-inline' &gt; &lt;input placeholder = 'Content' &gt; &lt;button class = 'btn btn-warning' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-unchecked'&gt; &lt;/span&gt; Content &lt;/button&gt; &lt;/form&gt; &lt;img src = 'img.png' &gt;</pre>	
	<pre>&lt;form class = 'form-inline' &gt; &lt;input placeholder = 'Content' &gt; &lt;button class = 'btn btn-success' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-tag'&gt; &lt;/span&gt; &lt;/button&gt; &lt;/form&gt; &lt;div class = 'page-header' &gt; &lt;h2&gt; Content &lt;/h2&gt; &lt;/div&gt;</pre>	
	<pre>&lt;button class = 'btn btn-danger' href = '#' &gt; Content &lt;/button&gt; &lt;button class = 'btn btn-danger' href = '#' &gt; &lt;span class = 'glyphicon glyphicon-ice-lolly'&gt; &lt;/span&gt; &lt;/button&gt; &lt;div class = 'alert alert-warning' &gt; Content &lt;/div&gt; &lt;h1&gt; Content &lt;/h1&gt;</pre>	

图 5.4 在手写数据集上的测试结果选例

## 第 6 章 结论和展望

### 6.1 结论

本文提出了一个端到端可微分的深度神经网络模型来解决从图片生成网络前端代码序列的任务，它可以自动接受图片输入，并输出无语法错误的、合理的 HTML 代码。这个模型采用 CNN 编码—LSTM 解码的结构，并结合了词嵌入空间，用深度卷积神经网络提取图片的固定长度特征向量，和经过了词嵌入的词向量一起逐个传递给 LSTM，LSTM 在每一时刻输出在嵌入空间上的概率分布  $p$ ，该模型被训练以最大化目标正确代码序列的概率值，而在推断的阶段采取集束搜索动态搜索最优解。

本文在这个任务上建立了两个数据集，分别是 1M 大规模机器生成的网页截图数据集，和收集的手写网页草图数据集，从语法正确率、BLEU 分数和人机交互编辑距离三个方面进行评估，本文验证了从网页截图和手写草图该模型都能够输出合理的代码内容，相比于之前的工作，本文的模型面对多样化的输入更加鲁棒。更重要的是，本文验证了从大规模数据集进行预训练后再迁移学习到手写数据集上，能有效提升在手写数据集上的泛化能力和稳定性。

本文用训练得到的模型和网页前后端搭建了完整的、用户体验良好的“Doodle2Code”软件，用户可以在页面上用画笔绘制想要的网页设计排版，经模型前馈得到输出代码后在文本框中显示出代码，并实时渲染出网页面面，用户可以修改文本框中的代码以修正模型的输出结果。本文在此基础上进行了用户研究，并且创新性地将编辑距离和编辑时间作为网络前端代码生成交互体验的评判标准。

### 6.2 未来工作展望

由于这个任务还处于起步阶段，相关的研究并不太多，后续的工作可以从很多角度入手，这里列举三个方面：

1. 生成更完整的网页内容。结合 LPN<sup>[28]</sup> 在代码生成上的经验，将代码分区、分类，每一部分的代码用不同的生成器进行预测，例如组件排列、样式部分的代码由一个类似于本文中的 RNN 解码器生成，文本内容由一个文字

识别器生成，而图片内容可以由最近发展迅速的对抗生成神经网络生成。

2. 网页图片的复杂度可以继续增加，例如增大网页的尺寸、增添左右排列、增添常见的网页排版等，甚至直接通过网页爬虫得到实际网站中的截图与代码。
3. 手写数据集收集的难度很大，是否可以用无监督的情况下进行迁移学习，或者构建在线学习的模型，能在人机交互之间进行学习从而改进模型，这会是很有趣的研究话题。

## 插图索引

图 1.1	从图片生成网页前端代码流程图 .....	2
图 2.1	RNN 与 CNN 共同作为编码器的方法 <sup>[31]</sup> .....	7
图 2.2	[32] 模型测试结果 .....	7
图 3.1	本文提出的网络模型架构图 .....	10
图 3.2	Inception-v3 网络结构图 .....	11
图 3.3	LSTM 网络结构图 .....	12
图 4.1	大规模网页截图数据集：选例 .....	15
图 4.2	大规模网页截图数据集：元素统计 .....	16
图 4.3	数据收集网站页面 .....	17
图 4.4	志愿者绘制手写数据 .....	17
图 4.5	手写数据集选例 .....	17
图 4.6	在大规模截图数据集上训练时的误差下降曲线 .....	18
图 4.7	在手写数据集上迁移训练时的误差下降曲线 .....	19
图 5.1	Doodle2Code 用户交互界面 .....	23
图 5.2	迁移学习的作用 .....	24
图 5.3	在大规模网页截图数据集上的测试结果选例 .....	26
图 5.4	在手写数据集上的测试结果选例 .....	27

## 表格索引

表 4.1	网页截图数据集比较 .....	15
表 5.1	实验结果 .....	23
表 5.2	推断参数设置对实验结果的影响.....	24

## 公式索引

公式 3-1 .....	9
公式 3-2 .....	9
公式 3-3 .....	11
公式 3-4 .....	13
公式 3-5 .....	13
公式 3-6 .....	13
公式 4-1 .....	18
公式 5-1 .....	21
公式 5-2 .....	22
公式 5-3 .....	22
公式 5-4 .....	22

## 参考文献

- [1] Farhadi A, Hejrati M, Sadeghi M A, et al. Every picture tells a story: Generating sentences from images[C]//European conference on computer vision. : Springer, 2010: 15-29.
- [2] Li S, Kulkarni G, Berg T L, et al. Composing simple image descriptions using web-scale n-grams[C]//Proceedings of the Fifteenth Conference on Computational Natural Language Learning. : Association for Computational Linguistics, 2011: 220-228.
- [3] Kuznetsova P, Ordonez V, Berg A C, et al. Collective generation of natural image descriptions [C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. : Association for Computational Linguistics, 2012: 359-368.
- [4] Ordonez V, Kulkarni G, Berg T L. Im2text: Describing images using 1 million captioned photographs[C]//Advances in neural information processing systems. 2011: 1143-1151.
- [5] Fang H, Gupta S, Iandola F, et al. From captions to visual concepts and back[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1473-1482.
- [6] Donahue J, Anne Hendricks L, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 2625-2634.
- [7] Kinghorn P, Zhang L, Shao L. A region-based image caption generator with refined descriptions [J]. Neurocomputing, 2018, 272: 416-424.
- [8] Kiros R, Salakhutdinov R, Zemel R S. Unifying visual-semantic embeddings with multimodal neural language models[J]. arXiv preprint arXiv:1411.2539, 2014.
- [9] Mao J, Xu W, Yang Y, et al. Deep captioning with multimodal recurrent neural networks (m-rnn)[J]. arXiv preprint arXiv:1412.6632, 2014.
- [10] Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator[C]// Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. : IEEE, 2015: 3156-3164.
- [11] Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3128-3137.
- [12] Mathur P, Gill A, Yadav A, et al. Camera2caption: A real-time image caption generator[C]// Computational Intelligence in Data Science (ICCIDDS), 2017 International Conference on. : IEEE, 2017: 1-6.
- [13] Liu C, Wang C, Sun F, et al. Image2text: a multimodal image captioner[C]//Proceedings of the 2016 ACM on Multimedia Conference. : ACM, 2016: 746-748.

- [14] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]//International Conference on Machine Learning. 2015: 2048-2057.
- [15] Chen L, Zhang H, Xiao J, et al. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning[J]. arXiv preprint arXiv:1611.05594, 2016.
- [16] Li L, Tang S, Deng L, et al. Image caption with global-local attention.[C]//AAAI. 2017: 4133-4139.
- [17] Pedersoli M, Lucas T, Schmid C, et al. Areas of attention for image captioning[C]//ICCV- International Conference on Computer Vision. 2017.
- [18] Wang Y, Lin Z, Shen X, et al. Skeleton key: Image captioning by skeleton-attribute decomposition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 7272-7281.
- [19] Lu J, Xiong C, Parikh D, et al. Knowing when to look: Adaptive attention via a visual sentinel for image captioning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): volume 6. 2017.
- [20] Ren Z, Wang X, Zhang N, et al. Deep reinforcement learning-based image captioning with embedding reward[J]. arXiv preprint arXiv:1704.03899, 2017.
- [21] Jia X, Gavves E, Fernando B, et al. Guiding the long-short term memory model for image caption generation[C]//Computer Vision (ICCV), 2015 IEEE International Conference on. : IEEE, 2015: 2407-2415.
- [22] Johnson J, Karpathy A, Fei-Fei L. Densecap: Fully convolutional localization networks for dense captioning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4565-4574.
- [23] Mao J, Huang J, Toshev A, et al. Generation and comprehension of unambiguous object descriptions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 11-20.
- [24] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.
- [25] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.
- [26] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]// Advances in neural information processing systems. 2014: 3104-3112.
- [27] Mou L, Men R, Li G, et al. On end-to-end program generation from user intention by deep neural networks[J]. arXiv preprint arXiv:1510.07211, 2015.
- [28] Ling W, Grefenstette E, Hermann K M, et al. Latent predictor networks for code generation [J]. arXiv preprint arXiv:1603.06744, 2016.
- [29] Karpathy A. Char-rnn: Multi-layer recurrent neural networks (lstm, gru, rnn) for character-level language models in torch[Z]. 2015.

- [30] Moran K, Bernal-Cárdenas C, Curcio M, et al. Machine learning-based prototyping of graphical user interfaces for mobile apps[J]. arXiv preprint arXiv:1802.02312, 2018.
- [31] Beltramelli T. pix2code: Generating code from a graphical user interface screenshot[J]. arXiv preprint arXiv:1705.07962, 2017.
- [32] Wallner E. Turning design mockups into code with deep learning[Z].
- [33] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.
- [34] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//European conference on computer vision. : Springer, 2014: 740-755.
- [35] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.
- [36] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous systems[EB/OL]. 2015. <https://www.tensorflow.org/>.
- [37] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [38] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [39] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. : Association for Computational Linguistics, 2002: 311-318.

## 致 谢

衷心感谢我的导师李春平教授在毕业设计期间对我的精心指导、鼓励与信任。感谢本文在采集数据期间给予过帮助的同学们。感谢香港科技大学的 Sung Kim 教授，我在港科的研修经历为本文提供了灵感。感谢加州大学伯克利分校的 Trevor Darrell 教授、商汤科技公司的技术主管钱晨，还有与我共同工作、给予我无数经验和指导的学长学姐们，感谢你们在学术道路上对我的引导。

感谢清华大学软件学院的各位老师、领导、学长学姐，还有四字班的同学们，谢谢你们在大学四年里对我的各种帮助和指教。特别感谢四字班辅导员曹越学长，在我最迷茫焦虑的时候，您的开导让我找到了前进的方向。特别感谢我的两位室友，谢谢你们每天带给我的陪伴和欢乐。

最后，感谢我的父母一直以来对我支持与关爱，你们是我努力生活的最大动力，我爱你们。

## 声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： 蒋梦青 · 日 期：2018.06.25

## 附录 A 外文资料的调研阅读报告或书面翻译

### 一种基于区域的精细图片描述生成器

**摘要:** 描述图像内容是一项具有挑战性的工作,为了能输出具体的描述,需要对图片中的物体、人、关系和相关的属性进行检测与识别。目前大多数现有的研究都基于全局整体的方法,却会丢失场景中一些重要的细节。为了解决这样的挑战,我们提出了一种全新的基于区域的深度学习架构以解决图片描述生成,它采用了一个区域性的物体检测器,基于循环神经网络(RNN)的属性检测,和一个嵌入了两个RNN的编码一解码结构的语言生成器来生成给定图片的精细具体描述。更重要的是,我们提出的系统关注于一个基于局部的方法来改进现有的整体方法,尤其涉及图片中人和物体的区域。我们使用IAPR TC-12数据集进行实验,我们所提出的系统的性能令人印象深刻,并且优于使用VAR评价指标的最先进的方法。特别是,当处理跨域室内场景图像时,我们所提出的系统显示出优于现有方法的性能。

## 1 简介

描述图片的内容对人来说是一件相对简单的任务,因为人可以识别、判断并描述物体、场景和动作,即使被很多别的因素影响,比如遮挡物、光线变化和姿态变化,而这些因素却会让机器很难去完成这个任务。最近的研究展示出最新的机器视觉算法已经在图片分类和分割等领域已经有相当强的能力,但是生成图片内容的精细描述仍然是一个困难的任务。

近年来,已经提出了许多方法来描述图像生成。然而,大多数相关的研究依赖于图像理解和实体识别的整体方法,这可能会丢失与图像的重要方面有关的细节。为了实现细化和详细描述,本研究的目的是提出一种新的用于图像描述生成的局部深度学习体系结构。它采用区域对象检测器、基于循环神经网络进行属性分类和基于编码器-解码器的RNN来生成图像内容的精细描述。最重要的是,所提出的系统侧重于基于局部的方法来改进现有的整体方法,这涉及到在给定图像中的人和对象的图像区域。

我们提出的系统包括四个关键阶段:(1)目标检测和识别;(2)属性预测;(3)

场景分类；(4) 描述生成。整个系统结构如图 1 所示。在第一阶段，利用大规模的深度卷积神经网络 (CNN) 来实现对象检测器，用于定位和分类图像中的人和物体，这个 CNN 提供边界框和对象类标签作为输出。在第二阶段，将上述 CNN 应用于检测区域，以提取用于后续属性预测的特征。实现了两个 RNN 用于局部区域的属性分类，其中一个专用于人类属性预测，另一个应用于对象属性预测。在第三阶段，机器学习的整体图像特征提取使用上述美国有线电视新闻网的场景分类。在第四阶段中，被识别的对象、场景、人及其相关的属性标签被传递给编码器-解码器结构，其由两个 RNN 组成，以将类和属性标签转换为完整的描述。

本文的主要贡献如下两点：

- 我们提出了一种用于图像描述生成的基于局部区域的深度学习体系结构。为了克服现有的整体方法的局限性，我们采用了区域对象检测器、基于 RNN 的属性预测和基于编码器-解码器的描述生成器。特别是，在本研究中，基于句子的描述生成的挑战被视为机器翻译问题。
- 我们使用 IAPR TC-12 数据集对所提出的系统进行了综合评价。实证结果表明，这个系统表现出令人印象深刻的性能，并且在几乎所有的评价指标上都优于最先进的相关研究。特别是，该系统在处理从 NYV2 语句数据集中提取的跨领域场景图像时表现出极大的优越性和效率，这与用于训练系统的数据集完全不同，被认为是对现有研究的巨大挑战。

论文的结构如下：第 2 节讨论相关的研究现状。在第 3 节中，我们提出了该系统，包括目标检测和识别、场景分类、基于 RNN 的属性预测和基于编码器-解码器的句子生成。在第 4 节中提供了综合评价。第 5 节总结了我们的工作，并确定了潜在的未来研究领域。

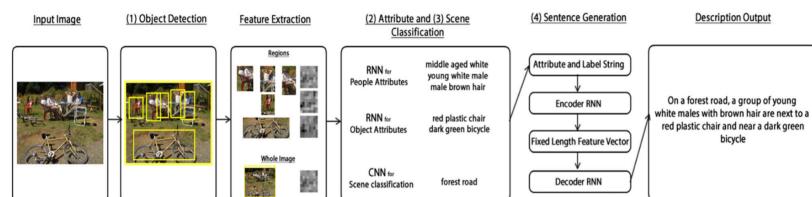


Fig. 1. The overall system architecture, which consists of (1) object detection and recognition, (2) attribute prediction, (3) scene classification and (4) sentence generation.

图 1

## 2 相关工作

在这一部分中，我们回顾了目前最先进的对象检测和识别方法、属性预测和图像描述生成。

### 2.1 物体检测和识别

Girshick 等人 [2] 提出一个物体检测和语义分割的算法，结合了带有 CNN 特征的候选区域，被称为 R-CNN，他们的系统提取出候选区域并且应用于 CNN 来得到后续的特征，为每一个候选区域学习生成一个固定维度的特征向量。线性支持向量机（SVMs）被用于对每个区域进行分类。他们的系统在物体检测上表现明显优于滑动窗 CNN 结构，比如 OverFeat。

### 2.2 属性预测

属性已经被广泛地应用于许多研究领域，或者通过向标签添加高级别信息或通过使用它们来帮助其他计算机视觉任务，例如场景或对象分类 [4-7]。Farhadi 等 [4] 根据对象的属性描述对象，使不能被分类为特定对象标签的对象按其形状、大小、颜色或纹理来描述，他们的系统利用边缘检测，使用 HOG 算子进行特征提取，并使用基于 LAB 的描述符进行颜色提取。属性被 Dhar 等人 [5] 用于预测美学和趣味性，他们的工作展示了成分属性的预测，他们还使用属性来表示对象、动物和/或人的存在以及光线环境。为了确定图像的“趣味性”，他们也提取了颜色直方图、HARR 特征和空间金字塔。Bourdev 等人的研究 [6] 使用姿势信息（称为 poselet）来确定人的属性，例如性别和服装，他们的系统由四个阶段组成：阶段 1 检测图像中的所有 poselet。在第 2 阶段，提取特征向量作为每个 poselet 类型。在第 3 阶段，使用 poselet 级别的属性分类器来预测属性的存在。在第 4 阶段，使用人级别的线性属性分类器组合来自所有身体部位的信息进行属性预测。Lang 和 Ling [7] 融合了图像特征和视觉属性来进行隐蔽照片分类，他们首先处理每个图像以产生空间金字塔，然后对于每个子图像提取特征用于视觉属性分类，他们采用了 13 种视觉属性，包括图像亮度、颜色丰富度等，然后将图像特征和视觉属性的融合用于隐蔽图像分类。

虽然上述相关工作获得了令人印象深刻的预测性能，但大多数这样的应用使用属性聚类的分类器集群，每个个体分类器在与相应属性标签有关的特定特征集上进行训练 [6][7]。这样的方法只能标记每个属性为当前的或其他的。本研究的目的是克服这种限制，通过采用 RNN 预测高度相关的属性以提高系统的描

述能力。

### 2.3 图片描述生成

近年来，自动图像描述引起了人们的广泛关注 [8—21]。KalPosiy 和 Li[8] 提出了一种对图像区域进行自然语言描述的系统，采用 CNN 从图像区域提取特征，采用 RNN 生成每个区域的短句子描述。他们的系统的一个版本，也称为 NoNalTalk，也能够为整个图像生成描述，他们使用 FLICKR8K、FLIKR30K 和 MSCCOO 数据集对工作进行了评估，并取得了令人印象深刻的性能。Vinyals 等人 [9] 还提出了一种基于整体方法的神经图像描述生成器，称为 NIC。它集成了深度 CNN 作为视觉特征学习的图像编码器和用于描述生成的 RNN，当使用 PASCAL、FLICKR8K、FLIKR30K 和 SBU 数据集进行评估时，BLEU 评分与其他最先进的系统相比显著提高。然而，他们的工作没有提取任何高层次的语义信息，如属性或关系的描述生成。

Lin 等人 [10] 提出了一种生成复杂室内场景多语言描述的框架。他们的工作主要集中在 3D 解析系统的开发上，以便生成场景的语义表示。具体来说，他们采用全局条件随机场 (CRF) 进行场景图生成。结果表明，基于 CRF 的描述生成 [11] 易受缺乏多样性的影响，因为它只是简单列出所有的结果来形成一个整体描述。Mathews 等最近的工作 [12] 试图通过引入情感元素来解决图像描述系统的局限性，他们系统的体系结构将基于 CNN 的“事实”描述生成与基于 RNN 的情感词生成相结合，以产生最终的描述。举个例子，他们的系统输出可以从“躺在床上的黑白猫”变成“躺卧在沙发上的可爱猫”。

作为一个著名的图像描述框架，Xu 等人的工作 [13]- “Show, Attend and Tell” 采用著名的 CNN + RNN 结构，并且在 RNN 内嵌入了注意力模型，允许模型学习下一个单词序列所需关注的显著区域，他们的注意力模型也可以被改变为硬或软注意机制。Lu 等人的工作 [14] 产生一个称为自适应注意的框架，他们的工作不需要注意模型对于每个单词的生成都是主动的。例如，不需要利用注意力和视觉信息来生成诸如“and”、“the” 以及可以从语言模型推断出、不需要视觉输入的其他单词。他们的模型在每个时刻确定是否需要视觉注意。此外，在 [13, 14] 中使用的两种注意力模型都试图模仿人类观看图像的方式，意图集中于图像的最重要方面。Liu 等人 [15] 提出了一种基于传统 CNN+RNN 结构的“Image2Text” 框架，然而它用检测到的对象表示替换 CNN，然后将其传递给 RNN 用于描述生成。

Johnson 等人提出的 DenseCap[16] 关注具体区域描述的生成上，而不是典型的全文描述，这个工作通过结合 Ren 等人的区域推荐网络（RPN）[17] 在显著区域延续了相似的原理，这个 RPN 组件是被插入到系统中，在基于 CNN 的特征提取和基于 RNN 的语言模型之间，作一个局部化的层。DenseCap 在效果上与 Mao 等人的工作 [18] 相似，目的都是描述图片中的具体物体或者区域。

You 等人最近的工作 [19] 旨在将自上而下和自底向上的方法在图像描述领域结合起来，通过 CNN 提取特征并使用它们使系统给出对图像的初步概览理解，然后这允许注意机制来关注图像中的认知线索，这些线索用于在它们的 RNN 结构中生成后续单词。为了生成属性，他们提出了两种方法，即（1）一种非参数方法，它利用具有成对数据的大语料库的最近邻图像检索，和（2）分别用排序损失和完全卷积网络训练的两个参数模型进行属性预测。如果需要的话，这两种方法可以结合使用。

### 3 本文提出的图片描述生成系统

如前面所讨论的，最新的方法很大程度上依赖于整体技术来提取字幕生成的图像特征 [8][9]。这种方法的主要缺点（例如 NIC）是会忽略本地信息。为了克服这种局限性，本研究提出了一种基于区域的深度学习方法，通过结合对象检测和识别、场景分类、属性预测和描述生成来捕获局部细节。此外，将描述生成的挑战视为机器翻译问题，因此提出了一种用于句子生成的编码器-解码器结构。所提出的系统具有优于现有的最先进的方法，特别是当处理图像描述任务的域外图像。我们在下面的小节中详细介绍了所提出的系统的每一个主要步骤。

#### 3.1 物体检测与识别

对于稳健的目标检测，我们实现了 R-CNN[2] 目标检测器，R-CNN 是一个传统的 CNN，但是有额外的输出来预测包围盒坐标。R-CNN 对 ILSVRC13 数据集进行训练，它能够检测、定位和分类 200 个对象类别。R-CNN[2] 使用 ImageNet 数据库 [22] 对 Krizhevsky 等人所采用的 CNN 进行训练 [23]。

具体而言，本研究中用于对象检测的 R-CNN 应用选择性搜索（Selective Search, SS）算法 [24] 从可能包含对象和/或人的图像中收集感兴趣区域（ROI），先采用贪婪搜索算法，然后分组相似的区域，并测量区域和它们的近邻之间的相似性，重复这个过程直到整个图像变成一个区域。在贪婪搜索算法中，区域 A

和 **B** 之间的相似度被定义为：

$$(a, b) = S_{size}(a, b) + S_{texture}(a, b) \quad (1)$$

它返回一个在  $[0, 1]$  之间的值。在等式 (1) 中,  $S_{size}(a, b)$  是 **A** 和 **B** 都占据的比例, 它趋向于使小区域合并, 而  $S_{texture}(a, b)$  是类 SIFT 测量之间的交集。该区域经变形以适应要求的 227×227 像素的输入。

对于每个候选区域, 提取 4096 维的特征向量。在测试期间, 将每个特征向量传递到代表每个对象类的 200 个支持向量机中的每一个, 以传递每个区域的分数。如果某个区域具有与预定义阈值相对较高的分数的联合 (IOU) 的交叉点, 非最大抑制 (NMS) 也将被用于处理该区域。

Girschick 等人实现的 R-CNN[2] 在本研究中被修改, 用于裁剪和存储图像、包围框、每个框各自的类标签和得分, 其中类标签被连接起来以形成检测列表, 然后将检测和区域再次裁剪并标记为对象或人, 该附加标签用于确定系统在后续处理中使用的哪些属性预测器 (例如, 对象或人属性预测器)。

### 3.2 场景分类

本研究采用场景分类的方法, 进一步提高了系统的描述性。我们的系统不是简单地陈述室内或室外, 而是通过场景分类产生语义场景标签 (例如公园或购物中心), 以便提供更精细的描述。

该场景分类采用混合 AlexNet CNN[23], 它被训练来检测 1183 个类别, 包括 205 个场景标签和 978 个对象类别。然而, 在我们的工作中, 这个网络不能作为对象检测器来应用。这是因为混合 CNN 网络不提供物体定位, 即包围框, 但是这对于我们将必不可少的。所以它被用来对 205 个场景标签进行分类, 并确保对描述生成的有效覆盖。总体而言, 它用相对较低的计算成本表现出引人注目的准确性, 同时极大地提高了所提出的系统的描述能力。

### 3.3 基于 RNN 的属性预测

传统上, 属性通常被聚类分类器进行分类, 其中一个分类器专用于一个属性 [4][6], 但是这样的分类器簇不仅需要更多的资源用于训练, 而且还倾向于仅指示属性的存在或不存在, 而不需要任何置信度度量。

在这项研究中, 我们采用基于 RNN 的属性分类以解决上述缺点。研究表明, RNNs 在机器学习的许多领域都是有用的, 包括图像描述和机器翻译 [9][25]。我们在这项工作中使用 RNN 是因为它们不仅能够对任意数量的人和对象的属性

进行分类，而且还能够有效地确定应该为给定的特征集动态地报告哪些属性。本研究采用两个 RNN 进行属性分类，其中一个专用于人类属性预测，另一个应用于对象属性分类。分割人和对象属性分类的目的是减少所需的训练数据量，以及防止人为属性生成对象的描述，反之亦然。

此外，由于 RNNs 在处理自然语言处理任务中的重要性和灵活性，因此已经被用于属性预测。RNN 与其他替代方法的初始比较也表明 RNN 比其他方法（如支持向量机的属性预测）具有更好的精度。

此外，在这项工作中使用的 RNNs 是“基于字的”，并采用长短期记忆网络结构（LSTM）[26]。在测试时，使用上述 CNN 提取图像特征，然后利用提取的特征逐个预测多个属性。属性预测是基于所提取的图像特征与先前生成的单词相结合，此过程继续生成相关属性，直到生成指定的停止字为止，即，当 RNN 确定没有其他属性可用于描述图像时，基于特征和所有先前生成的属性，系统生成该停止字。

在这项研究中，PubFig[27] 和 ImageNet[22] 的子集也分别用于基于 RNN 的人类和对象属性预测的训练。我们对 Krizhevsky 等人的 AlexNet[23] 稍加修改，删除掉了它的分类层用来提取训练 RNN 用的 CNN 特征。该网络从所需对象或人的先前裁剪图像中提取 4096 维图像特征向量，然后将它们与来自相应属性数据集的属性标签配对以进行训练。

### 3.3.1 人类属性

**Table 1**  
Human attributes employed in this research.

Hair color	Brown, blonde, black, gray, etc.
Hair style	Wavy, curly, straight, etc.
Age	Child, youth, middle aged, senior
Gender	Male, female
Ethnic	Asian, White, Indian
Accessories	Glasses, sunglasses, lipstick, necklace, necktie
Facial hair	Goatee, moustache

表 1

本研究采用 PubFig[27] 数据集对人类属性预测的 RNN 进行训练，共包含 10000 幅图像。它有 200 个独特的名人脸，每个标记有 73 个属性，如年龄，性别和发型。用于描述本研究中的人的属性是在 PubFig 数据集的子集，总共 26 个人类属性，如表 1 所示。

### 3.3.2 物体属性

ImageNet[22] 数据集在许多计算机视觉挑战中得到广泛应用，它也有一个较小的图像子集，有物体包围框和它们各自的属性完全标注。该子集由从 400 个集合中收集的大约 10000 个图像组成。每个同步集表示与特定的 WordNet[28]ID 相关联的一组 ImageNet 图像。每个图像与 25 个对象属性配对，并且所有这 25 个属性都被用于本研究中，它们在表 2 中示出。

**Table 2**

Object attributes employed in this research.

Color	Black, blue, brown, gray, green, orange, pink, red, violet, white, yellow
Pattern	Spotted, striped
Shape	Long, round, rectangular, square
Texture	Furry, smooth, rough, shiny, metallic, vegetation, wooden, wet

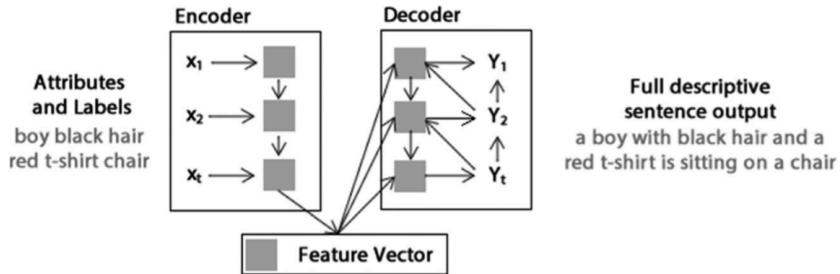
表 2

总的来看，与传统的基于支持向量机的属性分类相比，基于 RNN 的对象和人类属性预测对于高度相关属性的分类表现出很大的效率，并且提供了有效的灵活性和多样性，以帮助后续的句子生成。

### 3.4 句子生成

在这项研究中，我们把语言生成看作机器翻译问题，基于所提出的基于局部的方法，旨在生成的句子比其他现有的研究所产生的更具描述性：具有属性和标签细节。为此，IAPR TC-12 数据集 [29] 被用于训练和评估所提出的系统，因为它的描述与其他流行数据库（如 Flickr）所提供的相比更具体。

在该研究中，我们提出了一种编码器-解码器 RNN 结构，以克服机器学习领域中的典型挑战，如可变长度的输入。它包括两个 RNNs，一个用作编码器，另一个作为解码器，以解决输入长度变化问题。这种编码器-解码器体系结构最初是由 Bahdanau 等人提出 [25] 用于机器翻译问题，我们变换它来处理句子生成，并使用编码器 RNN 将名词（对象/人）和形容词（属性）关键字编码成固定长度向量。采用解码器 RNN 将这个固定长度的向量转换成完整的描述性句子。编码器-解码器语言生成器的结构如图 2 所示。



**Fig. 2.** The encoder-decoder architecture with example attribute and label inputs and the expected sentence generation output.

图 2

我们所提出的语言生成组件使用来自 MSCOCO[30] 的图片描述数据和 IAPR TC-12 (29) 的小子集来训练，以推断不同类型的句子结构。具体来说，MSCCOCO[30] 数据集中的每个图像与五个描述配对，IAPR TC-12 数据集每个图像只有一个句子，但是，它比由其他数据集提供的描述更具描述性（例如 Flickr30K[31]）。因此，我们提出的句子生成结构（即，编码器-解码器）对由上述训练数据集提供的描述的名词和形容词进行训练。每一个标题都是用 Python NLTK（自然语言工具包）(<http://www.nltk.org/>) 标记的第一个词条，可以提取和存储名词和形容词。这形成了一个由约 30000 个独特的单词组成的源训练集，和一个由约 20000 个独特的单词组成的目标训练集组成。训练集的整体大小超过 500000 个描述，超过 20000 个用作验证集。

**Table 3**  
Information used for sentence generation.

Recognized entities and attributes	Semantic labels
Scenes	205 scene labels from MIT scenes including playground, living room and bedroom, etc.
People	Present/non-present
Objects	200 object categories from ILSVRC13 including car, dog, orange, desk, etc.
Person attributes	A subset of 26 human attribute labels taken from the PubFig dataset
Object attributes	All 25 object attribute labels present in the ImageNet dataset

表 3

编码器和解码器 RNNs 都是基于 [25] 实现的。所有大于 50 个词的序列都被略去。每个 RNN 由单个隐藏层中的 1000 个隐藏单元组成，以计算下一个目标词的概率，利用 Adadelta 对这些 RNN 模型进行随机梯度下降训练，在训练过程中，每一个更新都输入 80 个句子的小批次。当 RNN 已被完全训练时，执行集束搜索以生成能最大化源句子匹配目标句子的条件概率的描述，即  $-\arg \max_y P(y|x)$ 。在测试阶段，在早期阶段生成的对象/人/场景和属性标签被用作输入源语言，以

完整的标题作为预期输出。总的来说，在句子结构中考虑的信息概括在表 3 中。

## 4 评测

为了评估图像描述生成系统的整体的有效性，现有的研究方法，包括 NeuralTalk、NIC、Show, Attend and Tell 和 Adaptive Attention 都被囊括进比较中。我们还比较了基于 RNN 的属性预测的效率。在这篇文章中我们第一个提出了这个评价指标。

### 4.1 评测指标

在本文中，BLEU、ROUGE 和 METEOR 被用于评价图片描述生成器。所有的这些方法都是基于衡量机器生成的描述和真值之间的相似程度。在下面的子章节中我们依次介绍这些方法。

#### 4.1.1 BLEU

BLEU 分数被广泛应用于评测机器翻译算法。它可以被用于衡量句子之间的语义相似程度。在本文中，他被用来计算机器生成的描述与多个高质量的人工生成的句子之间的相似程度。此外，在文献中，人类的表现被认为能在 BLEU 分数指标中达到 0.69。

#### 4.1.2 ROUGE

ROUGE-L 是所有可用的 ROUGE 评测方法的一个子集。它将两个序列作为输入，寻找两个序列的最长公共子序列（Longest Common Subsequence）。这里的最长公共子序列指在两个序列中都出现的子序列中长度最长的一个。在句子层面的 ROUGE-L 指标，从直觉而言，生成语句和真值语句的最长公共子序列越长，两个句子就越相似。

#### 4.1.3 METEOR

相比其他评测方法，得益于与人类受试者的注释具有高度相关性，METEOR 方法具有广泛的应用。他不仅计算了生成语句和真值语句之间的匹配距离，还尽可能发现关键词句的匹配。

## 4.2 评测结果

为了全面地评测所提出的系统，我们进行了全面的评测研究。首先我们用它们各自的测试集（如 ImageNet 和 PubFig 数据集）评估了基于 RNN 的属性分类系统。然后我们利用 IAPR TC-12 数据集的规模和复杂性对图像描述和描述生成器进行了实质性的评估。我们还用 NYUv2 语句数据集进行了一个小规模实验来测试处理超出范围的图像的有事和效率。NeuralTalk, NIC, Show, Attend 和 Tell 和 Adaptive Attention 等目前最先进的方法也用于图像描述生成的性能比较。

### 4.2.1 属性预测的评估

为了将基于 RNN 的属性分类器的输出与 PubFig 和 ImageNet 数据集的标注结果进行比较，我们使用了 BLEU 分数，因为它能将多个属性组合为一个语句。

Good	OK	Not good
 Round shiny wet yellow	 Long metallic rough	 red
 Furry spotted	 Square	 Brown rough

Fig. 3. Object attribute classification results using object attribute RNN.

图 3

PugFig 数据集和 ImageNet 数据集的子集被分别用于人和物体属性的分类。为了衡量物体属性分类，ImageNet 子集被分为约 7000 张图片的训练集、1500 张图的测试集和 1500 张图的验证集。在 1500 张测试集图片上，我们提出的方法达到了 0.62 的 BLEU-1 分数。在图 3 中展示了一些预测的结果。我们还将 PubFig 数据集分为了 6k、1k、1k 的三个子集分别用于训练、验证和测试，用于评测人物属性。基于 1000 张测试图片，我们提出的 RNN 结构达到了 0.61 的 BLEU-1 分数。显而易见，物体属性预测和人物属性预测的分数都和人类能在 BLEU 测试指标上达到的 0.69 的分数非常接近。图 4 展示了一些人物属性分类的例子。

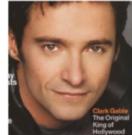
Good	OK	Not good
		
Female White Youth Wavy Hair Wearing Lipstick	Female White Brown Hair Wearing Necklace	Female White Brown Hair Wearing Lipstick
		
Male White Brown Hair Sunglasses Curly Hair	Male White Curly Hair	Female White Youth Wearing Lipstick

Fig. 4. Person attribute classification results using human attribute RNN.

图 4

#### 4.2.2 图片描述生成评估

**基于 IAPR TC-12 数据集的评测** 图像描述生成的评估使用多种度量方式，即 BLEU，ROUGE 和 METEOR。用于评估的主要数据集是 IAPR TC-12，由 20,000 张图像组成。鉴于我们生成的描述的特性，该数据集用于对现有数据集进行评估，包括 MSCOCO 或 Flickr8k/30k。上述的流行的数据集如 MSCOCO 和 Flickr8k/30k 通常由多个短小的描述组成。然而，所提出的系统倾向于产生比这些数据库提供的描述的长度长两倍以上的描述，这使得与这些数据集的比较不太具有相关性。此外，来自 IAPR TC-12 的图像更大，包含很多的物体和人的动作，以及比其他数据集（如 PASCAL，Flickr 和 MSCOCO）中的更长和更多的描述性注释。因此 TC-12 数据集被选中进行评估。

此外，对于训练所需要的数据，我们的研究所提出的方法也具有优势。我们所提出的系统已经在大型语料库上进行了训练，然而训练所需要的图片数远少于现有的其他方法，如 NIC。如前文所述，我们也用 NeuralTalk、NIC 和基于注意机制的方法（Show Attended and Tell 和 Adaptive Attention）进行了实验并用于性能比较。所有的方法都用它们最好的与星恋模型来测试。此外，NIC 是在 Flickr 数据集上训练，而其余三种对比方法是在 MSCOCO 数据集上训练。然而，我们提出的系统是在多个看上去毫无关联的数据集上训练的，例如用于物体检测的 ILSVRC13 数据集、用于物体属性和任务属性预测的 ImageNet 和 PugFig 数据集。为了确保比较的公平性，一个有 2400 张图的不出现在训练集中的 IAPR TC-12 数据集的子集用于评测所有方法。

所有模型生成的句子都与数据集中参考语句进行比较，然后通过 MSCOCO

评价脚本进行评测。该评价脚本用到了上述所有的评价指标，并能对比提出的方法和四个参考方法。IAPR TC-12 数据集中的每一张图片只对应一个描述，多样性的缺乏使得图片描述生成任务对于上述所有方法都更具有挑战性。

**Table 4**

Evaluation results for the IAPR TC-12 dataset using the MSCOCO evaluation script.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
This research	<b>0.231</b>	<b>0.099</b>	<b>0.046</b>	0.024	0.067	0.183
NeuralTalk [8]	0.133	0.072	0.041	<b>0.025</b>	0.067	<b>0.222</b>
NIC [9]	0.098	0.048	0.025	0.015	0.060	0.209
Show, Attend and Tell [13]	0.117	0.067	0.037	0.021	<b>0.077</b>	0.204
Adaptive Attention [14]	0.052	0.027	0.014	0.008	0.060	0.196

表 4

在 IAPR TC-12 数据集上的结果如表 4 所示。由结果可知，与 NeuralTalk、NIC、Show Attend and Tell 和 Adaptive Attention 对比，我们的系统能实现优秀的性能并在几乎所有评价指标上超过其余所有对比方法。具体而言，在于基于注意力机制的方法（如 Show Attend and Tell 和 Adaptive Attention）进行对比时，我们的工作在 BLEU 指标中显著超过其他两种方法，并在其余指标中仍具有竞争力。此外，值得指出的是，所有基准方法，尤其是 NeuralTalk 和 NIC 都是基于典型的 CNN+RNN 结构，作为生成描述的基础结构。然而我们提出的系统使用了一个基于局部的方法，不仅提出了局域的物体检测和识别，而且结合了给后续基于编解码器的描述生成器的属性预测。实验结果证明，我们系统生成的语句更具有描述性。与之对比的是，所有基准方法生成的描述更短。整体而言，实验结果和人类直觉显示我们系统生成的描述更具有描述能力。表格 5 展示了一些在 IAPR TC-12 数据集上我们的系统和四个基准系统输出的一些样本。

参考表格 4，值得指出的是，不论是相对更高的 BLEU 分数和可比较的 METEOR 分数，我们系统的 ROUGE-L 分数略低于其他系统。这可能因为我们生成的描述具有一些特定的特征。如前文所述，ROUGE-L 分数寻求计算机生成序列和真实序列之间的最长公共子序列，虽然属性预测丰富了描述的多样性，但描述偶尔会优于物体的标签添加新的属性。这可能因为截断公共子序列对 ROUGE 评价指标带来负面影响。在未来的工作中，我们的目标是收集类似 Flickr 或 MSCOCO 的数据集，并加入更多细节描述的语句，来进一步评价我们的方法和其他相似方法。

**Table 5**

Example system outputs as compared with reference descriptions and the outputs of all baseline methods for the evaluation of the IAPR TC-12 dataset.

IAPR TC-12		
Our system	View from above of a valley.	
Reference annotation	A river in a brown valley with many terraces; a yellowish-brown bush in the foreground	A sea cliff with a brown, red and brown cliffs and a wet seal behind it Gray and brown rocks in the foreground, a light brown sandy beach at the sea and wooded hills behind it, a blue sky in the background
NeuralTalk NIC Adaptive Att. Show, Attend and Tell	A dog is running through the woods A herd of elephants walking across a lush green field A group of people standing on the top of a lush green field A woman is walking down a rocky hill	A person on a beach with a surfboard A man is standing on a rock overlooking a lake A rocky beach with rocks on it A man is standing on a rock in the desert
IAPR TC-12		
Our system	Three people walking across a rope bridge	
Reference annotation	A woman is walking on a narrow rope bridge in the middle of a forest with many green trees and bushes	A hotel room with brown walls, a rectangular bed and a wooden sofa with a black, rectangular lamp behind it A single bed made of wood with a red pillow and a striped blanket, two small bedstands made of wood with a bedside lamp each, a light blue carpet and a wooden wall in the background
NeuralTalk NIC Adaptive Att. Show, Attend and Tell	A man is climbing a rock A man is standing in the woods holding a frisbee A brown and white dog standing next to a fence A man in a white shirt is standing in a wooded area	A dog is laying on the floor with a blue ball in its mouth A bed with a white comforter and a white blanket A bedroom with a bed and a dresser A woman in a pink striped shirt is sleeping on a bed
IAPR TC-12		
Our system	A construction site with a thunderous person at a round, brown wooden bicycle on a long wooden cart	
Reference annotation	A man with a blue cap and a greenish brown overalls is standing with a red machine in front of a gray wall, a black iron door on the right	A kitchen with red walls and brown vegetation on the white table A tray on bar with cut melons, oranges, mangoes and a coconut in form of a mouse head, a man behind it, three brown round doors and a white wall in the background
NeuralTalk NIC Adaptive Att. Show, Attend and Tell	A man sitting on a bench with a dog A man in a red shirt is sitting on a bench in a park A man standing next to a bicycle on a sidewalk A man in a blue shirt and blue hat is riding a bicycle	A woman is sitting on a couch with a man in a black shirt A table with a vase of flowers and a vase A woman is holding a stuffed animal in her hands A woman in a white shirt is sitting on a table with flowers and flowers

表 5

**基于 NYUv2 数据集的额外实验** 一个小规模的实验用到了跨模态的从场景描述任务中提取的图片。该实验的目的是验证我们的系统能多好地对超出范围的图片解决场景描述生成问题。这个 NYUv2 数据集完全由室内场景描述构成，并包含了物体属性和多个物体关系的标注。数据集中每个图片都对应一条描述。每条描述由三个句子组成。因此，该数据集中的描述可以很短也可以很长。

**Table 6**

Example system outputs as compared with reference descriptions and the outputs of all baseline methods for the evaluation of the NYUv2 sentence dataset.

NYUv2 sentence		
Our system	A hotel room with a long rectangular draped sofa and a very long, rectangular bookshelf awaits	An office setting with long rectangular red chairs next to a wooden table
Reference annotation	This is a living room with wooden floor. There is a big beige sofa on the left of the room with two pillows on top. Near the sofa is a black armchair. There is a book cabinet behind the sofa and the room is separated by a door.	This is a conference room with a long wooden table with red chairs around it, a projector mounted to the ceiling, and red window blinds. A multiline phone sits on the table.
NeuralTalk NIC Adaptive Att. Show, Attend and Tell	A man and a woman are sitting on a bench A living room with a couch and a TV A living room filled with furniture and a bookshelf A woman sitting on a couch in a library	A man is sitting on a bench in a room A living room with a couch and a TV A room with two windows and a window Two men are sitting on a couch

表 6

在这个小规模试验中，我们基于 MSCOCO 提供的描述数据训练了一个基于编解码器的语句描述生成组件。这个心生成的模型和其他所有模型都用 1400 张 NYUv2 数据集的图片进行测试。除了深度信息，实验显示对于提出的方法和对比方法，这是一个具有挑战性的任务。尽管在这个数据集上实现的整体评价分数比 IAPR TC-12 数据集更低，实验仍显示了在 BLEU-1 指标上，我们的模型比其余方法（NeuralTalk, NIC 和 Adaptive Attention）的性能分别高出 30%，70% 和 71%。在相同的指标下性能和 Show Attend and Tell 性能相近。一些我们的系统和相关方法生成的样本如表 6 所示。

#### 4.2.3 真实环境部署

为了确定我们提出的系统在现实生活或野外情况下的表现如何，我们使用人形 NAO 机器人的视觉系统进行了一些初始实验。由于所提出的系统在计算上是彻底的过程，我们不是将其部署到机器人平台，而是使用 GPU 服务器。机器人作为客户端通过无线网络与我们的 GPU 服务器进行通信。机器人的视觉 API 能够捕捉真实图像，然后将图像传输到远程 GPU 服务器并在云端进行计算。然后将生成的输出传回给机器人，使其能够描述环境。我们还使用真实场景对机器人进行了初步测试。结果表明，机器人可以识别许多物体，准确地描述环境布局和人物，但由于处理时间较长，所以在更复杂的场景中性能有限。在未来的工作中，我们致力于使机器人能够更快更准确地对场景作出反应，这可以用于医疗目的，例如，向心理疾病的护理提供者发出警报，或者这些患者需要帮助。

## 5 结论

在这项研究中，我们提出了一个基于深度学习的图像局部描述生成网络结构，用来描述和注释给定图像中的多个物体和人。它由物体检测和识别，场景分类，基于 RNN 的属性预测以及用于生成语句的 RNN 编解码器组成。我们提出的系统通过将给定图像中的人和物体的图像区域相关联的方法来减轻与整体问题的关联性，并可以获得更详细和更长的描述。实验结果表明，所提出的基于 RNN 的物体和人类属性预测器的性能非常优异。此外，整个系统中图像描述生成器也显示出其重要性。与 NeuralTalk, NIC, Show Attend and Tell 和 Adaptive Attention 等几种基准方法相比，IAPR TC-12 数据集评估了我们提出的系统可以输出更详尽的描述，在几乎所有的评估指标都优于这些最先进的方法。实验结果还表明，我们提出的系统比现有方法在处理 NYUv2 语句数据集的图像时，对超出边界的室内场景图像的描述生成具有优越性。

对于未来的工作，可以考虑更详细的属性，如相关文献中提到的与服装有关的属性，以进一步提高所提出的系统的描述能力。我们还致力于探索显著性检测，以图像的潜在焦点为重点，进一步提高系统的输出。从更长远的角度来看，我们的目标是将迁移学习和现有系统相结合，以处理卡通和油画等图像的图像描述生成。

### 书面翻译对应的原文索引

- [1] Kinghorn, Philip, Li Zhang, and Ling Shao. "A region-based image caption generator with refined descriptions." Neurocomputing 272 (2018): 416-424.

## 综合论文训练记录表

学生姓名	蒋梦青			学号	2014013443	班级	软件 42
论文题目	基于深度学习的图片自动生成网页前端代码方法						
主要内容以及进度安排	<p>主要内容:</p> <ol style="list-style-type: none"> <li>1. 本文首次提出从手写画生成对应网页 HTML 代码的任务，并提出了一个端到端的生成模型来学习网页的元素、排版和样式，这个生成模型基于深度学习框架，结合了卷积神经网络和循环神经网络，并使用了词嵌入。</li> <li>2. 建立了两个数据库：程序自动随机生成的大规模网页截图数据集和收集的手写网页草图。并且本文研究了模型在这两个数据集上的迁移学习问题。</li> <li>3. 本文用训练得到的模型和网页前后端搭建了"Doodle2Code" 软件，并在此基础上进行了用户研究。</li> </ol> <p>进度安排:</p> <p>4月：程序生成&amp;整理标准数据集；训练 baseline 和实验新的网络结构</p> <p>5月：搭建收集手绘数据 / 展示 demo 的网页平台；在手绘数据上进行实验</p> <p>6月：完善实验；撰写论文</p>						
	<p style="text-align: right;">指导教师签字: <u>李成平</u></p> <p style="text-align: right;">考核组组长签字: <u>刘洋</u></p> <p style="text-align: right;">2018年 3 月 22 日</p>						
中期考核意见	<p>蒋梦青同学的论文进行了方案调研和总体设计，完成了预定的阶段性工作，后续工作计划可行，答辩小组同意通过中期考核。</p>						
	<p style="text-align: right;">考核组组长签字: <u>刘洋</u></p> <p style="text-align: right;">2018 年 4 月 27 日</p>						

指导教师评语	<p>论文针对以手绘图自动生成对应的网页前端代码的方法进行了深入研究，提出了一种基于深度学习的视觉特征与序列生成相结合的生成式模型，并设计了网页前端代码自动生成算法。通过随机生成网页截图数据集以及手绘网页数据集，验证了算法的迁移学习能力和效果，实现了用户体验良好的软件系统，并给出了方法的评价。综合论文训练反映了蒋梦青同学已经具备扎实的计算机专业基础知识以及运用专业知识解决实际问题的能力。</p> <p>指导教师签字: <u>李春华</u></p> <p>2018年6月14日</p>
评阅教师评语	<p>论文针对深度学习和图像描述技术进行了深入研究，提出了一种基于深度神经网络模型通过手绘网页设计图自动生成对应网页前端代码的算法，创建了网页截图数据集和手绘网页数据集，对实现算法进行了验证与评价，开发了网页前端代码自动生成软件。同意蒋梦青同学本科毕业，取得学士学位。</p> <p>评阅教师签字: <u>宋祁旭</u></p> <p>2018年6月14日</p>
答辩小组评语	<p>蒋梦青同学的论文按计划完成了毕业设计工作，论文书写规范，达到本科生综合论文训练的要求。答辩过程思路清楚，回答问题正确，答辩小组一致同意通过论文答辩。</p> <p>答辩小组组长签字: <u>刘强</u></p> <p>2018年6月15日</p>

总成绩: 92  
教学负责人签字: 张慧

2018年6月19日