慧眼识物 中期报告

JUN HANAIZUMI, EDWARD RAYMOND, NAOMI MICHELLA GUNAWAN, 谢安琪

# 主要目标

## 1.1 开题背景

由于桌面杂乱、无收纳整理习惯等原因，在日常生活中经常存在一时无法从桌面上的众多物品中找到目标物品的情况。本程序希望能够在该情况发生时，通过AR技术在桌面上标记物品的可能位置;若未能找到，则对物品所处的位置进行预测，以帮助人们快速寻找目标物品，节省时间。

## 1.2 用户分析

考虑到几乎全年龄段的人群均可能遇到难以在桌面上找出物品的情况，本产品希望能给全年龄段人群提供桌面寻物的快速寻找方案。但由于在对项目进行测试时，小组较难获取到青年以外人群的大量桌面数据，因此现阶段主要针对青年人群进行产品设计和测试。

## 1.3 目标功能

本项目为主机平台，完成一款慧眼找物程序。

### 文字搜索

当用户不方便进行语音输入时，可在检索框中输入对目标物品的描述文本，程序将基于该文本输入寻找物品。

### **监控画面**

本项目需配备一固定位置的外置摄像头，对桌面进行扫描，以便跟踪记录桌面上各物体的位置。在电脑屏幕显示监控摄像头画面输出提示本项目提供语音助手，将语音作为输出，通知用户已在图中找到物品或预测目标物品位置。但考虑到部分用户不方便使用语音，会同步将语音提示信息在输出提示框中进行输出，方便用户读取。同时，与操作相关的部分提示信息也会在该提示框进行输出。

### 录入物品列表

系统支持用户录入物品，录入时需要提供物品的高清完整照片及自定义的物品名称。

如果用户希望寻找的物品在当前已录入的物品列表中，则将该物品与当前输入画面进行比对，并圈出目标物体。

### 语音输入

长按语音按钮，进行语音输入。程序将对输入进行处理，并提取出用户希望寻找的物品信息，并在输入画面中对类似物品进行检索。

### 单机存储

在找到物品后，用户可以单击物品的正确位置进行物品信息存储，存储后的物品进入物品列表中，方便用户下次寻找。

# 创新

小组成员经过讨论得出以下几个具有实现价值的创新功能

1. 如果寻找物体被遮挡，则系统会提示物体被遮挡，并描述该物体被什么东西遮挡

程序通过摄像头监测桌面并追踪每个物体的移动情况，因此当目标物体被遮挡时，能够实时更新位置关系，并告知用户

1. 如果物体被移开拍摄范围，系统也能提示物体被移开。同时，程序会监测物体的移动，判

断出移开拍摄范围的事件，将记录其从桌面被移除的时间并告知用户。

1. 可从物体的遮挡记录来预测用户打开时想要寻找的物体并进行列举。
2. 程序暂存各物体的遮挡事件的发生。当用户打开或切换至程序窗口时，程序会自动标出刚刚被遮挡的物体。用户还能够对物体的重要程度进行标记，便于程序区分优先级对物体进行罗列

# 3.设计思路

## 3.1前端设计

### 3.1.1前端功能

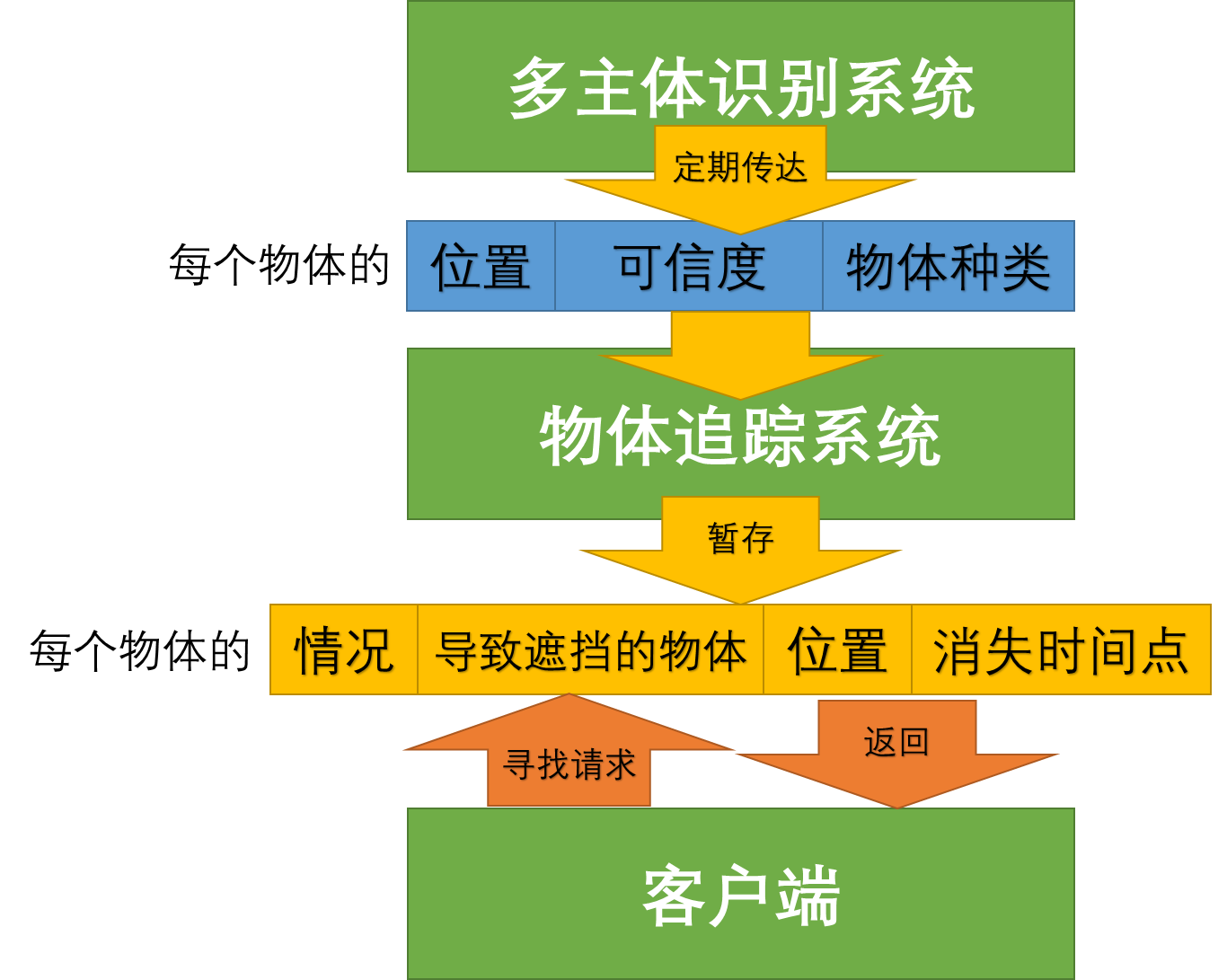
#### 3.1.1.1 物体追踪系统

##### 3.1.1.1.1 物体追踪系统功能目的

如果物体目前已出现在桌面上，那么用户自然可以自己找到，这个软件的用处将大大折扣。既然程序能一直观察着桌面，那应该可从先前的观察数据来追踪物体的移动历史，并利用该数据回答现在已被遮挡的物体的推测存在位置，与造成遮挡的物体，或者，它被拿开的话，去描述拿开的时间，这样才能体现这个软件的价值。

该物体追踪系统负责接受多主体识别系统发来的物体识别结果数据，处理该数据并做成物体移动的记录。当被请求寻找物体时，它应当能利用该记录判断物体的遮挡或拿开情况。如果从未出现在拍摄范围内，则说明从未出现。因为客户端需要在拍摄映像内标出目标物体的预测位置，因此还需要说明位置信息。从人的找东西的习惯上讲，还能给出物体消失的时间点会更贴心。

这些计算的结果应当返回给客户端，让客户端把这些信息转换为用户易于理解的信息提示。



图：物体追踪系统数据交互

##### 3.1.1.1.2 物体追踪系统实现

整个运行逻辑为：

1. 定期接受多主体识别系统的识别结果。
2. 监视物体的消失事件。
3. 记录每个物体的消失情况。
4. 在被要求寻找物体时，根据这个记录中的数据返回结果。

· 接受的原数据

多主体识别系统将识别结果以json格式的以下数据形式发送识别结果：

1. box数组 每个元素为物体的拍摄视野内矩形坐标（X(左上点X坐标)，Y（左上点Y坐标），W（矩形宽度），H（矩形长度）形式），数据类型为四个小数。
2. score数组 每个元素为物体识别可信度，值域0~1，数据类型为小数
3. class数组 每个元素为识别物体类型标签，值域>0数据类型为整数

上述的三个数组的大小为一致，共同下标代表了同一个识别结果的数据。

·接受数据处理步骤

1. 过滤掉可信度低于阈值（50%）的主体识别信息

经实验，返回结果的可信度取直方图时30%~90%区间几乎无数据形成谷，换言之很多时候只有很确信的和很不确信的识别结果。因此可以考虑利用该特征，过滤低于50%的低可信度的数据，简化后续处理。

1. 对比上一次的主体识别信息，检测物体消失

物体追踪系统每一次都会记录每个物体的出现的矩形位置信息，而如果这一次比上一次少了一些物体的出现时，应判断为这些物体以某种理由消失，能考虑到的情况为：拍摄环境突变（摄像头被遮挡、摄像头被移动）、物体激烈形变（变形、摧毁）、物体被遮挡、物体被移开拍摄视野。除去前两者极端情况，我们应考虑该物体应该是被遮挡或者被移开了当前拍摄区域，尽可能的服务最有可能发生的意外事件。

1. 判断消失情况

需判断是遮挡，还是被移开，可以从消失前最后确认到的物体中心点的位置，离拍摄视野边界有多近来去推测。物体中心点可从矩形数据 (xywh)获得的矩形中心点模拟，即

cx = x + w/2, cy = y + h/2

并使用坐标(cx, cy)作为物体的中心点。

接着计算该坐标与拍摄边界的最小水平/垂直距离：

minBordorDx = min(cx, screen – cx), minBordorDy = min(cy, screen – cy)

如果这两个值有一方小于我们设定的值，应当认为该物体是通过移开拍摄范围的方式从拍摄视野消失的。这个值的确定仍还需要实验。

如果判断情况不是移开拍摄范围，那么从上述的四种情况（拍摄环境突变、物体激烈形变、物体被遮挡、物体被移开拍摄视野）来考虑，应设计为认为是被遮挡最有用户需求。这时，可以从其它主体识别结果，还去推断目前导致该遮挡的物体是什么。如果能给客户端提供该信息，用户能从视野中的该物体出发去确认目标物体的位置，因而提高效率。

该功能实现思路为，使用刚刚的模拟中心点坐标(cx, cy)，判断哪些识别出来的主体的拍摄视野内矩形与该坐标重合。多个主体重合时，取中心点之间距离最近的一个。能得到结果时，将该主体认为是导致遮挡的物体，并将该信息提供给客户端。无法得到结果时，不做导致遮挡的物体的信息提供，客户端仅标出物体最后出现位置。

最后，作为设计贴心，应当返回一个物体最后出现时间。人有时找东西时会回想物体最后存在的时间，而该功能可以帮助人的这个思维工作。

1. 成员变量

综上所述，该程序所需的成员变量为

Struct ObjectRecord {

Float x, y, w, h; //拍摄视野内矩形坐标

Int classID; //分类标签

Long lastAppeardTime; //最后一次的出现时间

Status status; //当前情况

ObjectRecrod hideCause; //导致遮挡的物体

}

Enum Status {Normal, Hided, Removed} //Normal: 当前存在拍摄范围内, Hided: 被遮挡, Removed: 被移开拍摄范围

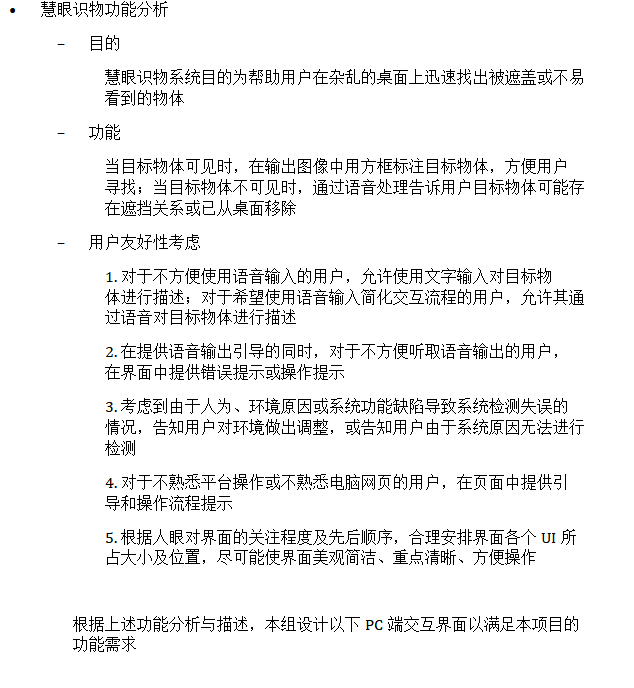
List<ObjectRecord> lastObjectRecords; //上一次的物体出现记录列表

1. 处理寻找请求

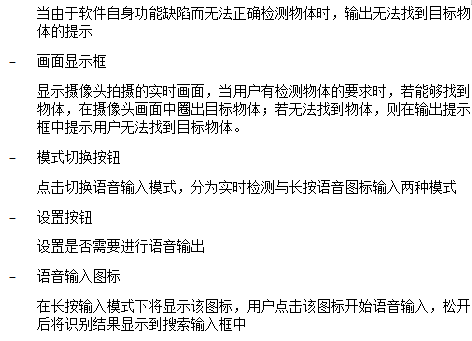
用户发出寻找命令时，该系统应收到需要寻找的目标物体的分类标签，整数数据类型。

该系统将返回对应的情况，导致遮挡的物体（如有）、位置（如确定）、最后出现时间（如有），也就是上述的ObjectRecord结构体。如果没有该物体的记录，则判定为从未出现，返回空。

### 3.1.2前端界面







## 3.2后端设计

### 3.2.1后端功能

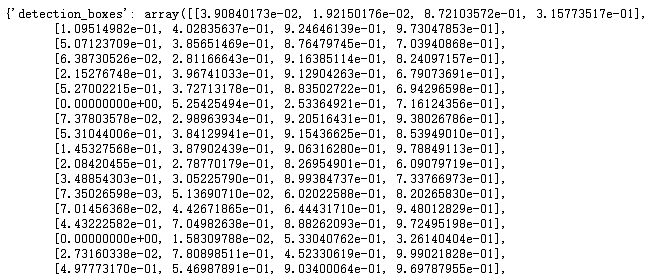
#### 3.2.1.1 Tensorflow识别图像dog: 930

图：Tensorflow图像识别性能展示

Tensorflow可对一个图像进行多主体识别。定期从摄像头获取拍摄映像，并对其进行处理。

对于每一个识别结果会有以下三个数据：

1. Detection\_boxes: 检测图像的具体位置大小，具体的表现形式为一个矩形
2. Detection\_scores:每个矩形框他是某一种类（Detection\_Class）的可能性
3. Detection\_classes:最有可能是哪一个物体类

多个识别结果的组合以如下图的json格式被发出

dtype=fIoat32) , 
o. 1396198 , 
o. 08091754, 
o. 0707489 
O. 063801B 
o. 05641 aga, 
o. 05425417, 
o. 05219474, 
o. 04822728, 
o. 04614913, 
o. 0416657 , 
o. 03988713, 
689882, 
. 3612375, 
o. 03572747, 
o. 0340262, 
o. 03349814, 
o. 0325323 , 
o. 03133166, 
LU. uuuuuuuue+UU, d. 
o. 13212034, o. 09975743, 
o. 07895234, o. 07818332, 
o. 05164081, o. 0506601 , 
0. 03677598, 0. 03671169, 
o. 03200769, o. 03199518, 
o. 03106266, o. 03057358, 
21 18, 
18, 62, 
l, 88, 47, 
, 18, 21, 
18, 47, 
' detection_scores' 
L. Odd4YYdUe—UL, O. 
IYld1Y41e—UIJJ, 
array ( [0. 9406307 
o. 9345027 , 
o. 23088235, 
o. 2251895 
, o. 17249912, 
, o. 06597456, 
o. 06050137, 
o. 05638802, 
o. 0540958 , 
o. 04720658, 
o. 04420188, 
o. 04137346, 
o. 03984368, 
3877512, o. 03853002, o. 03816223, 
o. 03610823, 
o. 03565428, 
O. 0345gg27, 
o. 03300408, 
o. 06404546, 
o. 05898514, 
o. 05500007, 
o. 05335933, 
o. 04673675, 
o. 04286546, 
o. 04096133, 
o. 03930986, 
o. 03601933, 
o. 03555554, 
O. 0338392g, 
o. 03294906, 
classes' 
o. 08992714, 
o. 07565752, 
o. 06387997, 
o. 058983 
o. 0549573 , 
o. 05279237, 
o. 05030194, 
o. 04671779, 
o. 04251266, 
o. 04053533, 
o. 0392389 , 
o. 03815907, 
o. 03654769, 
o. 0358645 , 
o. 03504348, 
o. 0338195 , 
o. 0328106 , 
o. 03172883, 
o. 03044927, 
, 63, 
18, 
18, 
O. 0888880B, 
o. 07331279, 
o. 172, 
o. 05761427, 
o. 05426416 
o. 05228513, 
o. 04837742, 
o. 04644245, 
0.0417434 , 
O. oaggog€3, 
o. 03742462, 
o. 03635332, 
o. 03582653, 
O. 034g5529, 
O. 03363g01, 
o. 03275439, 
O. 031d9137, 
o. 03041705) , 
dtype=fl oat 32) , 
' detectlon 
18, 
1B 1B 18, 
33, 
63, 
18, 
21, 
3, 18, 
62, 
18, 
62, 
18, 
13, 
62, 
21, 
1, 62, 
, 62, 21, 
1, 62, 62, 
18, 
57, 
' mm detectlons 
1, 21, 
62, 
62, 
62, 
84, 
1 ooh 
1, 84, 18 
dtype=int64) , 

之后这些数据会被传值前端的物体追踪系统，负责记录所有物体的位置与消失事件。

我们对该数据进行分析，发现这个图相对来说干扰较少，是一个歧义性较少的一个例子。

可以看出detection\_scores里，数值高于30%的可能性的猜测竟然只有两个，而这两个正是图1中的两个正确的结果。利用这个特征能帮助我们过滤不必要的信息。

#### 3.2.1.2 Tensorflow语音识别

**ASRT的语音识别系统：**

声学模型采用了深度全卷积神经网络，直接将语谱图作为输入。

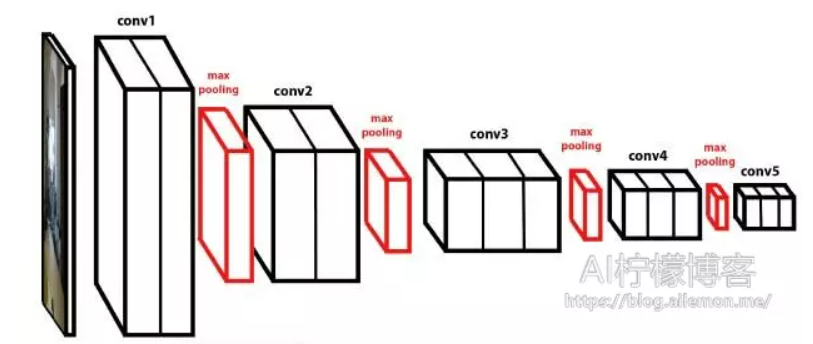
模型结构借鉴了图像识别中效果最好的网络配置VGG，有着很强的表达能力。

**系统流程：**

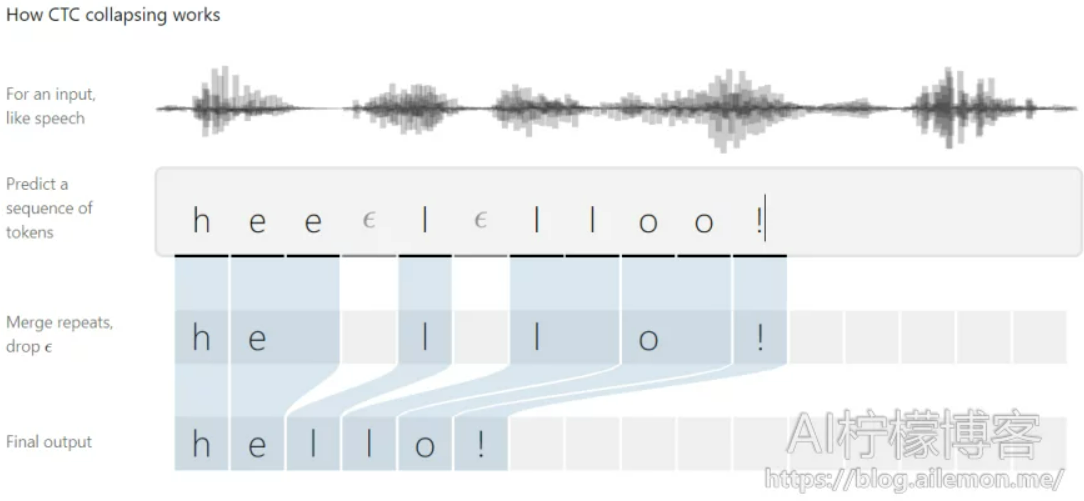
**1. 特征提取：**将普通的wav语音信号通过分帧加窗等操作转换为神经网络需要的二维频谱图像信号，即语谱图。



**2. 声学模型：**基于Keras和TensorFlow框架，使用这种参考了VGG的深层的卷积神经网络作为网络模型，并训练。



**3. CTC解码：**在语音识别系统的声学模型的输出中，往往包含了大量连续重复的符号，因此，我们需要将连续相同的符合合并为同一个符号，然后再去除静音分隔标记符，得到最终实际的语音拼音符号序列。



**4. 语言模型：**使用统计语言模型，将拼音转换为最终的识别文本并输出。拼音转文本的本质被建模为一条隐含马尔可夫链，这种模型有着很高的准确率。

# 4. 已完成工作

## 前端

### 物体追踪系统

目前已搭建基本数据，正在编写处理逻辑。

### 界面

**前端：**

PC前端UI界面已完备，能够访问摄像头获取视频。具体内容见前端部分。

## 后端

已完成服务器环境搭建，后端语音输入检测与图像目标检测部分待部署。

### 4.2.1 图像识别

基于tensorflow2.4.0与windows10环境，完成本地Object Detection API的部署和运行，支持检测单张图片和视频。

### 4.2.2 语言识别

1. 前端语音检测

在Vue使用MediaRecorder API 进行实时和录音的功能切换实现。

2. 语音识别系统

在决定使用ASRT的语音识别系统作为后续方案后，目前正在青云服务器上进行实施系统 的部署。

# 后续工作与分工

## 5.1**分工完成情况**

|  |  |
| --- | --- |
| Tensorflow识别图像本地 | 谢安琪 |
| Tensorflow语音识别本地 | Edward |
| Vue前端界面 | Naomi, Izumi |
| 遮挡检测算法设计 | Naomi，Izumi |

## **5.2后续分工**

|  |  |
| --- | --- |
| Tensorflow识别图像服务器部署 | 谢安琪 |
| Tensorflow语音识别服务器部署 | Edward |
| 前后端连接与功能测试 | Naomi, Izumi |
| 遮挡检测算法实现 | Naomi，Izumi |