OpenCV4.0 安装记录:

系统 : Ubuntu18.04.02

安装方法 : 编译安装

教程链接 : https://www.learnopencv.com/install-opencv-4-on-ubuntu-18-04/
该教程将提供一个 bash 脚本,用于安装 OpenCV4.0,同时将安装 C++和 Python3.6. 注意,本教程 采用的脚本将 OpenCV 安装在本地目录,而非整个系统.

1.安装 OpenCV4.0

打开 Ubuntu 终端,选择一个你想安装 OpenCV 的目录,进入目录,开始运行以下教程及代码.本样例,将 OpenCV 安装在 /home/name/ 目录下

步骤 0：选择要安装的 OpenCV 版本

```
echo "OpenCV installation by learnOpenCV.com"
#定义要安装的 OpenCV 版本（选择要安装的 OpenCV 版本,OpenCV4.0 代号为"master"）
cvVersion="master"
```

我们还将清理 `build` 目录并创建 `installation` 目录。

```
# 清理 build 目录
rm -rf opencv/build
rm -rf opencv_contrib/build
# 创建安装目录
mkdir installation
mkdir installation/OpenCV-"$cvVersion"
```

最后，我们将当前工作目录存储在 `cwd` 变量中。我们还将在此博客中将此目录称为 **OpenCV_Home_Dir,**即安装的文件夹.

```
# 保存当前的工作目录
cwd=$(pwd)
```

第 1 步：更新包

```
sudo apt  -y update
sudo apt -y upgr ade
```

第 2 步: 安装 OS 库

```
sudo apt -y remove x264 libx264-dev
## 安装依赖项
sudo apt -y install build-essential checkinstall cmake pkg-config yasm
sudo apt -y install git gfortran
sudo apt -y install libjpeg8-dev libpng-dev

sudo apt -y install software-properties-common
sudo add-apt-repository "deb http://security.ubuntu.com/ubuntu xenial-security main"
sudo apt -y update

sudo apt -y install libjasper1
sudo apt -y install libtiff-dev

sudo apt -y install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev
sudo apt -y install libxine2-dev libv4l-dev
cd /usr/include/linux
sudo ln -s -f ../libv4l1-videodev.h videodev.h
cd "$cwd"

sudo apt -y install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
sudo apt -y install libgtk2.0-dev libtbb-dev qt5-default
sudo apt -y install libatlas-base-dev
sudo apt -y install libfaac-dev libmp3lame-dev libtheora-dev
sudo apt -y install libvorbis-dev libxvidcore-dev
sudo apt -y install libopencore-amrnb-dev libopencore-amrwb-dev
sudo apt -y install libavresample-dev
sudo apt -y install x264 v4l-utils

# 可选的依赖项
sudo apt -y install libprotobuf-dev protobuf-compiler
sudo apt -y install libgoogle-glog-dev libgflags-dev
sudo apt -y install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen
```

第 3 步: 安装 Python 库

```
sudo apt -y install python3-dev python3-pip
sudo -H pip3 install -U pip numpy
sudo apt -y install python3-testresources
```

我们还将安装 virtualenv 和 virtualenvwrapper 模块来创建 Python 虚拟环境

```
cd $cwd
############ For Python 3 ############
# 创建虚拟环境
python3 -m venv OpenCV-"$cvVersion"-py3
echo "# Virtual Environment Wrapper" >> ~/.bashrc
```

```
echo "alias workoncv-$cvVersion=\"source $cwd/OpenCV-$cvVersion-py3/bin/activate\"" >>
~/.bashrc
source "$cwd"/OpenCV-"$cvVersion"-py3/bin/activate

# 现在在这个虚拟环境中安装 python 库
pip install wheel numpy scipy matplotlib scikit-image scikit-learn ipython dlib

# 退出虚拟环境
deactivate
```

第 4 步: 下载 opencv 和 opencv_contrib

```
git clone https://github.com/opencv/opencv.git
cd opencv
git checkout $cvVersion
cd ..

git clone https://github.com/opencv/opencv_contrib.git
cd opencv_contrib
git checkout $cvVersion
cd ..
```

(注:该下载会从 github 下载,若网络无法连接,可采用如下方式:

百度云链接 提取码：ad4p

将两个 zip 解压并重命名：

```
unzip opencv.zip
unzip opencv_contrib.zip
mv opencv-4.0.0 opencv
mv opencv_contrib-4.0.0 opencv_contrib
```

~~这里建议将 opencv 与 opencv_contrib 放到同一文件夹中，之前看相关博客说这需要在同一级文件夹中 Cmake 编译才能找到 opencv_contrib。~~

)

步骤 5：使用 contrib 模块编译并安装 OpenCV

```
cd opencv
mkdir build
cd build
```

接下来，我们开始编译和安装过程

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
        -D CMAKE_INSTALL_PREFIX=$cwd/installation/OpenCV-"$cvVersion" \
        -D INSTALL_C_EXAMPLES=ON \
        -D INSTALL_PYTHON_EXAMPLES=ON \
            -D WITH_TBB=ON \
            -D WITH_V4L=ON \
           -DOPENCV_PYTHON3_INSTALL_PATH=$cwd/OpenCV-$cvVersion-py3/
lib/python3.6/site-packages \
```

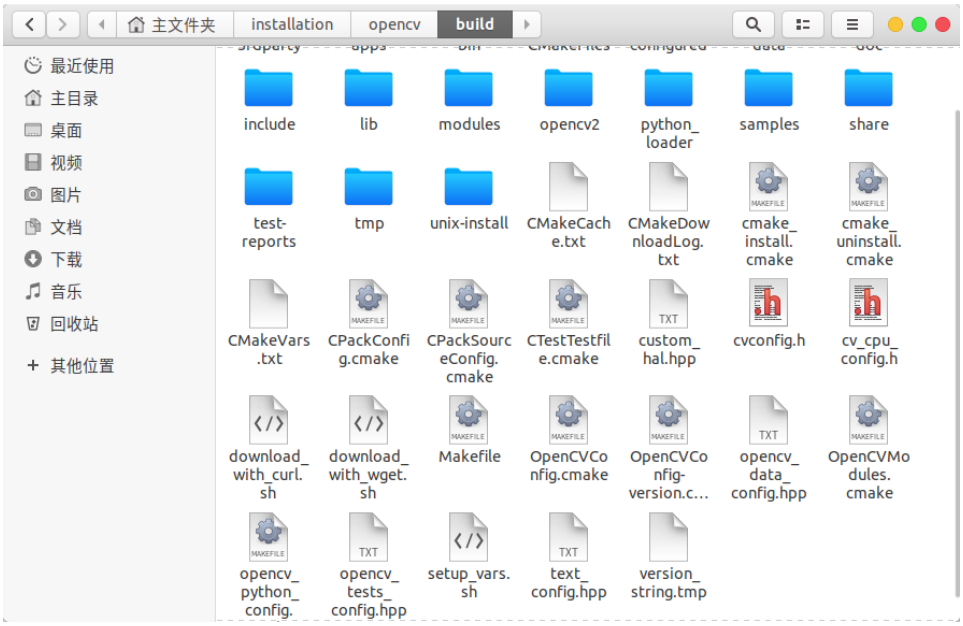
```
-D WITH_QT=ON \
-D WITH_OPENGL=ON \
-D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
-D BUILD_EXAMPLES=ON ..
```

编译安装

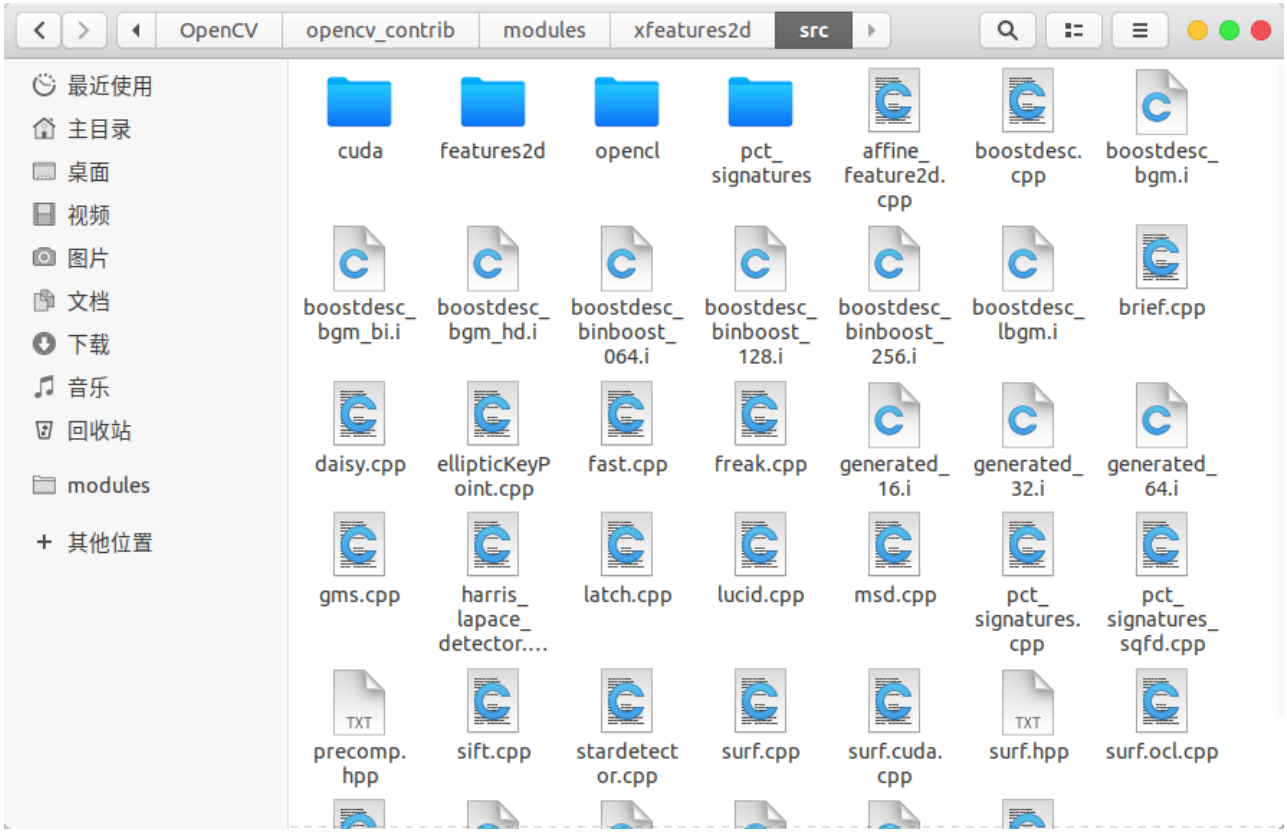
```
make -j4
make install
```

注:安装过程中可能会提示错误:

错误一: 提示某个目录或文件无法找到,此时,打开文件/build/CMakeDownloadLog.txt,



找到命令行提示中无法找到的文件对应的 log,打开对应的链接,手动进行下载,手动将文件导入到目录/opencv_contrib/modules/features2d/src.



注:缺少的文件的百度云链接:

百度云链接　提取密码: cbc8

将压缩包下载下来,并解压到/opencv_contrib/modules/xfeatures2d/src 文件中,不需要进行其他操作,再次编译.