

## PC 软件客户端制作技术栈：

### 一. 语音输入

#### 1.1 语音识别

- 使用Github开源项目ASRT代码，在服务器上部署
- 配置Nginx代理
- 客户端进行录音，并通过API访问服务器，服务器返回语音识别结果



#### 1.2 文本提取关键字

- 使用 Github Han Language Processing 项目，基于TensorFlow进行分词

```
hello.py  X
hello.py > ...
1 from hanlp_restful import HanLPClient
2 HanLP = HanLPClient('https://www.hanlp.com/api', auth=None, language='zh') # auth不填则匿名, zh中文, mul多语种
3
4 print(HanLP.parse("我的铅笔呢? "))
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

```
PS D:\Users\Dawn\Desktop\Python> d:; cd 'd:\Users\Dawn\Desktop\Python'; & 'D:/Edward/Tool/Coding/Python/Python39/python.exe'
cher' '51617' '--' 'd:\Users\Dawn\Desktop\Python\hello.py'
{
  "tok/fine": [
    ["我", "的", "铅", "笔", "呢", "? "],
    ],
  "tok/coarse": [
    ["我", "的", "铅", "笔", "呢", "? "],
    ],
  "pos/ctb": [
    ["PN", "DEG", "NN", "SP", "PU"],
    ],
  "pos/pku": [
    ["r", "u", "n", "y", "w"],
    ],
  "pos/863": [
    ["r", "u", "n", "u", "w"],
    ],
  "ner/msra": [
    []
    ],
  "ner/pku": [
    []
    ]
}
```

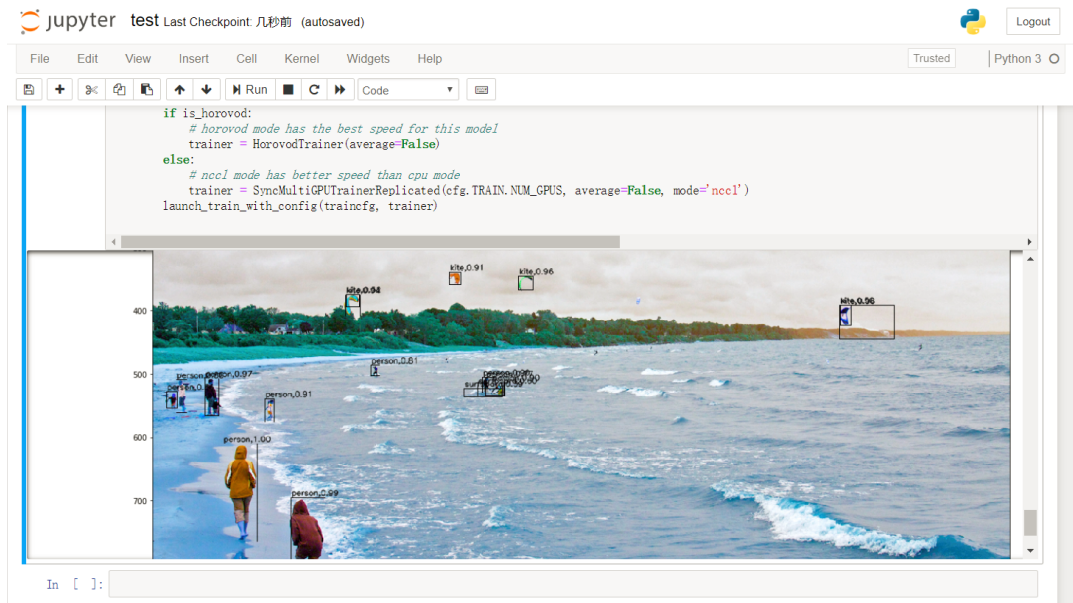
## 二. 服务器

## 二. 主体检测

- 使用Faster-RCNN 进行目标检测
  - 数据集
    - COCO 2017
  - 网络
    - ImageNet-R50 Backbone
  - 训练过程

```
python train.py --config MODE_MASK=False MODE_FPN=True
BACKBONE.WEIGHTS=/data/tensorpack/modelzoo/FasterRCNN/ImageNet-R50-
AlignPadding.npz DATA.BASEDIR=/data/tensorpack/coco/
```

- 样例图像检测结果



考虑到基于该数据集训练的模型可能存在不适用于当前项目的情况，我们并未选择直接使用 tensorflow 提供的现有 API，而是选择将源码、数据集等部署到服务器上，进行训练和部署，以便后续进行修改

### 三. 图像处理流程

- 假设：

- 用户很少在短时间内对物体进行迅速的移动
- 用户能够在较短时间内发现自己丢失了物品
- 用户同一时间不会在桌面上摆放过多的物品

- 处理算法

- 若未检测到语音输入，则更新当前物体位置信息，更新规则如下

- 用结构体记录主体信息，包括名称与位置。初始时，结构体数组为空。

用两个数组进行主体信息的记录

- 一组维护当前桌面上不断更新的主体信息，后简称数组1
- 一组维护一段时间未进行更新的主体信息，后简称数组2

- 对每个当前帧中的主体，得到上一帧中位置其最近的主体（包括数组1与数组2中的主体），计算两者的直线距离

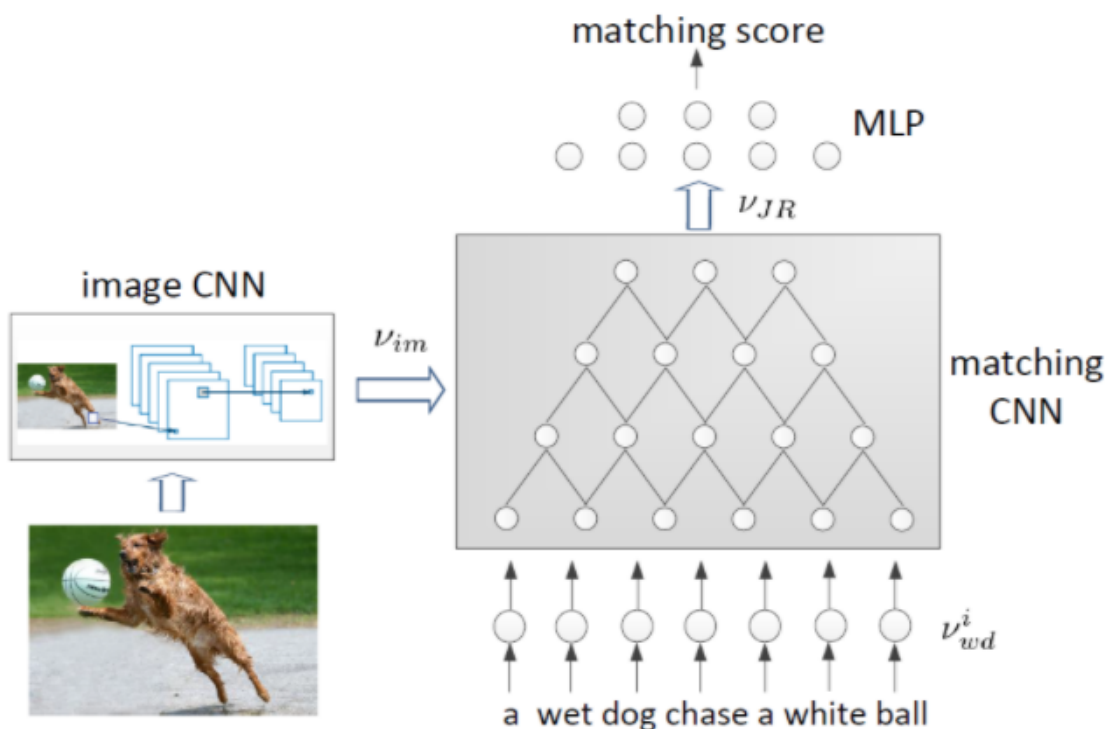
（1）若直线距离小于某一阈值且两者在样例图像检测结果的返回名称相同，则两者认定为同一主体，更新位置信息及其位置信息的最后更新时间，若物体原先在数组2中，将其转移到数组1中

（2）若直线距离小于某一阈值，但两者在样例图像检测结果中返回名称不同，则不进行匹配，按距离从小到大寻找下一项物体进行匹配，直到找到符合条件（1）的物体，并更新位置信息及其位置信息的最后更新时间，若物体原先在数组2中，将其转移到数组1中

（3）若在进行（2）后，在前一帧的结果中未找到能够与之匹配的物体，则将该主体信息加入数组1中，更新位置信息及其位置信息的最后更新时间

- 若数组1中主体节点信息超过5分钟未更新，说明可能存在被遮挡或不在桌面上的情况，将其转移到数组2中

- 若数组2中主体节点的信息超过  $7 * 24$  小时未更新，将其从数组2中删除
- 若检测到语音输入，对当前图像输入中的每个主体，先利用CNN网络进行如图所示的图像文本匹配
  - 当目标物体在桌面上并未被遮挡时，在所有主体中，有一个或多个物体的matching score 大于程序中所设定的阈值  
此时能够直接找到物体，将主题包围盒显示在屏幕上



- 当目标物体在桌面上，且被遮挡时，所有数组1主体中物体的matching score均低于所设定的阈值，查询位置关系图寻找目标为物体，查询方法如下：
  - 遍历数组2
    - 若数组2中存在与目标物体的matching score 高于设定阈值的物体，则将其最后更新的位置信息显示在屏幕上
    - 若数组2中不与目标物体的matching score 高于设定阈值的物体，提示用户未找到目标物体
- 可能存在的问题及朴素解决方案
  - 算法可能不足以支撑实时检测
    - 牺牲部分准确性，减少输入的频率
  - COCO2017数据集可能无法识别部分物体或识别错误
    - 选用合适的数据集，重新训练 Faster-RCNN

#### 参考链接

- 语音转文本
  - <https://blog.ailemon.me/2020/08/27/teach-you-how-use-asrt-deploy-chinese-asr-api-server/>
  - <https://github.com/nobody132/masr>
- 文本提取关键字

- <https://github.com/hankcs/HanLP/tree/doc-zh>
- 自然语言处理
  - <https://github.com/hankcs/HanLP/tree/doc-zh>
  - <https://github.com/fighting41love/funNLP>
  - <https://github.com/letiantian/TextRank4ZH>
  - [https://github.com/dongrixinyu/chinese\\_keyphrase\\_extractor](https://github.com/dongrixinyu/chinese_keyphrase_extractor)
  - <https://github.com/jbesomi/textthero>
- 服务器
  - <https://www.qingcloud.com/>
- 遮挡检测
  - <https://arxiv.org/pdf/1807.08407.pdf>
  - <https://arxiv.org/pdf/1711.07752.pdf>