



树莓派及其应用



什么是树莓派

1



什么是树莓派

- 树莓派（ Raspberry Pi ），是一款基于Linux的单片机电脑
- 它由英国的树莓派基金会所开发
- 目的是以低价硬件及自由的软件促进学校的基本电脑科学教育



树莓派
(一般来说指的是这个)



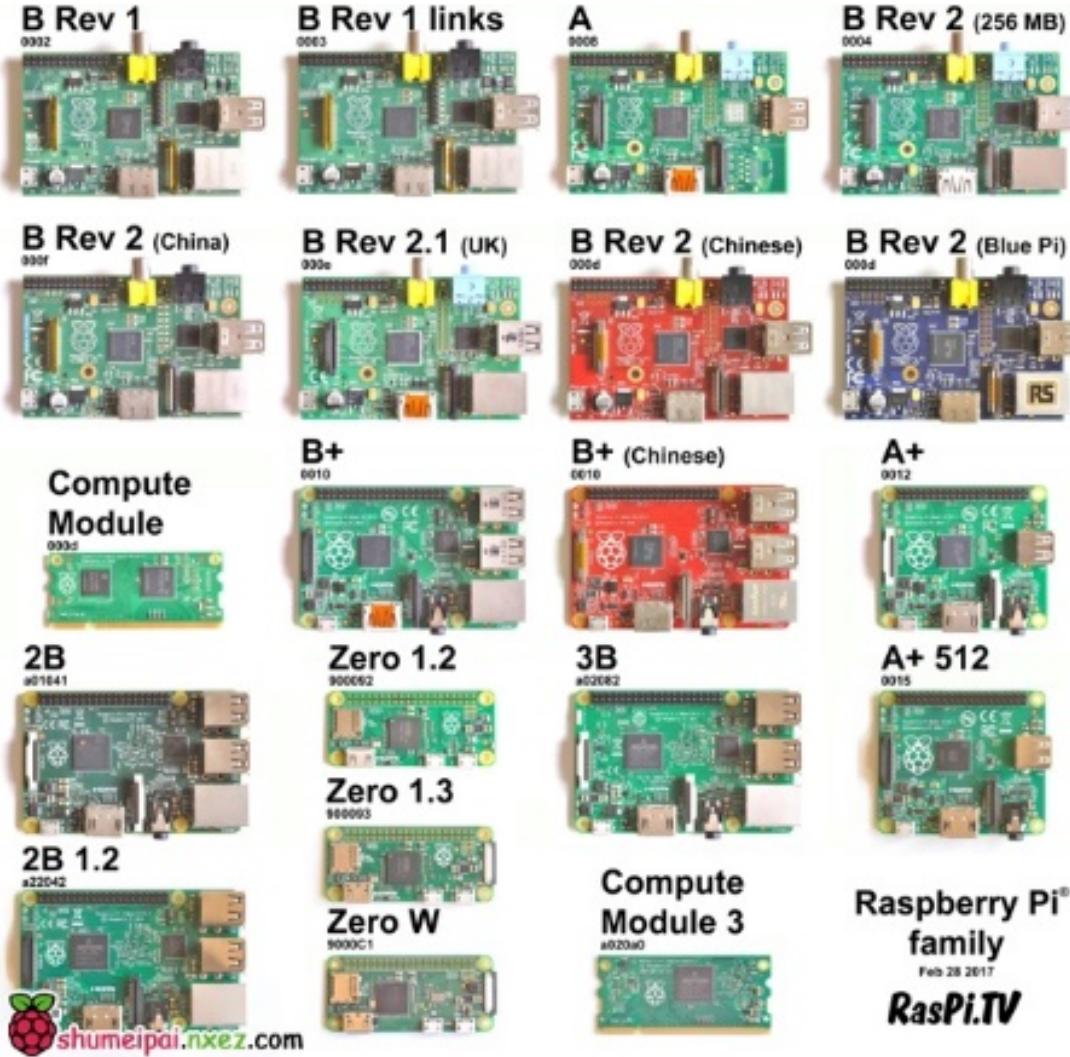
树莓派 Zero



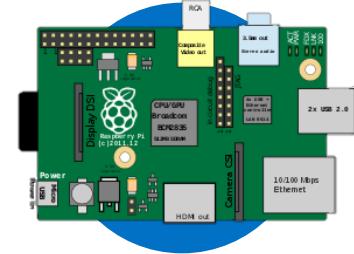
树莓派 计算模块
(相当于只有CPU)



树莓派的简要发展历程

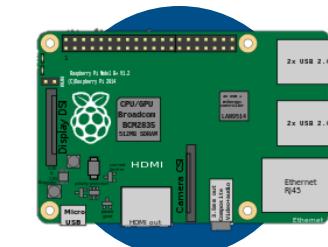


shumeipai.nxez.com



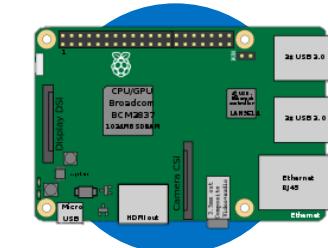
2012年 1B

256 / 512MB 内存 HDMI 输出
700MHZ CPU 支持 GPIO



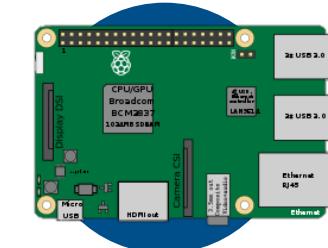
2015年 2B

更改布局 板载 1G 内存
更快的 CPU 增加 USB 数量



2016年 3B

64位处理器
板载 Wifi 和蓝牙



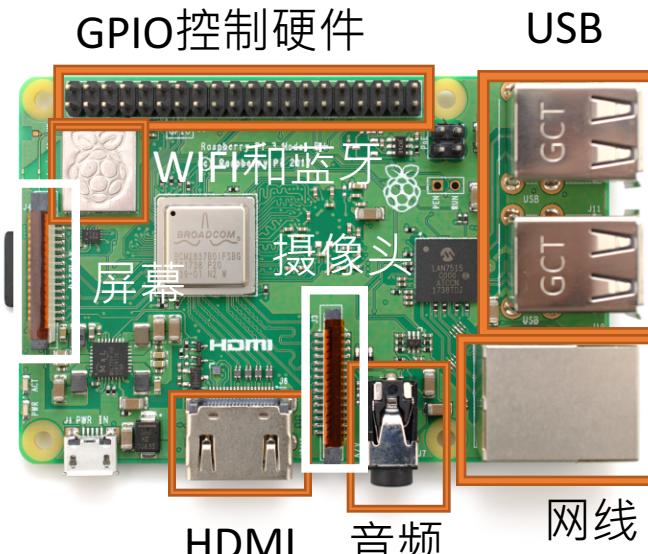
2018年 3B+

更快的CPU
更好的网络性能
支持蓝牙

为什么要用树莓派——“软硬皆施”

软件方面

1. 支持多种完整操作系统
 - Windows IoT
 - 多种 Linux 发行版
 - Android
2. 支持任意编程语言
 - Python
 - C/C++
 -
3. 支持多种外设
 - 键盘鼠标音响打印机摄像头
4. 支持大量软件
 - Linux下的各种软件
 - Docker
 -



树莓派可以看作
“电脑” + “硬件控制器”的结合体

硬件方面

1. 完整的GPIO支持带来无限可能
 - 完美兼容Arduino硬件
2. 比常规单片机处理能力强
 - 性能和功能均远超51单片机和Arduino
3. 体积小，易于嵌入到其他设备中作为计算核心
4. 自身功耗低，发热小
 - 待机5W左右
 - 满载10W左右
 - 充电宝即可带动



可以拿树莓派做什么

软件方面



计算集群



魔镜



交通指挥

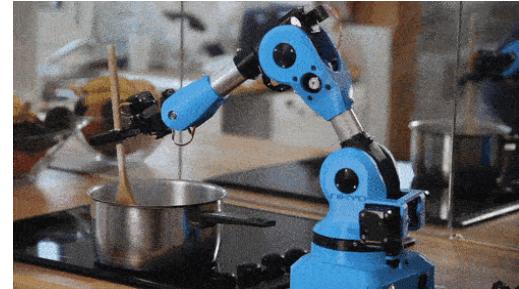


下载器/服务器



树莓派可以看作
“电脑” + “硬件控制器”的结合体

硬件方面



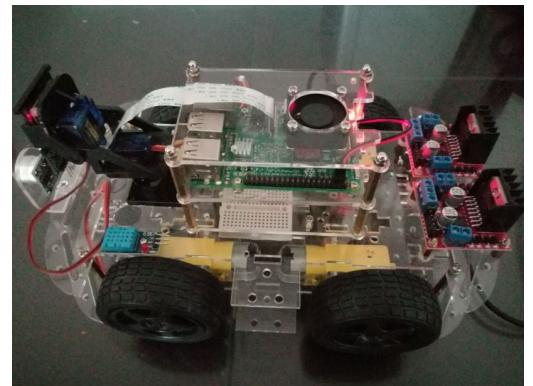
机械臂控制



PM2.5 监控



游戏机



智能小车

发现更多用法
<https://www.zhihu.com/question/20697024>
<http://shumeipai.nxez.com>

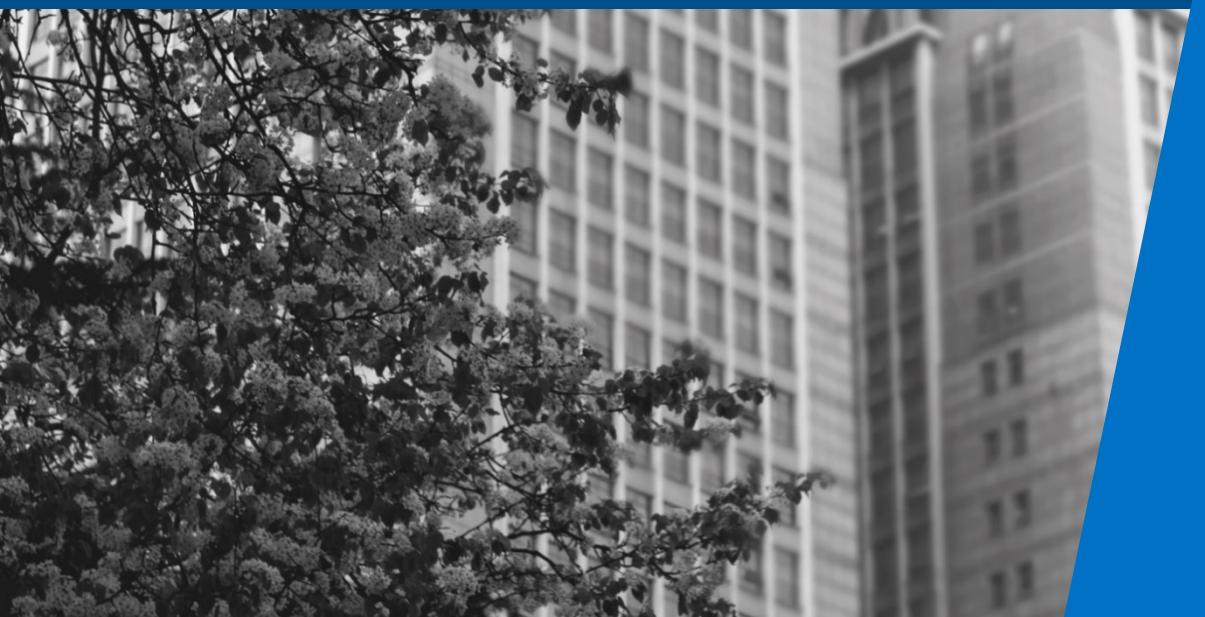


可以拿树莓派做什么





图片分类及应用

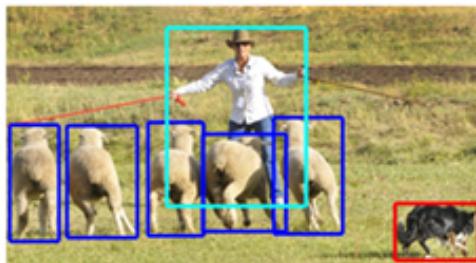




计算机视觉几种基本任务



(a) classification



图片分类任务

模型需要正确识别图片中的主要物体或场景

物体检测任务

模型不仅需要识别出物体，而且需要将其位置标示出来（“框出来”）

实例分割任务

模型不仅需要识别出物体，而且需要生成物体的掩膜（“描出来”）

难度递增
计算量递增



常用网络

适用于GPU等高性能设备

- VGG Net
- Inception Net
- ResNet
- DenseNet
-



适用于手持/小型/低性能设备

- MobileNet
- ShuffleNet
- SqueezeNet
-

大网络转小网络常用转换技巧

- 剪枝
- 量化
- 知识蒸馏
-



树莓派可用的深度学习框架

- Tensorflow
 - 无需自行编译，直接pip安装
 - Google团队开发
- PyTorch
 - 需要自行编译（尝试未成功）
 - Facebook团队开发
- Darknet
 - 直接编译即可（AlexyAB版本）
 - 有针对树莓派平台的优化
 - 个人开发，质量和更新不能完全保证

PYTORCH



如想自行编译，下面是参考笔记：

TensorFlow : <https://haoyu.love/blog503.html>

Darknet : <https://haoyu.love/blog511.html>



应用：Giorgio Cam



2016年，Google开发者大会上放出了一个名为
Giorgio Cam
的应用，可以将图片转化为简单的Rap



使用图片识别 API，为图片打标签
将标签转化为固定模板的歌词
用文本转语音（Text to Sound, TTS）引擎念出来
配合音乐旋律



这个Demo是基于在线 API /云服务 的
有兴趣的同学可以将它使用 树莓派+摄像头+显示屏+深度神经网络+PicоТTS 做成本地应用
需要注意模型大小，以权衡精度与速度

使用 Android 的同学请扫码体验





应用：神奇台灯



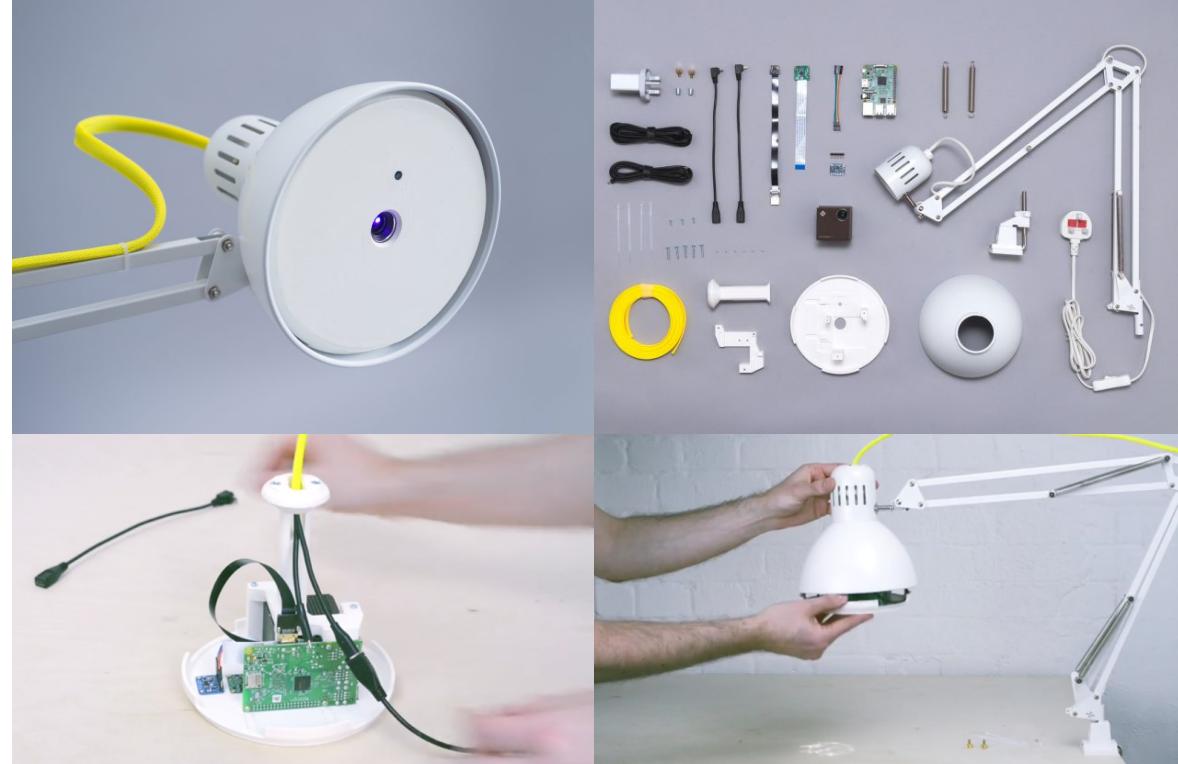
主要原料

- 台灯
- 树莓派
- 摄像头
- 投影仪



作用

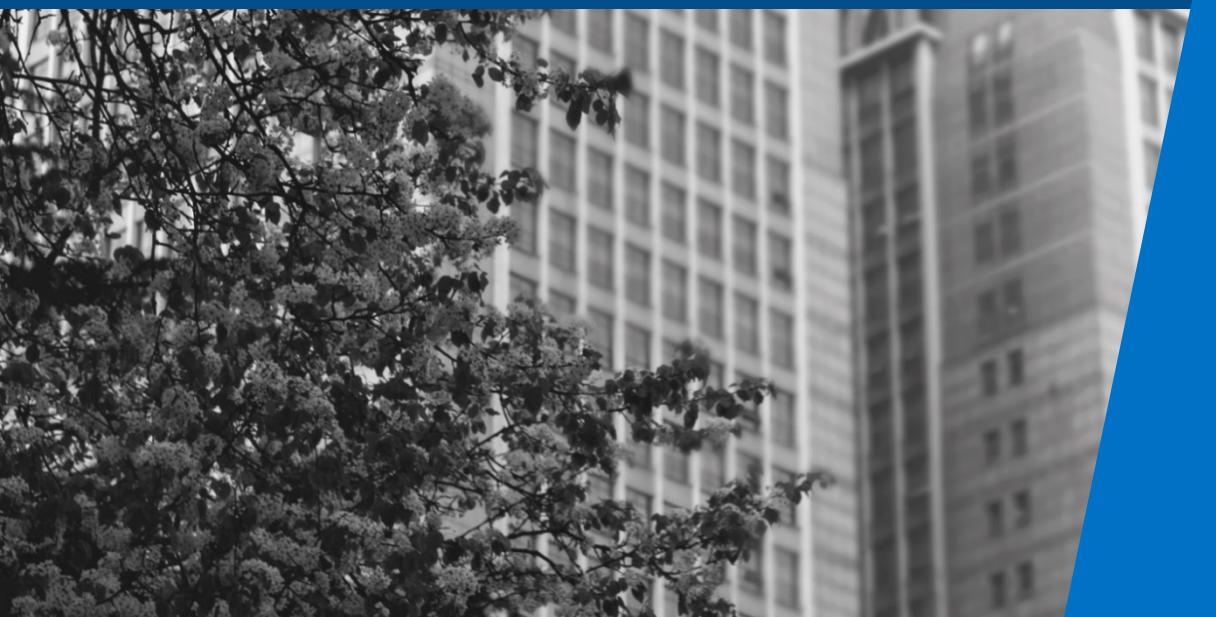
- 手影游戏
- 实体版“你画我猜”
- 手势翻书
- 隔空手势识别



2018年 Google 开发者大会上，除了手机版“猜画小歌”外，还有台灯实体版猜画小歌。装置即为此台灯，摄像头会实时捕捉绘画信息，并在树莓派上进行处理。有兴趣的同学可以进行尝试。需要注意的是训练数据收集难度较大。



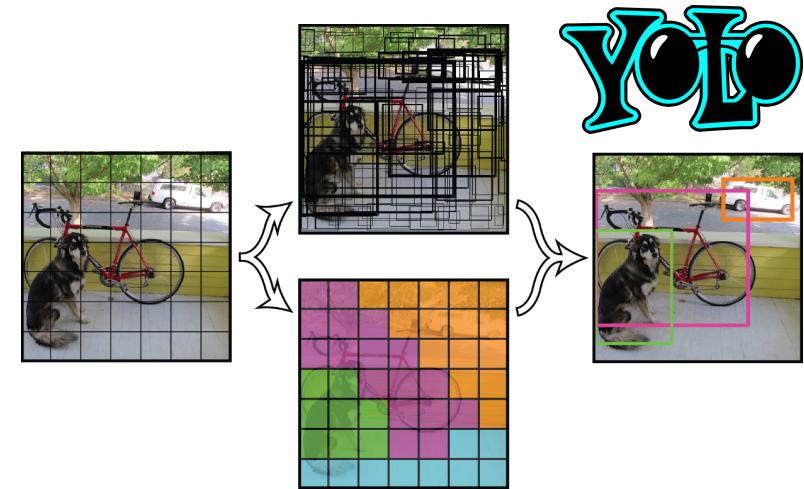
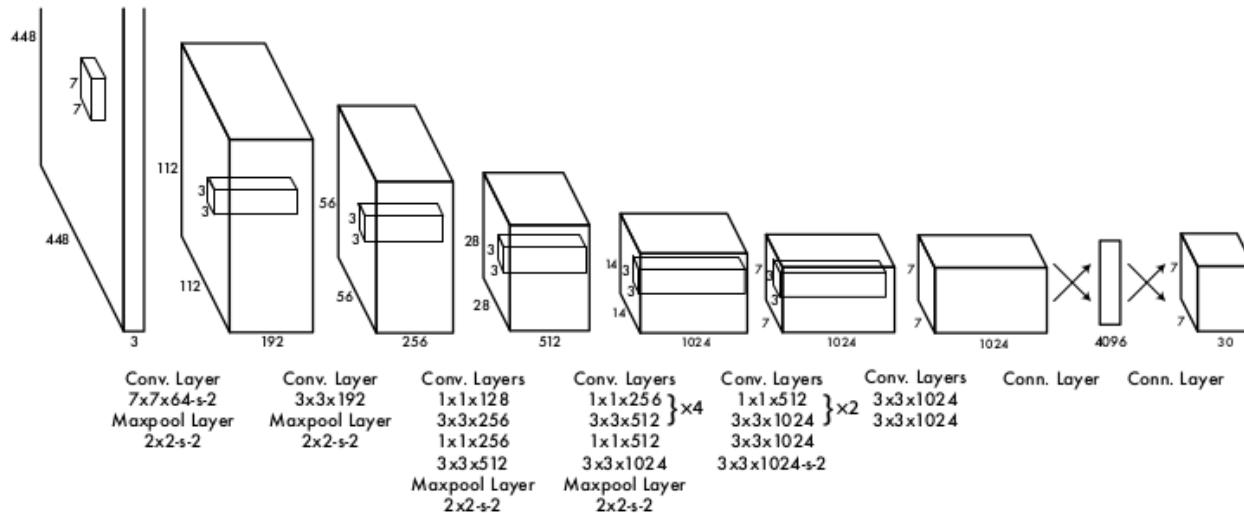
物休检测及应用





使用 YOLO 做物体检测

- 2014年，Ross Girshick 等人提出了基于深度神经网络的物体检测器 R-CNN。之后，经过一系列改进，有了Fast R-CNN和Faster R-CNN。
- 2016年，Joseph Redmon 提出了 You Only Look Once 物体检测器，相比于R-CNN，速度更快，使用GPU加速的 Tiny Yolo 可达到 200 帧/秒的处理速度





在树莓派上跑YOLO

在PC/服务器上做的事情

- 下载并编译Darknet
- 收集训练数据
- 定义网络，得到网络结构
- 训练网络，得到权重文件



在树莓派上做的事情

- 增加缓冲区 (Swap) 大小
- 下载并编译Darknet的树莓派特殊优化版本
- 加载网络结构，加载权重文件
- 进行图片分类/物体检测

由于树莓派性能较弱，所以一般用带有GPU的服务器进行网络训练，使用树莓派进行推理。

但即使如此，在树莓派等小型无GPU设备上跑神经网络依然有比较大的挑战，并且速度较慢。

除了使用特殊库（例如NNPack）进行加速外，我们还可以使用Intel神经计算棒来进行加速。

需要将它插入USB口，并配合相关驱动修改代码，流程略复杂。

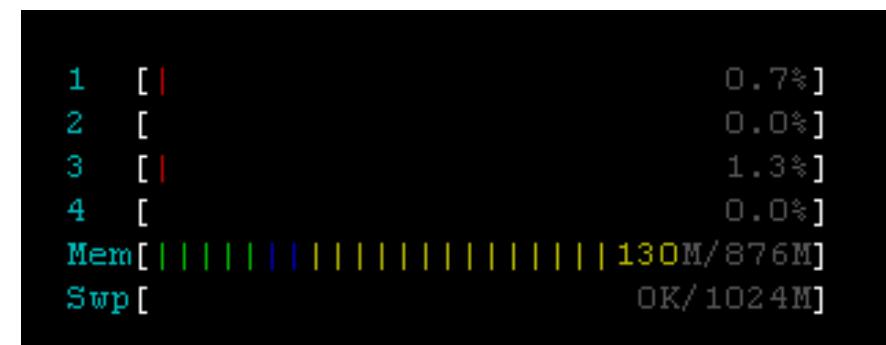


细节：增加Swap文件大小

- Swap文件相当于虚拟的内存
- 当内存不够用的时候，会把一部分数据临时放到Swap文件里
- Windows也有这个机制，叫“交换空间”
- 优点是超内存后程序不会崩溃，缺点是速度较慢

简要步骤：

- 打开 /etc/dphys-swapfile，找到 CONF_SWAPSIZE 变量
- 由默认的100改为 4096，SD卡小的话可以改为 1024。
- sudo /etc/init.d/dphys-swapfile stop
- sudo /etc/init.d/dphys-swapfile start





下载并编译 Darknet

- Darknet 有很多版本。在这里我们使用AlexyAB的版本。
- 如果想充分利用树莓派的多核心CPU以获得更高的速度，需下载并编译 darknet-nnpack 版本
- 注：树莓派是有GPU的，然而现在没有驱动程序支持此GPU做深度学习，也即GPU不可用，故只能使用CPU版本

简要步骤：

- git clone <https://github.com/AlexeyAB/darknet.git>
- cd darknet
- make all -j 5

树莓派可用的Github开源代码

Pjreddie/Darknet
AlexyAB/Darknet
digitalbrain79/darknet-nnpack
DarkFlow

各版本推理时间对比

框架	YOLO	Tiny YOLO
Darknet	560 s	25 s
Darknet-nnpack	15 s	1 s
TensorFlow	30 min	-



下载训练好的权重文件

- 如不想自己定义网络和训练模型，也可以直接使用现有模型
 - 这些模型未针对树莓派等设备做优化，速度比较慢
- 在 Joseph Redmon 的博客上可以下载到许多训练好的模型
- wget <https://pjreddie.com/media/files/yolov3-tiny.weights>
- 建议使用Tiny模型，一般十几秒可出结果
- 完整版YOLO需要检测一张图片需要5分钟左右

Performance on the COCO Dataset							
Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights

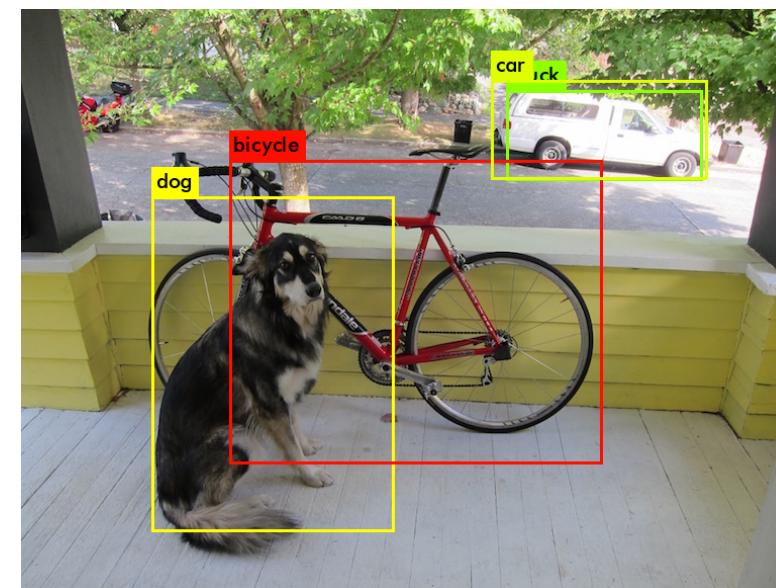


进行物体检测

- ./darknet detect cfg/yolov3-tiny.cfg yolov3-tiny.weights data/dog.jpg

```
layer    filters    size        input          output
  0 conv    16 3 x 3 / 1    416 x 416 x   3    ->  416 x 416 x   16 0.150 BF
  1 max      2 x 2 / 2    416 x 416 x   16    ->  208 x 208 x   16 0.003 BF
  2 conv    32 3 x 3 / 1    208 x 208 x   16    ->  208 x 208 x   32 0.399 BF
  3 max      2 x 2 / 2    208 x 208 x   32    ->  104 x 104 x   32 0.001 BF
  4 conv    64 3 x 3 / 1    104 x 104 x   32    ->  104 x 104 x   64 0.399 BF
  5 max      2 x 2 / 2    104 x 104 x   64    ->  52 x  52 x   64 0.001 BF
  6 conv    128 3 x 3 / 1    52 x  52 x   64    ->  52 x  52 x  128 0.399 BF
  7 max      2 x 2 / 2    52 x  52 x  128   ->  26 x  26 x  128 0.000 BF
  8 conv    256 3 x 3 / 1    26 x  26 x  128   ->  26 x  26 x  256 0.399 BF
  9 max      2 x 2 / 2    26 x  26 x  256   ->  13 x  13 x  256 0.000 BF
 10 conv   512 3 x 3 / 1    13 x  13 x  256   ->  13 x  13 x  512 0.399 BF
 11 max      2 x 2 / 1    13 x  13 x  512   ->  13 x  13 x  512 0.000 BF
 12 conv   1024 3 x 3 / 1   13 x  13 x  512   ->  13 x  13 x 1024 1.595 BF
 13 conv    256 1 x 1 / 1   13 x  13 x 1024  ->  13 x  13 x  256 0.089 BF
 14 conv    512 3 x 3 / 1   13 x  13 x  256   ->  13 x  13 x  512 0.399 BF
 15 conv    255 1 x 1 / 1   13 x  13 x  512   ->  13 x  13 x  255 0.044 BF
 16 yolo
 17 route   13
 18 conv    128 1 x 1 / 1   13 x  13 x  256   ->  13 x  13 x  128 0.011 BF
 19 upsample           2x   13 x  13 x  128   ->  26 x  26 x  128
 20 route   19 8
 21 conv    256 3 x 3 / 1   26 x  26 x  384   ->  26 x  26 x  256 1.196 BF
 22 conv    255 1 x 1 / 1   26 x  26 x  256   ->  26 x  26 x  255 0.088 BF
 23 yolo
Total BFLOPS 5.571
Loading weights from yolov3-tiny.weights...
seen 64
Done!
```

```
data/dog.jpg: Predicted in 25.380396 seconds.
dog: 57%
bicycle: 59%
car: 62%
truck: 56%
Not compiled with OpenCV, saving to predictions.png instead
```





人脸识别





使用OpenCV进行实时人脸检测和识别



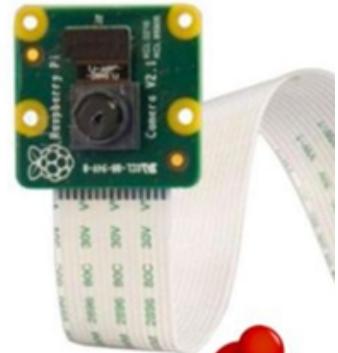
应用场景

人脸识别开门（张江计算机楼部分实验室正在使用）
人脸识别签到（VALSE 2018 会场使用人脸签到系统）
人脸门票（部分景点已上线人脸门票闸机系统）
.....



基本步骤

1. 硬件：加装摄像头
2. 环境：安装OpenCV
3. 代码：测试人脸检测
4. 代码：录入人脸，并进行识别





加装摄像头模块

专用摄像头模块

- 树莓派专用
- 可定制性强，价格适中
 - RGB
 - 红外线
 - 鱼眼
 - 补光灯
- 不能直接被 OpenCV 识别
- 安装在专用摄像头插座上，适合集成用



通用USB摄像头

- PC / 树莓派通用，兼容性好
- 可以直接被 OpenCV 识别
- 适合实验用





配置实验环境

项目	配置
树莓派	3B / 3B+
系统	Raspbian Stretch with desktop
摄像头	罗技 USB 摄像头
OpenCV	3.4.1
Python	3.5.3



简要步骤和代码：

- 将系统烧录到SD卡上，装入树莓派
- 系统升级
 - apt update; apt upgrade -y;
- 安装 OpenCV 及其依赖
 - sudo apt install libopencv-dev ;
 - sudo apt install libatlas-base-dev ;
 - sudo apt install libjasper-dev ;
 - sudo apt install qt4-dev ;
 - sudo pip3 install opencv-python;





测试摄像头

```
import cv2

cap = cv2.VideoCapture(0)          # 使用第 0 号摄像头

while True:
    ret, frame = cap.read()       # 获取摄像头画面
    cv2.imshow('frame', frame)    # 显示出来
cap.release()
cv2.destroyAllWindows()
```





进行人脸检测

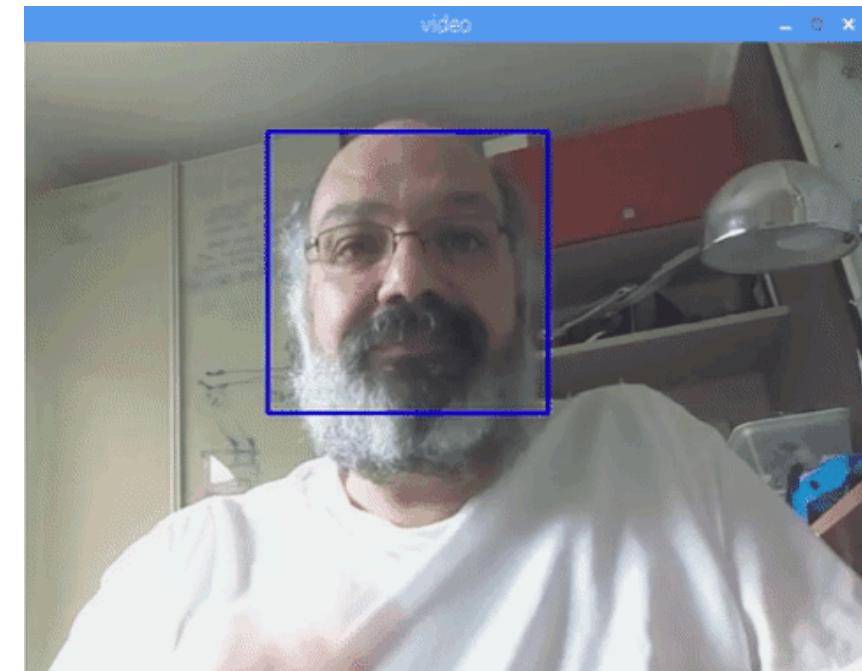
- 这里使用 OpenCV 内置的 Haar 级联分类器 进行人脸检测

```
import cv2

# 加载检测配置文件
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0) # 使用第 0 号摄像头

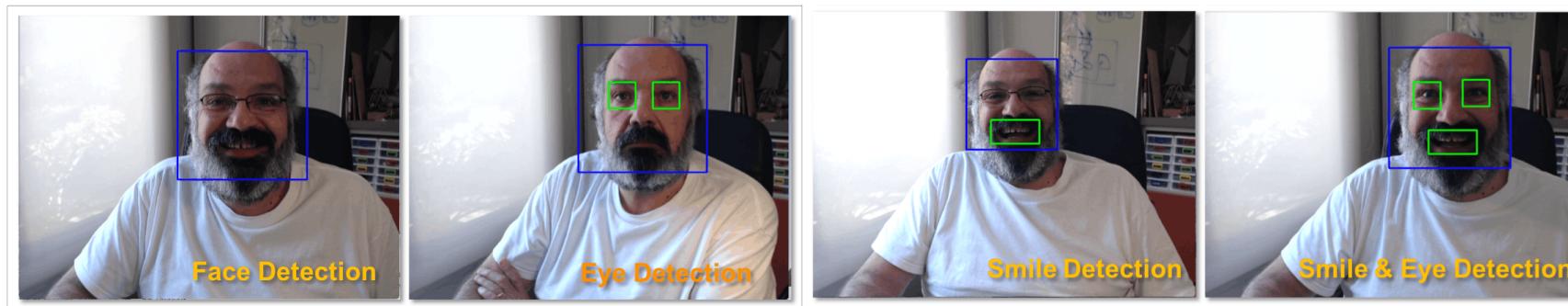
while True:
    ret, img = cap.read() # 获取摄像头画面
    faces = faceCascade.detectMultiScale( # 进行检测
        cv2.cvtColor(img, cv2.COLOR_BGR2GRAY), # 只接受灰度图
        scaleFactor=1.2, # 一些参数
        minNeighbors=5,
        minSize=(20, 20)
    )
    for (x,y,w,h) in faces: # 将检测结果框出来
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    cv2.imshow('video',img)
cap.release()
cv2.destroyAllWindows()
```





进行人脸检测

- 配置文件有很多，请大家自行尝试。
 - 检测眼睛、嘴巴、鼻子
- 可以对检测及结果进行叠加。
- 考虑到树莓派的运算性能，不建议超过 3 个检测器同时工作





进行人脸识别

1

采集人脸数据

每个人约采集30张图片

2

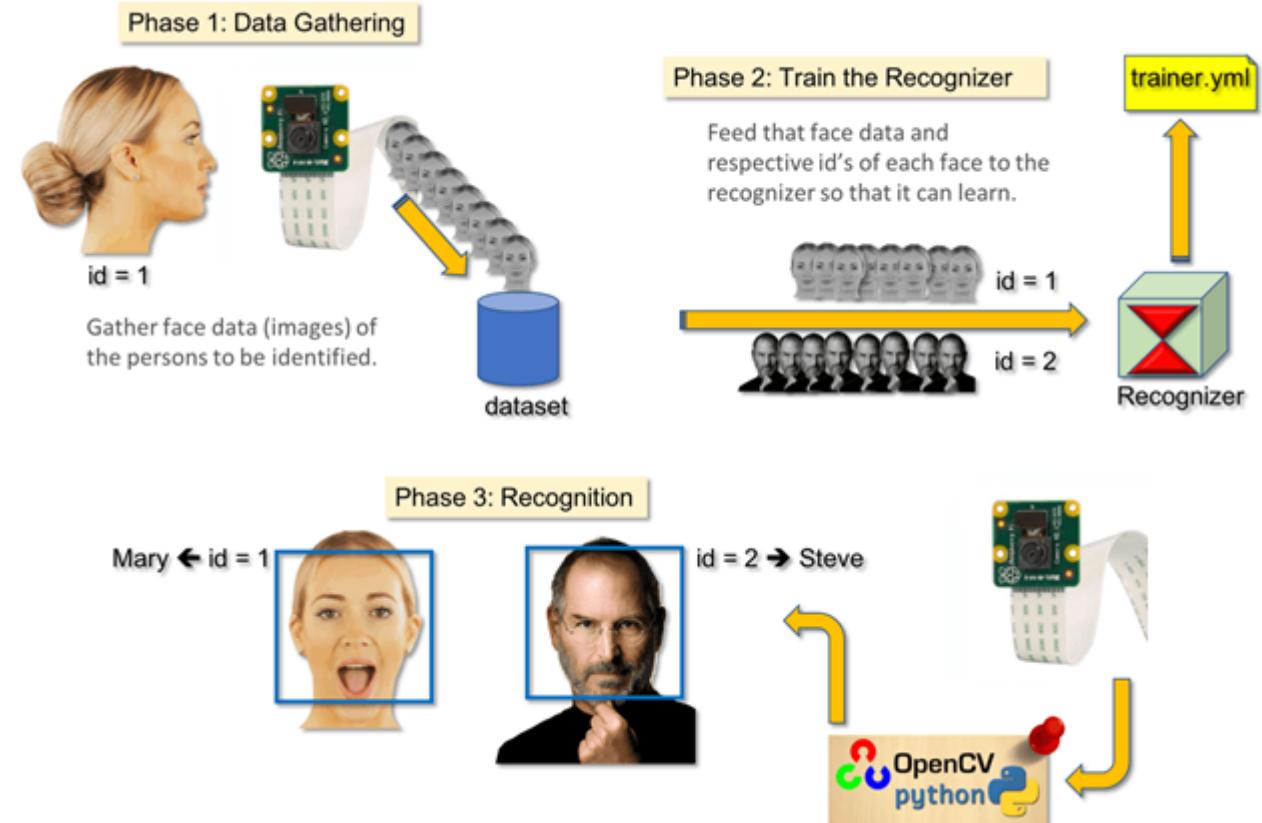
训练人脸分类器

得到人脸识别模型
建议在PC上进行

3

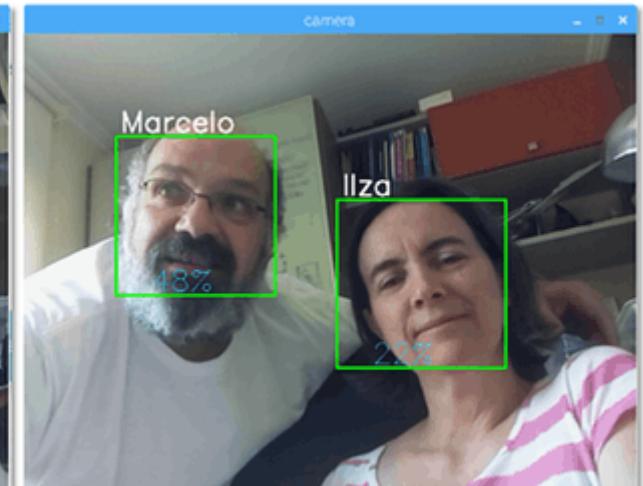
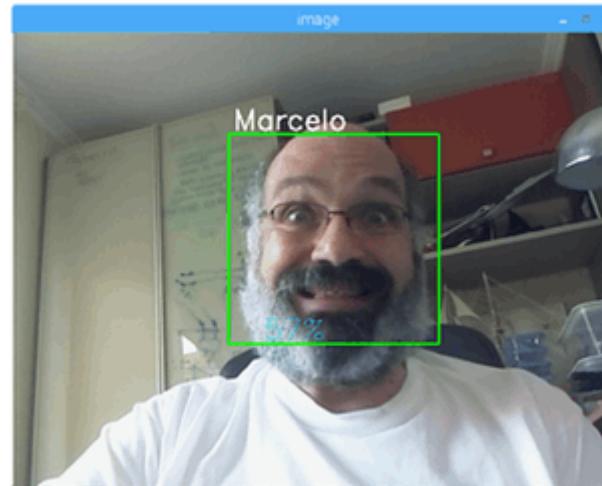
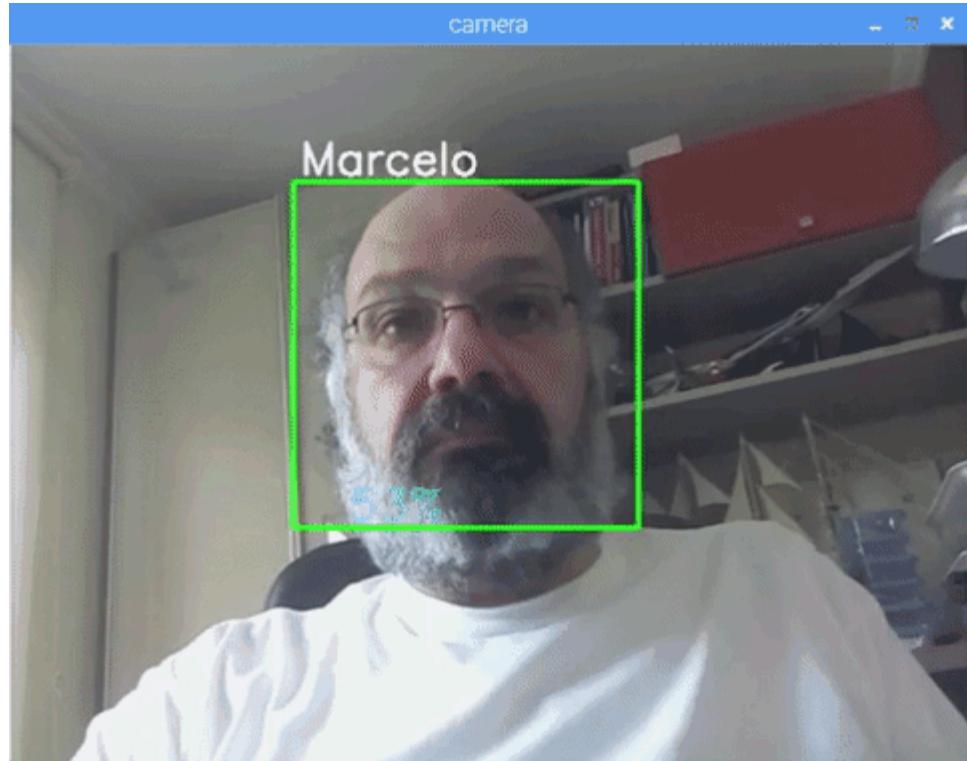
进行人脸识别

首先检测出脸部区域
然后识别是谁





进行人脸识别





其他综合应用

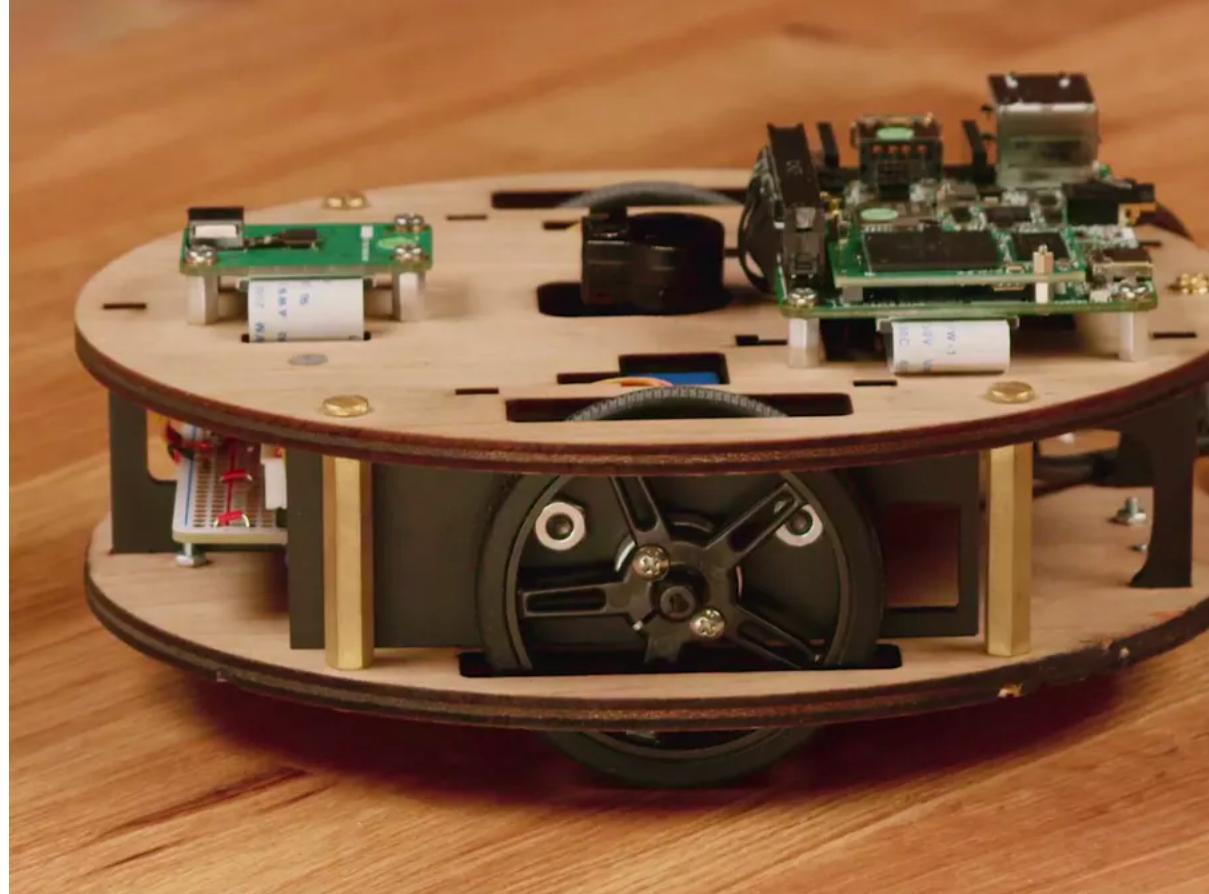




画图机器人（见视频）

基本原理与流程

- 摄像头采集图像
- 图像做降采样
- 图像做量化（本例中为4值量化）
- 小车扫描每一行
- 步进电机指导马克笔画图



此图中，开发版为AndroidThings专用开发版，不过树莓派也可以使用此系统，且二者GPIO引脚定义相同



魔镜魔镜

基本原理与流程

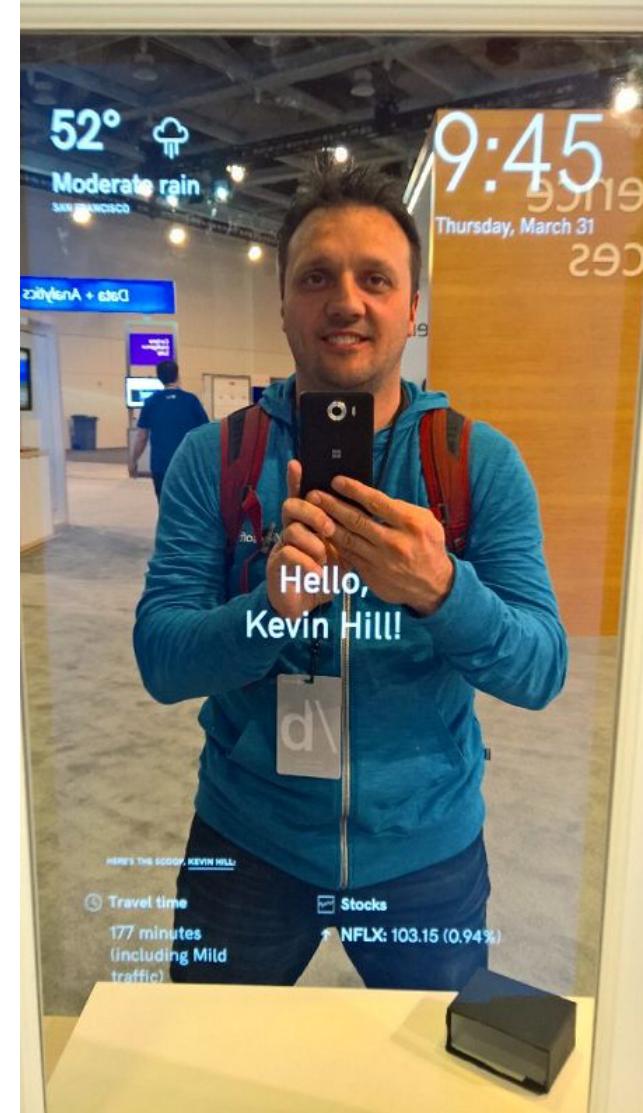
材料：

- 单面透镜（电影中“审讯室”常见的那种）
- 显示器
- 树莓派

流程：

- 显示器放在单面透镜后面
- 摄像头人脸识别互动（例如图中“Hello Hill！”）
- 声音识别进行互动（例如，“魔镜魔镜”）
- 设计显示样式（天气、时间、股票、日程等）

这是树莓派爱好者中最受欢迎的玩法





智能小车

基本原理与流程

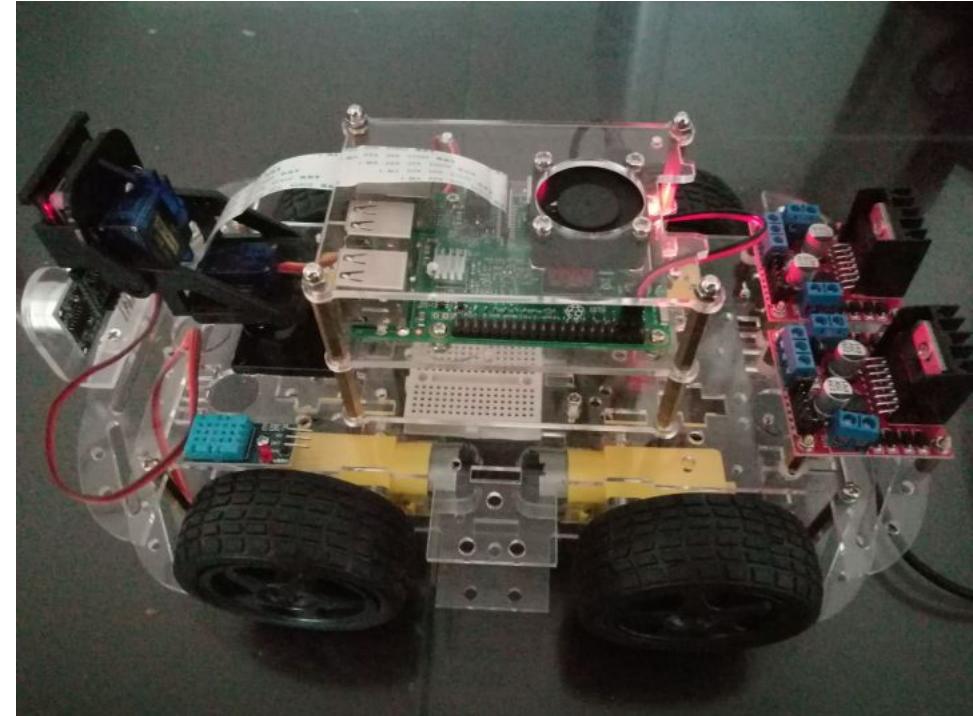
材料：

- 摄像头
- 寻迹模块
- 继电器
- 马达

流程：

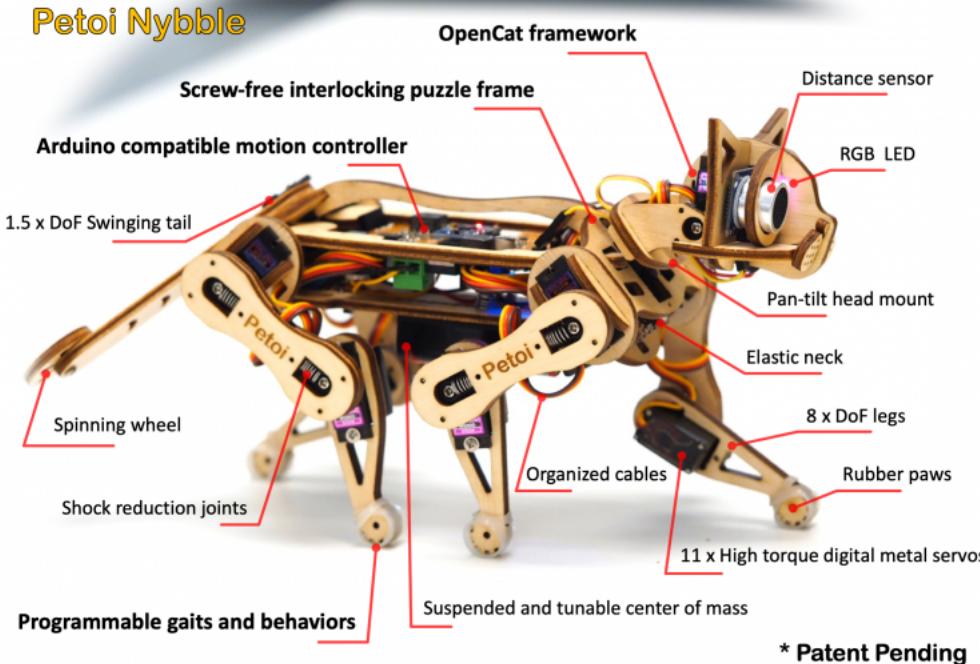
- 摄像头/寻迹模块获取当前道路信息
- 树莓派处理
- GPIO控制继电器，以驱动马达转动

这是树莓派爱好者中的常见玩法





机器人



基本原理与流程

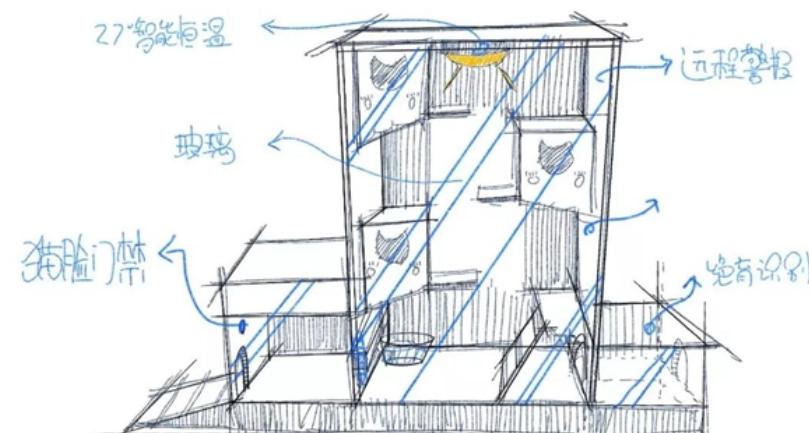
树莓派通过 GPIO 向各继电器发出指令，控制对应的步进电机（“关节”）转动，以使得机器人作出各种动作



人工智能猫窝

基本功能

1. 使用了x86平台
2. 刷脸开门
3. 温度传感器和加热系统
 - 保持27度恒温
4. 湿度传感器
5. 新风系统
 - 监测O₂、CO₂浓度并换气
6. 判断猫是否患病
7. 判断猫是否绝育
8. 远程报警





更多资料

- 知乎：有哪些对树莓派的有趣改造和扩展应用
 - <https://www.zhihu.com/question/20697024>
- 树莓派实验室
 - <http://shumeipai.nxez.com/>
- 官方网站
 - <https://www.raspberrypi.org>
 - 其官方微博也经常发布各种新颖的用法
- 避坑：尽量避免购买所谓“树莓派专用”硬件



THANKS



THANKS



平行信息罗列

Input your information here. Make it longer than the title..

插入

项目名称

一段描述的语言，长一点，超过一行会比较好
看。

相关

项目名称

一段描述的语言，长一点，超过一行会比较好
看。

图片

项目名称

一段描述的语言，长一点，超过一行会比较好
看。



排位信息罗列

Input your information here. Make it longer than the title..

1

简单的描述写在这里

一段描述的语言，不要太长，但是超过一行会比较美观。

3

简单的描述写在这里

一段描述的语言，不要太长，但是超过一行会比较美观。

2

简单的描述写在这里

一段描述的语言，不要太长，但是超过一行会比较美观。

4

简单的描述写在这里

一段描述的语言，不要太长，但是超过一行会比较美观。



案例理论结合

Input your information here. Make it longer than the title..



在左边插入一个合适的ICON，然后在这里写一段详细的描述，展示出这个案例的内容和基本结构。甚至你还可以在右边插入一张案例相关的图片。



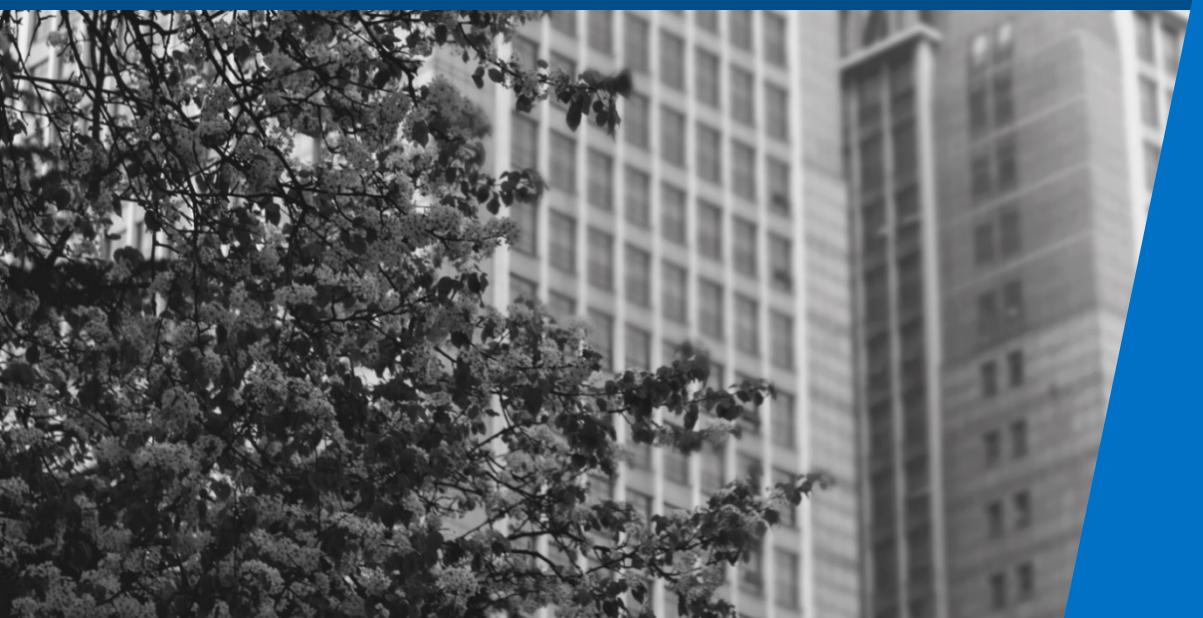
在左边插入一个合适的ICON，然后在这里写一段详细的描述，展示出这个案例的内容和基本结构。甚至你还可以在右边插入一张案例相关的图片。

理论对接

这里可以写清楚案例可以表达的理论，名称、概念、基本要素等等信息都可以写在这里。



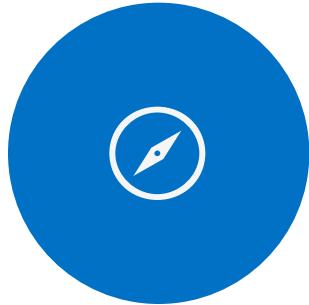
这里填写小标题





总结类型罗列

Input your information here. Make it longer than the title..



在这里输入小标题

在上面请插入和主题相关的ICON，用Google搜索关键词+ICON，运用高级搜索的图片筛选功能就可以找到没有北京的PNG图片了。

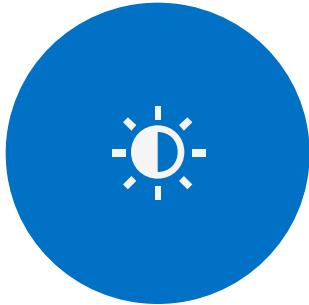
下面可以简单介绍这个标题下面的内容，用来引领或者回顾都是不错的选择。



在这里输入小标题

在上面请插入和主题相关的ICON，用Google搜索关键词+ICON，运用高级搜索的图片筛选功能就可以找到没有北京的PNG图片了。

下面可以简单介绍这个标题下面的内容，用来引领或者回顾都是不错的选择。



在这里输入小标题

在上面请插入和主题相关的ICON，用Google搜索关键词+ICON，运用高级搜索的图片筛选功能就可以找到没有北京的PNG图片了。

下面可以简单介绍这个标题下面的内容，用来引领或者回顾都是不错的选择。



在这里输入小标题

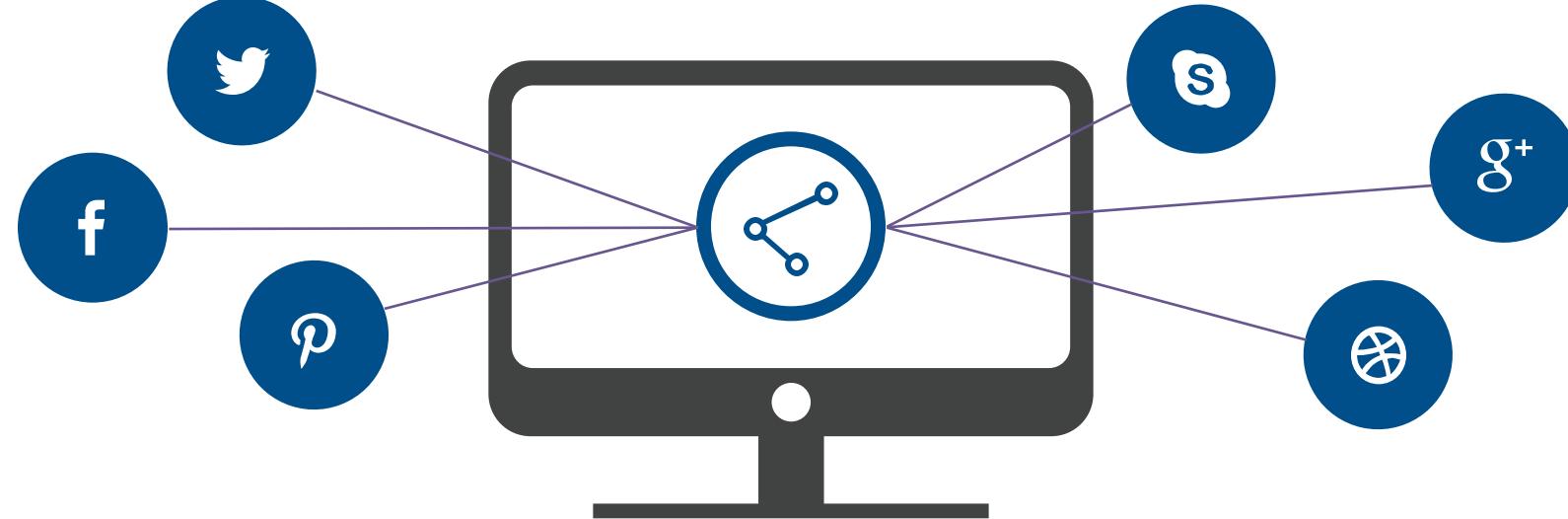
在上面请插入和主题相关的ICON，用Google搜索关键词+ICON，运用高级搜索的图片筛选功能就可以找到没有北京的PNG图片了。

下面可以简单介绍这个标题下面的内容，用来引领或者回顾都是不错的选择。



相关信息罗列

Input your information here. Make it longer than the title..

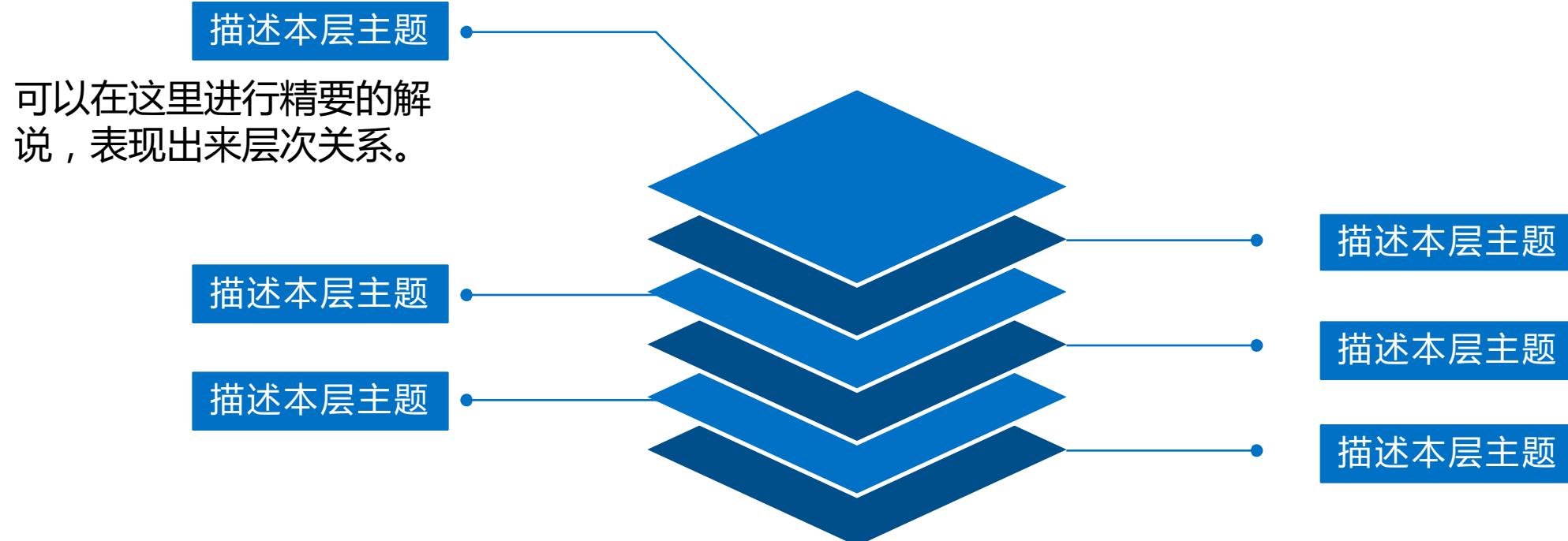


把白色的ICON换成主题相关的即可，Google图片能帮助你很快地找到它们。



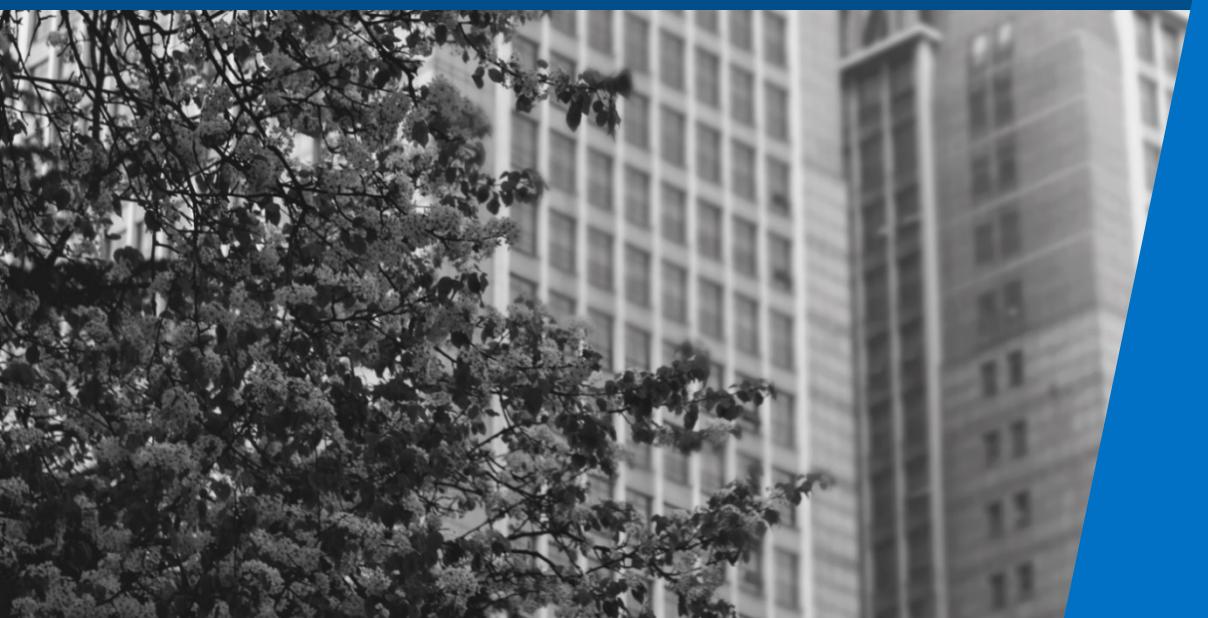
表里信息排列

Input your information here. Make it longer than the title..





这里填写小标题



3

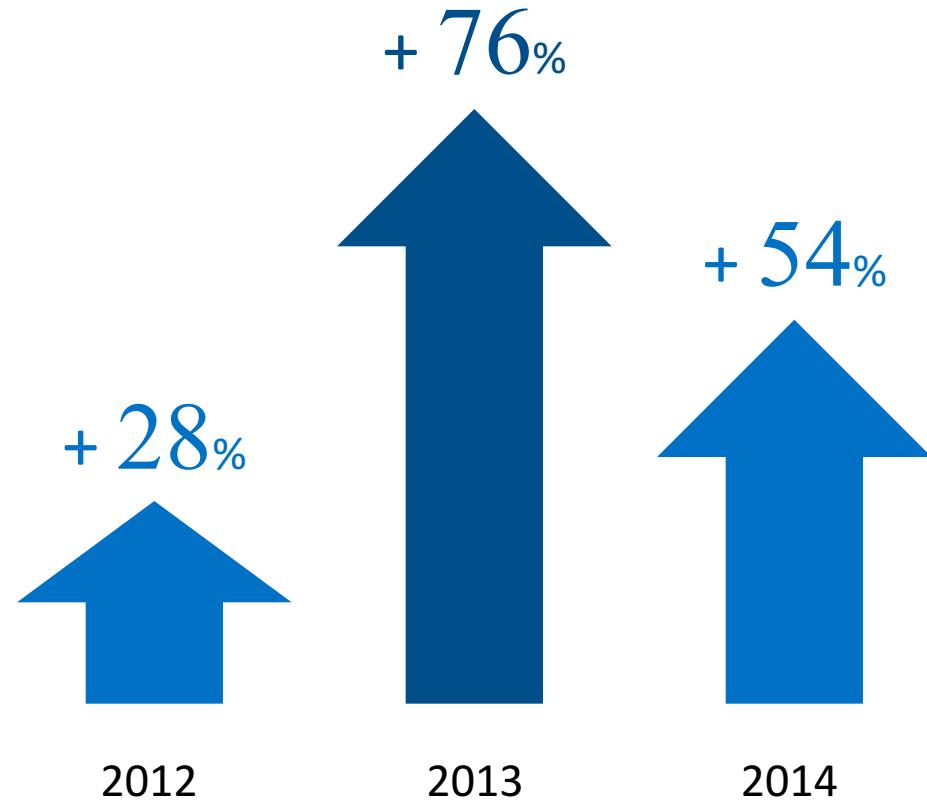


增长型柱状图

Input your information here. Make it longer than the title..

在这里输入图表主题

你可以改动数字，拉长压短箭头，或者复制粘贴产生新的箭头来表现更多的数据。





性别区分型调查分异

Input your information here. Make it longer than the title..



在这里插入男性对于话题的基本描述

你可以拖拽条形的长度，修改数字的大小来使之符合主题的需要。

在这里插入男性对于话题的基本描述

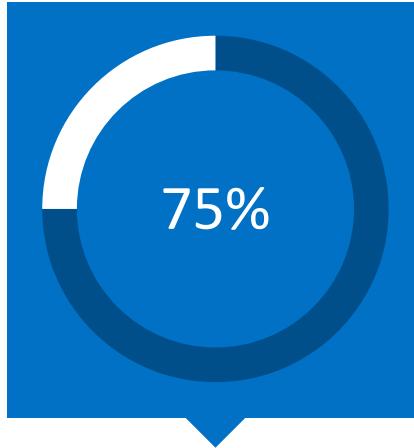
你可以拖拽条形的长度，修改数字的大小来使之符合主题的需要。





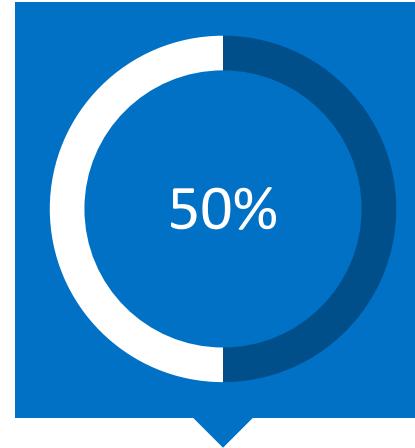
百分比型数据对比展示

Whoever evaluates your text cannot evaluate the way you write.



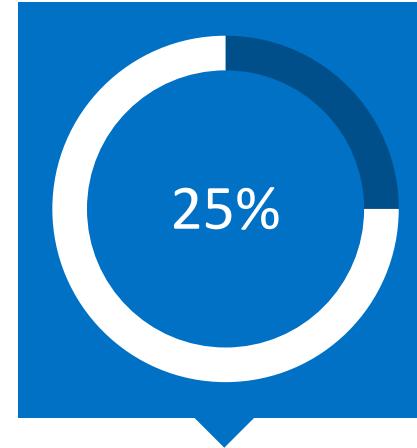
输入小主题

点击图表，选择「设计」
选项卡中的「编辑数据」
按钮即可修改数据。.



输入小主题

点击图表，选择「设计」
选项卡中的「编辑数据」
按钮即可修改数据。.



输入小主题

点击图表，选择「设计」
选项卡中的「编辑数据」
按钮即可修改数据。.



这里填写小标题





同一事物内部百分比展示

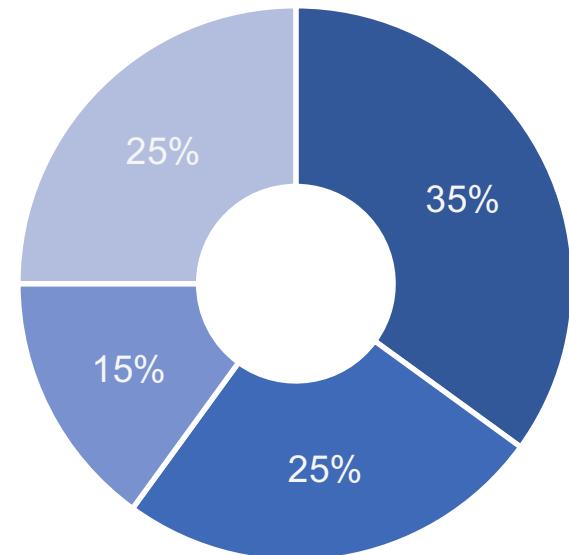
Whoever evaluates your text cannot evaluate the way you write.

在这里输入简短的主题描述

选中右边的饼状图，按照上一页的方法即可修改其中的数据使之更符合主题的需要。

下方的图注也是可以手动修改的，调节颜色记得根据右边的图表颜色一致哟。

● 2014 ● 2015 ● 2016 ● 2017





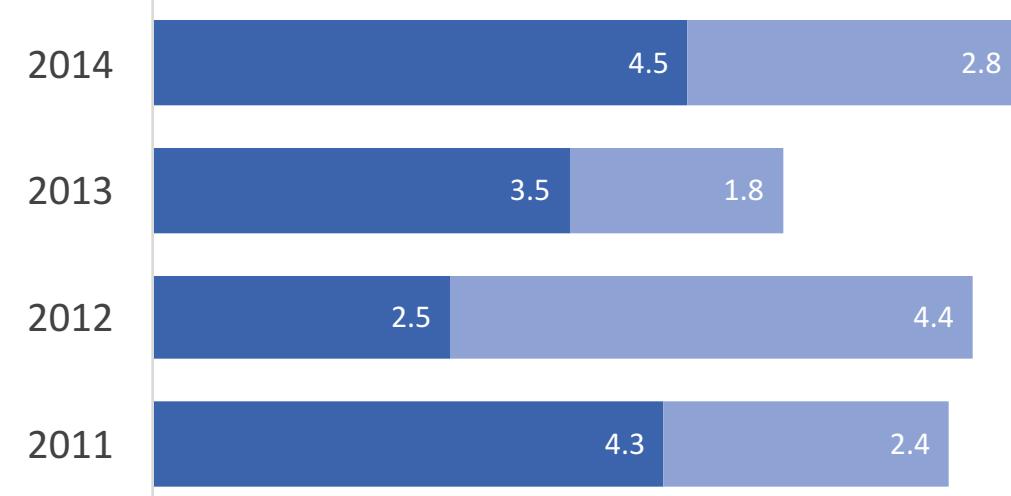
柱状图的纵向对比

Whoever evaluates your text cannot evaluate the way you write.

在这里输入小主题

同样地，右边是可以利用表格功能调整数据的柱状图。

简单的数据比对用这个模板可能会展示得更为清晰。





柱状图的横向对比

Whoever evaluates your text cannot evaluate the way you write.

1 输入主题



输入主题 1

2 输入主题



输入主题 2

3 输入主题



输入主题 3

4 输入主题



输入主题 4



THANKS