

# 利用人工智能解决移动端性能测试痛点

刘永刚 高级测试开发工程师

email: [liuyonggang@360.cn](mailto:liuyonggang@360.cn)



# 个人简介



刘永刚 奇虎360高级测开工程师

2017年加入奇虎360，现在主要从事手机卫士的功能自动化、性能测试及相关测试工具的开发工作。

曾就职于“LG数字电视研究所”、“乐视致新”，近10年的Android App开发、测试相关工作，积累了较丰富的移动端测试经验。

# 分享目的



本次分享通过App的启动时间为例，结合当前较火爆的人工智能技术对比传统的测试技术来展示人工智能在测试领域的优势。

详细介绍人工智能的核心概念、构造模型、训练模型、模型预测等知识点，协助相关童鞋快速理解人工智能的概念，写出自己的人工智能作品。本次分享实则抛砖引玉，希望更多的测试人去学习人工智能，使之在测试领域发挥更大的作用。

# 性能测试项



## 启动时间

应用冷启动时间  
应用热启动时间



## 资源消耗

cpu、内存、耗电等



## 响应时间

清理扫描、加速时间  
杀毒扫描、处理时间  
等等



## 系统影响

安装应用后  
手机系统启动时间  
三方应用启动时间



## 显示流畅度

各种扫描时的帧率变化以  
及丢帧情况



# 常规启动时间测试方法



# 基于人工智能的启动时间测试



基于人工智能的启动时间测试过程：



# 人工智能核心概念



# 人工智能核心概念



人工  
智能

- 人工 + 智能

机器  
学习

- 从数据中提取知识，它是统计学、人工智能和计算机科学交叉的研究领域，又叫“预测分析”或“统计学习”。

监督  
学习

- 人工将成对的已知输入和输出提供给算法，算法会找出一种方法，根据给定的输入给出预期输出。

无监督  
学习

- 没有已知输出，只有输入数据并需要从这些数据中提取知识。



# 人工智能核心概念



## 样本

- 每一对人工提供的数据就是一个“样本”。

## 特征值

- 样本的属性称之为“特征值”。

## 标签

- 样本的已知输出（即类别）。

## 模型

- 各机器学习库中包含很多算法用来从人工提供的数据对中提取特征值，每一个算法就是一个模型（每一个算法都有相关的类实现），模型使用前必须先实例化模型对象。

# 人工智能核心概念



## 训练

- 让模型进行学习的过程。

## 固化

- 将训练完后的模型对象保存到本地文件中，下次使用时直接加载即可。

## 分类

- 也叫预测，使用训练的模型对没有训练过的输入数据进行判定，对其输出（即类别）做出决策。

## 泛化

- 训练完毕的模型对象如果能够对未被学习的输入做出正确的判断则称该模型的泛化能力强，反之则是泛化能力弱。

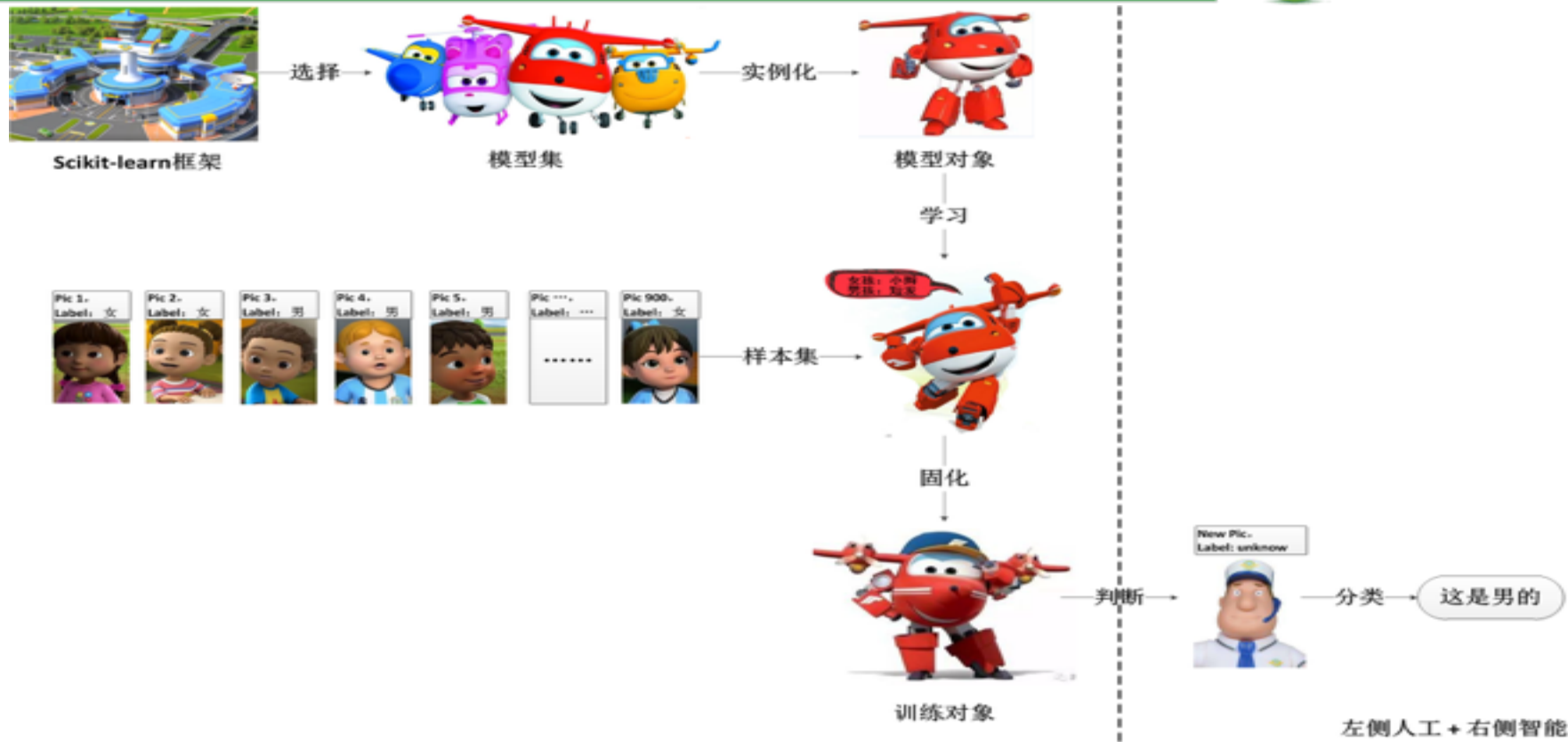
## Scikit-Learn

- 开源项目，免费使用和分发，轻松获取其源代码，包含许多最先进的机器学习算法(<http://scikit-learn.org/stable/documentation>)。

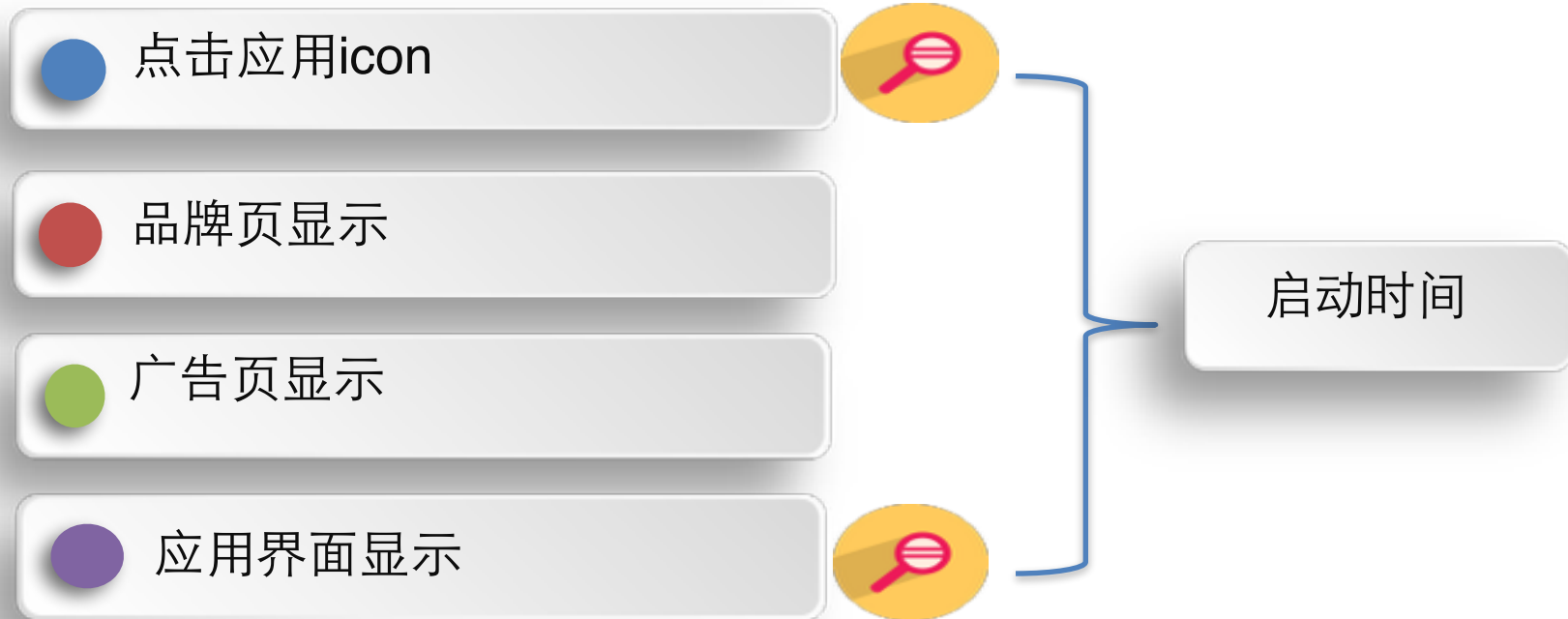
## Tensor Flow

- 另一个项目，其他略。

## 各概念关系图



# 应用启动需求分析

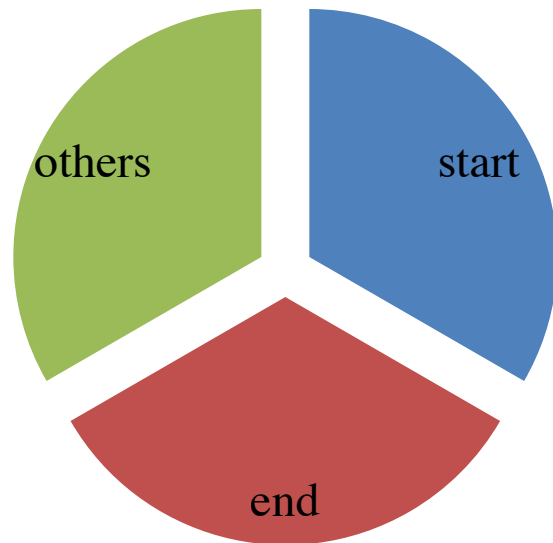


# 应用启动需求分析

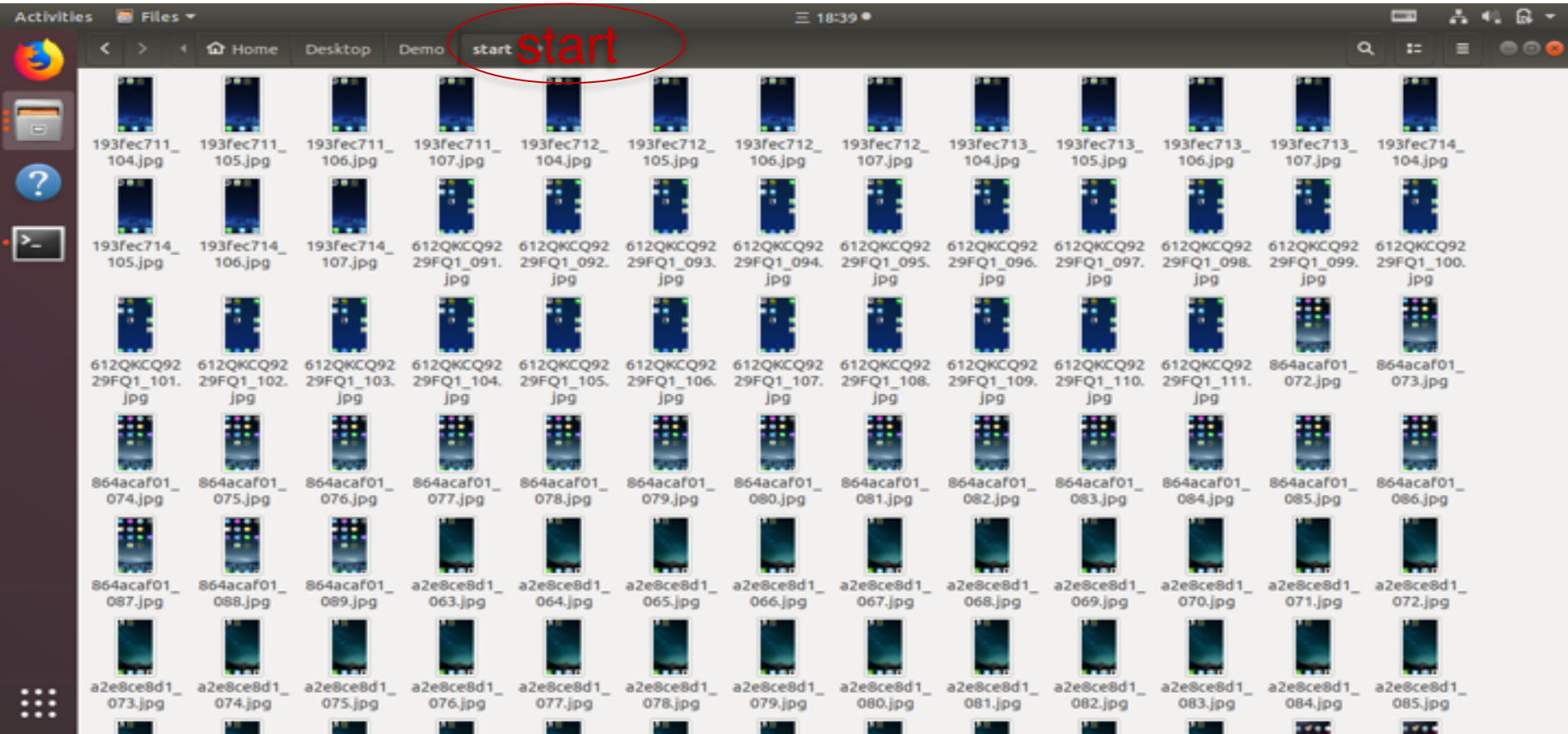


## 三分类问题

| 场景图片       | 标签     |
|------------|--------|
| App icon置灰 | start  |
| App home界面 | end    |
| 其他界面       | others |



# 样本收集



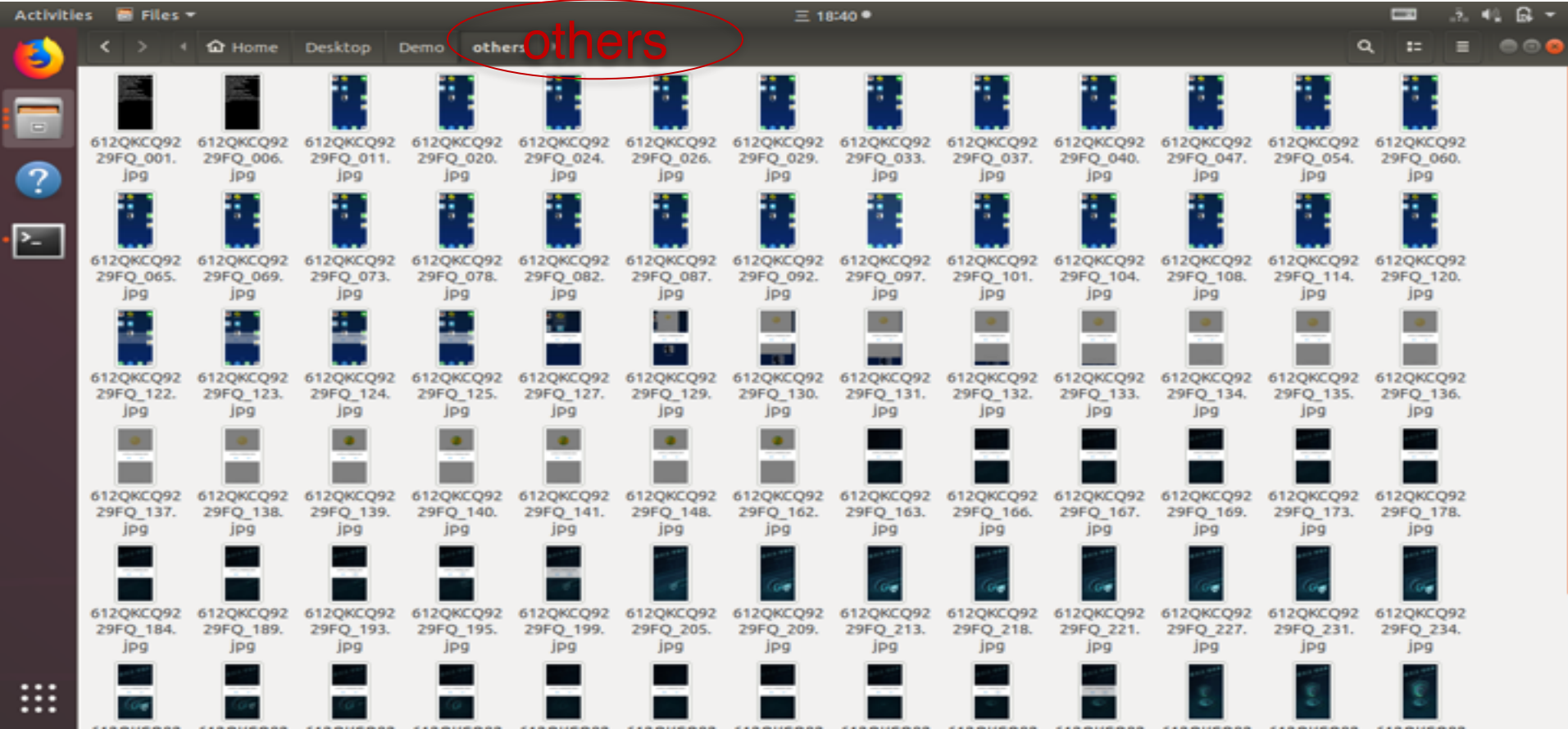


# 样本收集





# 样本收集



# 模型训练、固化



```
def training_model():  
    '''  
    tr_img是一个列表，其中每一项数据值都是从样本集的一个图片转成的数组数据；  
    tr_label是一个列表，其中每一项数据值都是对应着tr_img数据值的标签  
    '''  
  
    tr_img, tr_label = get_train_image_datas()  
    print ("tr_img={}, length={}".format(tr_img, tr_img.__len__()))  
    print ("tr_label={}, length={}".format(tr_label,  
tr_label.__len__()))  
    # 实例化模型对象  
    linear_svcClf = svm.LinearSVC()  
    # 提供学习样本给对象学习,也可以交过训练模型  
    linear_svcClf.fit(tr_img, tr_label)  
    # 固化模型  
    joblib.dump(linear_svcClf, "linear_svcClf_train_model.m")
```

```
adb shell screenrecord /sdcard/filename.mp4
```



```
ffmpeg -i filename.mp4 -s w*h -r X icname.jpg
```

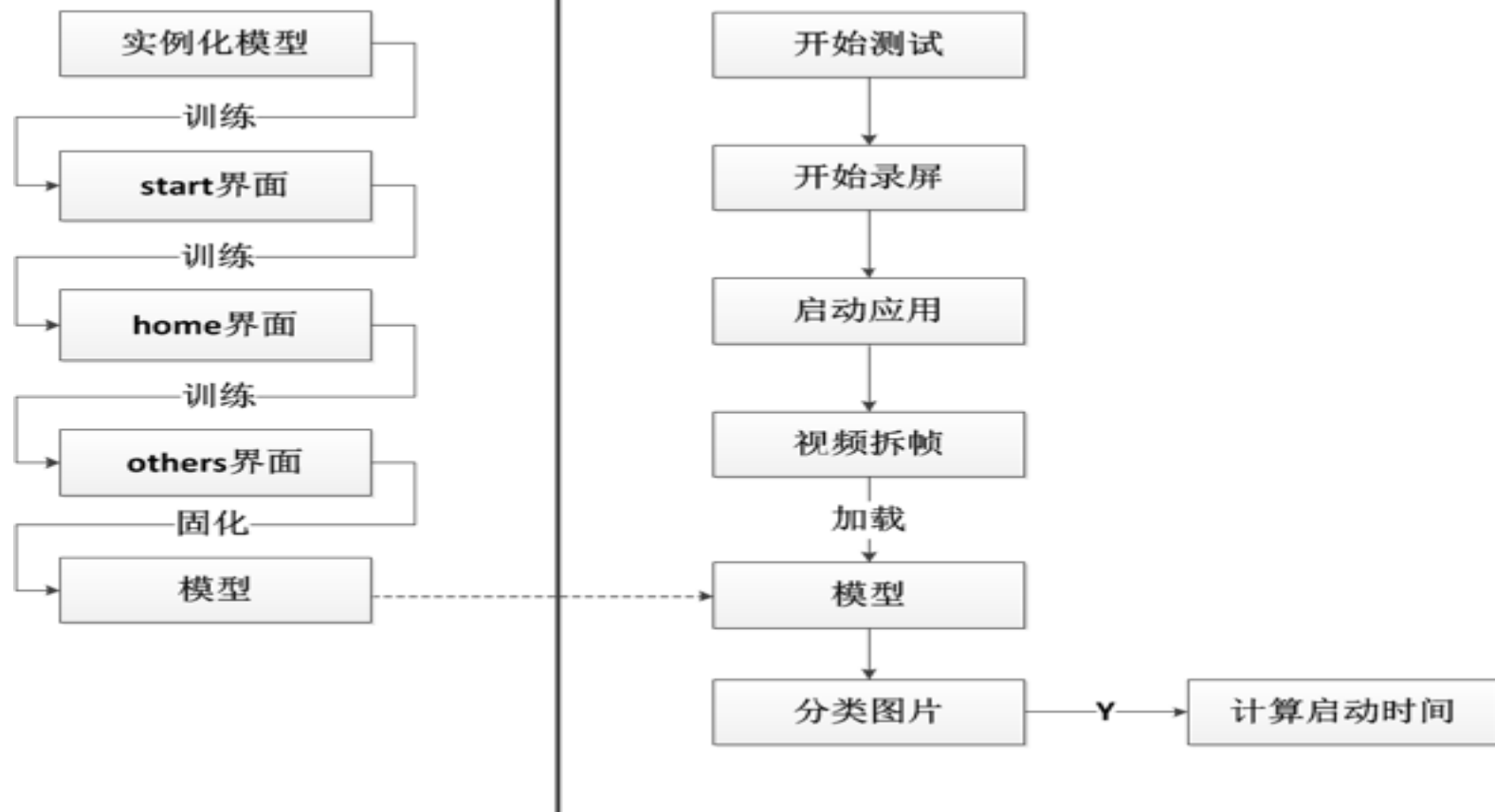
# 模型预测



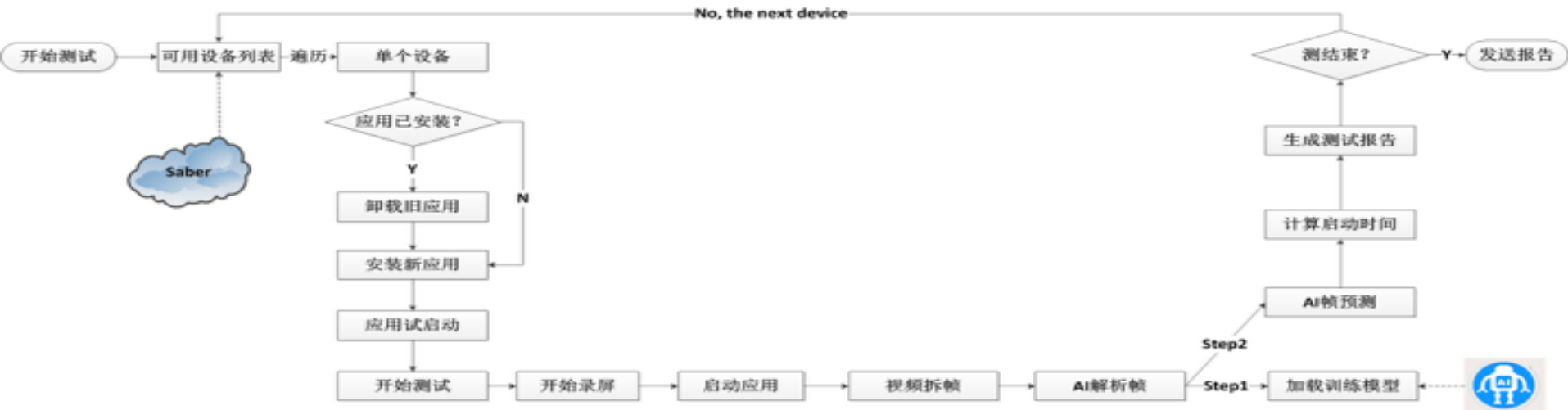
```
def learning_by_modle(model):  
    # 加载已固化模型  
    clf = joblib.load(model)  
    # tst_img是一个列表，其中每个数据值是拆帧视频得到的图片所转换成的数组  
    tst_img = get_test_image()  
    # 对数据执行分类工作  
    predicts = clf.predict(tst_img)  
    for index, result in enumerate(predicts):  
        if result == "start":  
            start_id = index + 1  
            break  
    for index1, result1 in enumerate(predicts):  
        if result1 == "end":  
            end_id = index1 + 1  
            break  
    return start_id, end_id
```

$$\text{启动时间} = (\text{end\_id} - \text{start\_id}) * (1000 / \mathbf{x})$$

# 核心逻辑



# 基于ai启动时间测试框架



- 单个手机每次启动时间
- 单个手机启动时间平均值
- 单个手机启动时间最大值
- 单个手机启动时间最小值
- 按品牌统计启动时间
- 按rom版本统计启动时间



谢谢！



360  
www.360.cn

安全第一