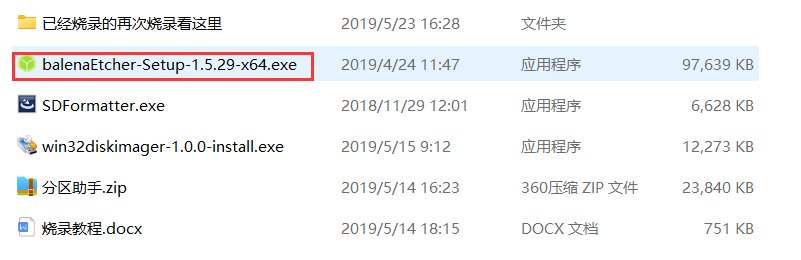
## 将镜像文件烧录至SSD卡

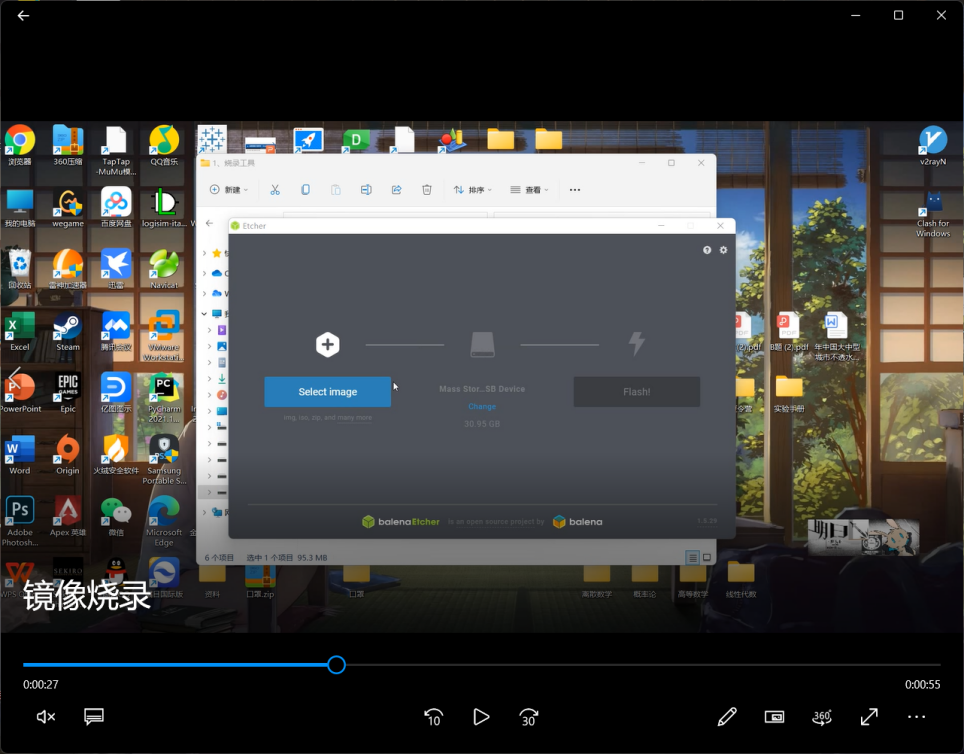
**（1）安装烧录软件**



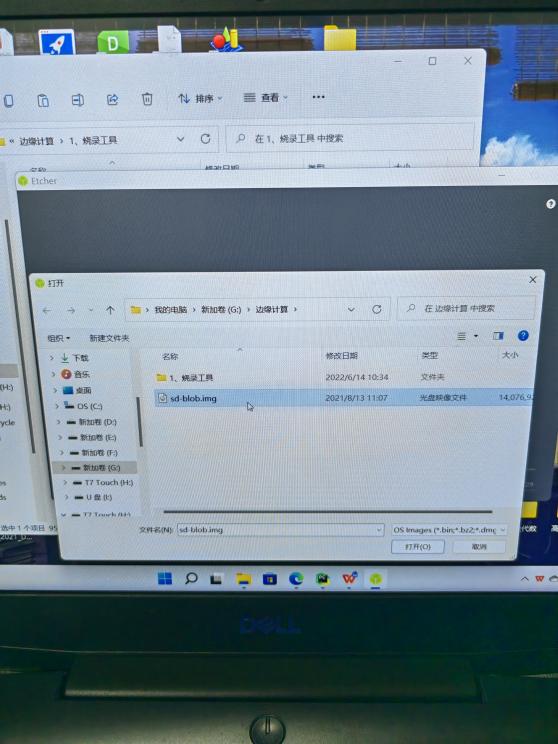
**（2）烧录所需的镜像文件**



**（3）安装好的balenaEtcher界面**



**（4）选择镜像文件**



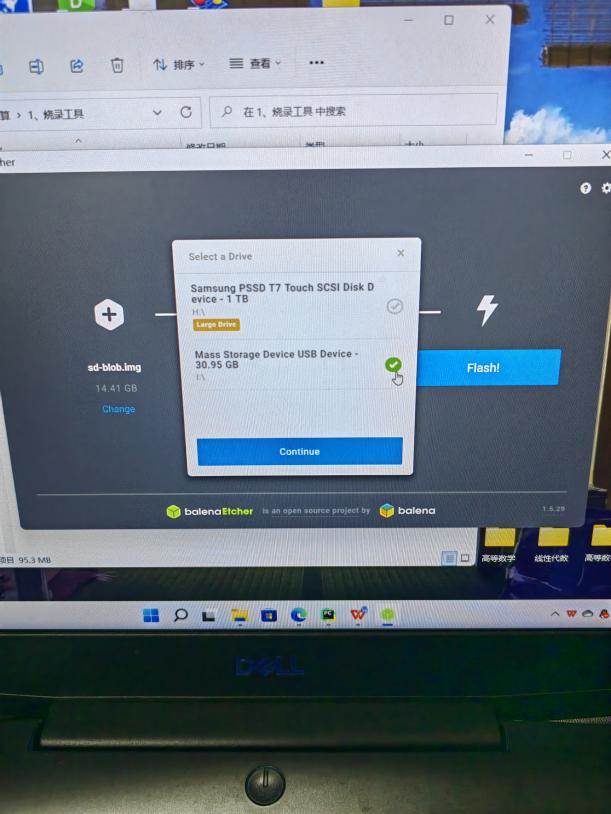
1. **去除带烧录的SSD卡和写卡器**



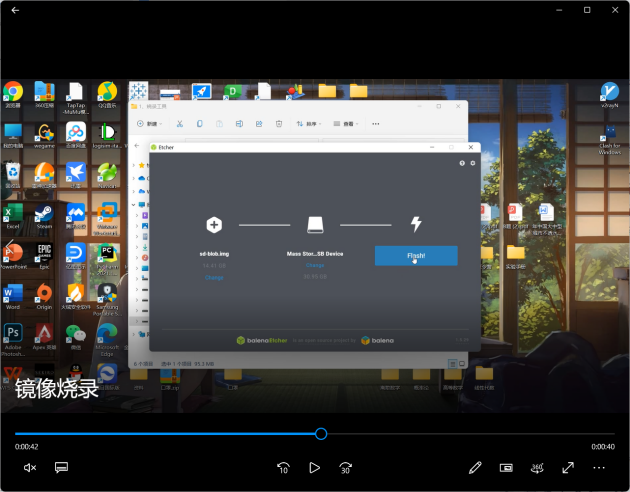
1. **将SSD卡插入写卡器**

****

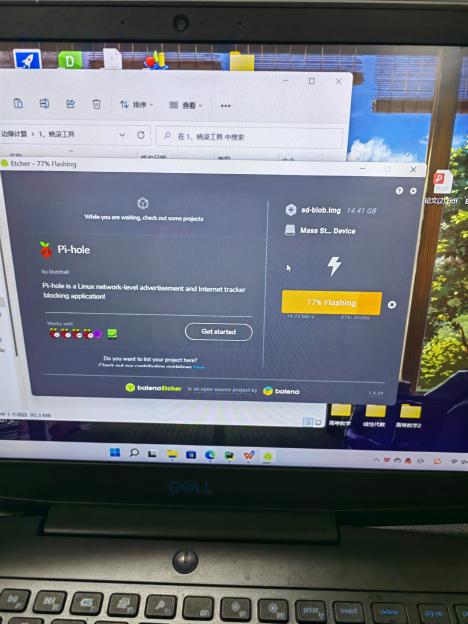
1. **选择待烧录的SSD卡**

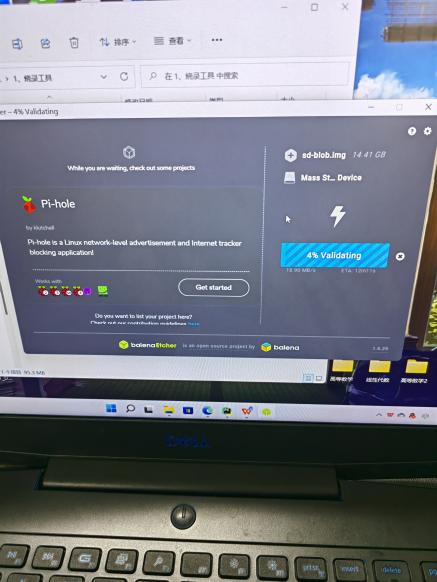


**（6）点击Finish开始烧录**

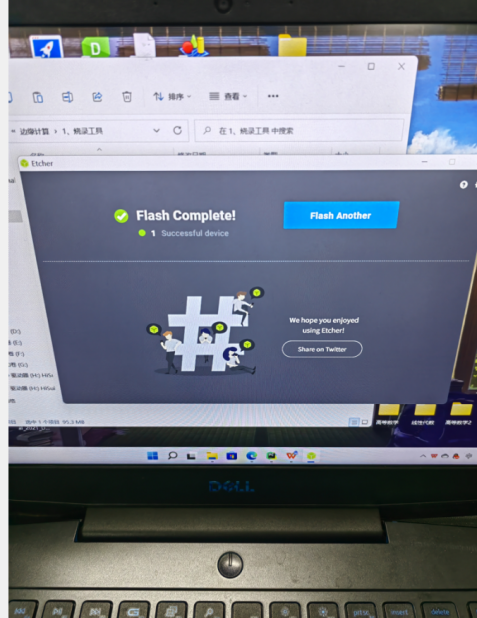


**（7）烧录界面，耐心等待**



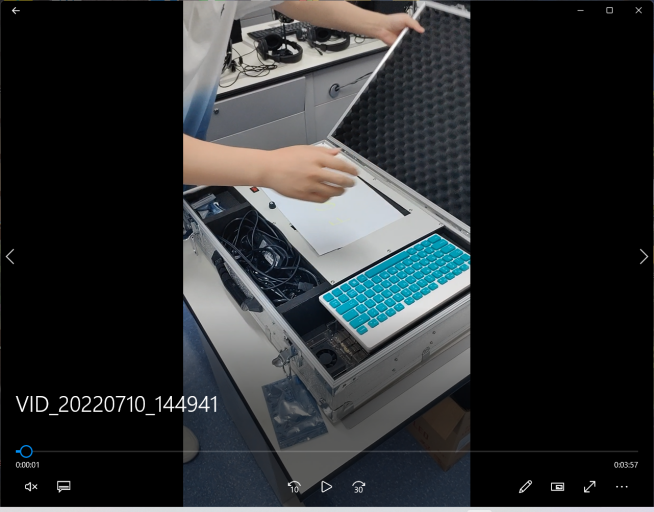
**（8）验证界面，耐心等待（如果失败，可以尝试直接重新烧录，或者换一台电脑烧录**

**（9）烧录成功**

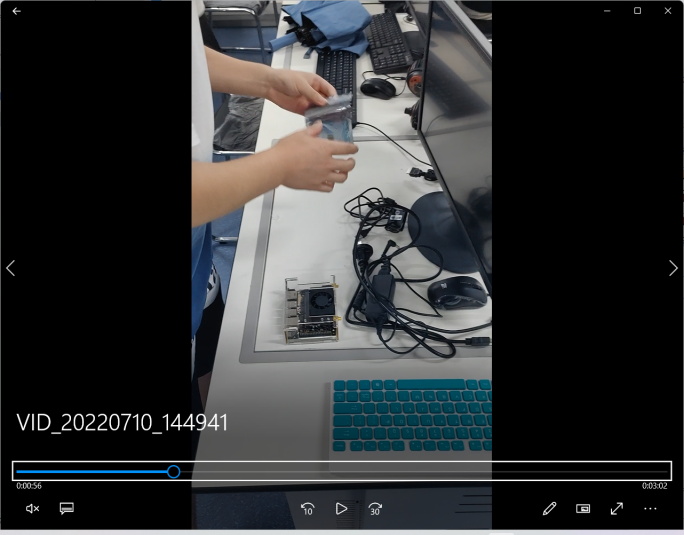


## 设备组装开机

1. **打开设备箱**



1. **取出鼠标、键盘、边缘计算机、电源线、HDMI线、烧录好的SSD卡**



1. **烧录好的SSD卡插入边缘计算机中。**



1. **组装电源线**

****

1. **插上边缘设备并通电**



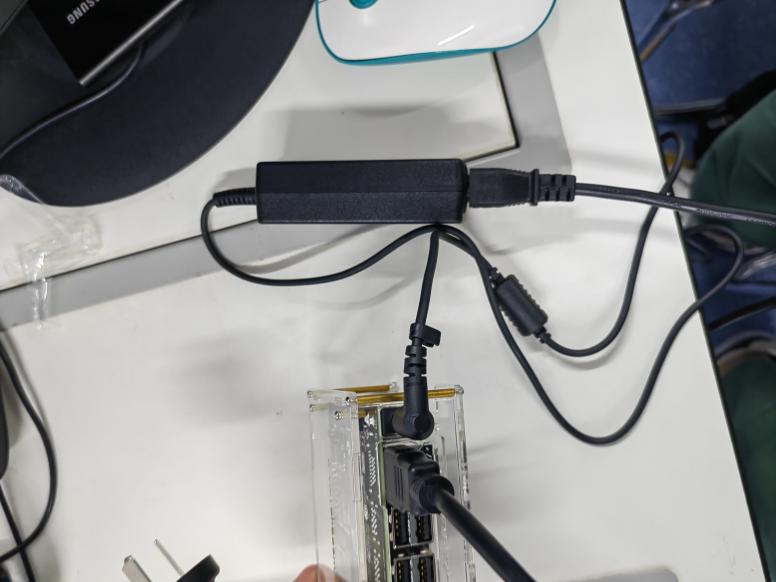
1. **拿出HDMI线**



1. **使用HDMI线将边缘计算设备连接到显示器上**



**（8）最终连接成这样**



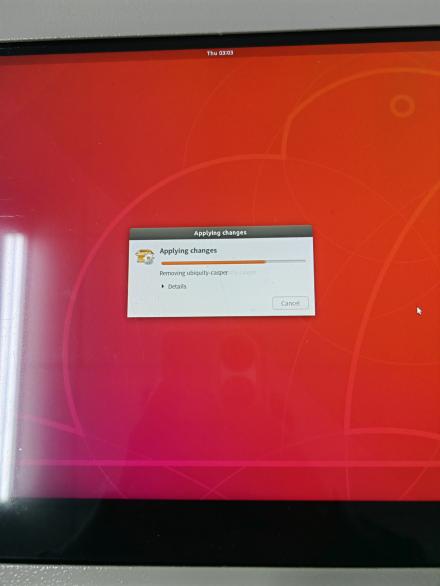
1. **扣开鼠标底盖，拿出无线鼠标和键盘的USB插卡**



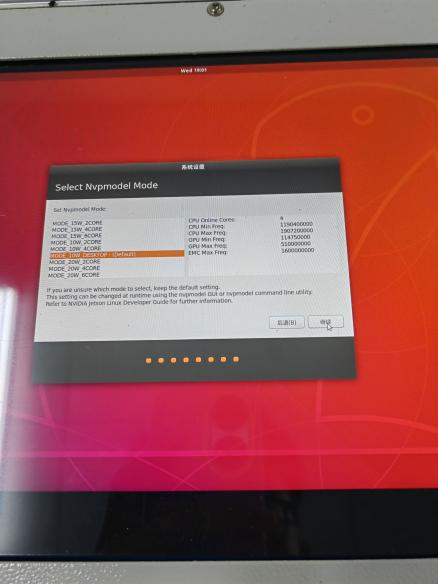
1. **插到边缘计算机的USB插槽上即可使用鼠标和键盘**



1. **初次启动配置加载**



1. **模块选择，这里选择default**



1. **分区大小选择默认值**



1. **用户名和密码设置，如果搭建集群，最好配置成相同的用户名**



1. **时区选择**



1. **配置网络，可以进入桌面后修改**



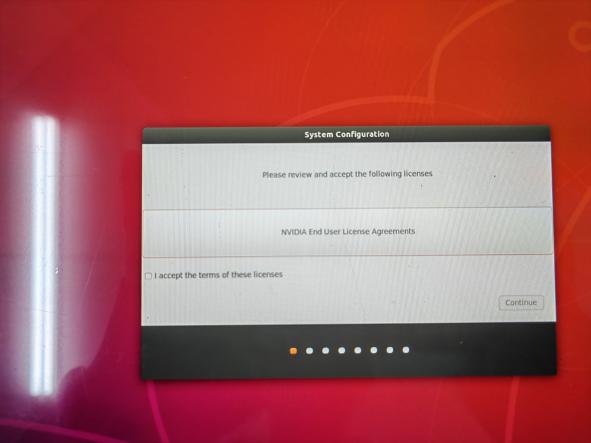
1. **键盘语言，选择英文之后无法输入中文（一般也没必要输入中文）**



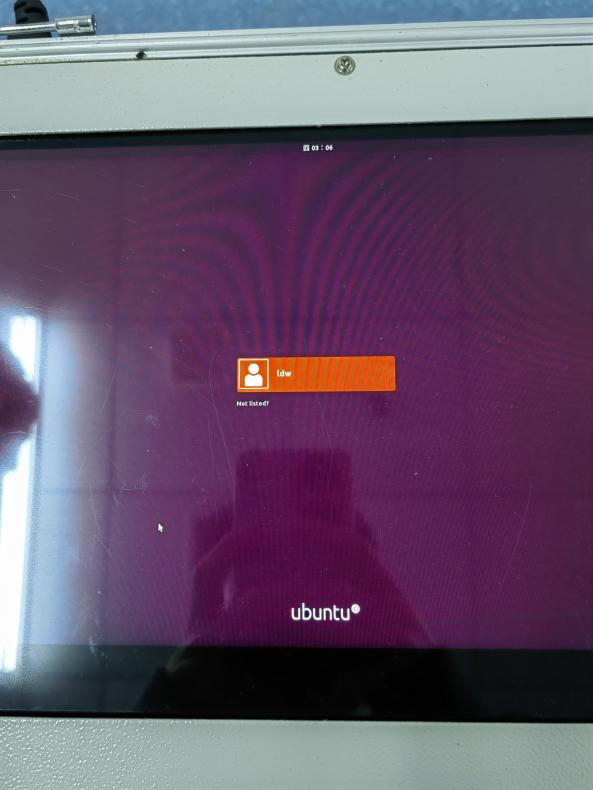
1. **语言选择中文简体**



1. **勾选接受，点击继续**



1. **插电、插卡后会自动开机，连接显示器后即可登入界面**



## 基础环境配置

**（1）配置CUDA环境（系统自带10.2版本）**

|  |
| --- |
| nvcc -V  # 没有则配置路径后再次验证：  sudo gedit ~/.bashrc  export CUDA\_HOME=/usr/local/cuda-10.2  export LD\_LIBRARY\_PATH=/usr/local/cuda-10.2/lib64:$LD\_LIBRARY\_PATH  export PATH=/usr/local/cuda-10.2/bin:$PATH  source ~/.bashrc |

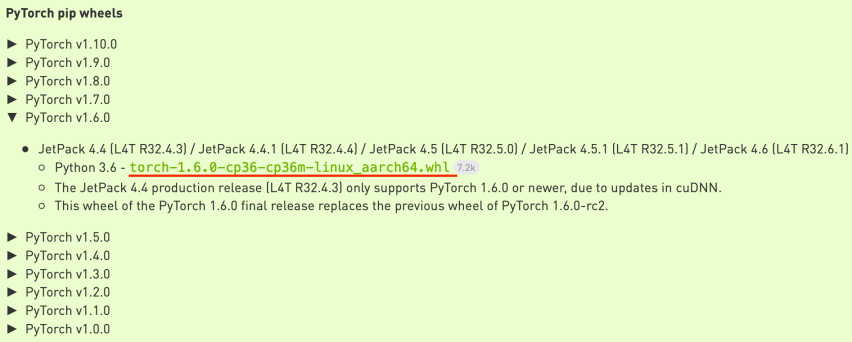
1. **安装实验一相关依赖**

（若无法访问下面多个链接，请访问百度网盘：https://pan.baidu.com/s/1h08FxQcct6k

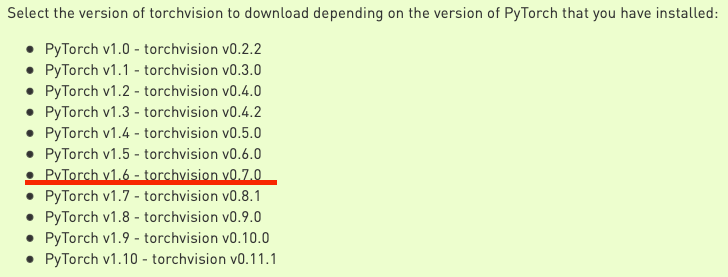
8E0rO7KUElA 密码: mis7）

1. 下载 pytorch，链接：https://forums.developer.nvidia.com/t/pytorch-for-jetso

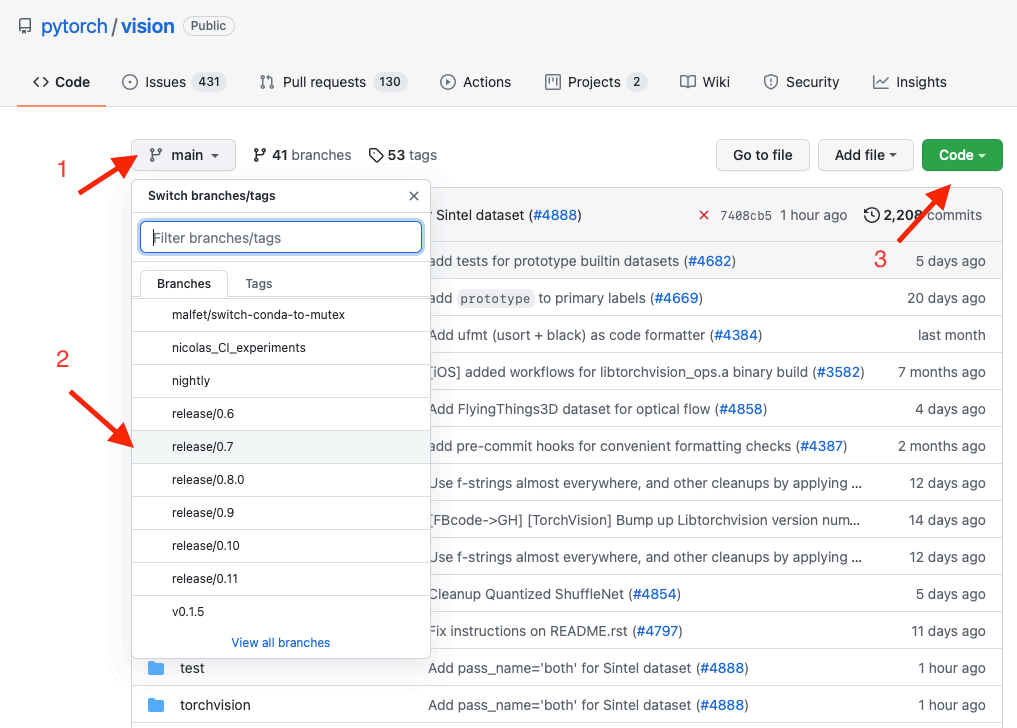
n-version-1-10-now-available/72048，下载版本为：torch-1.6.0-cp36-cp36m-linux\_aarch64.whl，



1. 下载 torchvision，链接：https://github.com/pytorch/vision，torchvision 与 torch 版本对应关



一定要切换到与 pytorch对应版本的 branch 之后下载源代码。



1. 安装相关依赖，apt-get 通常可以顺利安装，pip3 可能需要借助国内源，在下面每一个 pip3 命令后追加：-i https://pypi.doubanio.com/simple/ –trusted-host pypi.doubanio.com，即可从国内豆 瓣源安装相关第三方库，后续所有 pip3 均可如此添加，不再赘述。

|  |
| --- |
| sudo apt-get update  sudo apt-get install git cmake python3-dev  sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev  sudo apt-get install python3-pip  pip3 install -U pip testresources setuptools -i https://pypi.douban.com/simple  sudo apt-get install libopenblas-base libopenmpi-dev  sudo pip3 install mpi4py -i https://pypi.douban.com/simple  pip3 install Cython -i https://pypi.douban.com/simple  pip3 install protobuf==3.3.0 -i https://pypi.douban.com/simple  pip3 install numpy==1.16.1 -i https://pypi.douban.com/simple  pip3 install future==0.18.2 -i https://pypi.douban.com/simple  # 安装步骤1中下载的pytorch  pip3 install numpy torch-1.6.0-cp36-cp36m-linux\_aarch64.whl  sudo apt-get install build-essential python-dev python-scipy  sudo apt-get install python3-sklearn  pip3 install -U scikit-learn -i https://pypi.douban.com/simple  pip3 install scikit-image -i https://pypi.douban.com/simple #（**下载时间长，需要耐心等待**）  sudo apt-get install python3-h5py  sudo apt-get install libjpeg-dev zlib1g-dev libpython3-dev  sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev  # 安装步骤2中下载的torchvision  cd torchvision  export BUILD\_VERSION=0.7.0  python3 setup.py install --user |

1. dilb 以及 face\_recognition 安装

下载 dilb，链接：http://dlib.net/files，下载版本为：dlib-19.17.tar.bz2，如下图5所示。



（如果安装不成功，把整个文件夹包括安装包全部删除，重新下载安装）

|  |
| --- |
| tar jxvf dlib-19.17.tar.bz2  cd dlib-19.17  gedit dlib/cuda/cudnn\_dlibapi.cpp  # 在854行找到并注释//forward\_algo = forward\_best\_algo;  sudo python3 setup.py install  pip3 install face\_recognition -i https://pypi.douban.com/simple |

1. 进入实验一程序目录，执行python3 face\_recognition\_knn.py，等待一段时间后若弹出人脸检测结果，则实验一环境配置成功。
2. **安装实验六相关依赖**
3. apt-get 通常可以顺利安装，pip3 可能需要借助国内源，在下面每一个 pip3

命令后追加：-i https://pypi.doubanio.com/simple/ –trusted-host pypi.doubanio.com，即可从国内豆瓣源安装相关第三方库，后续所有 pip3 均可如此添加，不再赘述。

|  |
| --- |
| # 移除系统默认opencv  sudo apt purge libopencv\*  sudo apt update  # 安装前置依赖  sudo apt-get install -y build-essential cmake git gfortran libatlas-base-dev libavcodec-dev \  libavformat-dev libavresample-dev libcanberra-gtk3-module libdc1394-22-dev \  libeigen3-dev libglew-dev libgstreamer-plugins-base1.0-dev \  libgstreamer-plugins-good1.0-dev libgstreamer1.0-dev libgtk-3-dev \  libjpeg-dev libjpeg8-dev libjpeg-turbo8-dev liblapack-dev liblapacke-dev \  libopenblas-dev libpng-dev libpostproc-dev libswscale-dev \  libtbb-dev libtbb2 libtesseract-dev libtiff-dev libv4l-dev libxine2-dev \  libxvidcore-dev libx264-dev pkg-config python-dev python-numpy python3-sklearn \  python3-dev python3-numpy python3-matplotlib qv4l2 v4l-utils v4l2ucp zlib1g-dev |

1. 将 opencv 以及 opencv-contrib 解压并放置同一目录下，打开 terminal，更改一些设置以及代码和文件。

|  |
| --- |
| sudo chown -R jetson opencv-4.1.1  sudo chown -R jetson opencv\_contrib-4.1.1  cd opencv-4.1.1  mkdir build  #编辑opencv-4.1.1/modules/core/include/opencv2/core/private.hpp路径下的66行  edit file:opencv-4.1.1/modules/core/include/opencv2/core/private.hpp,line 66  #include <eigen3/Eigen/Core>  #把百度网盘里的补丁库文件复制到opencv\_contrib-4.1.1/modules/xfeatures2d/src下  move below files to opencv\_contrib-4.1.1/modules/xfeatures2d/src  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_bgm.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_bgm\_bi.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_bgm\_hd.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_binboost\_064.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_binboost\_128.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_binboost\_256.i  opencv\_3rdparty-contrib\_xfeatures2d\_boostdesc\_20161012/boostdesc\_lbgm.i  opencv\_3rdparty-contrib\_xfeatures2d\_vgg\_20160317/vgg\_generated\_48.i  opencv\_3rdparty-contrib\_xfeatures2d\_vgg\_20160317/vgg\_generated\_64.i  opencv\_3rdparty-contrib\_xfeatures2d\_vgg\_20160317/vgg\_generated\_80.i  opencv\_3rdparty-contrib\_xfeatures2d\_vgg\_20160317/vgg\_generated\_120.i  cd ..  #复制opencv-4.1.1/modules/features2d到build目录下  sudo cp -r opencv-4.1.1/modules/features2d opencv-4.1.1/build |

1. 设置 cmake 并 make 以及 install

步骤 4 需要注意 cmake 的时候 terminal 的输出信息，后续未能 make 成功很大程度上是网络问题

|  |
| --- |
| #在build下打开terminal  cd build  cmake -DCMAKE\_BUILD\_TYPE=Release -D CMAKE\_INSTALL\_PREFIX=/usr/local \  -D OPENCV\_ENABLE\_NONFREE=1 -D BUILD\_opencv\_python2=1 -D BUILD\_opencv\_python3=1 \  -D WITH\_FFMPEG=1 -D CUDA\_TOOLKIT\_ROOT\_DIR=/usr/local/cuda -D CUDA\_ARCH\_BIN=7.2 \  -D CUDA\_ARCH\_PTX=7.2 -D WITH\_CUDA=1 -D ENABLE\_FAST\_MATH=1 -D CUDA\_FAST\_MATH=1 \  -D WITH\_CUBLAS=1 -D OPENCV\_GENERATE\_PKGCONFIG=1 \  -D OPENCV\_EXTRA\_MODULES\_PATH=../../opencv\_contrib-4.1.1/modules \  ..  #编译并安装  5make -j4  sudo make install |

1. (可选) 该部分紧接步骤 2 的安装前置依赖之后

由于编译时间漫长，4 个 CPU 核心编译时长都有 2 小时，因此百度网盘也提供已经编译好的 opencv.tar.gz 压缩包，将压缩包下载并一定要放置在/home 目录下，并在该目录解压，解压得到 opencv 文件夹，里面包含了步骤三解压的 opencv-4.1.1 以及 opencv\_contrib-4.1.1 文件夹，该两个文件夹是已经 make 好的，因此只需在 opencv-4.1.1/build 下打开 terminal，并输入如下命令

|  |
| --- |
| # 在/home目录下解压  sudo tar -zxvf opencv.tar.gz  # opencv-4.1.1/build下编译  sudo make install |

1. 验证

|  |
| --- |
| python3  import cv2  cv2.\_\_version\_\_ |

1. 实验六

运行take\_image.py 程序，测试摄像头是否能正常启动，按下键盘上的”S”对当前帧进行保存，保存的路径为 data 目录下的 1 目录下的 left 和 right 文件夹，分别对应存放左右摄像头的图片。按下”C”展示下一帧，按下“Q”退出图像采集程序。

摄像头采集到图片样本放在/data/1 目录中，双目摄像头分为左右两个摄像头，故每次保存两张，分别存放在 left 和 right 中。识别并渲染后的图片存放在 disparities 目录下，combined 目录下是结合处理前和处理后的图片，便于进行比较对比。

运行程序 python3 main.py

## Hadoop安装

参考网址：[尚硅谷大数据Hadoop教程（Hadoop 3.x安装搭建到集群调优）\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1Qp4y1n7EN?p=25&vd_source=8cb51e3ee5b3eb79ab989a6ab8db803e)

1. 边缘设备没有防火墙，连接同一个热点后就可以ping通，也可以尝试使用交换机和高速网卡。
2. 配置hadoop集群时各节点用户名要相同（见注意事项（4）），建议每个节点上新建一个同名用户user。
3. 各节点之间配置ssh免密登录，注意在user下操作，保证免密登录的账户均为user。
4. 如视频中所示，需要对user添加到etc/sudoers中，不然hadoop启动时可能进入安全模式，无法上传文件。
5. 配置了JAVA\_HOME之后，还需要在hadoop安装目录的脚本中再次声明（见注意事项（2））
6. 不要使用b站教程中给出的jdk（见注意事项（3））

（7）hdfs如果启动失败，删除各个结点安装目录下的data和log文件夹，再次启动。

## 注意事项

1. **在完成试验六并按要求装好包之后，运行代码出现python：It seems that scikit-learn has not been built correctly的报错。**

**解决方法**：将原scikit-learn版本降为0.19.0即可。

1. **启动Hadoop，报错Error JAVA\_HOME is not set and could not be found。**

**解决方法：**在集群环境下，即使各结点都正确地配置了JAVA\_HOME，也会报如下错误，在hadoop-env.sh中，再显示地重新声明一遍JAVA\_HOME。

**（3）按照B站视频Hadoop安装课程，给出的JDK会出现问题，与系统不兼容。**

**解决方法**：使用sudo apt-get install openjdk-8-jdk，会自动安装到目录/usr/lib/jvm/java-1.8.0-openjdk-arm64，并与hadoop-3.1.3适配。

**（4）Hadoop安装后主节点无法访问其他其他节点。**

**解决方法：**Hadoop各节点的用户名必须一致。

**（5）Pytorch与当前环境不兼容，只能运行cpu版本的torch，cuda版本无法运行。**

**解决方法**：必须通过Jetson Xavier nx的官方网站（见文档）上下载与当前python环境（3.6.9）一致的.whl文件，然后使用pip进行安装。（torchvision的版本要与torch对应，下载地址同torch）。

**（6）高版本torch与python不兼容的问题**

**解决方法**：若需要用到高版本的torch，下载的whl文件很有可能与当前python不兼容，需要另外安装对应版本的python环境，并在该环境中安装对应的torch及torch vision（如python3.8 torch=1.8 torchvision=0.9.1）

**（7）高版本cuda（>10.2）无法正常使用的问题**

**解决方法**：该边缘计算设备的gpu算力有限，基本无法支持10.2以上版本的cuda（10.2版本cuda为系统自带，建议不要安装高版本cuda）

**（8）Opencv的cv.imshow无法显示图像的问题**

**解决方法**：要注意登录的用户需要拥有该图形化界面的使用权限，此时通过cv调用的图像才能在桌面上显示。

**（9）机器重启后无限循环在登录输入密码界面的问题**

**解决方法**：文件权限问题，主目录下的.Xauthority文件拥有者变成了root，从而以用户登陆的时候无法都取.Xauthority文件，将该文件删除即可（若之前在profile文件中引入了不相关的地址，需要将其注释掉）(无法进入桌面时，使用远程登录的方式进入，删除文件)

**（10）将用户的目录权限设置为777后，主机间两两无法免密登录的问题**

**解决方法**：sshd为了安全，对属主的目录和文件权限有所要求。如果权限不对，则ssh的免密码登陆不生效。将对应的文件夹和文件权限修正后，方可继续ssh连接。

**（11）protobuf编译器版本与库版本不匹配**

**解决方法**：通过查看系统中的protobuf版本和当前的版本，发现存在不匹配现象，需要将protobuf的版本降低为3.19.0。