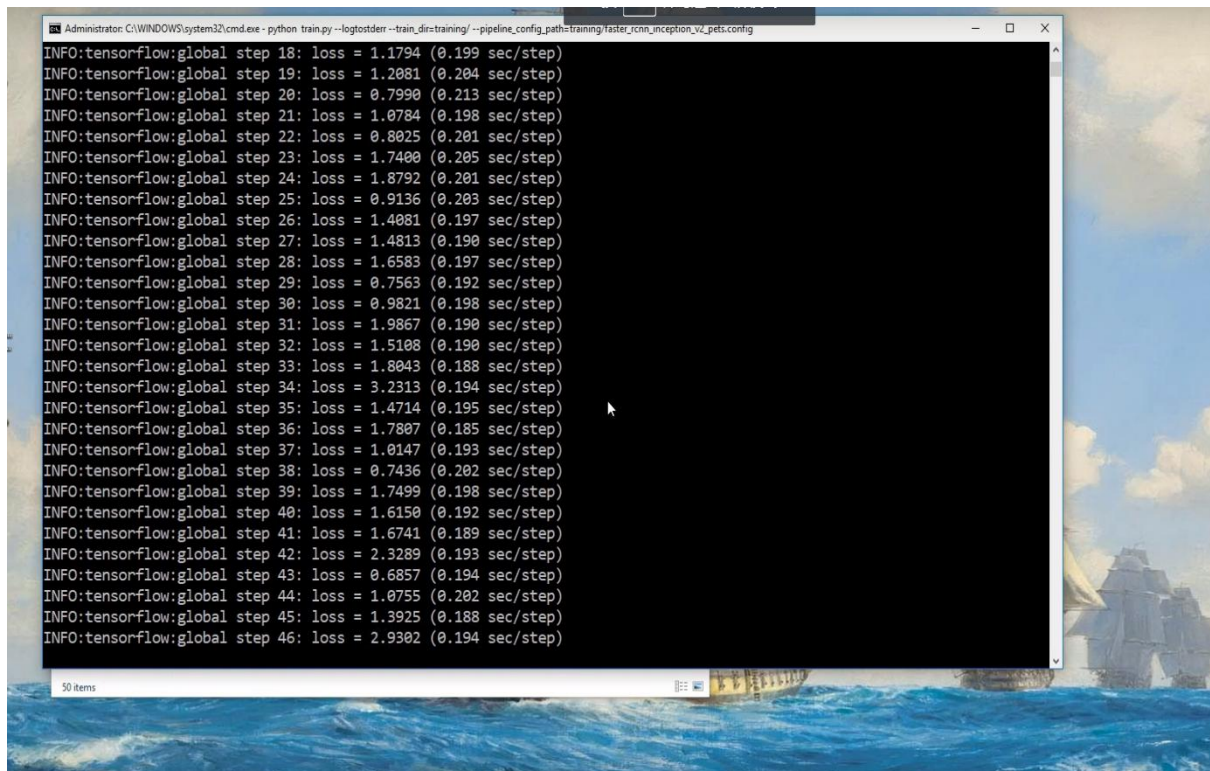


# “天眼”野外生物识别追踪系统解决方案

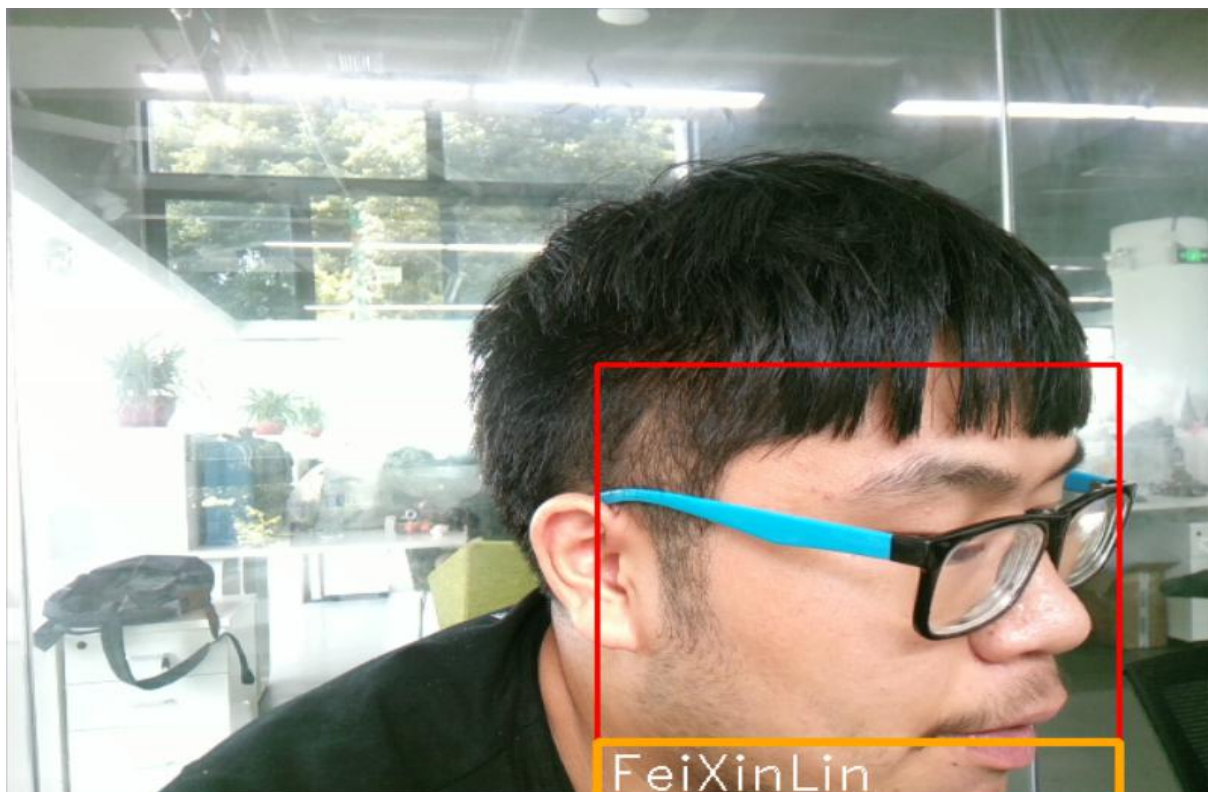
## 1. 感知层技术:

**核心：**通过本地摄像头机器识别人与动物。通过使用谷歌 tensorflow 卷积神经网络算法生成我们需要的模型，再调用训练好的模型，使我们的摄像头能够识别对应的人物，然后我们再通过 socket 协议将识别到的物体平面坐标传输到树莓派上，此时树莓派调用 GPIO 库，根据坐标实现对物体的追踪。

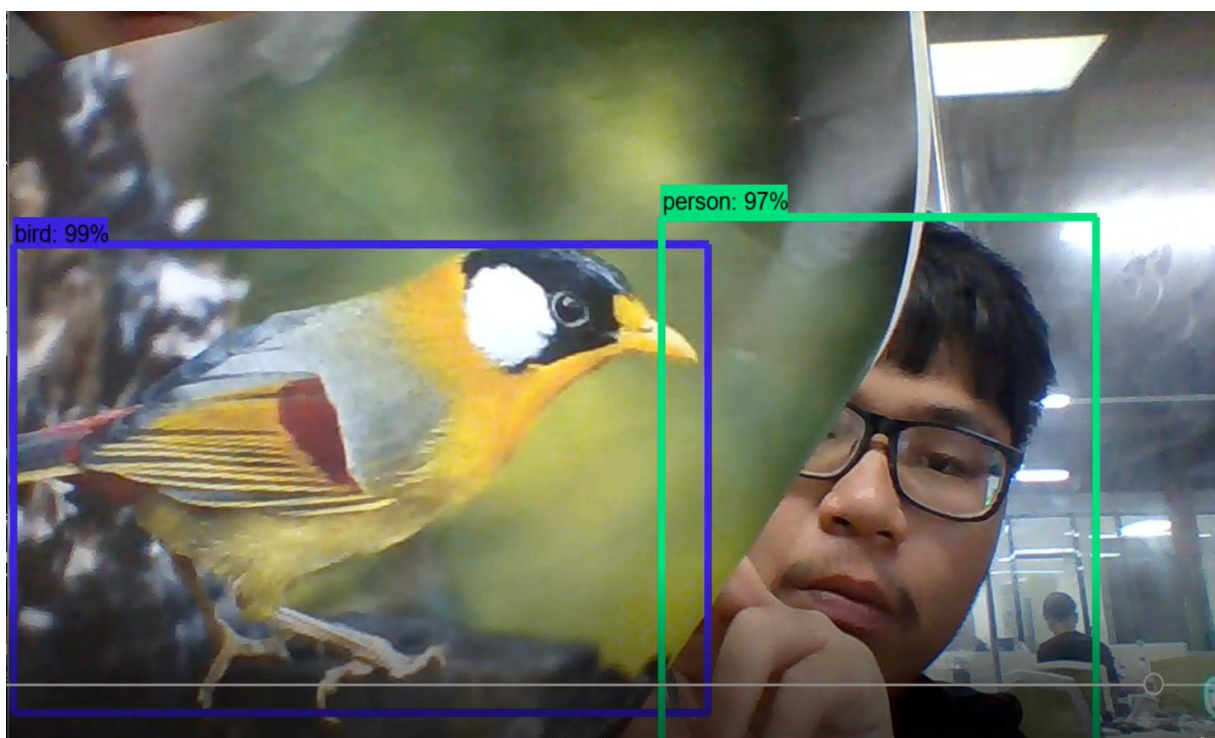


```
Administrator: C:\WINDOWS\system32\cmd.exe - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
INFO:tensorflow:global step 18: loss = 1.1794 (0.199 sec/step)
INFO:tensorflow:global step 19: loss = 1.2081 (0.204 sec/step)
INFO:tensorflow:global step 20: loss = 0.7990 (0.213 sec/step)
INFO:tensorflow:global step 21: loss = 1.0784 (0.198 sec/step)
INFO:tensorflow:global step 22: loss = 0.8025 (0.201 sec/step)
INFO:tensorflow:global step 23: loss = 1.7400 (0.205 sec/step)
INFO:tensorflow:global step 24: loss = 1.8792 (0.201 sec/step)
INFO:tensorflow:global step 25: loss = 0.9136 (0.203 sec/step)
INFO:tensorflow:global step 26: loss = 1.4081 (0.197 sec/step)
INFO:tensorflow:global step 27: loss = 1.4813 (0.190 sec/step)
INFO:tensorflow:global step 28: loss = 1.6583 (0.197 sec/step)
INFO:tensorflow:global step 29: loss = 0.7563 (0.192 sec/step)
INFO:tensorflow:global step 30: loss = 0.9821 (0.198 sec/step)
INFO:tensorflow:global step 31: loss = 1.9867 (0.190 sec/step)
INFO:tensorflow:global step 32: loss = 1.5108 (0.190 sec/step)
INFO:tensorflow:global step 33: loss = 1.8043 (0.188 sec/step)
INFO:tensorflow:global step 34: loss = 3.2313 (0.194 sec/step)
INFO:tensorflow:global step 35: loss = 1.4714 (0.195 sec/step)
INFO:tensorflow:global step 36: loss = 1.7807 (0.185 sec/step)
INFO:tensorflow:global step 37: loss = 1.0147 (0.193 sec/step)
INFO:tensorflow:global step 38: loss = 0.7436 (0.202 sec/step)
INFO:tensorflow:global step 39: loss = 1.7499 (0.198 sec/step)
INFO:tensorflow:global step 40: loss = 1.6150 (0.192 sec/step)
INFO:tensorflow:global step 41: loss = 1.6741 (0.189 sec/step)
INFO:tensorflow:global step 42: loss = 2.3289 (0.193 sec/step)
INFO:tensorflow:global step 43: loss = 0.6857 (0.194 sec/step)
INFO:tensorflow:global step 44: loss = 1.0755 (0.202 sec/step)
INFO:tensorflow:global step 45: loss = 1.3925 (0.188 sec/step)
INFO:tensorflow:global step 46: loss = 2.9302 (0.194 sec/step)
```

### 1.1. 训练过程



1.2. 目标识别和追踪



1.3 多对象多物种同时识别



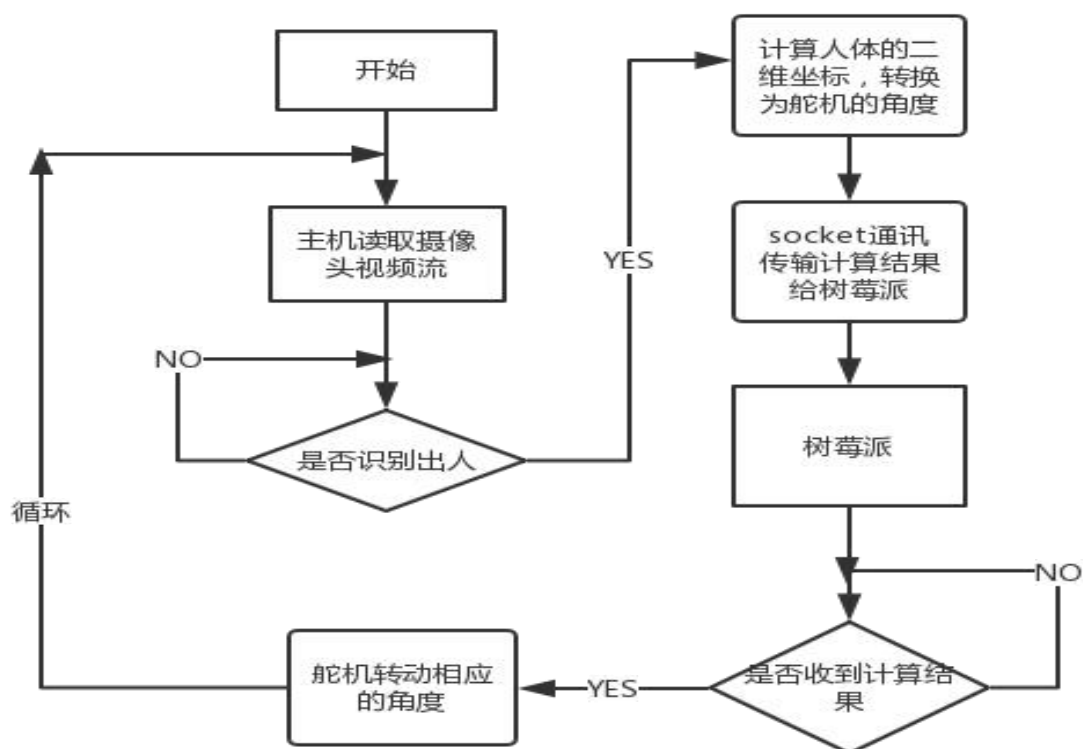
#### 1.4 遮挡下的效果识别

识别效果演示链接

[https://v.youku.com/v\\_show/id\\_XMzczNjczNjczMg==.html?x&sharefrom=android&sharekey=9f1f665514e48e9316f686048a8e2c8f9](https://v.youku.com/v_show/id_XMzczNjczNjczMg==.html?x&sharefrom=android&sharekey=9f1f665514e48e9316f686048a8e2c8f9)

#### 2. 应用层技术:

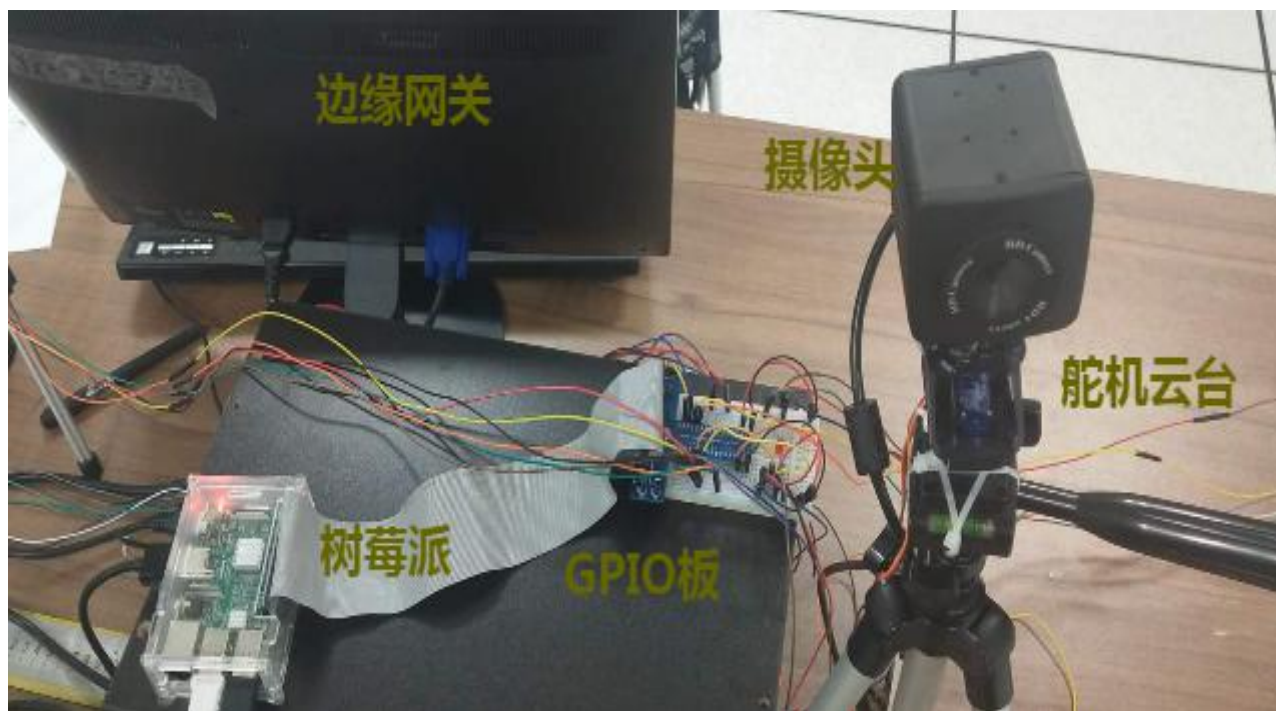
如图 2.1 所示系统工作流程图。我们的系统主要由高性能 Win10 主机作边缘网关，RaspberriPi 3B+, SG90 二自由度舵机，800 万像素高清摄像头等设备组成。



2.1 “天眼”识别追踪系统流程图

依托具备强大运算能力的边缘网关，通过 Python 运行训练好的 TensorFlow Models 实现识别人与动物。同时 python 通过 socket 协议将识别到的物体平面坐标传送到树莓派，树莓派根据通过 decode() 解码得到真实的坐标，再利用 RPi.GPIO, PWM.ChangeCutycycle() 方法控制舵机转动，实现追踪离去方向。





2.2 系统硬件设备



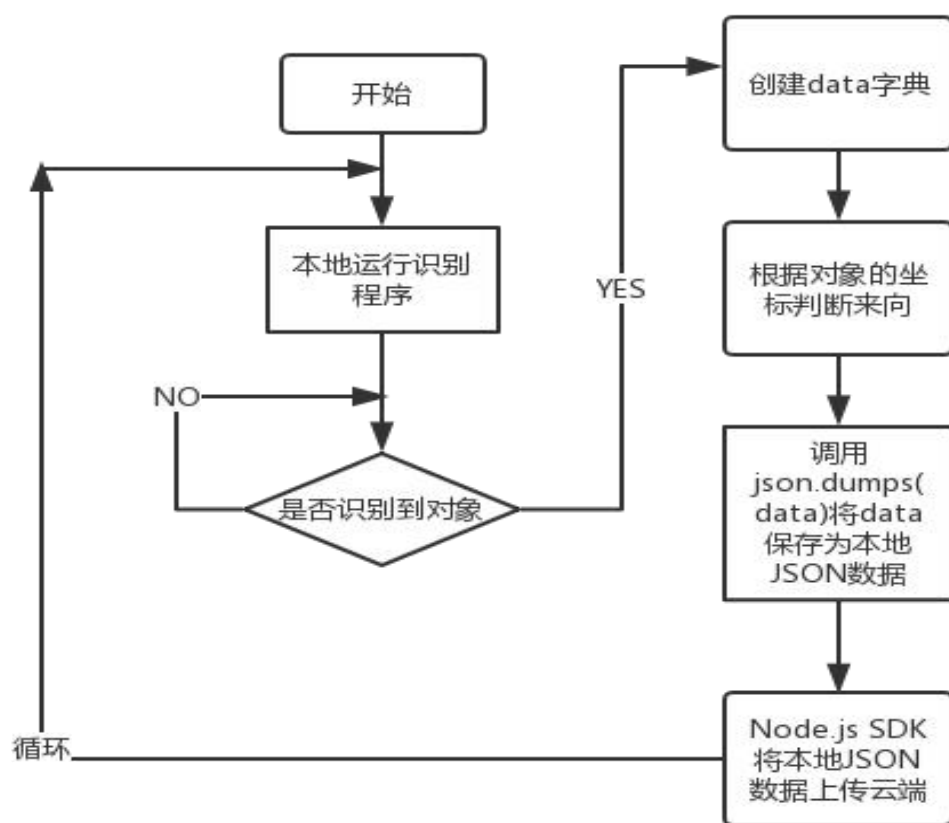
2.3 四摄联合监控场景摄像头通过边缘网关组网拓扑

追踪效果演示链接:

[http://www.meipai.com/media/1021188274?uid=1510885906&client\\_id=1089857302&utm\\_source=meipai\\_share&utm\\_term=meipai\\_android&utm\\_content=test&viewCount=1&shareCount=1](http://www.meipai.com/media/1021188274?uid=1510885906&client_id=1089857302&utm_source=meipai_share&utm_term=meipai_android&utm_content=test&viewCount=1&shareCount=1)

### 3. 表示层技术:

如图 3.1 所示流程图, 我们通过 Node.js SDK 绑定设备激活凭证, 将识别到的对象名、对象出现的时间、摄像头所在的小地名、对象出现的方向、离去的方向、逗留的时间等属性信息通过 `json.dumps()` 方法将 Python 数据保存为本地 JSON 文件。



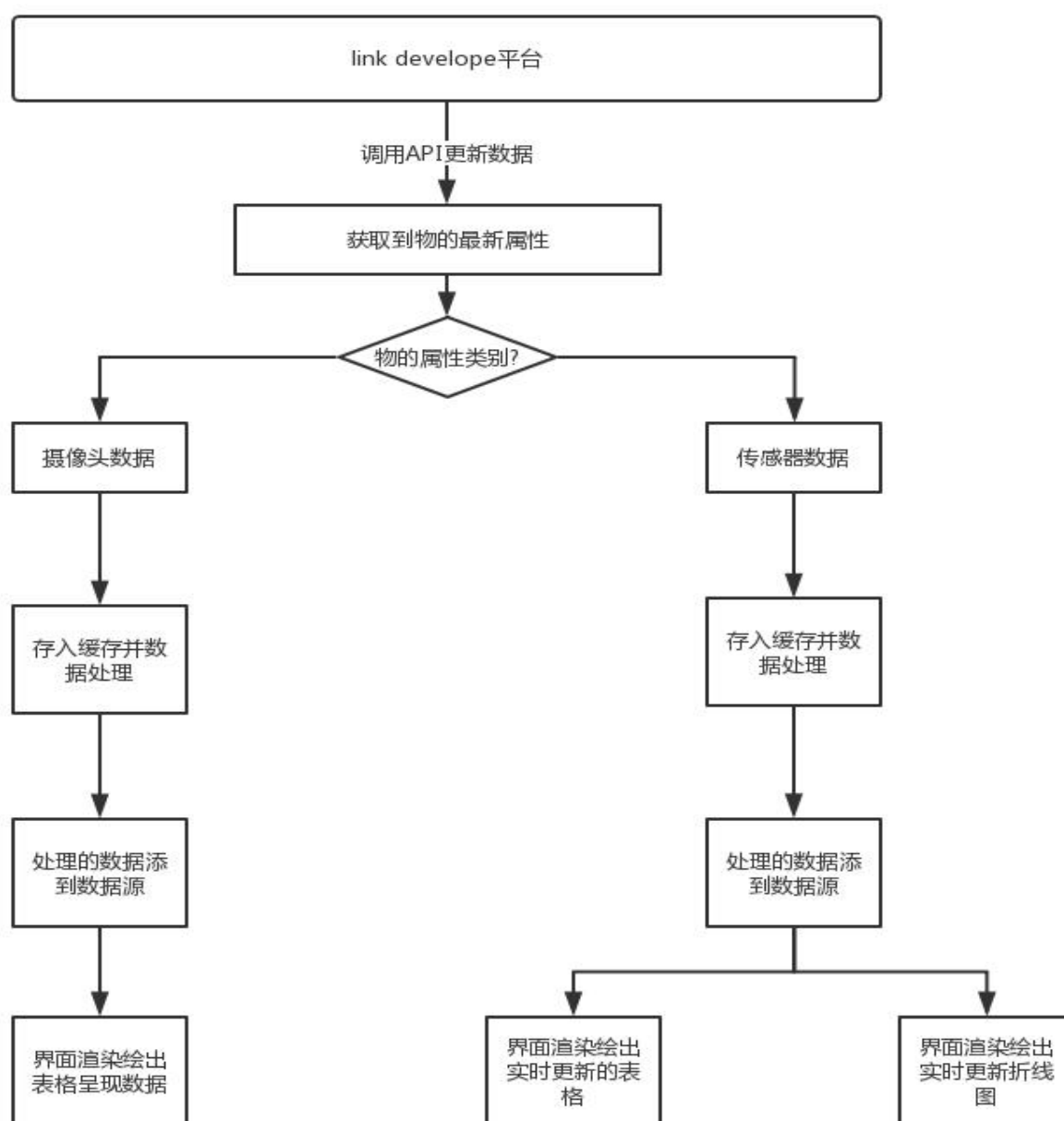
3.1 保存 JSON 数据流程图

### 4. 数据链路层技术:

利用 mqtt 协议调用 node.js 的设备接入 SDK, 实现与平台的链接。通过 node-schedule 计时器实现定时更新从 python 接受到的数据, 从而使我们上传 link Develop 平台的数据具备实时性。

## 5. Web 应用层技术:

Bone 框架下做界面渲染，运用 Viser-react 可视化制作图表。构造多层对象分布图，并在被识别的对象所对应的分布图上**加深识别发生次数较多的区域**。最终实现识别数据与对象分布情况**可视化输出**。



5. 1Web 应用端架构图

