Install

Reference:

<https://learnopencv.com/opencv-dnn-with-gpu-support/>

Windows:

<https://learnopencv.com/how-to-use-opencv-dnn-module-with-nvidia-gpu-on-windows/>

<https://machinelearningprojects.net/build-opencv-with-cuda-and-cudnn/#Step_2_%E2%80%93_Download_Visual_Studio>

1. Install anaconda
2. Clone opencv & opencv-contib repository

git clone <https://github.com/opencv/opencv.git>

git clone <https://github.com/opencv/opencv_contrib.git>

A screenshot of a computer screen

Description automatically generated

1. Run Cmake GUI

* In Where is the source code, Select the main opencv extracted folder.
* In Where to build the binaries, Select the empty build folder we created above.

A screenshot of a computer

Description automatically generated

* Click on the Configure button.
* I have Visual Studio 2022 installed, so I selected Visual Studio 17 2022.
* In the Optional platform for the generator, select x64.

A screenshot of a computer program

Description automatically generated

* Click on Finish and it will start configuring.
* Once done you will see a screen like this.

A screenshot of a computer

Description automatically generated

***Now search the following***…

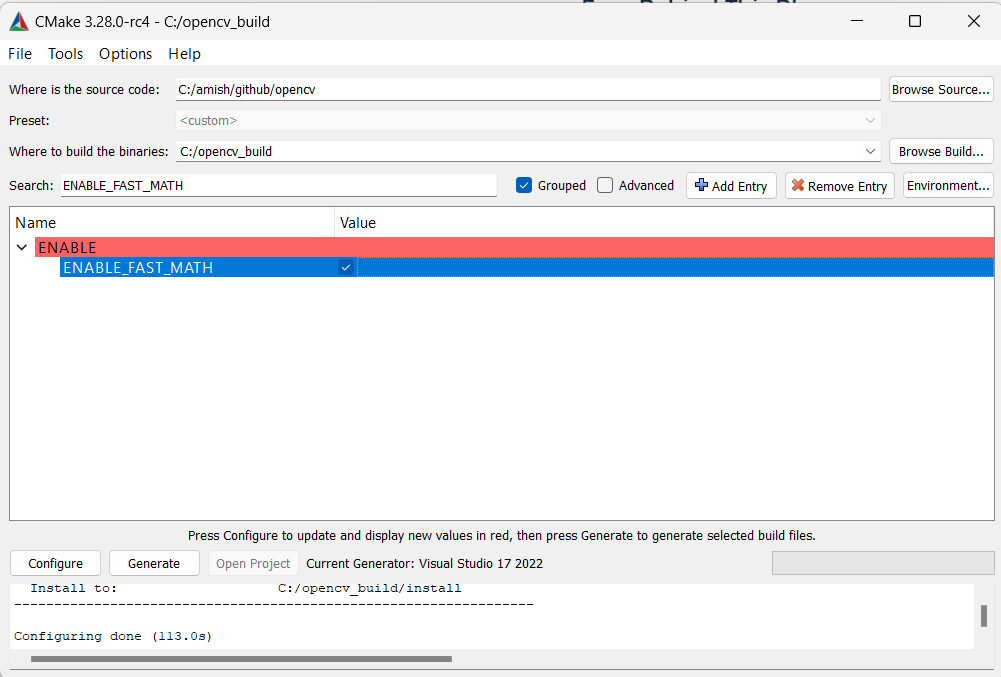
* **WITH\_CUDA**and tick/check it.



* **OPENCV\_DNN\_CUDA**and tick/check it.



* **ENABLE\_FAST\_MATH**and tick/check it.



* **OPENCV\_EXTRA\_MODULES\_PATH:** and browse to the module folder in opencv-contrib we extracted in step 3.



**And now again hit on the Configure button.**

A screenshot of a computer

Description automatically generated

* Once you see, **Configuring done**, check **CUDA\_FAST\_MATH** also.

A blue line on a pink background

Description automatically generated

* Now go to [this link](https://en.wikipedia.org/wiki/CUDA#:~:text=GPUs%20supported%5Bedit%5D) and check your compute capability against your graphic card.
* Mine is **NvidiaRTX 4060**, so my arch is **8.9**

A close up of a white background

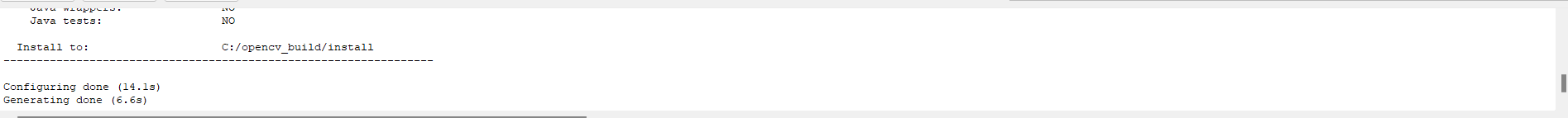
Description automatically generated

* Delete all values except your Compute Capability.

A screen shot of a computer

Description automatically generated

* **Again hit on the Configure button for the final time.**
* **Once done, finally hit the Generate button.**



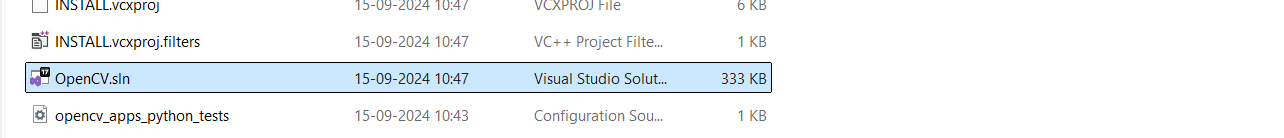
* Now you will see a lot of files in your **build**folder.

A screenshot of a computer

Description automatically generated

**Step 4 – Build OpenCV with CUDA**

* We will have a file like this **OpenCV.sln**.

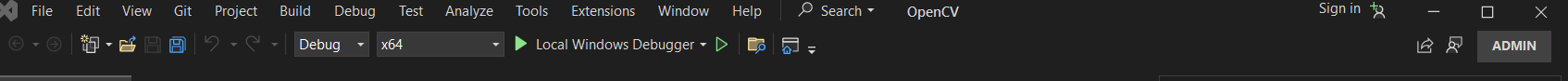


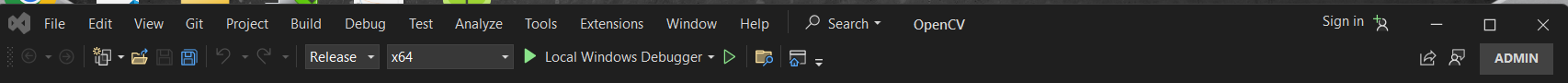
* Now open the CMD Terminal in **Administrator mode** from the start menu.
* Change the directory to your build folder using **cd**command and run **OpenCV.sln**.
* It will open up Visual Studio.

A screenshot of a computer

Description automatically generated

* Once Visual Studio is opened, change**Debug to Release**.





* Now in the right sidebar, open the**CMake Targets** dropdown and you will see **ALL\_BUILD**.
* Right click on it and select **build**and it will start building our binaries.
* This process will take some time.

A screenshot of a computer

Description automatically generated

* Once done it will prompt this.

A screenshot of a computer

Description automatically generated

* Now right click on **INSTALL**and build.

A screenshot of a computer

Description automatically generated

Step 5 – Check OpenCV Installation.

* Congratulations, you have successfully installed OpenCV with Cuda support.
* Now let’s check if python is detecting cv2 or not.
* Open cmd and paste the following commands.



A screenshot of a computer

Description automatically generated

* Now let’s check if cv2 is detecting CUDA or not.
* Create a**test.py**file and paste the following code in it and run it.

import numpy as np

import cv2 as cv

import time

npTmp = np.random.random((1024, 1024)).astype(np.float32)

npMat1 = np.stack([npTmp,npTmp],axis=2)

npMat2 = npMat1

cuMat1 = cv.cuda\_GpuMat()

cuMat2 = cv.cuda\_GpuMat()

cuMat1.upload(npMat1)

cuMat2.upload(npMat2)

start\_time = time.time()

for i in range(100):

cv.cuda.gemm(cuMat1, cuMat2,1,None,0,None,1)

print("CUDA --- %s seconds ---" % (time.time() - start\_time))

start\_time = time.time()

for i in range(100):

cv.gemm(npMat1,npMat2,1,None,0,None,1)

print("CPU --- %s seconds ---" % (time.time() - start\_time))

A black screen with white text

Description automatically generated