需要安装：

opencv   X11

在网上找到源代码：

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

while(1):

# get a frame

ret, frame = cap.read()

# show a frame cv2.imshow("capture", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break cap.release()

cv2.destroyAllWindows()

拷贝到docker容器里执行：

报错缺少cv2模块

查找资料需要安装opencv-python

步骤一：安装opencv-python

在docker内部执行

pip install opencv-Python

如果python版本较低，会报错：

Getting requirements to build wheel ... error

  ERROR: Command errored out with exit status 1:

   command: /usr/bin/python /usr/local/lib/python2.7/dist-packages/pip/\_vendor/pep517/\_in\_process.py get\_requires\_for\_build\_wheel /tmp/tmpK9j8nr

       cwd: /tmp/pip-install-dVEO5J/opencv-python

  Complete output (22 lines):

  Traceback (most recent call last):

·····

、直接安装 apt-get install python3.6 ，失败

root@91d2d47e8aee:/# apt-get install python3.6

Reading package lists... Done

Building dependency tree

Reading state information... Done

E: Unable to locate package python3.6

E: Couldn't find any package by glob 'python3.6'

E: Couldn't find any package by regex 'python3.6'

2、添加ppa到系统，执行

add-apt-repository ppa:jonathonf/python-3.6

1

失败，问题：add-apt-repository找不到

对于ppa的解释：<https://www.cnblogs.com/EasonJim/p/7119331.html>

3、执行

apt-get update

1

4、执行以下

apt-get install python-software-properties

apt-get install software-properties-common

5、重复步骤

add-apt-repository ppa:jonathonf/python-3.6

安装完成！！！

6、查看Python版本以及指向

ls -l /usr/bin | grep python

1

7、删除原有Python链接

rm /usr/bin/python

8、建立新连接

ln -s /usr/bin/python3.6 /usr/bin/python

再次安装：

pip install opencv-Python

再次报错：

ImportError: No module named 'pip.\_interna

解决如下：

wget https://bootstrap.pypa.io/get-pip.py  --no-check-certificatesudo //报错就去掉选项

python get-pip.py

安装opencv依赖：

apt install libopencv-dev

之后再次安装：

成功！

步骤二：

打开docker使用x11权限：

xhost+

导出之前创建的镜像：

docker commit video video

创建新的容器并加上有关参数：

docker run -itd --name video1 --hostname video1  --device=/dev/video0 -e DISPLAY=unix$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix video

/dev/video0是摄像头挂载名

再次运行代码：

成功！

rtmp推流

<https://blog.csdn.net/zong596568821xp/article/details/92790502>

opencv播放rtmp推流代码：

<https://www.cnblogs.com/sirxy/p/12126383.html>

opencv播放本地或远程视频

<https://www.cnblogs.com/sirxy/p/12123426.html>

配置rtmp推流：

在docker镜像中安装ffmpeg nginx

1、nginx服务器搭建：

sudo apt-get update

sudo apt-get install openssl libssl-dev

sudo apt-get install libpcre3 libpcre3-dev

编译源码：

在工作空间下，新建一个nginx文件夹，用来存放需要下载nginx和nginx-rtmp-module两个安装包源码

nginx[下载链接](http://nginx.org/en/download.html)，这里我下载了1.8.1版本的源码，解压文件，生成nginx-1.8.1文件夹

在nginx目录下，下载nginx-rtmp-module

git clone https://github.com/arut/nginx-rtmp-module.git

然后编译安装nginx，cd进nginx的目录

cd nginx-1.8.1

./configure --add-module=../nginx-rtmp-module

make

make install

configure报错：

进入/objs/Makfile，找到gcc编译行，将参数-Werror删除，再次编译成功

make

报错：

make[1]: \*\*\* [objs/src/event/ngx\_event\_openssl.o] 错误 1

查阅发现是openssl版本不对，参照如下教程修改：

<https://blog.csdn.net/qq_39720249/article/details/84655501>

再次编译，找不到openssl，需要加上路径：

./configure --with-openssl=/usr/local/openssl-1.0.10 --add-module=../nginx-rtmp-module

再次make报错

cc1: all warnings being treated as errors

objs/Makefile:510: recipe for target 'objs/src/core/ngx\_murmurhash.o' failed

make[1]: \*\*\* [objs/src/core/ngx\_murmurhash.o] Error 1

make[1]: Leaving directory '/home/nginx/nginx-1.8.1'

Makefile:8: recipe for target 'build' failed

make: \*\*\* [build] Error 2

再次打开objs/Makefile删去Werror

make

make install

成功安装！

**测试nginx**

进入安装目录/usr/local/nginx，运行以下命令

./sbin/nginx

**配置rtmp**

编辑/usr/local/nginx/conf/nginx.conf文件

rtmp {

server {

listen 1935; #服务端口--默认

chunk\_size 4096; #数据传输块的大小--默认

*#设置直播的application名称是 mylive*

application mylive{

live on; #live on表示开启直播模式

}

}

}

#请在http里面找到server

http{

...# 这里有一些其他的配置

server {

listen 8080;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

location /pop/video {

alias /var/video;

}

location /info {

rtmp\_stat all;

rtmp\_stat\_stylesheet stat.xsl;

}

location /stat.xsl {

root html;

}

}

}

配置完之后，需要重启nginx

/usr/local/nginx/sbin/nginx -s reload

推流代码：

import cv2

import subprocess

#rtsp = "rtsp://admin:a12345678@10.10.8.101:554/h264/ch1/main/av\_stream"

rtmp = 'rtmp://localhost:1935/mylive/test'

# 读取视频并获取属性

cap = cv2.VideoCapture(rtsp)

size = (int(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH)), int(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)))

sizeStr = str(size[0]) + 'x' + str(size[1])

command = ['ffmpeg',

'-y', '-an',

'-f', 'rawvideo',

'-vcodec','rawvideo',

'-pix\_fmt', 'bgr24',

'-s', sizeStr,

'-r', '25',

'-i', '-',

'-c:v', 'libx264',

'-pix\_fmt', 'yuv420p',

'-preset', 'ultrafast',

'-f', 'flv',

rtmp]

pipe = subprocess.Popen(command

, shell=False

, stdin=subprocess.PIPE

)

while cap.isOpened():

success,frame = cap.read()

if success:

'''

对frame进行识别处理

'''

if cv2.waitKey(1) & 0xFF == ord('q'):

break

pipe.stdin.write(frame.tostring())

cap.release()

pipe.terminate()

本部分参考： <https://blog.csdn.net/zong596568821xp/article/details/92790502>

安装ffmpeg:

echo "deb [check-valid-until=no] http://archive.debian.org/debian jessie-backports main" > /etc/apt/sources.list.d/jessie-backports.list

sed -i '/deb http:\/\/deb.debian.org\/debian jessie-updates main/d' /etc/apt/sources.list

apt-get -o Acquire::Check-Valid-Until=false update

apt-get -y --force-yes install yasm ffmpeg

安装成功之后就可以运行推流代码，推流到localhost:1935

docker run -itd --name video4 --hostname video4 -p 1935:1935 -p 8080:80 --device=/dev/video0 -e DISPLAY=unix$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix quagga\_snmp:v7

docker run -itd --name leo11 --hostname leo11 -p 1935:1935 -p 8080:80 --device=/dev/video0 -e DISPLAY=unix$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix quagga\_snmp:v7

docker run -itd --name leo13 --hostname leo13 -p 1900:1900 -p 8000:81 --device=/dev/video0 -e DISPLAY=unix$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix quagga\_snmp:v7

实验流程：

运行终端后，按照上面的指令开启nginx服务，发送端设置rtmp\_send.py文件，rtmp链接后的ip改为拉流节点的ip，接收端设置rtmp\_video.py，rtmp链接修改对应端口

发送端和接收端分别运行：

python rtmp\_send.py

python rtmp\_video.py