JK Jung's blog

How to Install OpenCV (3.4.0) on Jetson TX2

Aug 7, 2017

2018-08-08 update: Verified opencv-3.4.2 built and run OK under JetPack-3.3.

2018-03-27 update: Update for JetPack-3.2.

2018-01-29 update: 1. Updated to opencv-3.4.0; 2. Added more apt package clean-up steps at the beginning; 3. Using Qt/OpenGL backend instead of GTK+ (this resolves the ‘protobuf double free’ issue for caffe: reference); 4. Install python3/python2 ‘matplotlib’ before building opencv (this is recommended by opencv’s official documentation).

2017-10-20 update: I added installation steps for python2, as well as the resulting cmake configuration on my Jetson TX2.

When I started programming with python3 on Jetson TX2, I found that `import cv2` did not work. It appeared that the pre-installed OpenCV4Tegra (2.4.13.1) only provided bindings for python2.7. After some research, I found the recommended solution to this problem was to re-compile OpenCV.

I was kind of reluctant to remove OpenCV4Tegra and re-build OpenCV by myself. But anyway I finally decided to do it. I documented the procedure below.

Prerequisite

- Complete installation of JetPack-3.3 (or JetPack-3.2.1 or JetPack-3.1) on the target Jetson TX2.

Reference

- Installing OpenCV Guide for Jetson TK1
- Ubuntu 16.04: How to install OpenCV
- Install OpenCV-Python in Fedora
- Building OpenCV with OpenGL support?
- Resolved: Matplotlib figures not showing up or displaying

Installation Steps

I’d start by cleaning up older opencv packages and installing necessary dependencies for building opencv.

Regarding the python `matplotlibrc` modifications below, refer to this StackOverflow thread for more details.

```
### Remove all old opencv stuffs installed by JetPack (or OpenCV4Tegra)
$ sudo apt-get purge libopencv*
### I prefer using newer version of numpy (installed with pip), so
### I'd remove this python-numpy apt package as well
$ sudo apt-get purge python-numpy
### Remove other unused apt packages
$ sudo apt autoremove
### Upgrade all installed apt packages to the latest versions (optional)
$ sudo apt-get update
$ sudo apt-get dist-upgrade
### Update gcc apt package to the latest version (highly recommended)
$ sudo apt-get install --only-upgrade g++-5 cpp-5 gcc-5
### Install dependencies based on the Jetson Installing OpenCV Guide
$ sudo apt-get install build-essential make cmake cmake-curses-gui \
                       g++ libavformat-dev libavutil-dev \
                       libswscale-dev libv4l-dev libeigen3-dev \
                       libglew-dev libgtk2.0-dev
### Install dependencies for gstreamer stuffs
$ sudo apt-get install libdc1394-22-dev libxine2-dev \
                       libgstreamer1.0-dev \
                       libgstreamer-plugins-base1.0-dev
### Install additional dependencies according to the pyimageresearch
### article
$ sudo apt-get install libjpeg8-dev libjpeg-turbo8-dev libtiff5-dev \
                       libjasper-dev libpng12-dev libavcodec-dev
$ sudo apt-get install libxvidcore-dev libx264-dev libgtk-3-dev \
                       libatlas-base-dev gfortran
$ sudo apt-get install libopenblas-dev liblapack-dev liblapacke-dev
### Install Qt5 dependencies
$ sudo apt-get install qt5-default
### Install dependencies for python3
$ sudo apt-get install python3-dev python3-pip python3-tk
$ sudo pip3 install numpy
$ sudo pip3 install matplotlib
### Modify matplotlibrc (line #41) as 'backend      : TkAgg'
$ sudo vim /usr/local/lib/python3.5/dist-packages/matplotlib/mpl-data/matplotlibrc
### Also install dependencies for python2
### Note that I install numpy with pip, so that I'd be using a newer
### version of numpy than the apt-get package
$ sudo apt-get install python-dev python-pip python-tk
$ sudo pip2 install numpy
$ sudo pip2 install matplotlib
```

```
### Modify matplotlibrc (line #41) as 'backend      : TkAgg'
$ sudo vim /usr/local/lib/python2.7/dist-packages/matplotlib/mpl-data/matplotlibrc
```

Before downloading and building opencv-3.4.0, I'd first do some modifications according to this post, in order to fix OpenGL related compilation problems . More specifically, I'd modify `/usr/local/cuda/include/cuda_gl_interop.h` and fix the symbolic link of libGL.so.

```
$ sudo vim /usr/local/cuda/include/cuda_gl_interop.h
$ cd /usr/lib/aarch64-linux-gnu/
$ sudo ln -sf tegra/libGL.so libGL.so
```

Here's how the relevant lines (line #62~68) of `cuda_gl_interop.h` look like after the modification.

```
//#if defined(__arm__) || defined(__aarch64__)
//#ifndef GL_VERSION
//#error Please include the appropriate gl headers before including cuda_gl_interop.h
//#endif
//#else
 #include <GL/gl.h>
//#endif
```

Next, download opencv-3.4.0 source code, cmake and compile. Note that opencv_contrib modules (cnn/dnn stuffs) would cause problem on pycaffe, so after some experiments I decided not to include those modules at all.

```
### Download opencv-3.4.0 source code
$ mkdir -p ~/src
$ cd ~/src
$ wget https://github.com/opencv/opencv/archive/3.4.0.zip \
       -O opencv-3.4.0.zip
$ unzip opencv-3.4.0.zip
### Build opencv (CUDA_ARCH_BIN="6.2" for TX2, or "5.3" for TX1)
$ cd ~/src/opencv-3.4.0
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local \
        -D WITH_CUDA=ON -D CUDA_ARCH_BIN="6.2" -D CUDA_ARCH_PTX="" \
        -D WITH_CUBLAS=ON -D ENABLE_FAST_MATH=ON -D CUDA_FAST_MATH=ON \
        -D ENABLE_NEON=ON -D WITH_LIBV4L=ON -D BUILD_TESTS=OFF \
        -D BUILD_PERF_TESTS=OFF -D BUILD_EXAMPLES=OFF \
        -D WITH_QT=ON -D WITH_OPENGL=ON ..
$ make -j4
$ sudo make install
```

Just for reference, here's the resulting opencv-3.4.0 cmake configuration for my Jetson TX2 system.

```
-- General configuration for OpenCV 3.4.0 =====================================
--   Version control:               unknown
--
--   Platform:
--     Timestamp:                   2018-01-29T07:58:45Z
--     Host:                        Linux 4.4.38-tegra aarch64
--     CMake:                       3.5.1
--     CMake generator:             Unix Makefiles
--     CMake build tool:            /usr/bin/make
--     Configuration:               RELEASE
--
--   CPU/HW features:
--     Baseline:                    NEON FP16
--       required:                  NEON
--       disabled:                  VFPV3
--
--   C/C++:
--     Built as dynamic libs?:      YES
--     C++ Compiler:                /usr/bin/c++  (ver 5.4.0)
--     C++ flags (Release):         -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor -Werror=address -Wer
--     C++ flags (Debug):           -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor -Werror=address -Wer
--     C Compiler:                  /usr/bin/cc
--     C flags (Release):           -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor -Werror=address -Wer
--     C flags (Debug):             -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor -Werror=address -Wer
--     Linker flags (Release):
--     Linker flags (Debug):
--     ccache:                      NO
--     Precompiled headers:         YES
--     Extra dependencies:          dl m pthread rt /usr/lib/aarch64-linux-gnu/libGLU.so /usr/lib/aarch64-linux-gnu/libGL.so
--     3rdparty dependencies:
--
--   OpenCV modules:
--     To be built:                 calib3d core cudaarithm cudabgsegm cudacodec cudafeatures2d cudafilters cudaimgproc cuda
--     Disabled:                    js world
--     Disabled by dependency:      -
--     Unavailable:                 java ts viz
--     Applications:                apps
--     Documentation:               NO
--     Non-free algorithms:         NO
--
--   GUI:
--     QT:                          YES (ver 5.5.1)
--       QT OpenGL support:         YES (Qt5::OpenGL 5.5.1)
--     GTK+:                        NO
--     OpenGL support:              YES (/usr/lib/aarch64-linux-gnu/libGLU.so /usr/lib/aarch64-linux-gnu/libGL.so)
--     VTK support:                 NO
--
--   Media I/O:
--     ZLib:                        /usr/lib/aarch64-linux-gnu/libz.so (ver 1.2.8)
--     JPEG:                        /usr/lib/aarch64-linux-gnu/libjpeg.so (ver )
--     WEBP:                        build (ver encoder: 0x020e)
--     PNG:                         /usr/lib/aarch64-linux-gnu/libpng.so (ver 1.2.54)
--     TIFF:                        /usr/lib/aarch64-linux-gnu/libtiff.so (ver 42 / 4.0.6)
--     JPEG 2000:                   /usr/lib/aarch64-linux-gnu/libjasper.so (ver 1.900.1)
--     OpenEXR:                     build (ver 1.7.1)
--
--   Video I/O:
--     DC1394:                      YES (ver 2.2.4)
--     FFMPEG:                      YES
--       avcodec:                   YES (ver 56.60.100)
--       avformat:                  YES (ver 56.40.101)
--       avutil:                    YES (ver 54.31.100)
--       swscale:                   YES (ver 3.1.101)
--       avresample:                NO
--     GStreamer:
--       base:                      YES (ver 1.8.3)
--       video:                     YES (ver 1.8.3)
--       app:                       YES (ver 1.8.3)
--       riff:                      YES (ver 1.8.3)
--       pbutils:                   YES (ver 1.8.3)
--     libv4l/libv4l2:              1.10.0 / 1.10.0
--     v4l/v4l2:                    linux/videodev2.h
--     gPhoto2:                     NO
--
--   Parallel framework:            pthreads
--
--   Trace:                         YES (built-in)
--
--   Other third-party libraries:
--     Lapack:                      NO
--     Eigen:                       YES (ver 3.2.92)
--     Custom HAL:                  YES (carotene (ver 0.0.1))
--
--   NVIDIA CUDA:                   YES (ver 8.0, CUFFT CUBLAS FAST_MATH)
--     NVIDIA GPU arch:             62
--     NVIDIA PTX archs:
--
--   OpenCL:                        YES (no extra features)
--     Include path:                /home/nvidia/src/opencv-3.4.0/3rdparty/include/opencl/1.2
--     Link libraries:              Dynamic load
--
--   Python 2:
--     Interpreter:                 /usr/bin/python2.7 (ver 2.7.12)
--     Libraries:                   /usr/lib/aarch64-linux-gnu/libpython2.7.so (ver 2.7.12)
--     numpy:                       /usr/local/lib/python2.7/dist-packages/numpy/core/include (ver 1.14.0)
--     packages path:               lib/python2.7/dist-packages
--
--   Python 3:
--     Interpreter:                 /usr/bin/python3 (ver 3.5.2)
--     Libraries:                   /usr/lib/aarch64-linux-gnu/libpython3.5m.so (ver 3.5.2)
--     numpy:                       /usr/local/lib/python3.5/dist-packages/numpy/core/include (ver 1.14.0)
--     packages path:               lib/python3.5/dist-packages
--
--   Python (for build):            /usr/bin/python2.7
--
--   Java:
--     ant:                         NO
--     JNI:                         NO
--     Java wrappers:               NO
--     Java tests:                  NO
--
--   Matlab:                        NO
--
--   Install to:                    /usr/local
-- -----------------------------------------------------------------
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/nvidia/src/opencv-3.4.0/build
```

To verify the installation:

```
$ ls /usr/local/lib/python3.5/dist-packages/cv2.*
/usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-aarch64-linux-gnu.so
$ ls /usr/local/lib/python2.7/dist-packages/cv2.*
/use/local/lib/python2.7/dist-packages/cv2.so
$ python3 -c 'import cv2; print(cv2.__version__)'
3.4.0
$ python2 -c 'import cv2; print(cv2.__version__)'
3.4.0
```

Bonus:

With opencv-3.4.0 properly installed on the Jetson TX2, we could use a python script to capture and display live video from either the Jetson onboard camera, a USB webcam or an IP CAM. Just follow along this post: How to Capture and Display Camera Video with Python on Jetson TX2.

82 Comments **https://jkjung-avt.github.io/** 1 **Login**

♡ **Recommend** 11 Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Gale • 8 months ago

Typo in code to verity installation:

```
$ ls /usr/local/lib/python3.5/dist-packages/cv2.*
/usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-aarch64-lin...
$ ls /usr/local/lib/python2.7/dist-packages/cv2.* // TYPO: usr instead of use
/use/local/lib/python2.7/dist-packages/cv2.so
$ python3 -c 'import cv2; print(cv2.__version__)'
3.4.0
$ python2 -c 'import cv2; print(cv2.__version__)'
3.4.0
```

5 ⌃ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Gale • 8 months ago

Thanks for pointing it out. I've corrected it accordingly.

⌃ | ⌄ • Reply • Share ›

Vince Dfr • 7 months ago

Hi @Jiun-Kuei Jung:disqus and everyone,
thanks a lot for your several posts, really a great help.
I'm currently trying to use tracker algorithms and I'm facing a problem with cv2.TrackerKCF_create() when I run this script (https://www.learnopencv.com... I have this error occuring each time:
" module object has no attribute 'Tracker_create()'

Is there something to add in your installation procedure to fix it ? I am using Jetson TX2 with OpenCV3.4.0, Python2.7.12 and Python3.5.2.

Thanks in advance

1 ⌃ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Vince Dfr • 7 months ago

OpenCV's tracking API seems to be part of the 'contrib' package: https://github.com/opencv/o...

You can probably reference the following page about how to install OpenCV with the 'contrib' stuffs: https://www.pyimagesearch.c...

I chose not to install 'opencv_contrib' because its dnn stuffs caused conflict with my own caffe build.

⌃ | ⌄ • Reply • Share ›

Vince Dfr ➦ Jiun-Kuei Jung • 7 months ago

Thank you for answering, I finally managed to make it work properly !

⌃ | ⌄ • Reply • Share ›

Chandrakanta ➦ Vince Dfr • 5 months ago

hi Vince, did you able to install opencv-contrib , can you guide me the process or blog you have followed . i am trying cv2.face.createLBPHFaceRecognizer() , getting error "module cv2 has no attribute face" , my ubuntu system is working fine with opencv-contrib ,but issues is with jetson TX2

1 ⌃ | ⌄ • Reply • Share ›

João Paulo Abdalla ➦ Chandrakanta • 4 months ago

I'm having the same problem, my Mac OS and Ubuntu systems are working fine with Opencv_contrib, using 'cv2.bgsegm.createBackgroundSubtractorMOG()' and on Jetson TX2 I receive the same king of message 'module cv2 has no attribute bgsegm'

⌃ | ⌄ • Reply • Share ›

Chandrakanta ➦ Chandrakanta • 5 months ago

hi , i build it with opencv-contrib and it works ,

⌃ | ⌄ • Reply • Share ›

João Paulo Abdalla ➦ Vince Dfr • 4 months ago

How you did it? I'm having the same kind of issue, but with the module 'bgsegm' and I'm having issues to build the OpenCV with this tutorial during trying to add '-D OPENCV_EXTRA_MODULES_PATH= <path to="" modules="" from="" opencv_contri="">

^ | v • Reply • Share ›

Jiun-Kuei Jung Mod ➦ João Paulo Abdalla • 4 months ago

Be sure to download the same version of opencv_contrib as opencv. For example, if you have opencv_contrib-3.4.0 unzipped as ~/src/opencv_contrib-3.4.0/, then you could add the following to your cmake flags:

```
-D OPENCV_EXTRA_MODULES_PATH=${HOME}/src/opencv_contrib-3.4.0/modules
```

^ | v • Reply • Share ›

Vince Dfr ➦ João Paulo Abdalla • 4 months ago

Hi Joao,
sorry it's been a while I did it and I'm not even use it anymore but from what I can remember, I purged my OpenCV installation and did it again using the Github repository indicated by Jiun (first response to my post).

It's really just a matter of adding some line in the Make file before making it but you have to rebuild OpenCV entirely.
Hope it will help

^ | v • Reply • Share ›

JHS • 13 days ago

잘보고 가유 ~
감사합니다

^ | v • Reply • Share ›

Hướng Đỗ Văn • 2 months ago

Hello,
Thanks you for your sharing, I completed to build OpenCV in Jetson TX2. However, when I tried to use opencv in virtual environment in Jetson TX2. It's seem not works! Do you know how to fix this issue? I'm looking forward to hear your advice~!

The below description as detail of my situation:

```
nvidia@tegra-ubuntu:~$ python3
Python 3.5.2 (default, Nov 12 2018, 13:43:14)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> quit()
nvidia@tegra-ubuntu:~$ python2
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> quit()
```

see more

^ | v • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Hướng Đỗ Văn • 2 months ago

You just need to copy the cv2 python library file to your virtual environment. Taking python3 as example, do the following and `import cv2` should work for python3 in your `env_tensorflow` virtual environment.

```
$ cp /usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-aarch64-linux-gnu.so $HOME/env_tensorflow/lib/python3.5/site-packages/
```

^ | v • Reply • Share ›

Roman Sandler • 3 months ago

I am trying to use multiple OpenCV windows in python on my Jetson TX2. However, I get the error: "ASSERT: "false" in file qasciikey.cpp, line 495"

Below is the minimum reproducible code:

```
import cv2
import numpy as np

img1=np.random.randn(300,400)
img2=np.random.randn(600,400)

cv2.imshow('win1', img1)
cv2.imshow('win2', img2)
```

And here is the error trace:

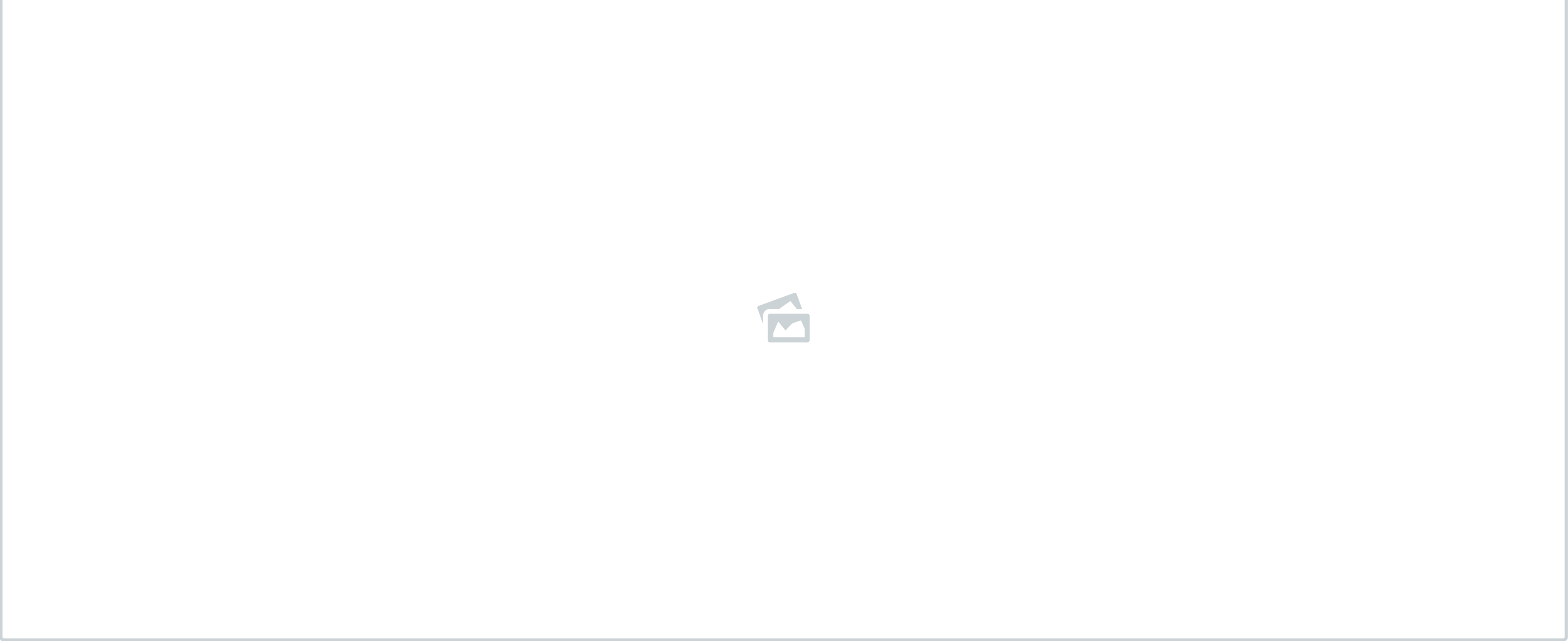
```
Could not initialize OpenGL for RasterGLSurface, reverting to RasterSurface.
QXcbConnection: XCB error: 145 (Unknown), sequence: 164, resource id: 0, major code: 139 (Unknown), minor code: 20
Could not initialize OpenGL for RasterGLSurface, reverting to RasterSurface.
ASSERT: "false" in file qasciikey.cpp, line 495
Aborted (core dumped)
```

Please help. Thanks!

^ | v • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Roman Sandler • 3 months ago

I was not able to reproduce this error. I added a line of 'cv2.waitKey(0)' at the end of your python script. It ran OK on my JTX2.



^ | ⌄ • Reply • Share ›

Roman Sandler ➦ Jiun-Kuei Jung • 3 months ago
@Jiun-Kuei Jung I realized the error only happened when I SSH'd into the Jetson and viewed the output via X11. If I viewed the output directly from the Jetson, I did not get the error..
^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Roman Sandler • 3 months ago
I wonder whether installing OpenGL related libraries on your remote Linux PC would solve the problem. For example,

```
### Note: The following is for x64 PC, not Jetson TX2.
$ sudo apt-get update
$ sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev
```

^ | ⌄ • Reply • Share ›

Roman Sandler ➦ Jiun-Kuei Jung • 3 months ago
Unfortunately, my remote PC is Windows (I am using a MobaXterm client), so not sure how to install these there..
^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Roman Sandler • 3 months ago
Maybe you can try to use VNC to operate the Jetson TX2 remotely.

Reference: https://devtalk.nvidia.com/...
^ | ⌄ • Reply • Share ›

Roman Sandler • 3 months ago
This was extreemely helpful. Thanks!
^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Roman Sandler • 3 months ago
You are very welcome. And thanks for letting me know.
^ | ⌄ • Reply • Share ›

Jorge Alberto HERNANDEZ TAPIA • 4 months ago
Hello, excellent guide, one question, how can ibuild openCV 3.4.0 with support for cv::cuda?
^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Jorge Alberto HERNANDEZ TAPIA • 4 months ago
Since we are building opencv with these cmake flags in this tutorial: '-D WITH_CUDA=ON -D CUDA_ARCH_BIN="6.2" -D CUDA_ARCH_PTX="" '. The resulting opencv library should support cv::cuda functions already.
^ | ⌄ • Reply • Share ›

João Paulo Abdalla • 5 months ago
How can I add modules from the 'Opencv_contrib' on the original OpenCV?
^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ João Paulo Abdalla • 4 months ago
In that case, I think you cannot use the opencv libraries which come from JetPack installation or from `apt-get install`. You'll have to remove your old installation of opencv, download source code and build/install opencv (remember to specify -D OPENCV_EXTRA_MODULES_PATH=...) by yourself.
^ | ⌄ • Reply • Share ›

smit • 6 months ago
Not able to open video stream using cvCaptureFromFile() function in C opencv 3.4.1 on jetson . Get following error :

```
[ WARN:0] cvCreateFileCaptureWithPreference: backend FFMPEG doesn't support legacy API anymore.
NvMMLiteOpen : Block : BlockType = 261
TVMR: NvMMLiteTVMRDecBlockOpen: 7907: NvMMLiteBlockOpen
NvMMLiteBlockCreate : Block : BlockType = 261
TVMR: cbBeginSequence: 1223: BeginSequence 1280x720, bVPR = 0
TVMR: LowCorner Frequency = 100000
TVMR: cbBeginSequence: 1622: DecodeBuffers = 10, pnvsi->eCodec = 4, codec = 0
TVMR: cbBeginSequence: 1693: Display Resolution : (1280x720)
```

TVMR: cbBeginSequence: 1694: Display Aspect Ratio : (1280x720)
TVMR: cbBeginSequence: 1762: ColorFormat : 5
TVMR: cbBeginSequence:1776 ColorSpace = NvColorSpace_YCbCr601
TVMR: cbBeginSequence: 1904: SurfaceLayout = 3
TVMR: cbBeginSequence: 2005: NumOfSurfaces = 17, InteraceStream = 0, InterlaceEnabled = 0, bSecure = 0, MVC = 0 Semiplanar = 1, bReinit = 1, BitDepthForSurface = 8 LumaBitDepth = 8, ChromaBitDepth = 8, ChromaFormat = 5
TVMR: cbBeginSequence: 2007: BeginSequence ColorPrimaries = 2, TransferCharacteristics = 2, MatrixCoefficients = 2
Allocating new output: 1280x720 (x 17), ThumbnailMode = 0
OPENMAX: HandleNewStreamFormat: 3464: Send OMX_EventPortSettingsChanged : nFrameWidth = 1280, nFrameHeight = 720

see more

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ smit • 6 months ago

For opening an RTSP stream with opencv on Jetson TX2, I'd recommend using gstreamer pipelines. You can find examples in my later posts.

^ | ⌄ • Reply • Share ›

smit ➦ Jiun-Kuei Jung • 6 months ago

Not able to open video stream as well. Can u tell what exactly the error is and how to resolve it. It used to work fine in opencv 3.3.0 . After updating Opencv to 3.4.1 i am getting this error

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ smit • 6 months ago

Could you try the following from a command line to make sure you have all necessary gstreamer elements and libraries installed properly on your JTX2 system?

```
$ gst-launch-1.0 rtspsrc location=rtsp://XXX:YYY@x.y.z.u:554/cam/realmonitor?channel=10&subtype=0 ! rtph264depay ! h264p
```

^ | ⌄ • Reply • Share ›

Yoko • 8 months ago

Heyy. Thanks for sharing!! When I was try to cmake and compile, confuguring incomplete and this error occured. Im new with JetsonTX2. What can I do for solve this problem ?

Looking for ccache - not found
-- Found ZLIB: /usr/lib/aarch64-linux-gnu/libz.so (found suitable version "1.2.8", minimum required is "1.2.3")
-- Found ZLIB: /usr/lib/aarch64-linux-gnu/libz.so (found version "1.2.8")
CMake Warning at cmake/OpenCVFindLibsGUI.cmake:18 (find_package):
By not providing "FindQt5Core.cmake" in CMAKE_MODULE_PATH this project has
asked CMake to find a package configuration file provided by "Qt5Core", but
CMake did not find one.

Could not find a package configuration file provided by "Qt5Core" with any
of the following names:

Qt5CoreConfig.cmake
qt5core-config.cmake

Add the installation prefix of "Qt5Core" to CMAKE_PREFIX_PATH or set
"Qt5Core_DIR" to a directory containing one of the above files. If

see more

^ | ⌄ • Reply • Share ›

akmal Hisyam • 8 months ago

Hi everyone,

Firstly, many thanks to **@Jiun-Kuei Jung** for this complete guide on how to install OpenCV (3.4.0) on Jetson TX2 with CUDA support.

I did the installation steps and I verified the installation, then I wanted to compile the "samples" folder found in opencv-3.4.0 with "make -j4" after doing "cmake ." in the "samples" folder, but I got this :

see more

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ akmal Hisyam • 8 months ago

Once you have opencv built and installed properly on JTX2, you can build individual opencv sample programs by doing something like this (taking the gpu version of `hog` for example):

```
$ cd ~/src/opencv-3.4.0/samples/gpu
$ g++ -o hog -O2 -Wall hog.cpp -lopencv_core -lopencv_imgproc -l opencv_flann \
                      -l opencv_imgcodecs -lopencv_videoio -lopencv_highgui -lopencv_ml \
                      -lopencv_video -lopencv_objdetect -lopencv_photo -lopencv_features2d \
                      -lopencv_calib3d -lopencv_stitching -lopencv_videostab -lopencv_shape \
                      -lopencv_cudaobjdetect -lopencv_cudawarping -lopencv_cudaimgproc
$ ./hog --video ../data/vtest.avi
```

Alternatively you can go to the opencv build directory, redo cmake with BUILD_EXAMPLES=ON, and build all sample programs at once.

```
$ cd ~/src/opencv-3.4.0/build
$ cmake (...other flags omitted...) -D BUILD_EXAMPLES=ON ..
$ make -j4
```

^ | ⌄ • Reply • Share ›

akmal Hisyam ➦ Jiun-Kuei Jung • 8 months ago

Thanks a bunch !

^ | ⌄ • Reply • Share ›

robertzzz • 8 months ago

Never mind - the file on disk was corrupted. I used `debsums -s` and it found it. I retired the suspect hardware and it's working fine now.

Hi,
I'm trying to install version 3.4.1 on my Jetson with the latest JetPack release, 3.2.
Has anyone seen the following problem? Any idea what could be the culprit?

[code]
make -j 4
...
[31%] Linking CXX shared library ../../lib/libopencv_cudev.so
/usr/bin/ld: error: /usr/local/cuda-9.0/lib64/libnppitc.so: ELF section name out of range

collect2: error: ld returned 1 exit status
modules/cudev/CMakeFiles/opencv_cudev.dir/build.make:86: recipe for target 'lib/libopencv_cudev.so.3.4.1' failed
make[2]: *** [lib/libopencv_cudev.so.3.4.1] Error 1
...
[/code]

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ robertzzz • 8 months ago

I have tried to build opencv-3.4.1 on Jetson TX2 before, but do not recall seeing similar problems. (I prefer opencv-3.4.0 since YOLOv2/YOLOv3 doesn't compile against opencv-3.4.1.) Maybe you could also post this question onto NVIDIA's developer forum.

^ | ⌄ • Reply • Share ›

Aaron Romain • a year ago

Hi,

I figured out how to make this work on Jetpack 3.2 developer preview

Change

$ sudo apt-get purge libopencv4tegra-python libopencv4tegra-dev \
libopencv4tegra
$ sudo apt-get purge libopencv4tegra-repo

to

$ sudo apt-get purge libopencv-python libopencv-dev \
libopencv
$ sudo apt-get purge libopencv-repo

and change

$ sudo vim /usr/local/cuda-8.0/include/cuda_gl_interop.h

to

$ sudo vim /usr/local/cuda-9.0/include/cuda_gl_interop.h

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Aaron Romain • a year ago

Thanks a lot for this sharing.

^ | ⌄ • Reply • Share ›

Louis-D Cypher • a year ago

Compiles however with errors - should something be done for this or simply ignored.

Example:

cc1plus: warning: /home/nvidia/src/opencv-3.4.0/build/modules/ml/precomp.hpp.gch/opencv_ml_RELEASE.gch: not used because `OPENCV_TRAITS_ENABLE_DEPRECATED' is defined [-Winvalid-pch]

^ | ⌄ • Reply • Share ›

Jiun-Kuei Jung Mod ➦ Louis-D Cypher • a year ago

I saw the same warning during the build process. I think it could be safely ignored.

^ | ⌄ • Reply • Share ›

Robert Abramson • a year ago

Thanks for this article! I was able to use this procedure with JetPack 3.2 and CUDA 9.0, which also ships OpenCV without Python 3 bindings. The only difference is the location of the CUDA/GL header file that must be patched: it's located at `/usr/local/cuda-9.0/include/cuda_gl_interop.h` since there's a new CUDA version.

Reply • Share ›

mike ➦ Robert Abramson • 8 months ago

Hi Robert! I'm getting similar errors. Do you know what files I would have to change?

Reply • Share ›

Robert Abramson ➦ mike • 8 months ago

Hi Mike,
It's been a while since I played with this and I've honestly forgotten most of what I did. The patching thing has to do with this patching procedure, which is referenced in the article: https://devtalk.nvidia.com/...

The critical bit is that I built mine with CUDA 9.0, so when I patched this file, it was located at `/usr/local/cuda-9.0/include/cuda_gl_interop.h` instead of `/usr/local/cuda-8.0/include/cuda_gl_interop.h`. The content of the patch was the same.

I have no idea whether this advice still works! It worked 4 months ago...

Reply • Share ›

Kai Shang Wang • a year ago

Thanks for sharing, I followed the step and everything worked good before I reboot TX2,
After I reboot it, here comes Error when "import cv2", here is the error say:

```
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: /usr/lib/python3.5/dist-packages/cv2.cpython-35m-aarch64-lin...: undefined symbol: _ZTIN2cv3dnn19experimental_dnn_v15LayerE
```

need some help , any suggestion, thank you~

Reply • Share ›

Jiun-Kuei Jung Mod ➦ Kai Shang Wang • a year ago

I'm not exactly sure why you encountered this error. But if you are not going to use dnn stuffs in opencv, you can safely disable that module by adding '-D BUILD_opencv_dnn=OFF' into the cmake command and rebuild opencv.

By the way, which version of opencv were you using?

Reply • Share ›

Kemal Bozkurt • a year ago

MY program gives me blelow error with eclipse C++

```
Failed to connect to Mir: Failed to connect to server socket: No such file or directory
Unable to init server: Could not connect: Connection refused

(Frame 1:31791): Gtk-WARNING **: cannot open display:
```

Reply • Share ›

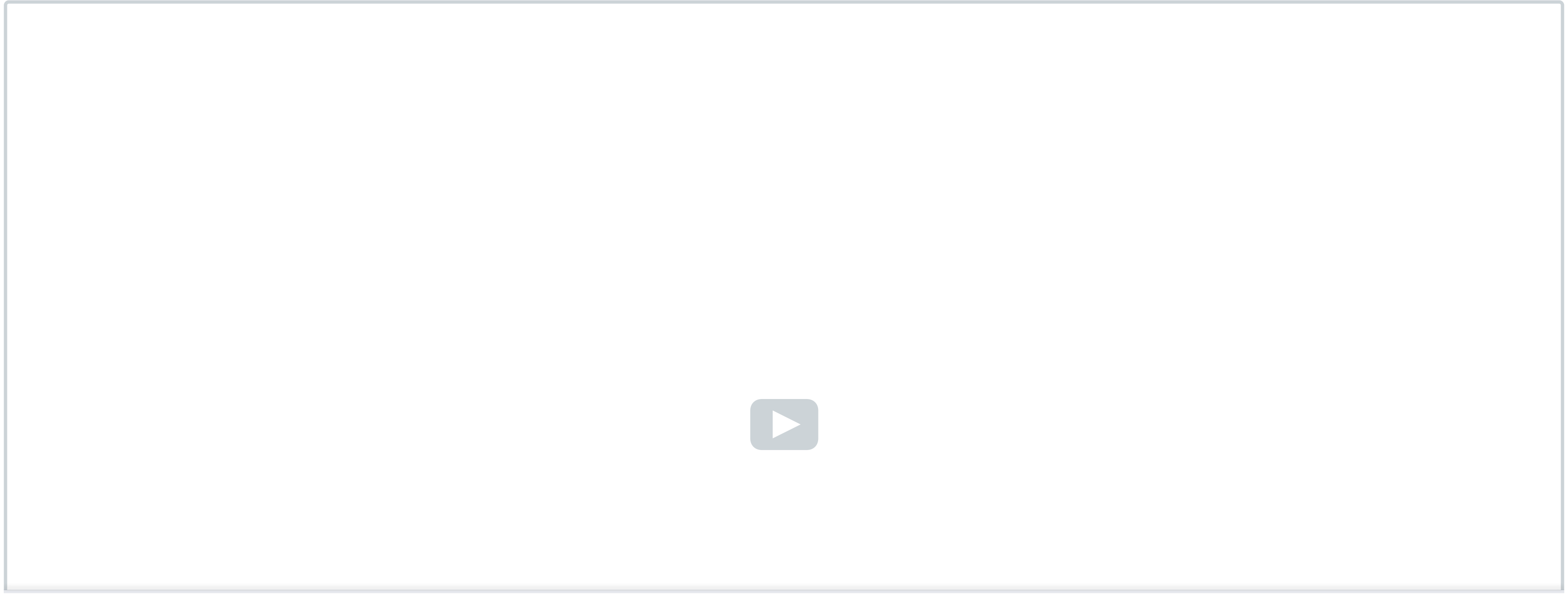
Jiun-Kuei Jung Mod ➦ Kemal Bozkurt • a year ago

I think you probably need to do 'export DISPLAY=:0'. Please google for details.

Reply • Share ›

Kemal Bozkurt ➦ Jiun-Kuei Jung • a year ago

I try



see more

Reply • Share ›

Kemal Bozkurt • a year ago

Have you got different opencv-3.3.0.zip file
I think the problem is here

Reply • Share ›

Jiun-Kuei Jung Mod ➦ Kemal Bozkurt • a year ago

In addition to the GitHub link shown in the post, you could also download opencv-3.3.0 source code from this page: https://opencv.org/releases...

Reply • Share ›

Load more comments

ALSO ON **HTTPS://JKJUNG-AVT.GITHUB.IO/**

Training a Fish Detector with NVIDIA DetectNet (Part 2/2)

19 comments • a year ago

Avatar **Jiun-Kuei Jung** — Glad to know that you've found the answer by yourself.

YOLOv3 on Jetson TX2

42 comments • 10 months ago

Avatar **Jiun-Kuei Jung** — My colleague has applied TensorRT on SSD. As a result, we are able to achieve pretty accurate object detection at >30 fps on JTX2. You can refer to …

Adapting the Hand Detector Tutorial to Your Own Dataset

2 comments • 5 months ago

Avatar **Jiun-Kuei Jung** — The learning rates in TensorFlow detection model zoo examples and in my hand detector example should be good starting points to try. You …

Multi-threaded Camera Caffe Inferencing

2 comments • 8 months ago

Avatar **Jiun-Kuei Jung** — You're welcome. I'm really delighted to see this comment.

Subscribe **Add Disqus to your siteAdd DisqusAdd** **Disqus' Privacy PolicyPrivacy PolicyPrivacy**

blog built using the cayman-theme by Jason Long. LICENSE