

OpenCV Python

Windows Installation Manual



e-con Systems

Your Product Development Partner

Version 1.1

e-con Systems

8/14/2020

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

INTRODUCTION TO OPENCV	3
PREREQUISITES	3
DESCRIPTION	3
BUILDING OPENCV FOR PYTHON	4
STEP 1 – INSTALLING PYTHON	4
STEP 2 – LAUNCHING CMAKE WINDOW	6
STEP 3 – VISUAL STUDIO VERSION SELECTION	7
STEP 4 – CONFIGURE AND GENERATE WITH CMAKE	7
STEP 5 – VIDEOIO FILE REPLACE	9
STEP 6 – BUILDING BASECLASSES	9
STEP 7 – BUILD OPENCV IN VISUAL STUDIO	12
CONFIGURATION FOR PYTHON	13
TROUBLESHOOTING	16
SUPPORT	17

Introduction to OpenCV

Open Source Computer Vision Library (OpenCV) is an open source computer vision and machine learning software library. OpenCV libraries are used to communicate with Cameras. APIs introduced in the OpenCV can be supported with all e-con Systems cameras.

This document helps you to install OpenCV in Windows and build a sample code to access the camera with OpenCV.

Prerequisites

The prerequisites are as follows:

- Click here(<https://cmake.org/download/>) to download CMake.
- Download OpenCV from here(<https://github.com/opencv/opencv>).

Using git

Click on code option and copy the URL

```
$ git clone <OpenCV_URL>
$ cd opencv
$ git checkout <opencv_version(3.3.1 or 3.4.1)>
```

Direct Download

For 3.3.1 (<https://github.com/opencv/opencv/archive/3.3.1.zip>),
For 3.4.1 (<https://github.com/opencv/opencv/archive/3.4.1.zip>).

- Create a source directory in the opencv folder and move all the files to the source folder
- Create a build directory in the opencv/
- Build OpenCV in your PC using Visual Studio 2017.

Description

The following steps have been tested on Windows 10. It relatively works on other versions of Windows OS.

Building OpenCV for Python

OpenCV for Python is a sample python script used to demonstrate some of the features of the e-con Systems cameras with OpenCV APIs.

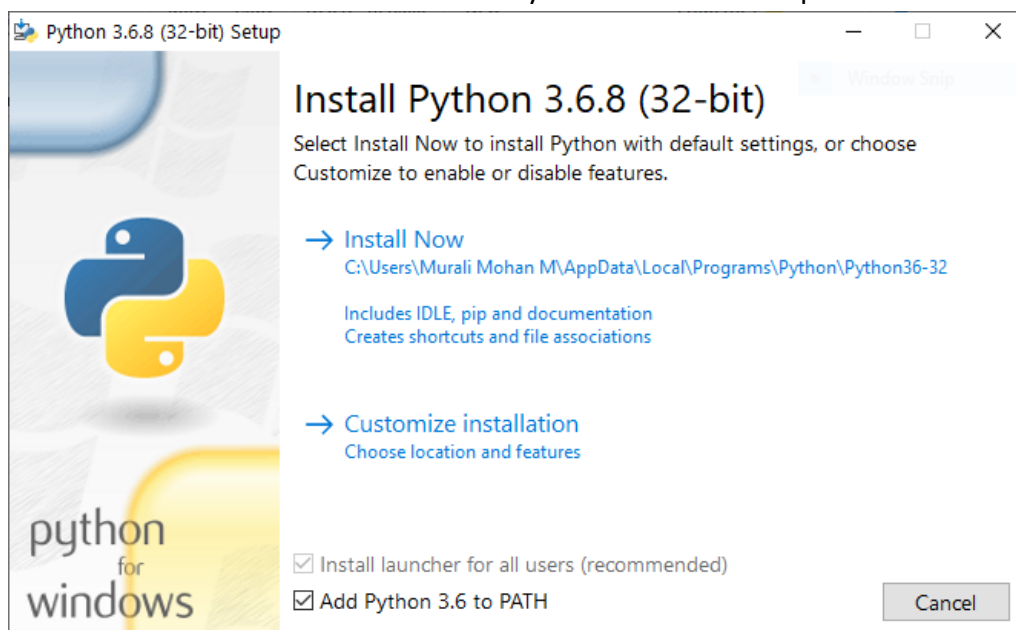
- Step 1. [Installing Python](#)
- Step 2. [Launching CMake Window](#)
- Step 3. [Visual Studio Version Selection](#)
- Step 4. [Configure and Generate with CMake](#)
- Step 5. [Replace Videoio File](#)
- Step 6. [Building BaseClasses](#)
- Step 7. [Build OpenCV](#)

Step 1 – Installing Python

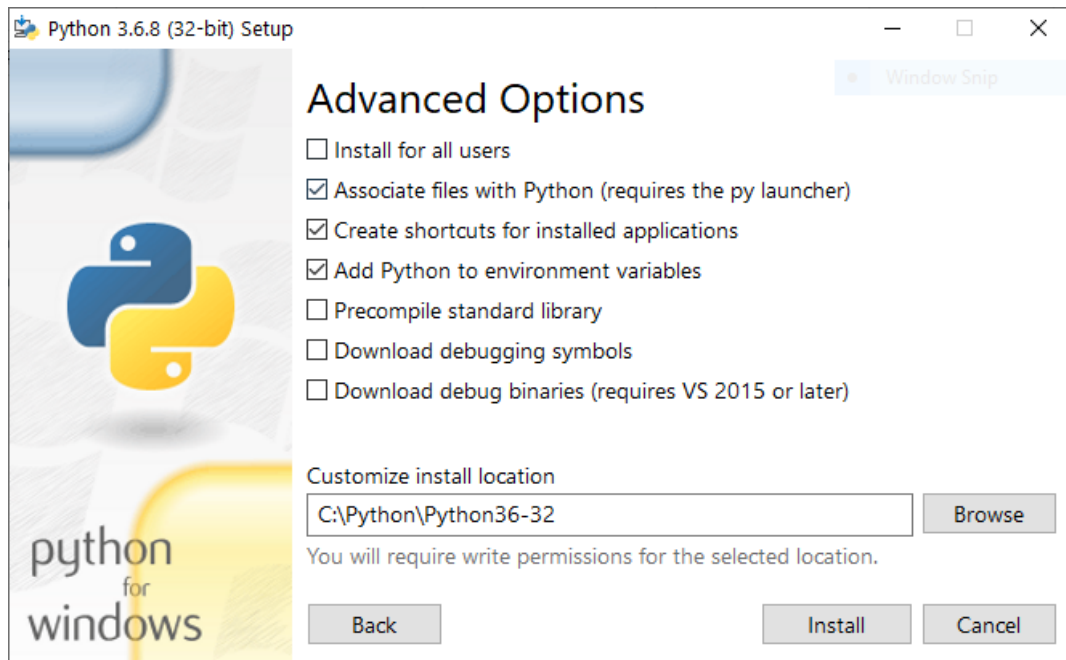
The steps to install the python in your PC as follows:

1. Installation of the python 3.6.8:

- a) Download the python 3.6.8 installer for windows in below link
<https://www.python.org/downloads/release/python-368/>
- b) Follow the steps while installing the python installer.
 - i. Check the “Add Python 3.6 to PATH” option.



- ii. Click customize installation.



III. Brows the path and click next.

2. If Multiple Python installers (32 & 64 bit) installed:

- a) If in your PC more python versions installed then make sure you are working python version path should be in top of the User variables.

Search for **Edit the system environment variables**



Click **Environment Variables**



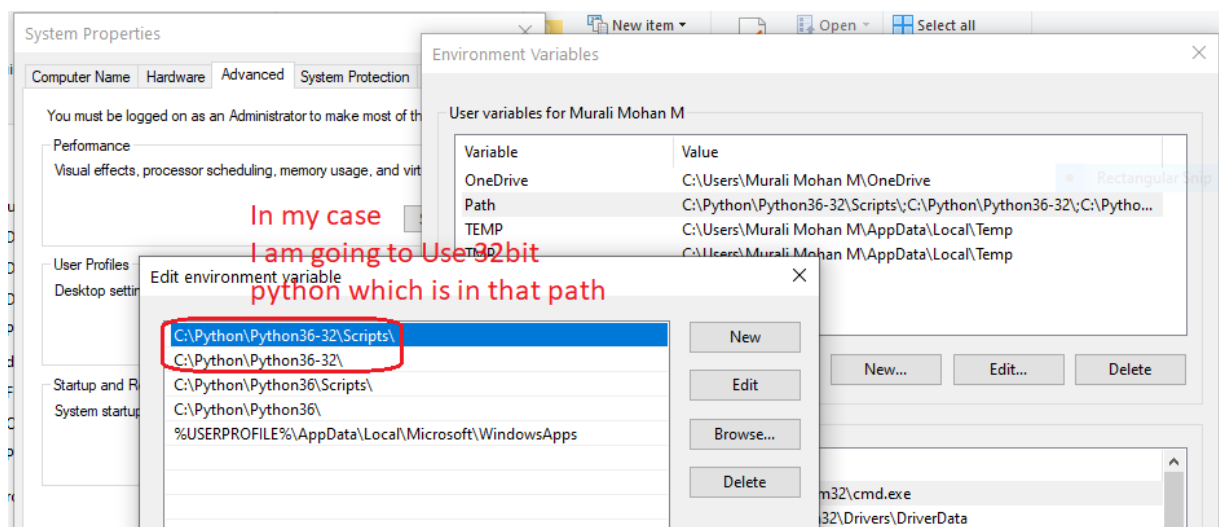
Under **User variables**



Click on **Path** and **Edit**

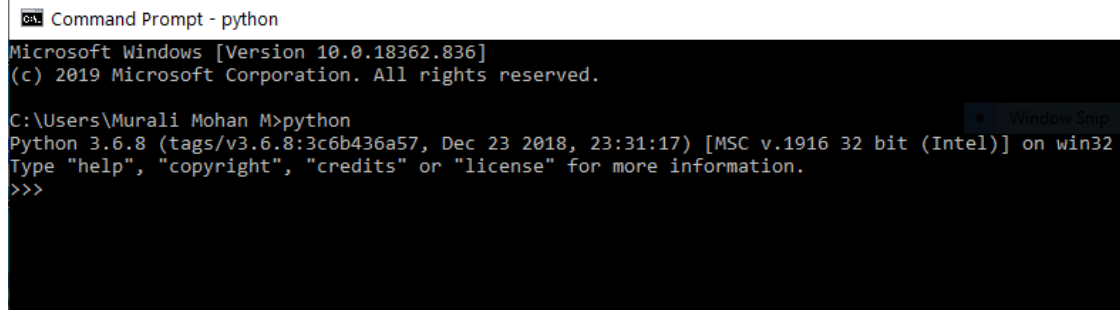


Adjust the Paths with **Move UP** or **Move Down**



3. Install the opencv for python (cv2):

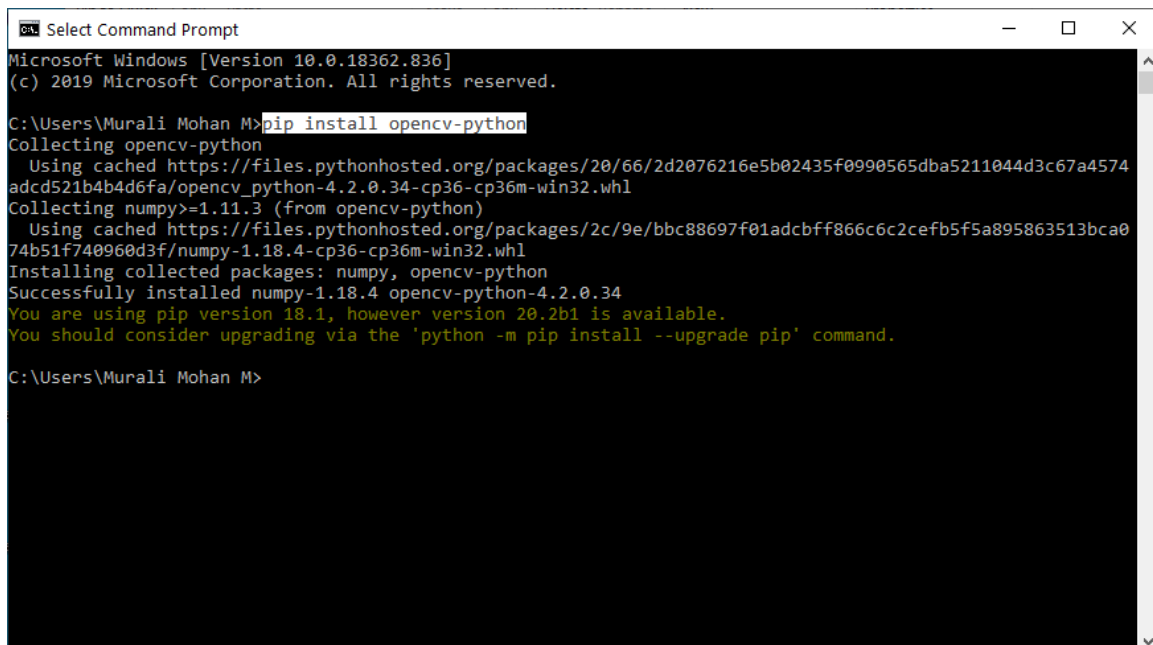
- I. Open Command prompt.
- II. Type python and give enter to check the python version.
- III. Make sure the installed python version is 3.6.8.



```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Murali Mohan M>python
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 23 2018, 23:31:17) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- IV. Give exit () to exit from python mode.
- V. Type "**pip install opencv-python**" and click enter, it will install the opencv for python.



```
Select Command Prompt
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Murali Mohan M>pip install opencv-python
Collecting opencv-python
  Using cached https://files.pythonhosted.org/packages/20/66/2d2076216e5b02435f0990565dba5211044d3c67a4574adcd521b4b4d6fa/opencv_python-4.2.0.34-cp36-cp36m-win32.whl
Collecting numpy>=1.11.3 (from opencv-python)
  Using cached https://files.pythonhosted.org/packages/2c/9e/bbc88697f01adcbff866c6c2cefb5f5a895863513bca074b51f740960d3f/numpy-1.18.4-cp36-cp36m-win32.whl
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.18.4 opencv-python-4.2.0.34
You are using pip version 18.1, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Murali Mohan M>
```

- VI. Type python and give enter to set the python mode.
- VII. Give import cv2, if this will not show any error then the opencv was successfully installed.
- VIII. Type "**cv2.__version__**" it will print the installed cv2 version.
- IX. It will show the latest cv2 version.

Step 2 – Launching CMake Window

In CMake window, select the OpenCV sources as source folder and OpenCV/build as build folder and click **Configure** button.

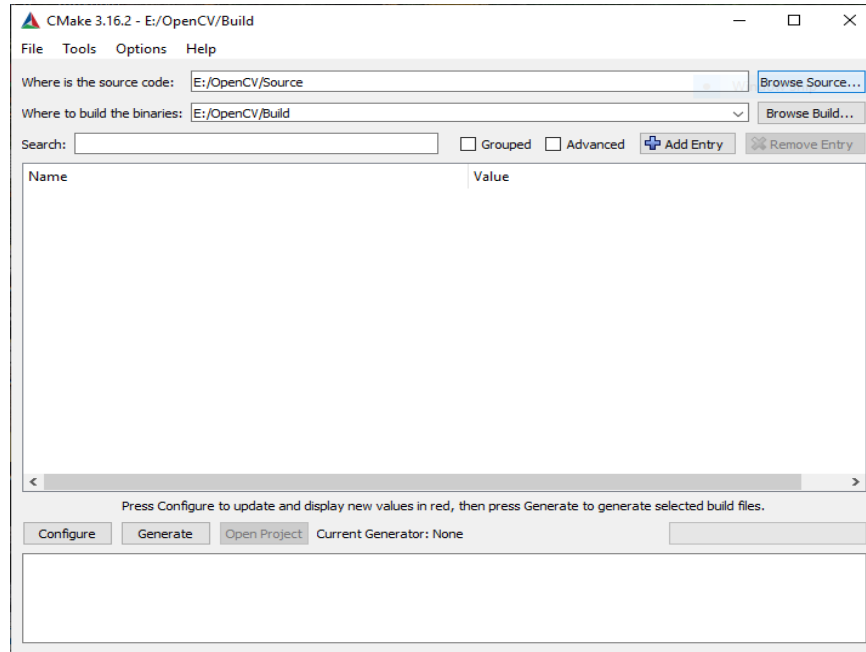


Figure 1: CMake Source and Build Directory Specification Window

Step 3 – Visual Studio Version Selection

A window prompting to select current Visual Studio version (VS2017) in your PC and x32 and x64 version appears. Select the appropriate options as shown below.

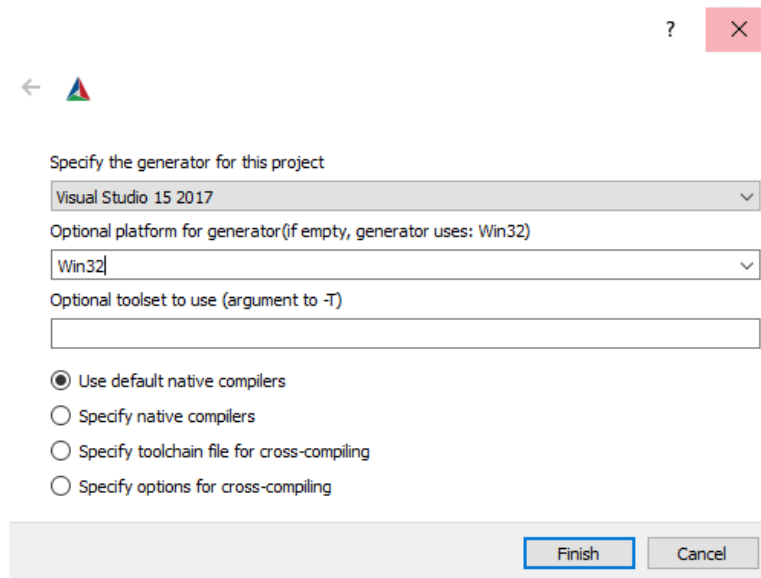
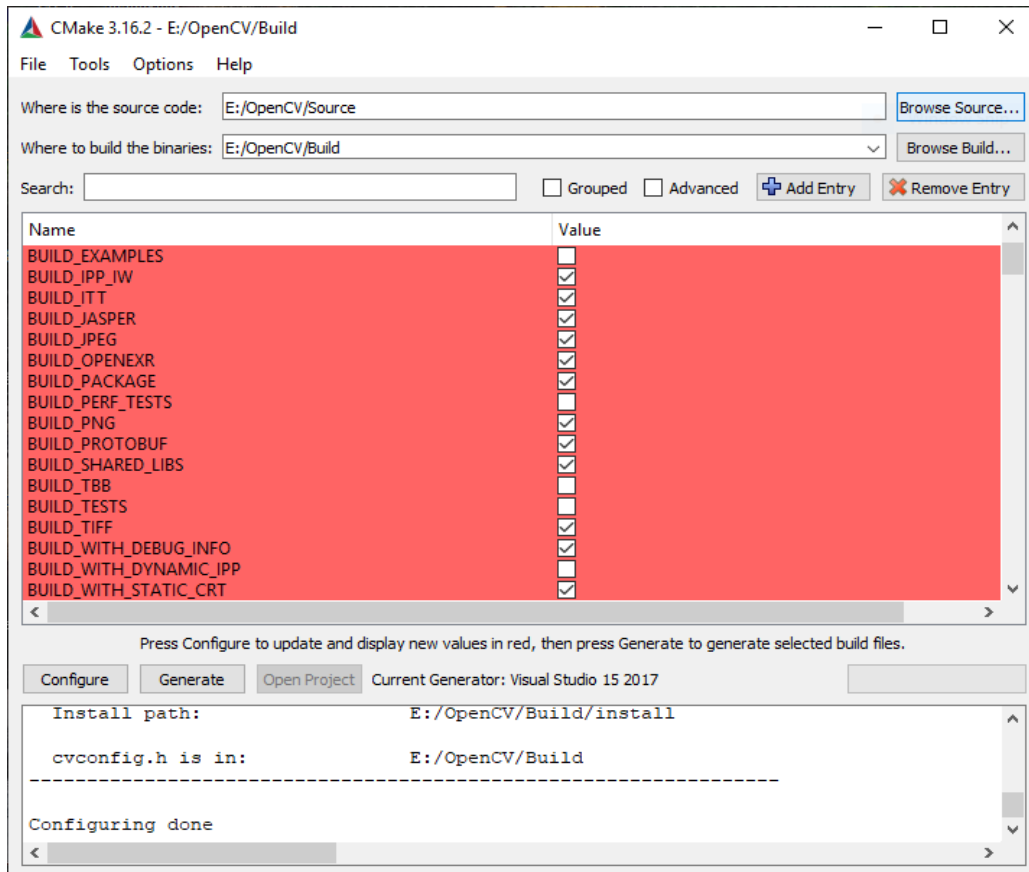


Figure 2: Visual Studio Version selection in CMake

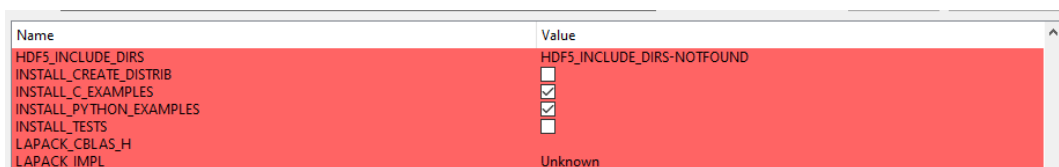
Step 4 – Configure and Generate with CMake

The steps to configure and generate CMake are as follows:

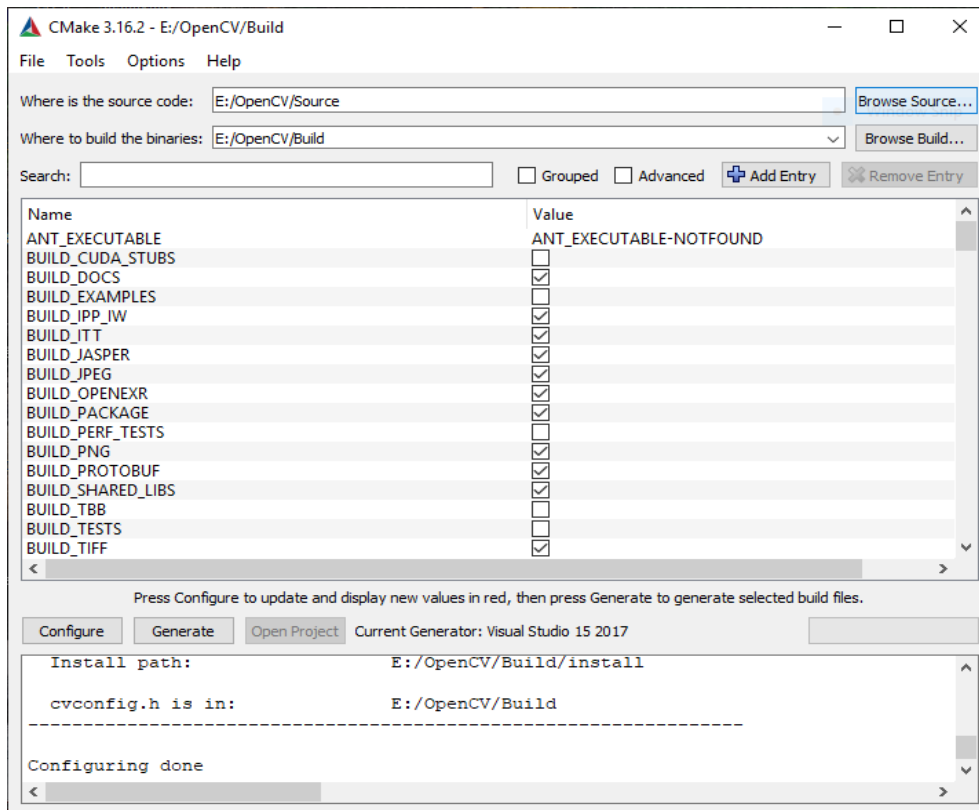
1. Click **Configure** after selecting the Visual Studio version.



2. Uncheck BUILD_PERF_TESTS and BUILD_TESTS



3. Conform the **BUILD_opencv_python3** checked or not.
4. Check **INSTALL_C_EXAMPLES** and **INSTALL_PYTHON_EXAMPLES**.
5. Check the **BUILD_opencv_world** to build single binary, including all the modules instead of a collection of separate binaries.
6. Now click configure to apply these changes.
7. Click **Configure** till all red flag goes off.



8. Click **Generate** to create Visual Studio solution file in the OpenCV build directory.

Step 5 – Videoio File Replace

Replace the **videoio** folder with the folder downloaded from the e-con's GitHub (<https://github.com/econsystems/opencv/tree/master/Source>) with **OpenCV/Sources/modules/** location.

Step 6 – Building BaseClasses

The prebuild binaries of the BaseClasses Library is also available in the below link. The BaseClasses library can either be build or referenced as directly.

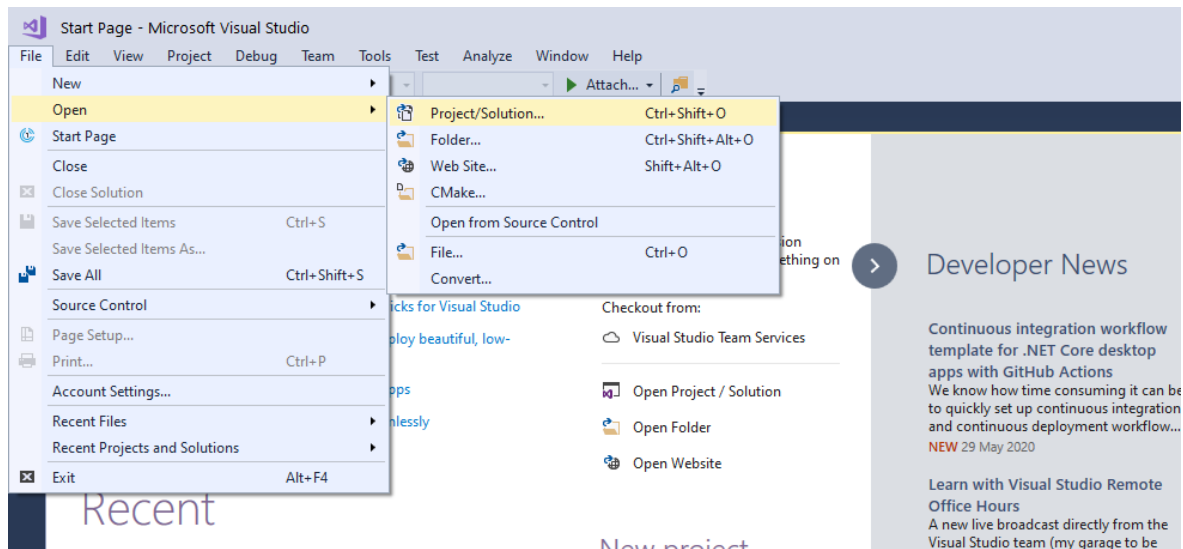
(<https://github.com/econsystems/opencv/tree/master/Binary/Lib>)

To Build BaseClasses library files:

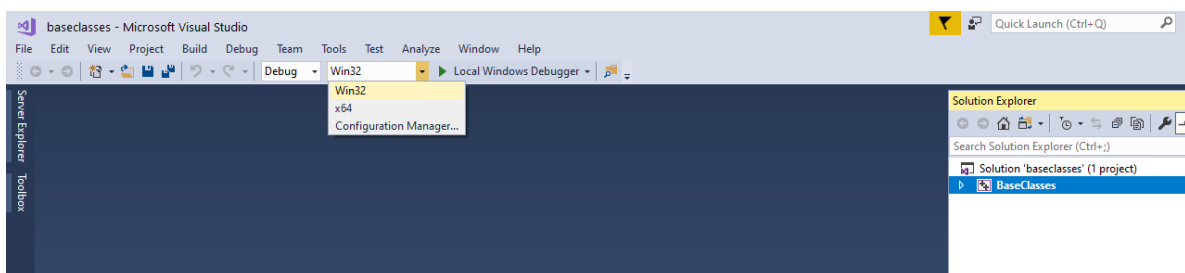
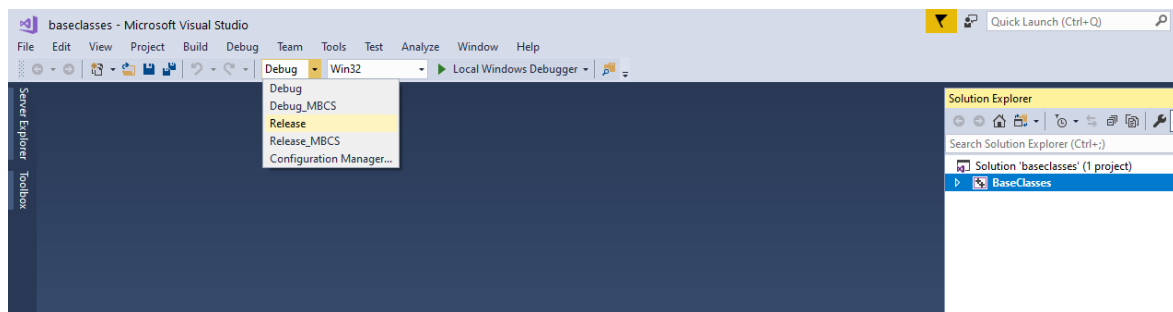
Download BaseClasses project e-con's GitHub

(<https://github.com/econsystems/opencv/tree/master/Source>)

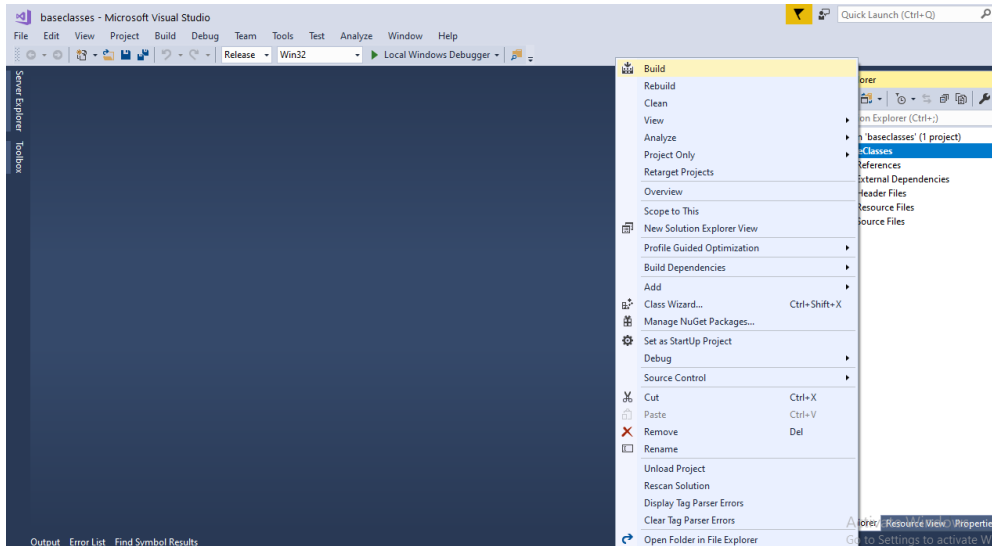
1. Open and Configuring the **baseclasses** project:
 - a) Open the new instance of visual studio (2017).
 - b) Click **File-> Open -> Project/Solution**



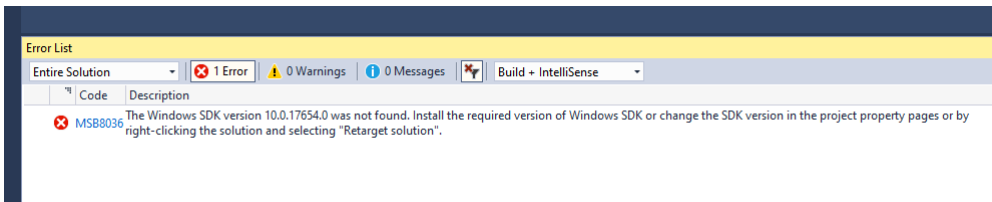
- c) Browse the baseclasses project and select **baseclasses.sln**.
- d) Choose Solution configuration (Debug / Release) and Solution Platform (Win32 or x64) (based on your requirement).



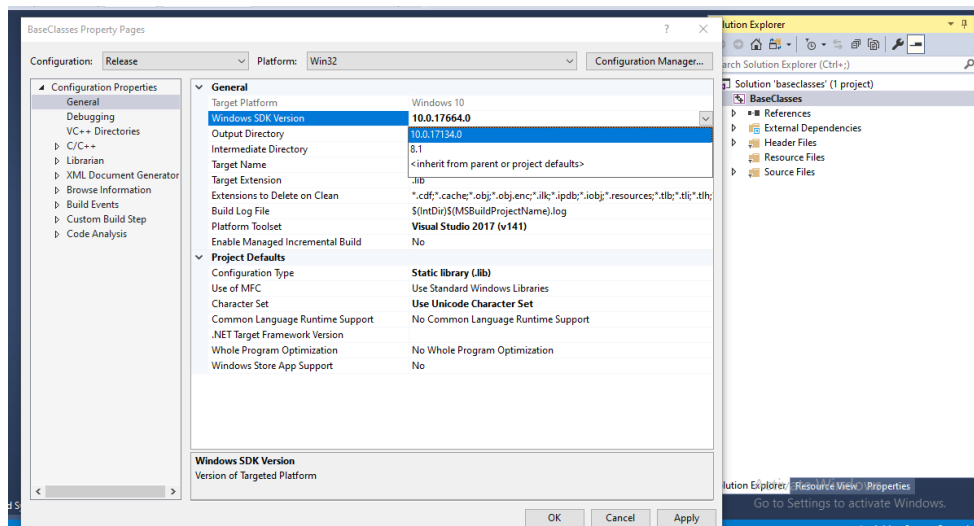
2. Build the baseclasses project:
 - a) Right Click on baseclassess solution and select Build Solution (or Rebuild Solution).



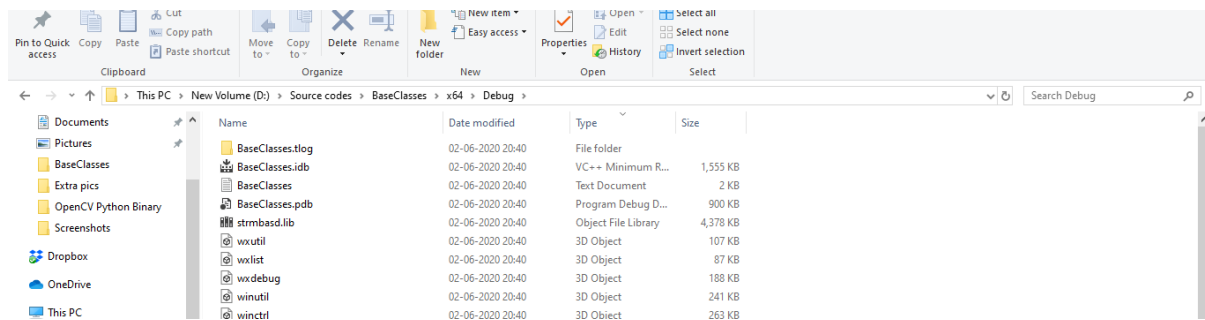
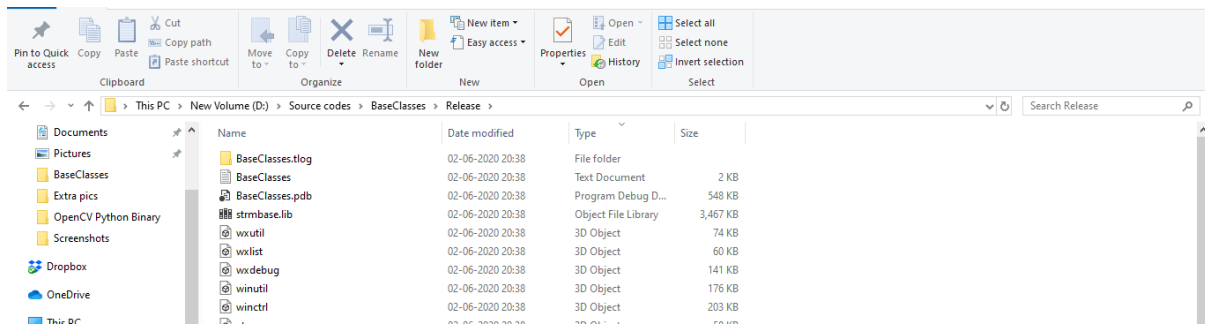
- b) While building the solution, if you get the following error change the Configuration Properties as stated below.



- c) **BaseClasses -> Properties -> Configuration Properties -> General -> Windows SDK Version** choose available SDK version.



- d) Build the solution.
e) After Building you can find the strmbase.lib (for Release), strmbasd.lib (for Debug) in respective folders.



f) The **strmbase.lib** / **strmbasd.lib** has to be linked in the OpenCV Project.

Step 7 – Build OpenCV in Visual Studio

The steps to build OpenCV in Visual Studio are as follows:

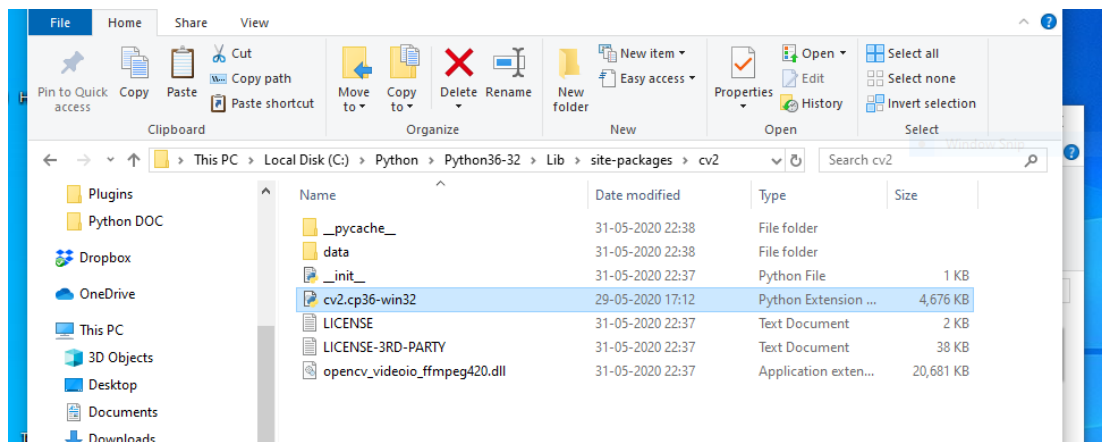
1. Run the **OpenCV.sln** found in the build directory of OpenCV using Visual Studio.
2. Add **setupapi.lib** in modules/opencv_world properties tab under Linker > Input > Additional Dependencies.
3. Add **strmbase.lib** for **release** mode and **strmbasd.lib** for **debug** mode from the **BaseClasses** solution directory, in modules/opencv_world properties tab under **Linker->General->Additional Library directories** and mention the lib name in **Linker->Input->Additional dependencies**.
4. Build **CMakeTargets/All Build** and **CMakeTargets/Install** separately in both the Debug/Release Configuration of the Visual Studio.

Configuration for Python

This section describes about how to Configure the python for sample python code.

1. Replacing the cv2 file (changing the cv2 version)

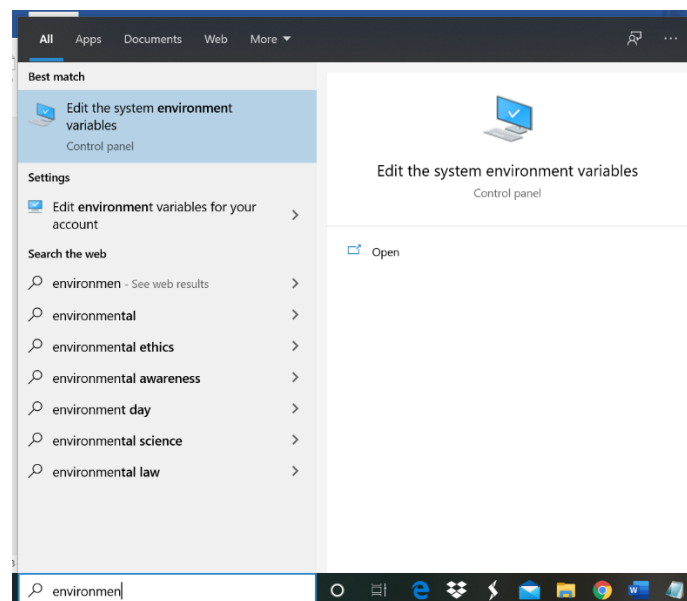
- a) After building the opencv code for python, **cv2.cp36-win32** file will generate in "OPENCV_BUILD_PATH\Lib\python3\Release\"
- b) replace the "**cv2.cp36-win32**" file to
"PYTHON_INSTALLED_PATH\Lib\site-packages\cv2\"



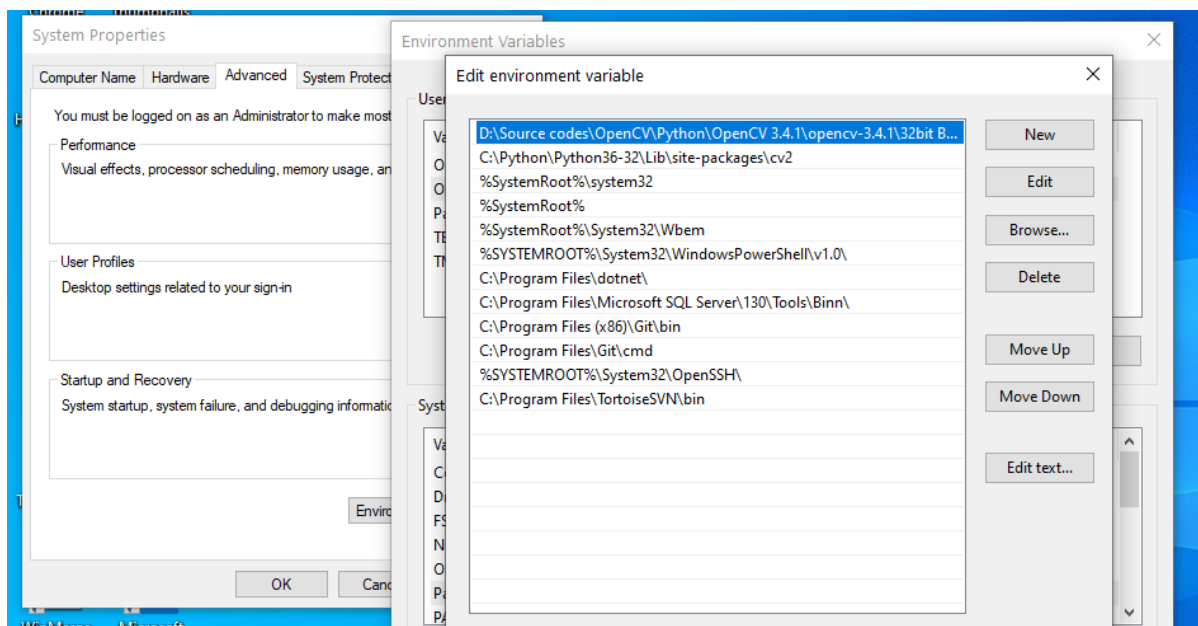
2. Update System Environment Variables

1. Update environment variable – PATH

- a) First of all, we will add OpenCV dll files path to our system variable PATH. Press Windows Super key, search for "environment variables".

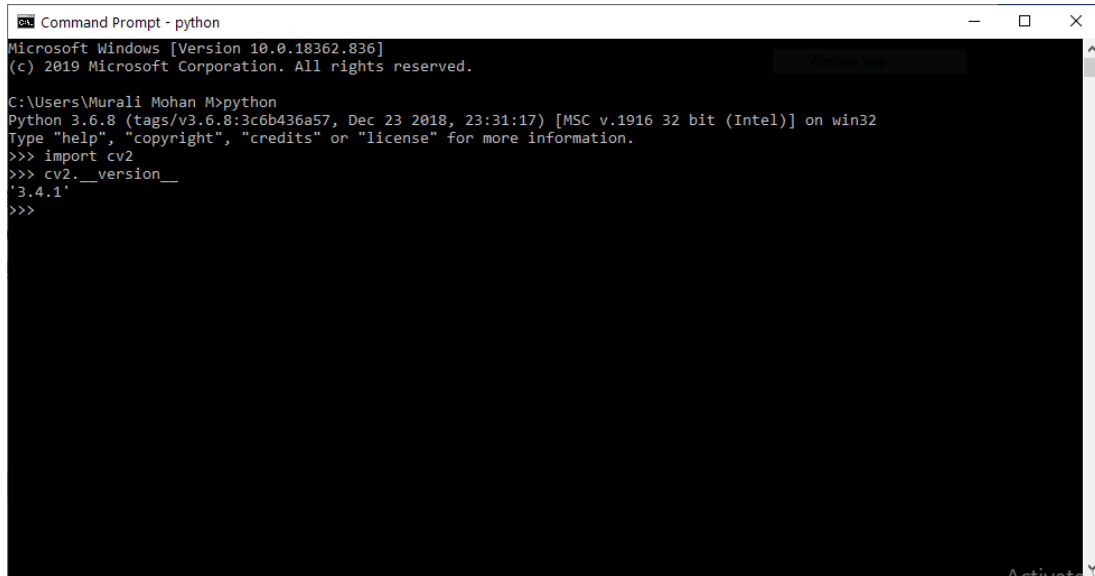


- b) Click Environment Variables in System Properties window.
- c) Under System Variables, Select Path and click edit.
- d) Click New, and give path to **OPENCV_PATH\build\install\x86\vc15\bin** (for OpenCV 3.4.1) **OPENCV_PATH\build\install\bin** (for OpenCV 3.3.1) and click Ok. Depending upon where you have kept opencv-3.4.1 folder and what version of Visual Studio you used to compile OpenCV, this path would be different.



3. Check the cv2 version

- a) Type python and give enter to set the python mode.
- b) Give import cv2, if this will not show any error then the opencv was successfully down loaded.
- c) Type "cv2.__version__" it will print the installed cv2 version
- d) Make sure after replace we see the version '3.4.1' (or '3.3.1').



```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Murali Mohan M>python
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 23 2018, 23:31:17) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.1'
>>>
```

4. Run the Python Script

- a) Open the command prompt window with script path.

Run the python script with “**Python main.py**”.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

Linker issues relating to **setupdi*** while building.

Add **setupapi.lib** in the modules/opencv_world properties tab under **Linker> Input > Additional dependencies**.

There is no install folder present in the opencv<version>/build/

Build the CMakeTargets or **install project** in both Debug and Release configurations.

HID settings are not shown in the command line application.

Change Use Unicode Character set in the Application-> configuration properties -> General -> Project defaults -> character set

In Opencv version 3.4.1, Opencv_test_namespace related errors while building.

Unload the tests accuracy and tests performance projects from the opencv and start the building process again.

IAMVIDEOCONTROL related error while building Opencv.

Copy the **strmbase.lib**, if using release mode or strmbasd.lib, if using debug mode from the BaseClasses project path and paste it in the **build/lib/release (or) debug** directory of the OpenCV. Also based on the x86 or x64 architecture, the libs should be copied and pasted. If the strmbase.lib or strmbasd.lib is not present. Then build the **baseclasses.sln** using visual studio 2017.

If git is not recognized as an internal or external command error

You need to download and install the git exe from internet with the following link.
<https://gitforwindows.org/>

Contact Us

If you need any support on OpenCV product, please contact us using the Live Chat option available on our website - <https://www.e-consystems.com/>

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - <https://www.e-consystems.com/create-ticket.asp>

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - <https://www.e-consystems.com/RMA-Policy.asp>

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - <https://www.e-consystems.com/warranty.asp>

Revision History

Rev	Date	Description	Author
1.0	13-August-2020	Initial Draft	Murali Mohan. M
1.1	14-August-2020	Added changes.	Murali Mohan. M