

实战三：Quick, Draw! Google涂鸦识别挑战项目(上)

导师：GAUSS

目录

1/ 背景简介

2/ 数据解读

3/ 思路分析

4/ 模型构建

知识树

Knowledge tree



深度之眼
deepshare.net

实战三：Quick, Draw! Google涂鸦识别挑战项目(上)

背景简介

数据解读

思路分析

模型构建

背景简介

"Quick, Draw!" 简介

"Quick, Draw!" 作为一种实验性游戏发行，以有趣的方式向公众宣传AI的工作原理。游戏提示用户绘制描绘特定类别的图像，例如“香蕉”，“桌子”等。游戏生成了超过1B的图纸，其中的一部分数据集被公开发布，**该数据集包含5000万个图纸，其中包含340个标签类别。**

听起来很有趣，对不对？这挑战就是：由于训练数据来自游戏本身，所以图纸可能不完整或与标签不匹配。您需要构建一个识别器，该识别器可以有效地从这些嘈杂的数据中学习，并在来自不同分布的手动标记的测试集上表现良好。

现在的任务是通过现有的数据集为Quick, Draw建立更好的分类器！这将对手写识别及其在OCR（光学字符识别），ASR（自动语音识别）和NLP（自然语言处理）等领域的强大应用产生直接影响。

"Quick, Draw!" 网站: <https://quickdraw.withgoogle.com/>



神经网络能学会辨识涂鸦吗？

只要将你的涂鸦作品添加到[全球最大的涂鸦数据集](#)，即可协助训练神经网络辨识涂鸦。这个数据集会公开共享，以便协助进行机器学习研究。

开始涂鸦吧！

评分公式

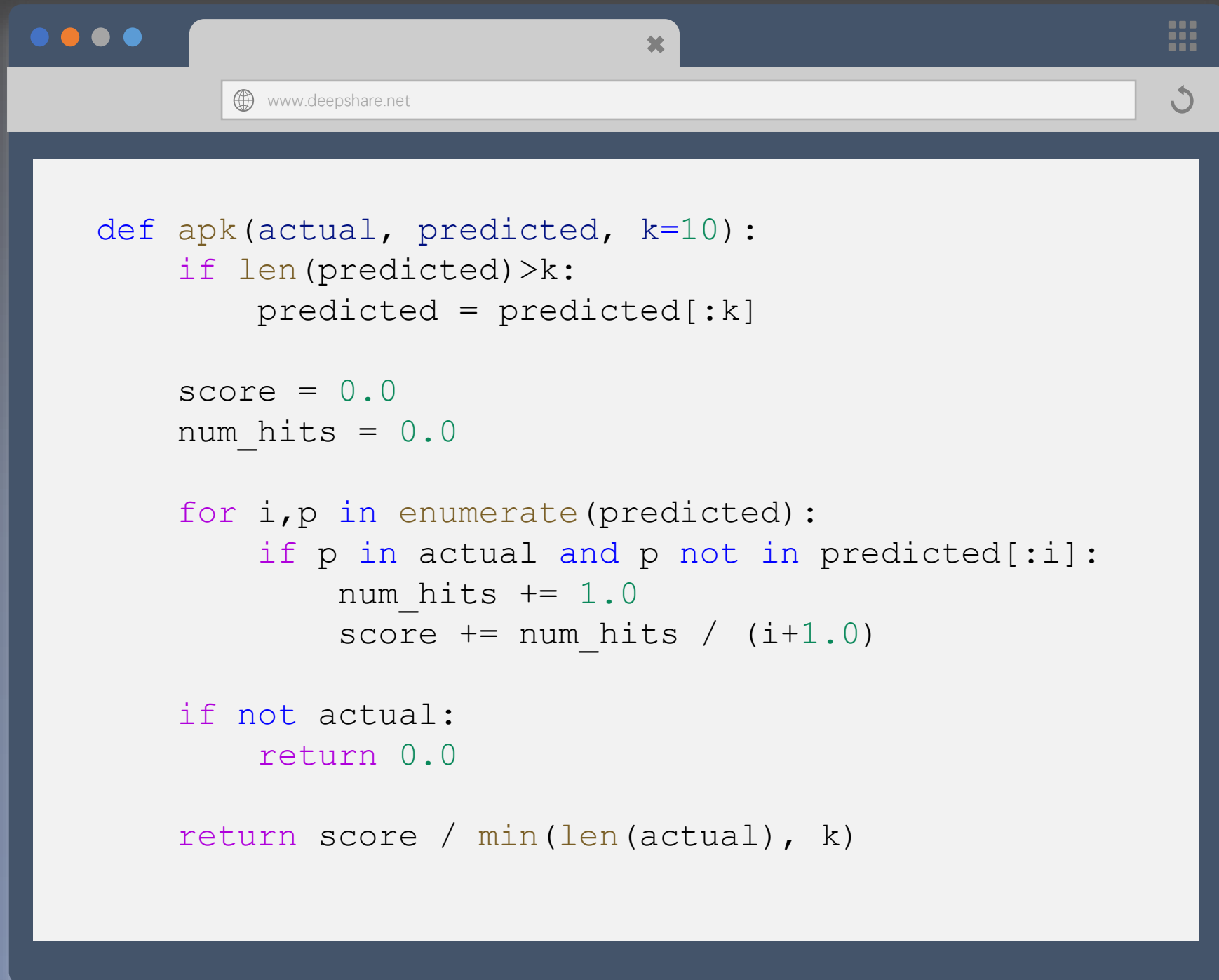
模型效果评估指标 Mean Average Precision@3 (MAP@3):

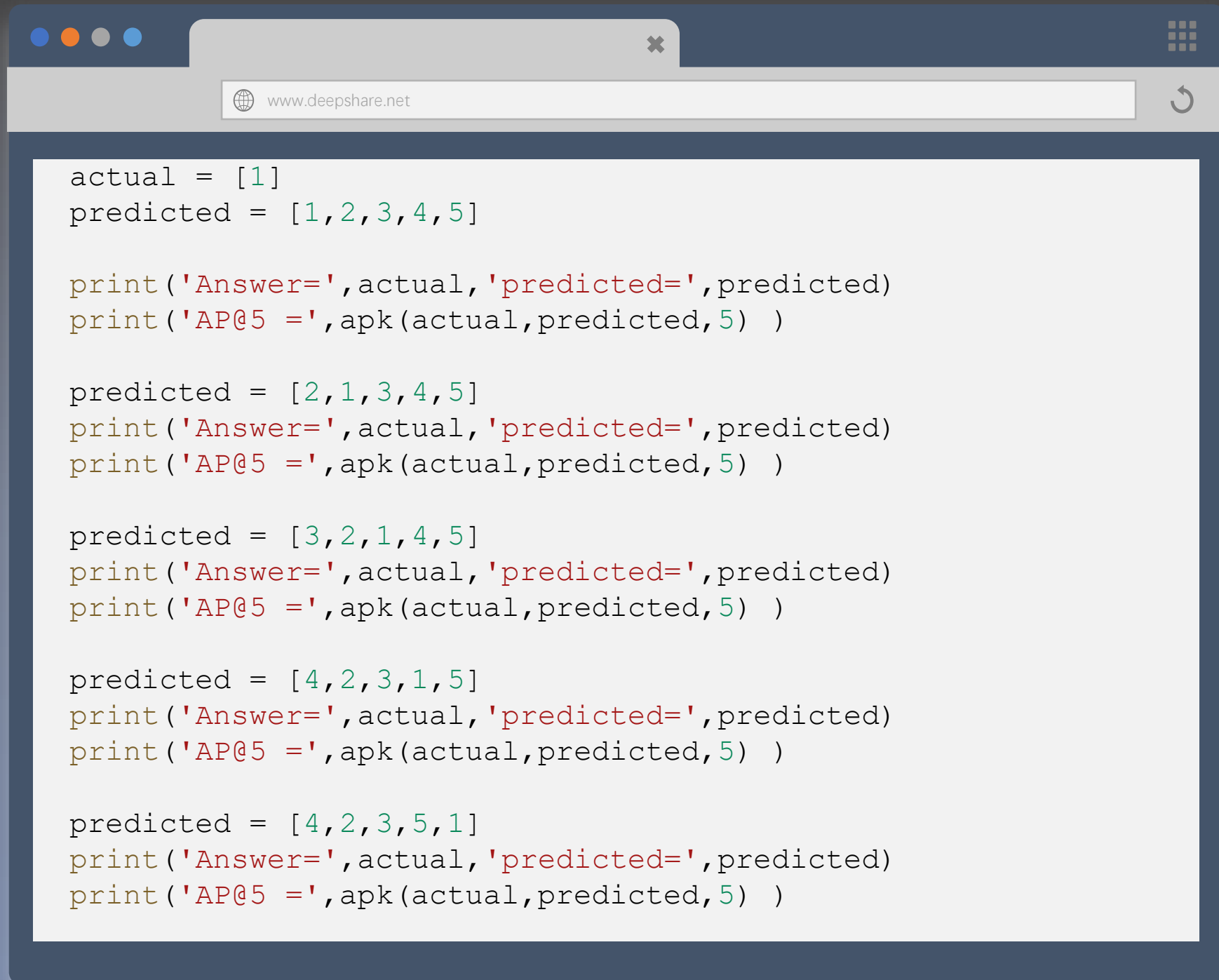
$$MAP@3 = \frac{1}{U} \sum_{u=1}^U \sum_{k=1}^{\min(n,3)} P(k)$$

其中U是测试数据中计分的图的数量，P(k) 是截止k时的精度，n是每个图的预测数量。
这里k=3。

<https://www.youtube.com/watch?v=pM6DJ0ZZee0>

https://github.com/benhamner/Metrics/blob/master/Python/ml_metrics/average_precision.py





```
actual = [1]
predicted = [1, 2, 3, 4, 5]

print('Answer=', actual, 'predicted=', predicted)
print('AP@5 =', apk(actual, predicted, 5) )

predicted = [2, 1, 3, 4, 5]
print('Answer=', actual, 'predicted=', predicted)
print('AP@5 =', apk(actual, predicted, 5) )

predicted = [3, 2, 1, 4, 5]
print('Answer=', actual, 'predicted=', predicted)
print('AP@5 =', apk(actual, predicted, 5) )

predicted = [4, 2, 3, 1, 5]
print('Answer=', actual, 'predicted=', predicted)
print('AP@5 =', apk(actual, predicted, 5) )

predicted = [4, 2, 3, 5, 1]
print('Answer=', actual, 'predicted=', predicted)
print('AP@5 =', apk(actual, predicted, 5) )
```


数据解读

数据集介绍

- sample_submission.csv - 格式正确的样本提交文件
- test_raw.csv - raw矢量格式的测试数据
- test_simplified.csv - simplified矢量格式的测试数据
- train_raw.zip - raw矢量格式的训练数据；每个label一个csv文件
- train_simplified.zip - simplified矢量格式的训练数据；每个label一个csv文件

数据集网址: <https://www.kaggle.com/c/quickdraw-doodle-recognition/data>

数据介绍



深度之眼
deepshare.net

Data Explorer

227.61 GB

- ▶ train_raw
- ▶ **train_simplified**
 - sample_submission.csv
 - test_raw.csv
 - test_simplified.csv

Summary

- ▶ 683 files
- ▶ 4088 columns

Download All

< train_simplified (340 files)



The Eiffel Tower.csv
45.74 MB



The Great Wall of China...
84.51 MB



The Mona Lisa.csv
81.25 MB



airplane.csv
62.87 MB



alarm clock.csv
76.09 MB



ambulance.csv
92.85 MB

数据介绍

simplified数据集示例:

```
files_directory = os.listdir("../input/train_simplified")
file = files_directory[0]
train_simplified = pd.read_csv('../input/train_simplified/' + file, nrows=10)
train_simplified.head()
```

	countrycode	drawing	key_id	recognized	timestamp	word
0	US	[[[11, 14, 35, 42], [8, 36, 121, 189]], [[0, 4...	5352250580926464	True	2017-01-27 22:19:37.987450	oven
1	TH	[[[41, 30], [0, 94]], [[40, 190, 237, 247, 254...	6701726566449152	True	2017-03-29 17:09:50.120520	oven
2	US	[[[180, 53, 29, 13, 5, 0, 4, 18, 145, 254, 255...	5713783643176960	True	2017-03-02 00:44:13.105340	oven
3	US	[[[128, 46, 6, 11, 0, 10, 128, 227, 228, 224, ...	4989314201026560	True	2017-01-24 18:02:13.390310	oven
4	GB	[[[3, 25, 27, 31, 81, 193, 193, 207, 212, 219,...	4956239173779456	True	2017-03-26 15:43:44.787210	oven

数据介绍

raw数据集示例:

```
files_directory = os.listdir("../input/train_raw/")
file = files_directory[0]
train_raw = pd.read_csv("../input/train_raw/" + file, nrows=10)
train_raw.head()
```

	countrycode	drawing	key_id	recognized	timestamp	word
0	US	[[[263, 263, 264, 267, 271, 276, 282, 288, 295...	5352250580926464	True	2017-01-27 22:19:37.987450	oven
1	TH	[[[544, 542, 542, 541, 540, 539, 538, 536, 534...	6701726566449152	True	2017-03-29 17:09:50.120520	oven
2	US	[[[866, 840, 835, 829, 821, 814, 807, 799, 791...	5713783643176960	True	2017-03-02 00:44:13.105340	oven
3	US	[[[600, 591, 586, 579, 567, 552, 535, 519, 504...	4989314201026560	True	2017-01-24 18:02:13.390310	oven
4	GB	[[[172, 172, 176, 181, 185, 187, 190, 193, 196...	4956239173779456	True	2017-03-26 15:43:44.787210	oven

数据含义

每列数据含义：

列名	数据类型	解释
word	string	提示玩家绘制类别
recognized	boolean	这个类别(word)是否被游戏识别。
timestamp	datetime	绘制时间
countrycode	string	所在位置的国家/地区代码
drawing	string	绘图的JSON数组

数据含义

每一行包含一个图形。这是单个绘图的示例：

```
{ "word": "nose",  
  "countrycode": "AE",  
  "timestamp": "2017-03-01 20:41:36.70725 UTC",  
  "recognized": true,  
  "drawing": [[[129, 128, 129, 129, 130, 130, 131, 132, 132, 133, 133, 133, 133, ...]]]  
}
```




数据含义

右图是一条**drawing**的数据：

其中x和y是像素坐标，t是自第一个点以来的时间（以毫秒为单位）。x和y是实数值，而t是整数。

由于用于显示和输入的设备不同，原始图形的边界框和点数可能会大大不同。

```
[  
  [ // First stroke  
    [x0, x1, x2, x3, ...],  
    [y0, y1, y2, y3, ...],  
    [t0, t1, t2, t3, ...]  
  ],  
  [ // Second stroke  
    [x0, x1, x2, x3, ...],  
    [y0, y1, y2, y3, ...],  
    [t0, t1, t2, t3, ...]  
  ],  
  ...  
  // Additional strokes ]
```



两者数据区别

给出了两个版本的数据。**raw数据是从所述用户绘画记录确切的输入，而simplified版本从矢量信息删除不必要的点。**（例如，一条直线可能记录了8点，但是由于只需要2点就可以唯一地标识一条线，因此可以删除6点。）这些simplified文件要小得多，并且可以有效地提供相同的信息。

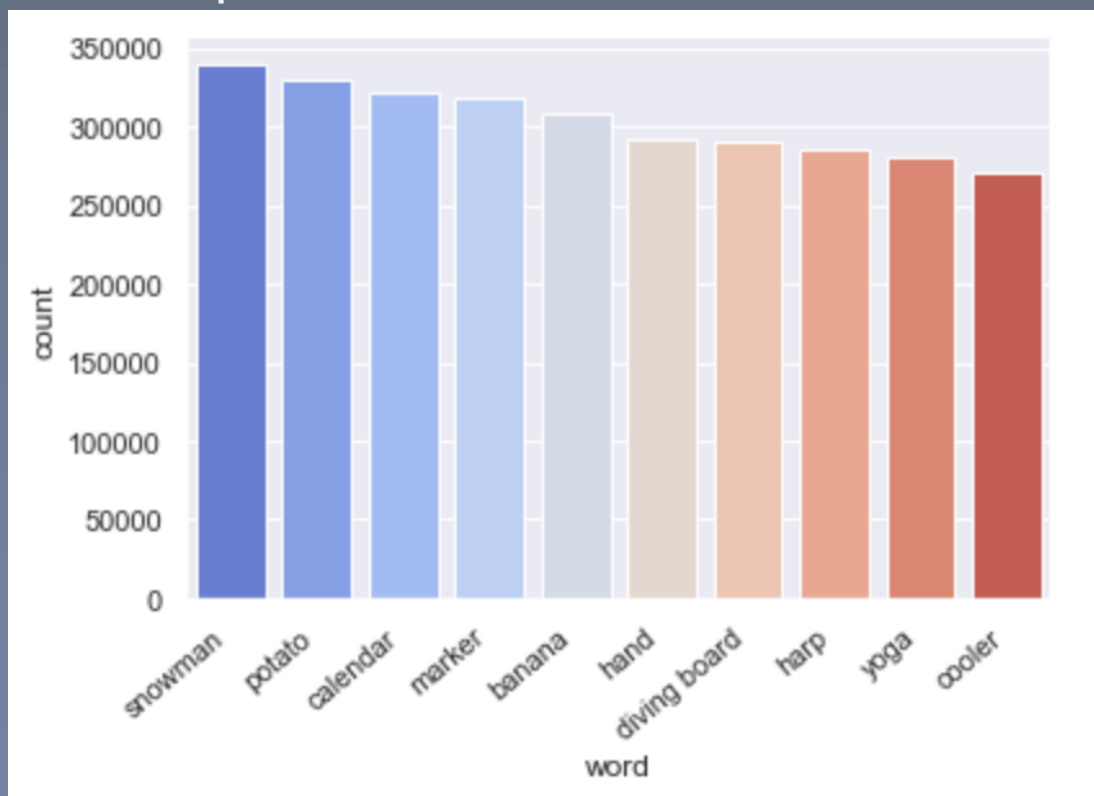


数据勘察

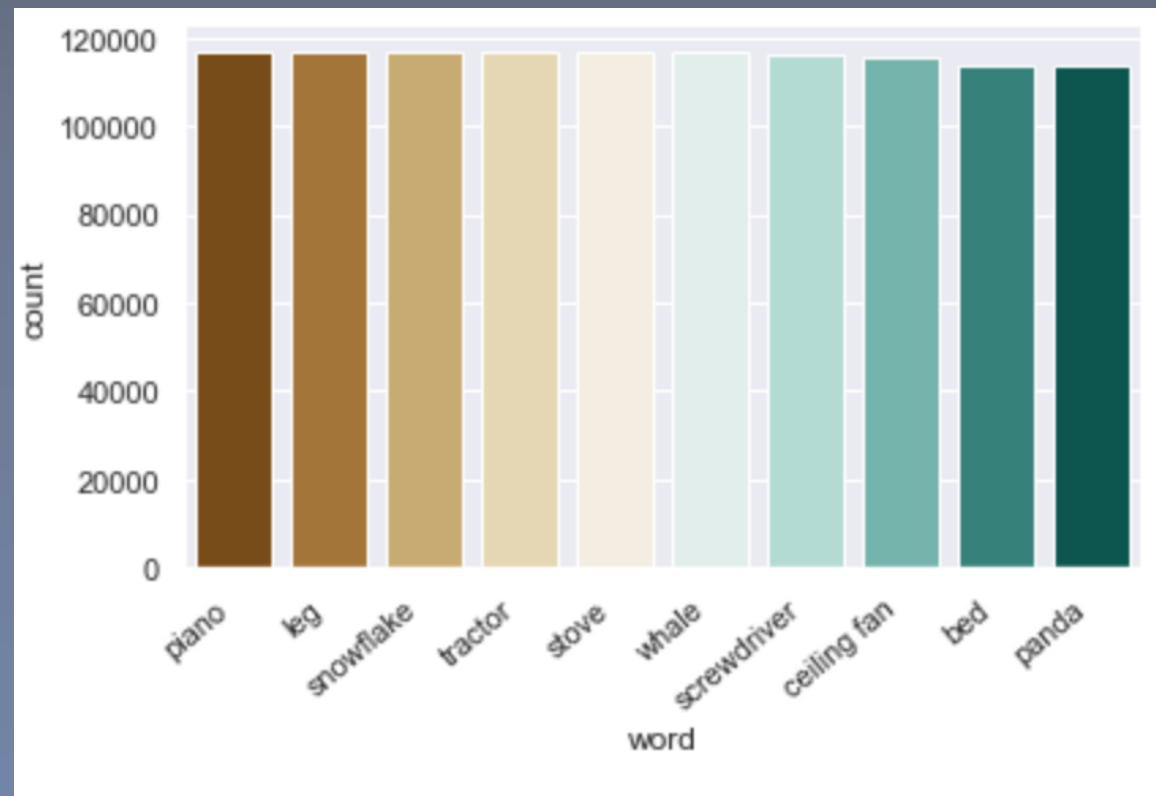
数据量：49707579条

类别数：340类

左图：top10类别占比



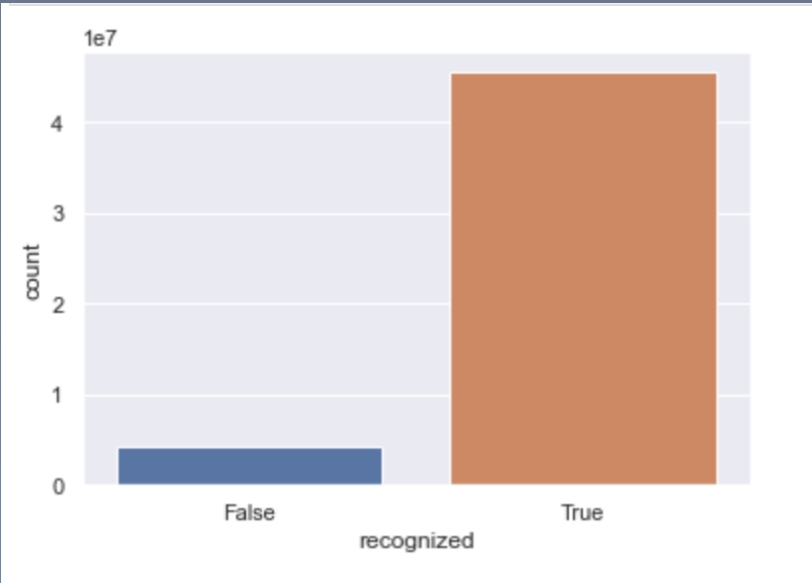
右图：bottom10类别占比



数据勘察

训练集

recognized=True : 45512752
recognized=False : 4194827
对应比例为92%：8%

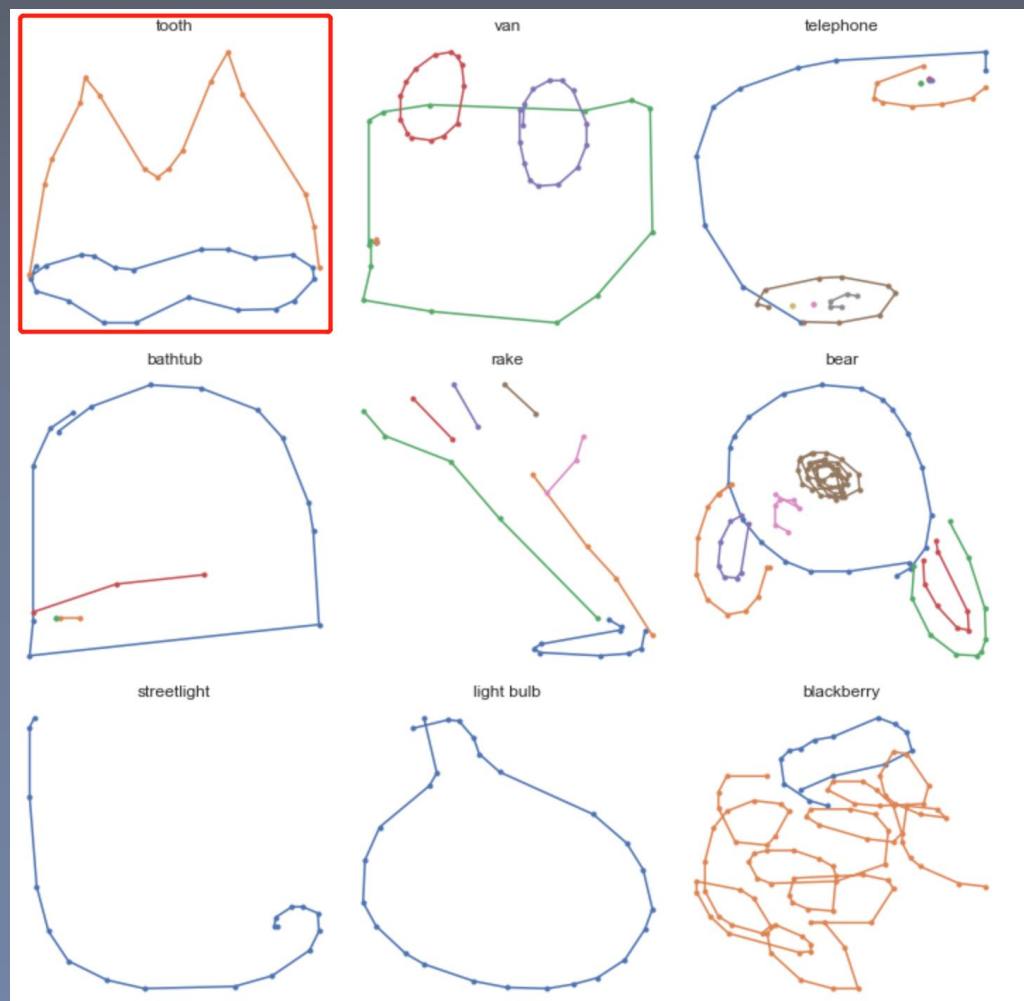
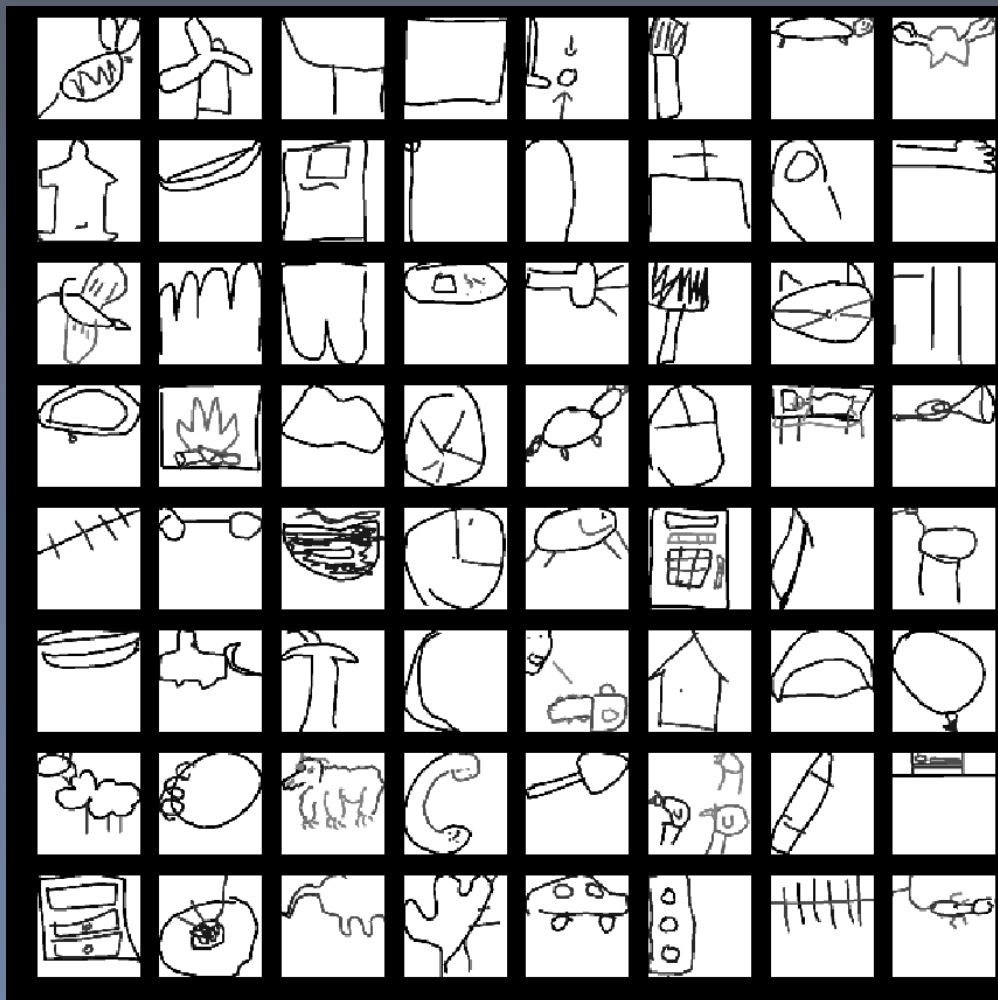


	recognized_true	count_true	recognized_false	count_false
word				
The Eiffel Tower	True	130433	False	4368
The Great Wall of China	True	171064	False	21951
The Mona Lisa	True	111580	False	9803
airplane	True	135823	False	15800
alarm clock	True	113911	False	9488
...
wine glass	True	124895	False	7407
wristwatch	True	140672	False	21973
yoga	True	258493	False	21949
zebra	True	127159	False	17449
zigzag	True	116635	False	3437

数据可视化



深度之眼
deepshare.net



思路分析

思路分析

drawing数据集转化：黑白图片 或者RGB图片

数据处理：未识别的数据如何处理？

数据增强：旋转、翻转等

模型建模：CNN模型、预训练模型、迁移学习

参数调整：图片大小、batch大小等

模型构建

模型构建

- 数据读取：tfrecord读取、csv文件读取、from_generator读取
- 数据处理：黑白图片、RGB图片
- 模型思路：简单的CNN模型、预训练模型、迁移学习
- 结果分析：模型效果分析

预训练建模

AlexNet

VGG 系列: VGG11、VGG13、VGG16、VGG19

MobileNet 系列: MobileNetV1、MobileNetV2

ShuffleNet 系列:

ResNet 系列: ResNet18、ResNet34、ResNet50、ResNet101、ResNet152

DenseNet 系列: DenseNet121、DenseNet161、DenseNet169

SENet 系列

Inception 系列

EfficientNet 系列

总结

本节小结

Summary

实战三： Quick, Draw! Google涂鸦 识别挑战项目	背景简介	
	数据解读	数据含义以及数据分布
	思路分析	如何快速构建baseline?
	模型构建	模型构建

结语

——我 说——

看过千万代码，不如实践一把！





深度之眼
deepshare.net

联系我们：

电话：18001992849

邮箱：service@deepshare.net

QQ：2677693114



公众号



客服微信

