树莓派安装 openCV

1、树莓派安装 python-opencv

树莓派自带 python2 和 python3，不需要在安装 python,直接安装 OpenCV 即可，采取简单的方案安装 OpenCV，避免复杂的源码编译。

1、更新树莓派系统

sudo apt-get update

sudo apt-get upgrade

2、安装 python－OpenCV

sudo apt-get install libopencv-dev

sudo apt-get install python-opencv

该种安装方式不知道安装那个版本的 OpenCV

3、测试 opencv

终端命令行打印 OpenCV 的版本号，本教程资料会增加一个使用 OpenCV 获取 usb 摄像头实时视频流的程序。该种 OpenCV 的安装方式，比较简单和快速，适合刚上手树莓派 OpenCV 的读者。

```
E: Couldn't find any package by regex 'python-opencv-3.4.0'
pi@raspberrypi:~ $ python
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'2.4.9.1'
>>>
```

2、源码编译方式安装 OpenCV

源码编译安装 OpenCV 比较繁琐，安装之后的 OpenCV 可以采用多种语言进行开发，是一种全面的安装，安装过程费时费力，后续会有专门一节讲述。

源码安装 opencv+opencv_contrib 编译

树莓派源码安装 opencv+opencv_contrib 编译,源码安装时间较久大家编译的时候一定要用质量好的电源，并把风扇插上，防止编译时因为亏电重启或者烧坏树莓派。

一、首先我们要安装 OpenCV 所依赖的各种工具和图像视频库

打开终端（Raspbain 和 Ubuntu 的快捷键都是 Ctrl + Alt +t），然后依次执行下列命令，网上很多教程都是放到一起执行的，但是一起执行往往会出现错误，也不好查错。

1.软件源更新

sudo apt-get update // 升级本地所有安装包，最新系统可以不升级

sudo apt-get upgrade// 升级树莓派固件

2.安装构建 OpenCV 的相关工具：

// 安装 build-essential、cmake、git 和 pkg-config

sudo apt-get install build-essential cmake git pkg-config

3.安装常用图像工具包：

// 安装 jpeg 格式图像工具包

sudo apt-get install libjpeg8-dev // 安装 tif 格式图像工具包

sudo apt-get install libtiff5-dev // 安装 JPEG-2000 图像工具包

sudo apt-get install libjasper-dev // 安装 png 图像工具包
sudo apt-get install libpng12-dev
4.安装视频 I/O 包（注意最后一个包的数字“4”后面是“L”）：
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
5.安装 gtk2.0
树莓派很可能下载错误，更换中科大或者清华源即可，ubuntu 有可能出现包版本过高的情况，需要将依赖包降级安装
sudo apt-get install libgtk2.0-dev （务必安装该包后再编译 opencv）
sudo apitude install libgtk2.0-dev
查看有没有安装成功
dpkg --listfiles libgtk2.0-dev #dpkg --listfiles package
6.优化函数包：
sudo apt-get install libatlas-base-dev gfortran

二、获取 OpenCV 源代码
恭喜你，执行到这里就把 OpenCV 的依赖包全部安装好了，之后要开始编译 OpenCV 源代码了，请大家用 wget 工具下载到用户目录下（源码要放在有执行权限的位置，不是安装位置），命令如下：
1、使用 wget 获取 OpenCV 源码
// 下载 OpenCV3.4.0
wget -O opencv-3.4.0.zip https://github.com/Itseez/opencv/archive/3.4.0.zip
// 解压 OpenCV
unzip opencv-3.4.0.zip
2、使用 wget 获取 OpenCV_contrib 库
// 下载 OpenCV_contrib 库：
wget -O opencv_contrib-3.4.0.zip https://github.com/Itseez/opencv_contrib/archive/3.4.0.zip
// 解压 OpenCV_contrib 库：
unzip opencv_contrib-3.4.0.zip
三、源码方式编译 OpenCV
1、进入源码文件夹
cd opencv-3.4.0
之后我们新建一个名为 release 的文件夹用来存放 cmake 编译时产生的临时文件：
2、 新建 release 文件夹
mkdir release
3、进入 release 文件夹
cd release
4、设置 cmake 编译参数
安装目录默认为/usr/local ，注意参数名、等号和参数值之间不能有空格，但每行末尾“\”之前有空格，参数值最后是两个英文的点：
编译参数如下：
/** CMAKE_BUILD_TYPE 是编译方式
* CMAKE_INSTALL_PREFIX 是安装目录
* OPENCV_EXTRA_MODULES_PATH 是加载额外模块
* INSTALL_PYTHON_EXAMPLES 是安装官方 python 例程

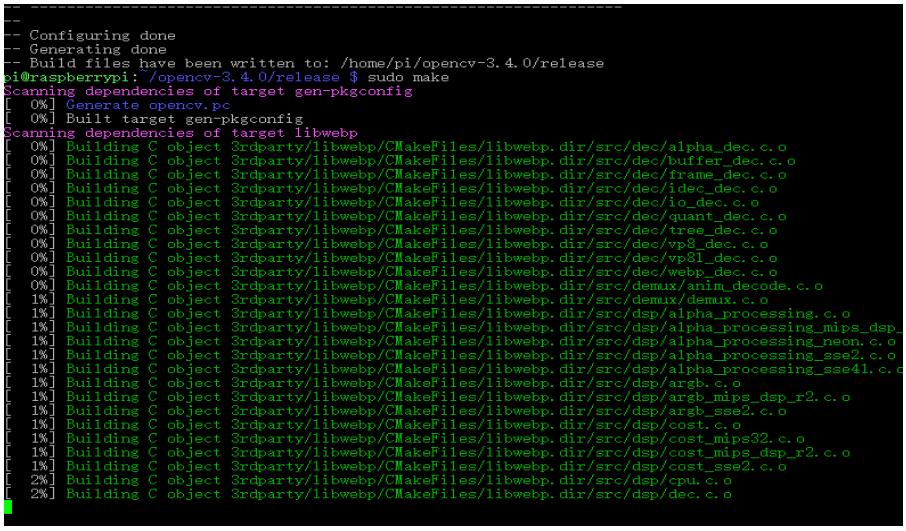
* BUILD_EXAMPLES 是编译例程（这两个可以不加，不加编译稍微快一点点，想要 C 语言的例程的话，在最后一行前加参数 INSTALL_C_EXAMPLES=ON \）
**/
编译指令如下：

```
sudo cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.4.1/modules \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=ON ..
```

之后开始正式编译过程。
5、编译器设置检查完毕进入编译

```
// 编译，以管理员身份，否则容易出错
sudo make// 安装
sudo make install// 更新动态链接库
sudo ldconfig
```

到这里，OpenCV 的编译完成，已经可以正常使用了。

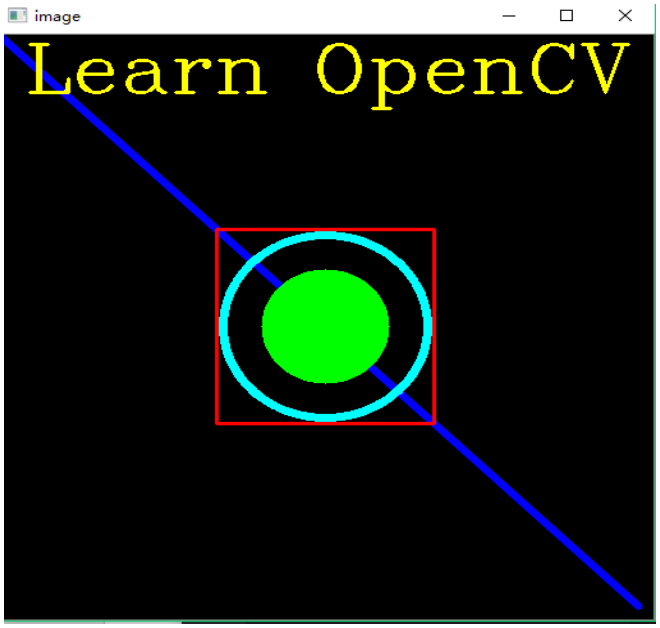


四、代码测试
1、python 测试：
下面我提供一个 Python 语言的测试程序，用来测试 OpenCV 是否正常：

```
#-*-coding:UTF-8 -*-
# @author: zdl
'''
使用 OpenCV 函数绘制常用图形
在一幅图片上画线，画圆，画矩形和文字。
'''

import cv2
import numpy as np

img = np.zeros((512,512,3), dtype=np.uint8)          #创建一幅图像
```

```
cv2.line(img,(0,0),(500,500),(255,0,0),5)          #绘制直线
cv2.circle(img,(255,255),50,(0,255,0),-1)          #绘制填充圆
cv2.circle(img,(255,255),80,(255,255,0),5)         #绘制非填充圆
cv2.rectangle(img,(170,170),(340,340),(0,0,255),2) #绘制矩形

# 绘制文本
cv2.putText(img,'Learn
OpenCV',(20,50),cv2.FONT_HERSHEY_COMPLEX,2,(0,255,255),2)
cv2.imshow('image', img)     #显示图像

#等待释放窗口
cv2.waitKey(0)
cv2.destroyAllWindows()
```

运行之后，显示的图片如下：



2、C++测试：

采用 GCC 编译器编译.cpp 文件

编译指令：测试是否安装成功,你可以使用以下的命令行编译位于源代码包中的 test.c 例子:

g++ test.c `pkg-config opencv --libs --cflags opencv` -o test

运行指令：sudo ./test ball.jpg

测试代码：

test.cpp

```
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <iostream>
using namespace cv;
```

```
using namespace std;
int main (int argc, char **argv)
{
	Mat image, image_gray;
	image = imread(argv[1], CV_LOAD_IMAGE_COLOR );//执行需要外部传入一个参数
	if (argc != 2 || !image.data) {
		cout << "No image data\n";
		return -1;
	}

	cvtColor(image, image_gray, CV_RGB2GRAY);
	namedWindow("image", CV_WINDOW_AUTOSIZE);
	namedWindow("image gray", CV_WINDOW_AUTOSIZE);

	imshow("image", image);
	imshow("image gray", image_gray);

	waitKey(0);
	return 0;
}
```