

文件编号: MALLOC(0)-SWC2018-20180081

受控状态: ☒ 受控 ☐ 非受控

保密级别: ☐ 公司级 ☐ 部门级 ☒ 项目级 ☐ 普通级

采纳标准: CMMI DEV V1.2



# 速写思维导图

SketchMind

## 项目开发文档

Version 1.0

2018/11/19

Written by malloc(0)

All Rights Reserved

# 目录

<b>1</b>	<b>引言 .....</b>	<b>1</b>
1.1	编写目的 .....	1
1.2	项目概述 .....	1
1.3	项目背景 .....	1
1.4	术语和缩略语 .....	1
1.5	参考资料 .....	1
<b>2</b>	<b>问题聚焦 .....</b>	<b>2</b>
2.1	问题描述 .....	2
2.2	问题抽象 .....	2
2.2.1	输入 .....	2
2.2.2	纠错 .....	2
2.2.3	输出 .....	2
2.3	问题定位 .....	3
2.4	问题评估 .....	3
2.5	问题分解 .....	3
2.5.1	优化 .....	3
2.5.2	分割 .....	3
2.5.3	分类 .....	3
2.5.4	手写字符 OCR .....	4
<b>3</b>	<b>相关工作 .....</b>	<b>4</b>
3.1	MINST DATABASE .....	4
3.2	Microsoft Azure 手写体认知服务 .....	4
3.3	传统光学字符识别 (OCR) .....	4
<b>4</b>	<b>技术方案 .....</b>	<b>5</b>
4.1	技术方向 .....	5
4.2	模型选择 .....	5
4.2.1	模型设计 .....	5
4.2.2	模型结构 .....	5
4.2.3	数据集 .....	6
4.3	结果期望 .....	6
4.3.1	卷积神经网络 (CNN) 的 Gadget 部件分类 .....	6
4.3.2	循环神经网络 (RNN) 手写体识别 .....	7

## 记录更改历史

[illegible]

# 1 引言

## 1.1 编写目的

为了确保本小组所进行的创新和工作有意义,不存在技术上不可解决的问题,同时为了使得在开发的过程中较少遇到技术方面的阻碍性问题,在正式开发之前我们对于涉及到技术栈方面的问题进行了较为完整的研究,确保了项目的可行性、可实施性之后,才继续进行后续步骤。

## 1.2 项目概述

SketchMind 可以通过一张手绘的思维导图生成精美的思维导图电子文稿格式。此项目意在以较高的容错率将小组讨论中信笔划下的思维草稿转化成精美易读且容易编辑的电子版思维导图,并以多种通用格式输出作为选择给予用户。

## 1.3 项目背景

掏出纸笔,圈圈点点写写画画是小组讨论中经常能遇见的场景。然而,最终得到的结果往往还需要手动打开 XMind、Visio 等思维导图软件再进行一次誊抄,费时费力。我们希望能够通过这份粗糙的手稿,最大精度地还原思维的流向,减少无谓的重复劳动。

## 1.4 术语和缩略语

[1] 思维导图: 又称脑图, Mind Map。一种图像式思维的工具以及一种利用图像式思考辅助工具来表达思维的工具。

[2] XMind: 一个由香港 XMind 公司开发的脑力激荡法和心智图的软件工具,其主要用途为帮助用户捕捉想法,组织各类报表。

[3] OpenCV: 一个基于 BSD 许可发行的跨平台计算机视觉库,轻量级而且高效,同时提供了 Python、Ruby、MATLAB 等语言的接口,实现了图像处理和计算机视觉方面的很多通用算法。

## 1.5 参考资料

[1] 质量管理体系国家标准理解与实施(2008 版)

[2] ISO 9001 质量体系——设计、开发、生产、安装和服务的质量保证模式

[3] 系统开发规范与文档编写.徐惠民.中央广播.2010.9

- [4] 心智图. 维基百科. <https://zh.wikipedia.org/wiki/心智图>
- [5] XMind. 维基百科. <https://zh.wikipedia.org/wiki/XMind>
- [6] OpenCV. <https://opencv.org>
- [7] THE MNIST DATABASE. 手写体识别数据库. <http://yann.lecun.com/exdb/mnist/>
- [8] Tensorflow 之 MNIST 解析. <https://www.cnblogs.com/lizheng114/p/7439556.html>
- [9] Microsoft Azure 机器学习云服务. 认知服务. 手写体识别.  
<https://azure.microsoft.com/zh-cn/services/cognitive-services/computer-vision/#handwriting>.
- [10] 汉王 OCR. <http://www.hw99.com/jssq/index.jhtml>
- [11] 卷积神经网络模型结构. <https://www.cnblogs.com/pinard/p/6483207.html>
- [12] CS231n Convolutional Neural Networks for Visual Recognition, Stanford.  
<http://cs231n.github.io/convolutional-networks/>

## 2 问题聚焦

### 2.1 问题描述

我们希望填平的是手绘思维导图和电子文稿思维导图之间的障碍。手绘的导图虽然易画且表意清楚，适合快速的记录闪念想法，但却不适合做成出版物进行分发和想法的交流。电子文稿思维导图具有精美简洁的优点，却又在制作和使用上有着较高门槛。我们希望能够通过此工具，让手绘思维导图到可出版的电子稿之间的过渡更为简单明白。

### 2.2 问题抽象

#### 2.2.1 输入

本问题的输入相当简单：用户只需要上传一张质量较好的拍摄或扫描得到的手绘思维导图图片即可。

#### 2.2.2 纠错

在完成后端的抽象识别之后，用户可以在网页提供的简易编辑器中纠正大部分机器所犯的错误，如：错连的线，错分的层级，错误识别的字符等。

#### 2.2.3 输出

我们预计将支持如下几种输出格式：位图图片（如流行的 JPEG、PNG 等格式）、矢量图片（如流行的 SVG 格式）和特有的思维导图文稿格式（如 FreeMind、XMind 等格式）。

## 2.3 问题定位

此问题的后端部分主要属于计算机图形学进行分块处理和机器学习用于手写体识别的组合问题。在对图像中元素进行分析时，用到了传统计算机图形学中的各类算法来完成分块分析；在手写体内容识别时则使用自设计的模型进行了手写体的识别训练。前端部分难度较低，主要需要实现抽象思维导图数据结构的显示和修改和多种文件格式的存储。预备会使用较为广泛使用的 D3 图像库结合自制的数据结构和脑图编辑器完成这一部分工作。

## 2.4 问题评估

由于传统计算机图形学已经相当成熟，发展出了许多成熟的图形算法且得到了多种优秀的实现。最著名的跨平台库就是我们所使用的基于 C++ 的 OpenCV。

## 2.5 问题分解

此问题本质上分为两个部分：其一是思维导图部件（Gadget）和关系（Relationship）的识别和分类；其二是部件内的手写体的识别。

第一个问题又分为几个主要部分：优化、分割与分类。

### 2.5.1 优化

首先对传入的图片进行降噪、锐化和二值化。降噪可以减少噪点产生的无效部件（小黑点），增加识别正确概率。锐化可以使得边缘更加清晰，减少错分部件的概率。二值化则是为了简化后续的图像处理步骤，方便后面做图像进一步算法处理。

### 2.5.2 分割

我们主要采用目前较为成熟的图像算法来对图片中的部件进行分割。选择的图像处理库为 OpenCV，其主要优势是开源且跨平台，轻量级且高效——适合这个单个数据规模不是很大的小项目。

首先利用 Erode（边缘腐蚀膨胀算法）将同一部件的笔画相连，这样没有边框的部件也可以被识别在一起。同时小心地调整膨胀系数，防止部件之间产生粘连。

随后，使用 Thresh 算法将每一个相连通的部分划分为一个部件，并进行初步处理。如：完全包含与被包含的两个部件，仅保留最外一个部件（认定为边框）；小于一定面积的部件认定为噪点进行去除。

分割完成之后的部件以 (x, y, width, height) 格式进行存储，并为下一步分类做准备。

### 2.5.3 分类

上一步我们已经利用传统的计算机图像识别办法进行了分割。这一步我们将进行部件的

分类。部件的种类主要分三种：文本框、箭头和花括号。在这里我们将使用 Keras 作为框架，并利用 Intel 官方提供的 AI Dev Cloud 进行这部分训练。

对于文本框，需要到下一步再将其进行手写体识别，本步骤仍然使用图片进行存储；箭头则需要能够识别出大致指向；花括号则需要识别出包被的范围。

#### 2.5.4 手写字符 OCR

最后一步仍然需要使用机器学习框架，完成最后一步：部件内容识别与填充。完成之后，就可以初步生成一张电子思维导图了。

### 3 相关工作

目前已有的手写体识别方案主要包括：

#### 3.1 MINST DATABASE

MINST DATABASE 需要自行设定训练数据集，对英文和数字的识别准确率较好，经简单测试可以达到 90%以上。但对于中文汉字的训练模型效果不是特别好，推测原因可能是复杂的笔画会加大识别难度，另一方面是上下文关系不像英文那样确定。

#### 3.2 Microsoft Azure 手写体认知服务

Microsoft Azure 是老牌云服务提供商，在近年 AI 和机器学习的大潮下也推出了在线的认知服务：其中就包括手写体的识别和图像情景的识别。其优势是具有大量优质的训练数据，且经过特殊算法得以适用于不同的图面和背景，如白纸、黄色便签和白板。然而由于这是一项商业服务，因此无法轻易使用自己的数据进行训练，API 设计相对固化，难以自行扩展。且其计算收费对于小项目而言较为高昂。

#### 3.3 传统光学字符识别（OCR）

金山 OCR、汉王 OCR 都是传统老牌的汉字 OCR 识别工具。他们之中有的才刚刚转向机器学习辅助中文 OCR 的道路，有的则已经结合了中文语义识别来减少错误识别字。但不可否认他们在复杂条件下的字符识别和类字的识别，以及针对汉字的识别优化都经过了十几年的积累，在这方面优于 3.1 和 3.2 两种尚处于 Preview 版本，还没有支持中文的解决方案。在我们进行开发的过程中，也需要时刻学习这门行业多年所积累的经验，尽可能提高自己的识别水平。

## 4 技术方案

### 4.1 技术方向

经过讨论，我们初步决定采用利用卷积神经网络（CNN）进行分割后部件的分类。原因是分划所产生的类型较为有限（预计仅仅包括四到五种）、变化不大且便于给出数据标签。另外，将会使用循环神经网络（RNN）来进行手写体文本的识别，由于手写体具有不分段、不定时序的特点，因此选择了此种神经网络。

### 4.2 模型选择

#### 4.2.1 模型设计

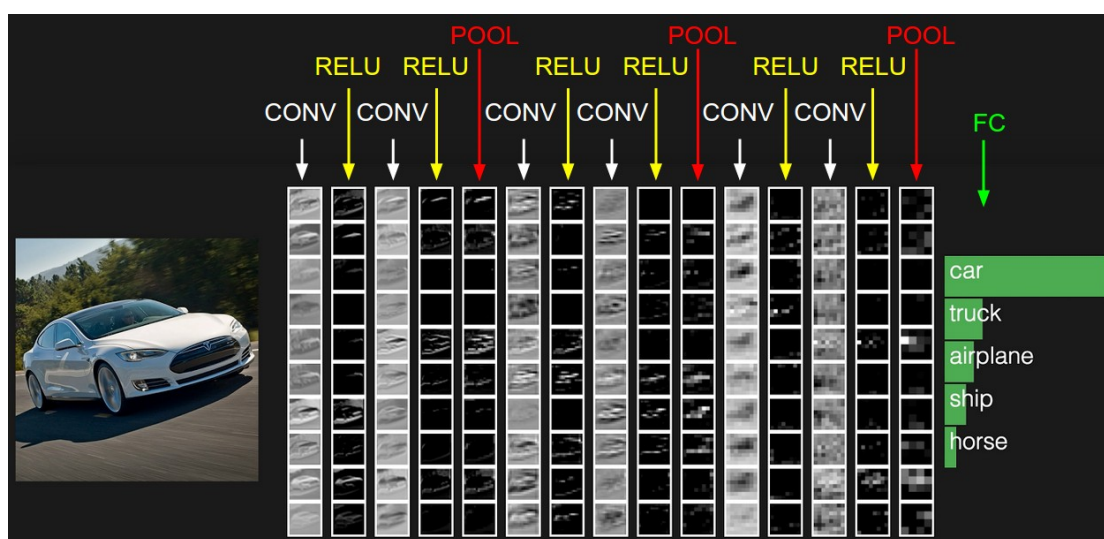
如 4.1 中所述，在此项目中我们将使用 CNN 和 RNN 两种神经网络结合来完成产品的开发。其具体结构见下。

#### 4.2.2 模型结构

对于 CNN 卷积神经网络而言，我们首先将输入的 Gadget 图片（其尺寸一般不大，以不超过  $100 \times 100$  为宜）转化为一个二维矩阵。使用 ReLu 激活函数做卷积。常用的 ReLu 激活函数采用

$$\text{ReLU}(x) = \max(0, x)$$

之后进行池化（Pooling）。池化的主要目的是缩小原矩阵的规模。在反复进行卷积+池化操作之后，进行全联接（Full Connection）进行输出操作。



（图源[10]）



### 4.2.3 数据集

首先，我们将从自身环境及网络之中获取大量手绘思维导图图片素材，并使用计算机图形学（SketchMind.ML.parsing 文件夹）内容对其进行分块。

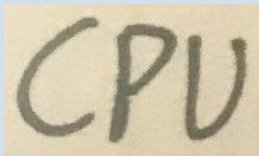
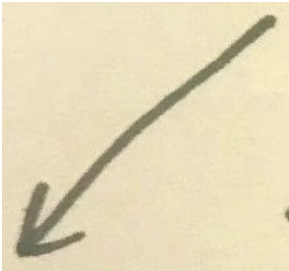

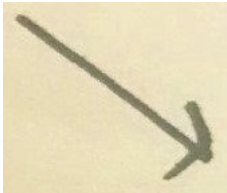
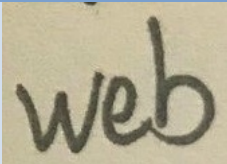
分块之后，我们将使用人工贴标签的方式训练分类部分的 CNN 网络。由于此部分的类型数目较少，因此人工标签方式较为可行且具有较高准确率。

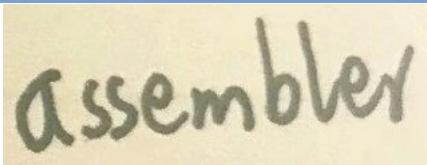
此步骤的测试集从第一次训练之中可信度较高的数据中选择，除去测试集之后利用剩下的训练集进行第二次训练得到最终训练结果，第二次训练集中最高可信度的部分作为验证集。

这样，所有获取的素材就分成了训练集、验证集和测试集了。

## 4.3 结果期望

### 4.3.1 卷积神经网络（CNN）的 Gadget 部件分类

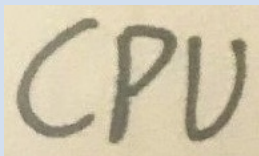
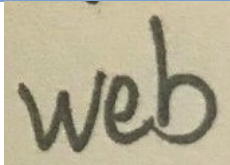
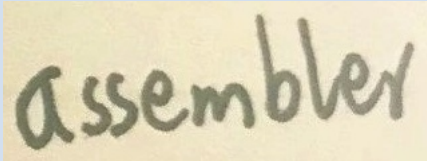
输入	输出
	"Word"
	"Arrow" "Angle": 226
	"Arrow" "Angle": 270
	"Arrow" "Angle": 330
	"Word"

	“Word”
---	--------

### 4.3.2 循环神经网络（RNN）手写体识别

初步我们将首先实现对西文字母和阿拉伯数字的识别，在时间充分的后期开发阶段再着力提高汉文字的识别率。

预期的训练结果如下：

输入	输出
	“CPU”
	“web”
	“assembler”