**Documentation: Installation of OpenCV with CUDA using Microsoft Visual Studio in Windows**

***Prepared by: Lim Jun Jie***

1. **Prerequisites**
2. Installed Python
3. Installed NVIDIA GPU drivers and CUDA development Kit
4. Installed Microsoft Visual Studio and its relative Build Tools
5. Installed C-Make
6. **Installation of prerequisites**
7. Installation of OpenCV and OpenCV-contrib

* Download zip file of OpenCV from <https://opencv.org/releases/>, choose “Sources” under the version of OpenCV you want to download.
* Download zip file of OpenCV-contrib from <https://github.com/opencv/opencv_contrib/releases>, choose the relative version with the zip file of OpenCV (opencv-4.1.2 with opencv\_contrib-4.1.2)

1. Installation of Python and CUDA development Kit

* Refer to this YouTube link <https://www.youtube.com/watch?v=qrkEYf-YDyI&list=PLjy4p-07OYzulelvJ5KVaT2pDlxivl_BN> if you haven’t set up for CUDA development Kit

1. Installation of Microsoft Visual Studio and Build Tools of Visual Studio

* Installation link: <https://visualstudio.microsoft.com/downloads/>

1. Installation of C-Make

* Installation link: <https://cmake.org/download/>

1. **Build OpenCV using C-Make**
2. Extracting OpenCV and OpenCV-contrib to any location you want, then open CMake-gui, set the

* Where is the source code: <path-to-file>/opencv-x.x.x
* Where to build the binaries: <path-to-file>/opencv-x.x.x/build

1. Click Configure, then

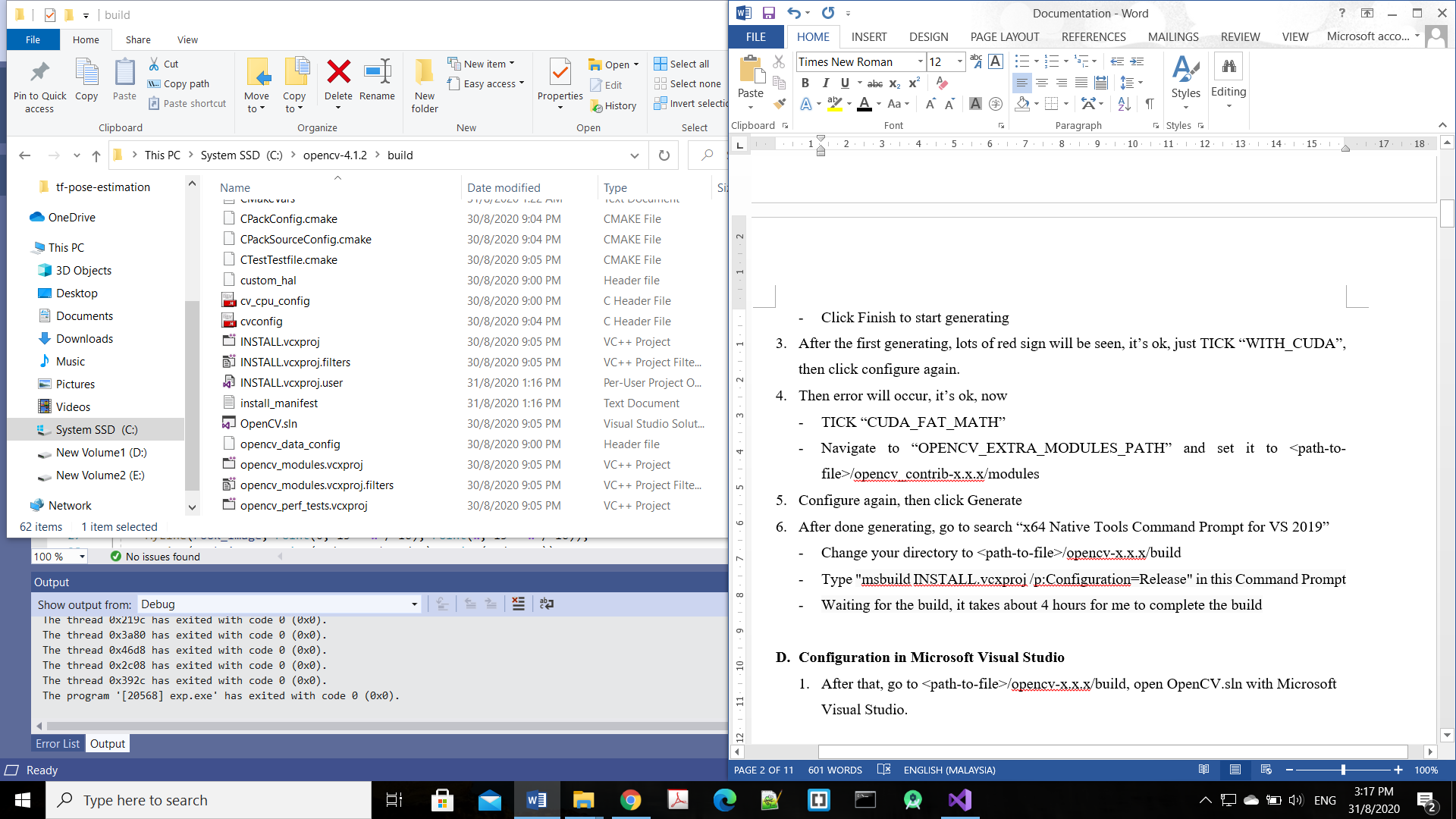
* Specify the generator for the project according to your Visual Studio version
* Optional platform for generator is depending on your OS, for example x64
* Choose “Use default native compilers”
* Click Finish to start generating

1. After the first generating, lots of red sign will be seen, it’s ok, just TICK “WITH\_CUDA”, then click configure again.
2. Then error will occur, it’s ok, now

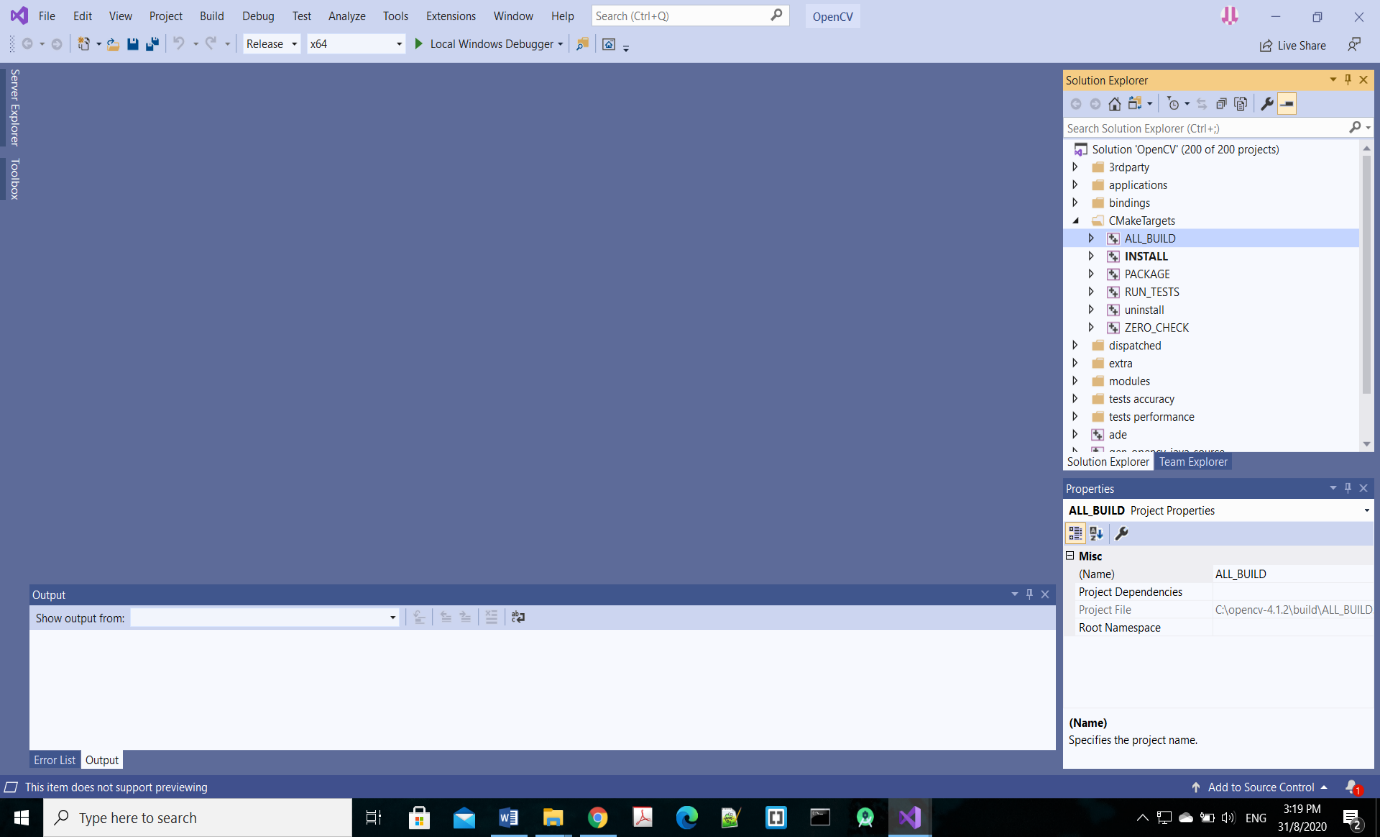
* TICK “CUDA\_FAT\_MATH”
* Navigate to “OPENCV\_EXTRA\_MODULES\_PATH” and set it to <path-to-file>/opencv\_contrib-x.x.x/modules

1. Configure again, then click Generate
2. After done generating, go to search “x64 Native Tools Command Prompt for VS 2019”

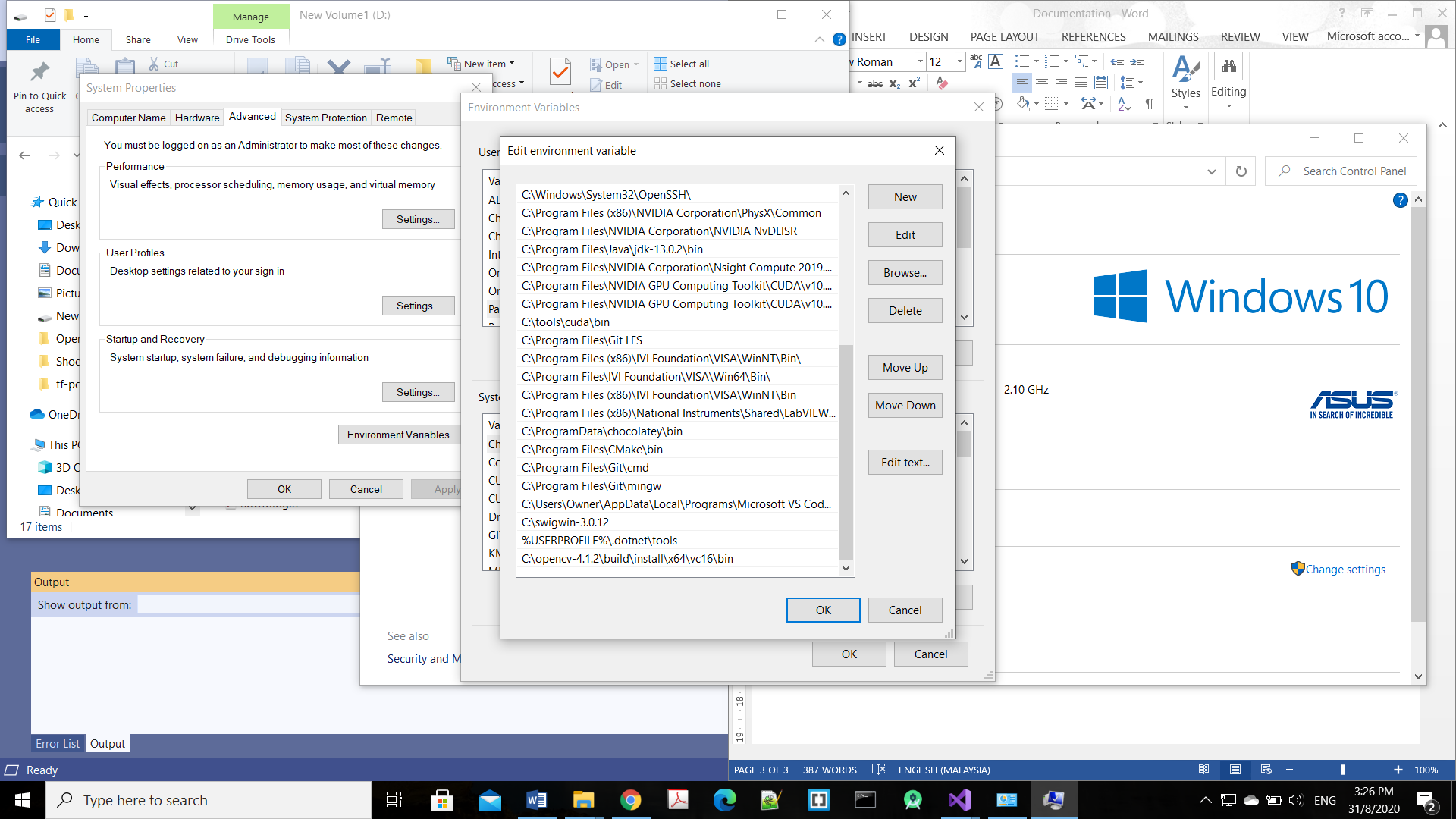
* Change your directory to <path-to-file>/opencv-x.x.x/build
* Type "msbuild INSTALL.vcxproj /p:Configuration=Release" in this Command Prompt
* Waiting for the build, it takes about 4 hours for me to complete the build

1. **Configuration in Microsoft Visual Studio**
2. After that, go to <path-to-file>/opencv-x.x.x/build, open OpenCV.sln with Microsoft Visual Studio.

* Set your VS environment as Release & x64 (red box)
* Then, right click “ALL\_BUILD”, click “Set as startup project”, then click “Build”, it takes long time too. (green box)
* After that, repeat the same process by choosing INSTALL. (green box)

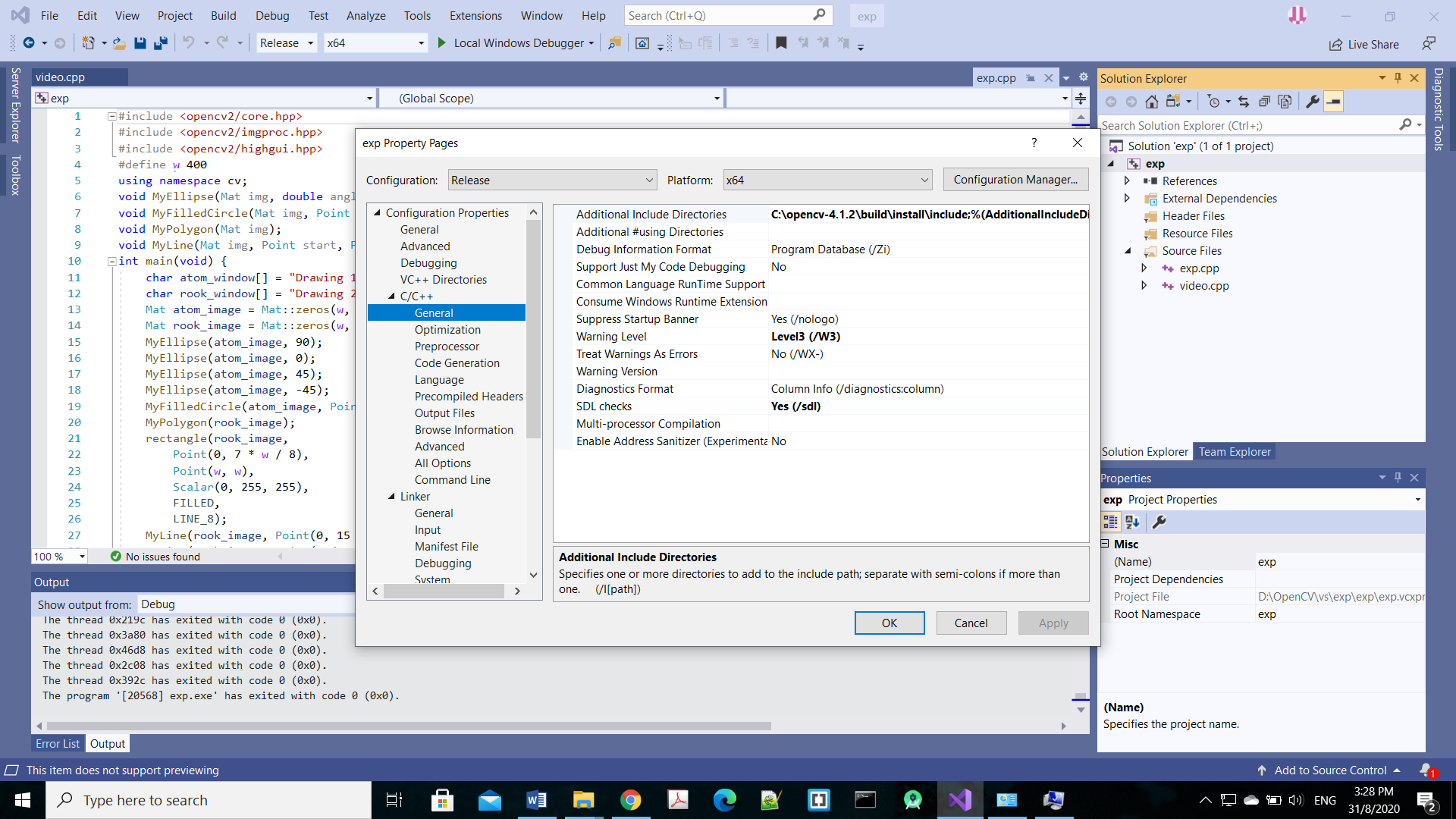


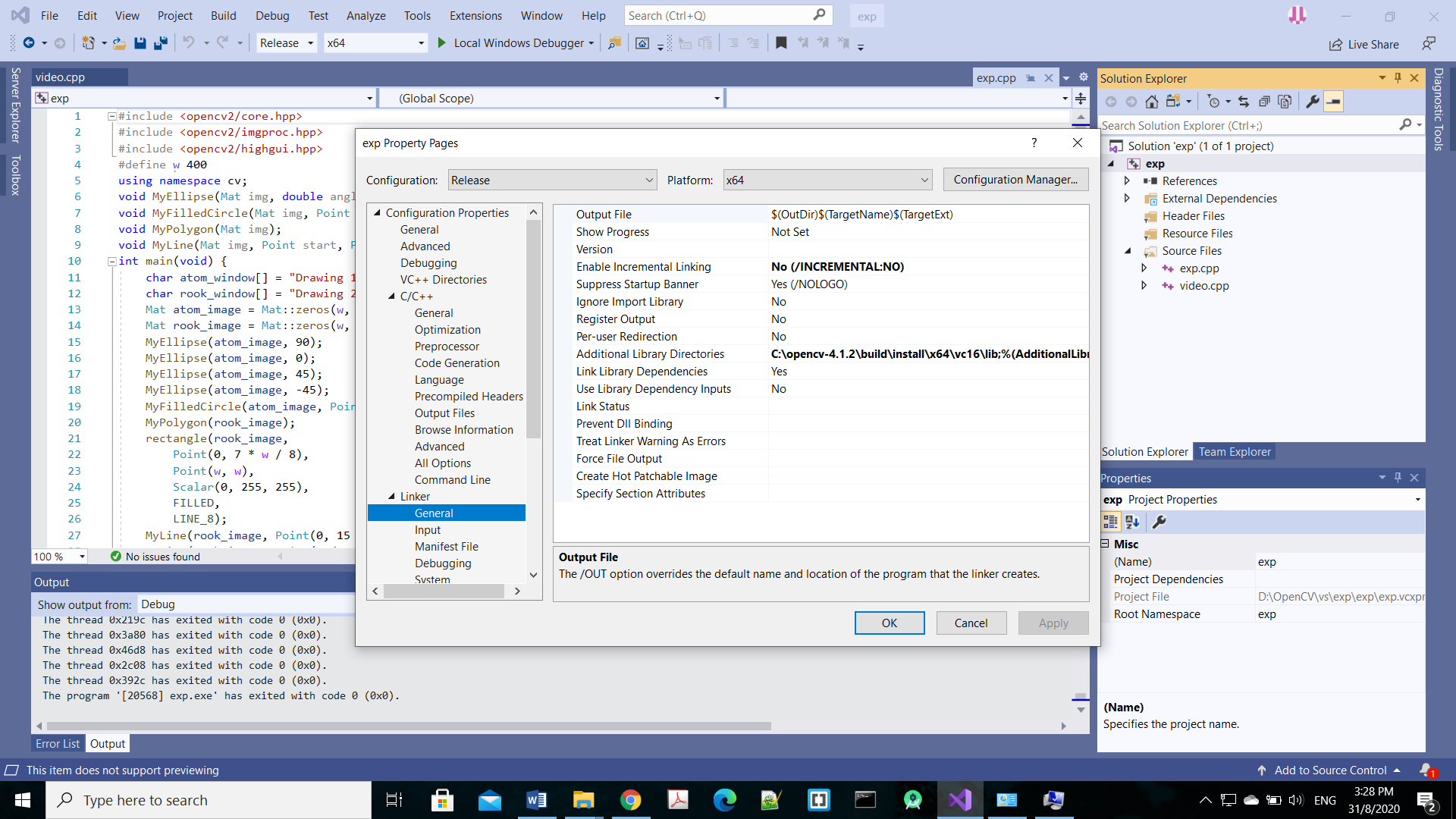
1. **Adding the path to Environment Variable**



# Add path <file-to-path>/opencv-x.x.x/build/install/x64/vc16/bin

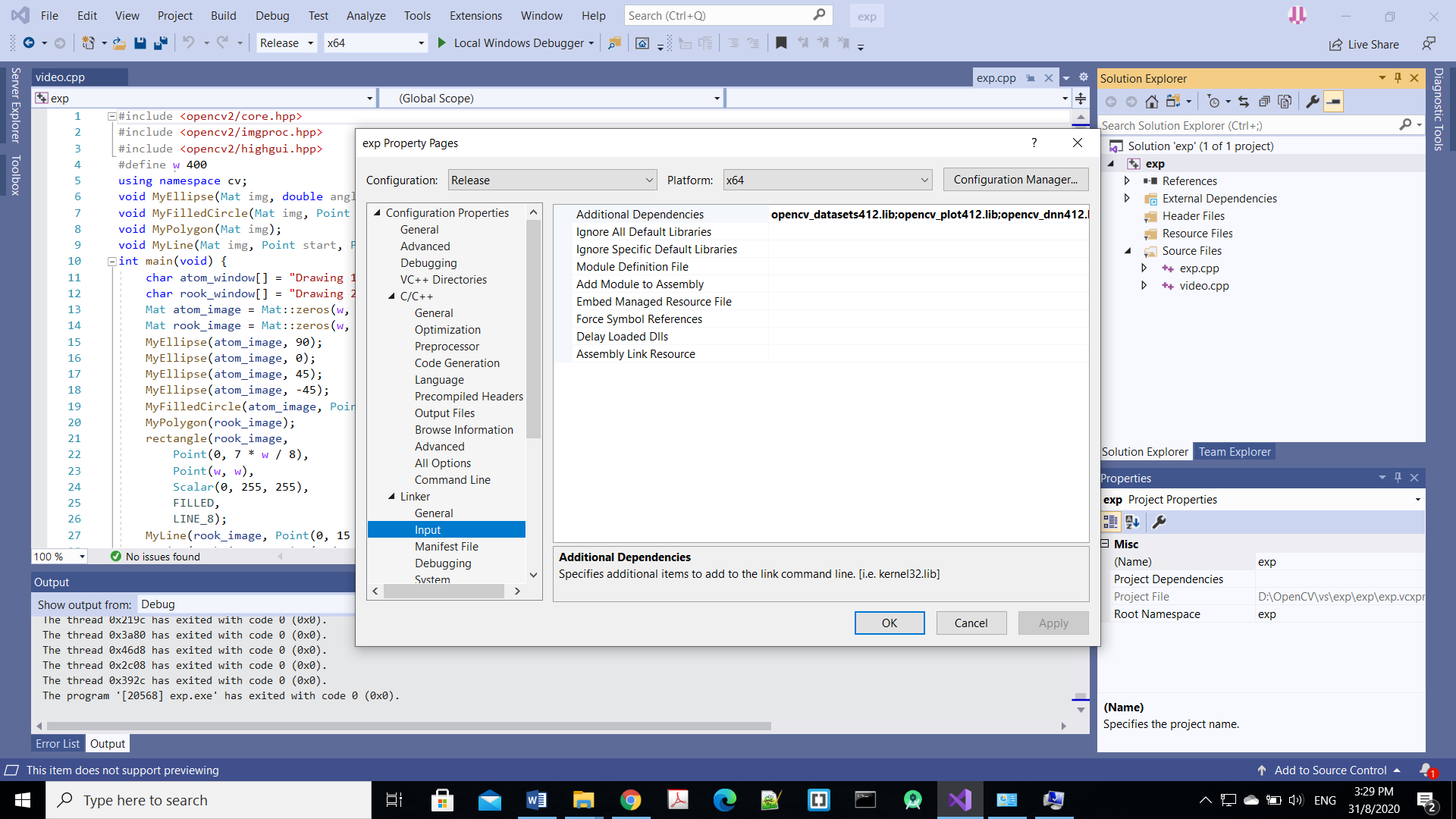
1. **Setting the dependencies of Microsoft Visual Studio**

* Now, you can create new OpenCV project by setting the relevant dependencies



# Set path as <path-to-file>/opencv-x.x.x/build/install/include (Purple box)

# Set path as <path-to-file>/opencv-x.x.x/build/install/x64/vc16/lib (Light blue box)



# Add all \*.lib files under <path-to-file>/opencv-x.x.x/build/install/x64/vc16/lib (black box)

### For the details of part E and F, can refer to the documentation of setting up the OpenCV using Microsoft Visual Studio without CUDA support in my github link https://github.com/JJLim99/OpenCV-with-and-without-CUDA-using-Micosoft\_VS.git

Main Reference: <https://www.youtube.com/watch?v=TT3_dlPL4vo>