# Documentation to start the program

## 1.0 Pre-request

Before you start the program, you need some application installed in your device

1. Python 3.6 or above (mandatory)
2. Opencv library (will discuss on the following section)
3. Pyqt5 (will discuss on the following section)

## 2.0 Configure and install the libraries

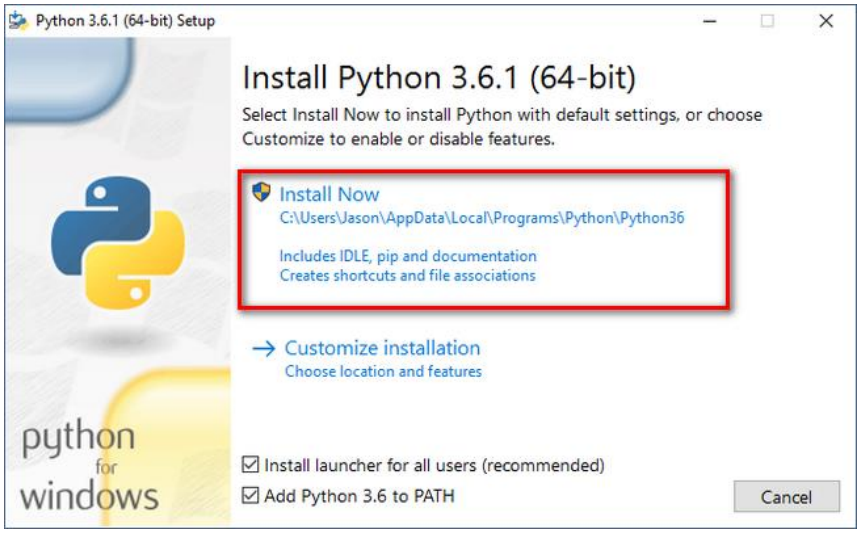
### 2.1 Python Installation steps

Step 1: To start the program, you need to install Python 3.6 or above. You can get the installer through the link provided below

Link: https://www.python.org/


1. If you are Windows user, select Windows
2. Click python 3.6.5 as shown above

Step 2: After completed download the installer, double click the installer, select Install Now (Remainder: select “Add Python 3.6 to PATH” before you proceed)




Step 3: Click “Next” and remain all setting as default setting

### 2.2 Opencv installation

Step 1: Go to [https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv](https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv), search opencv, and download opencv python-3.4.1 cp36-cp36m-win32.whl (if your python is 64bit, please select opencv python-3.4.1 cp36-cp36m-win amd64.whl)

**Index by date:** h5py quickfix spglib kiwisolver jupyter pillow matplotlib gevent pendulum moderngl numpy-quaternion aiohttp zstd tifffile cx_oracle sounddevice tats
netifaces pandas netcdf4 cftime dipy pygresql debug-information-files statsmodels twisted numexpr chompack cvxopt numpy vitables rapidjson aggdraw lxml pyopen
image scikit-learn menpo astropy swiglpk simplejson btrees faulthandler thriftpy zipline gmpy orange hyperspy zope.interface sqlalchemy brotli arctic pip gdal logb
numcodecs discretize dulwich py-lmdb tornado mercurial simpleitk mod_wsgi jpype lz4 biopython fastparquet tensorflow lsqfit indexed_gzip pyodbc bokeh javabridg
wordcloud meshpy tomopy cobra sfepy cytoolz blist cheetah basemap xylib-py cyrasterize pyswisseph openexr pulp grpcio gensim pymongo cantera cchardet raster
cdecimal crcmod crc16 pycld2 planar autopy pyx pywin32 iminuit rtmidi-python pycosat pyflux opencv mkl-service postgresadapter datrie polygon py-earth lightnin
fisx fisher ffnet fasttext iris pymc hddm hmmlearn heatmap jsonlib intbitset sasl bsddb3 flann fiona msgpack cartopy pyfits scikits.odes regex louvain-igraph pyt
scandir fast-histogram pycifrw pyzmq coverage lp_solve zodbpickle aspell-python transformations vlfd chebyfit vidsrc psf akima pykinsol pyodeint pycvodes mayavi
chaco enable traits noise scikits.vectorplot scikit-fmm rtree python-levenshtein python-lzo pyspharm pyminuit pymetis pymcubes pylzma pyhook pyeda pyfmi reportla
pyaudio apsw mysqlclient greenlet pymvpa thrift pyicu python-snappy atom pyemd enaml shapely pypmc wrf_python fabio pyyaml quantlib slycot babel mkl_rand
polylearn blosc libsbml simpleaudio pylibtiff line_profiler persistent pywavelets cx_freeze videocapture pygame pycuda pyproj boost.python fastrlock minepy fann2 n
scikit-umfpack pillow-simd openpiv czifile scs veusz cvxpy gr qutip sympy pyarrow scikit-misc pycorrfit pyside vigra grako kivy pyjnius imaged11 python-cjson
nipy qimage2ndarray guiqwt qt_graph_helpers pyqwt pyqt4 multiprocess libtfr nitime lfdfiles mathutils cvxcanon pyvrml97 pythonmagick yappi pyfftw pyviennacl py
cyassimp sima pymca friture pycogent pysqlite blaze scikits.audiolab la bazaar dynd genshi python-sundials glumpy pyamf libxml-python cellcognition pymcmc pyk
bio_formats pysfml pyexiv2 pylibdeconv iocbio pymix umysql lazyflow mmlib scikits.timeseries casuarius wxpython ilastik pywcs scientificpython vpython nmoldyn
carray llvmpy cgkit pymedia scipy-cluster scikits.scattpy scikits.samplerate scikits.ann pyxml pytst delny mysql-python htseq pyusb-ftdi silvercity steps pysparse pyro

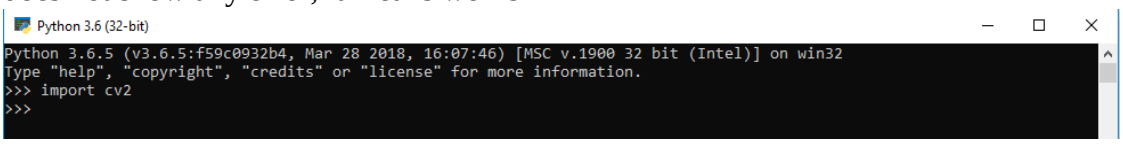
**OpenCV**, a real time computer vision library.

opencv_python-2.4.13.5-cp27-cp27m-win32.whl
opencv_python-2.4.13.5-cp27-cp27m-win_amd64.whl
opencv_python-3.1.0-cp34-cp34m-win32.whl
opencv_python-3.1.0-cp34-cp34m-win_amd64.whl
opencv_python-3.4.1+contrib-cp35-cp35m-win32.whl
opencv_python-3.4.1+contrib-cp35-cp35m-win_amd64.whl
opencv_python-3.4.1+contrib-cp36-cp36m-win32.whl
opencv_python-3.4.1+contrib-cp36-cp36m-win_amd64.whl
opencv_python-3.4.1+contrib-cp37-cp37m-win32.whl
opencv_python-3.4.1+contrib-cp37-cp37m-win_amd64.whl
opencv_python-3.4.1-cp35-cp35m-win32.whl
opencv_python-3.4.1-cp35-cp35m-win_amd64.whl
opencv_python-3.4.1-cp36-cp36m-win32.whl
opencv_python-3.4.1-cp36-cp36m-win_amd64.whl
opencv_python-3.4.1-cp37-cp37m-win32.whl
opencv_python-3.4.1-cp37-cp37m-win_amd64.whl

Step 2: Go to Command Prompt, and type the following command, wait the installation complete

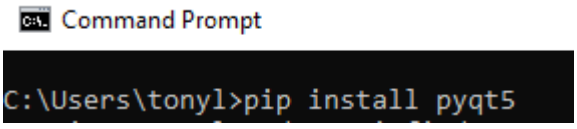
```
C:\Users\tonyl>pip install opencv_python-3.4.1-cp36-cp36m-win32.whl
```

Step 3: Go to python 3.6 and type the following command, if the command does not show any error, it means works

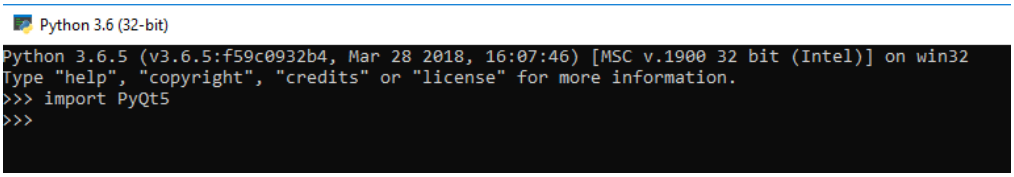



### 2.3 Pyqt5 installation

Step 1: Go to Command Prompt, and type the following command, wait the installation complete



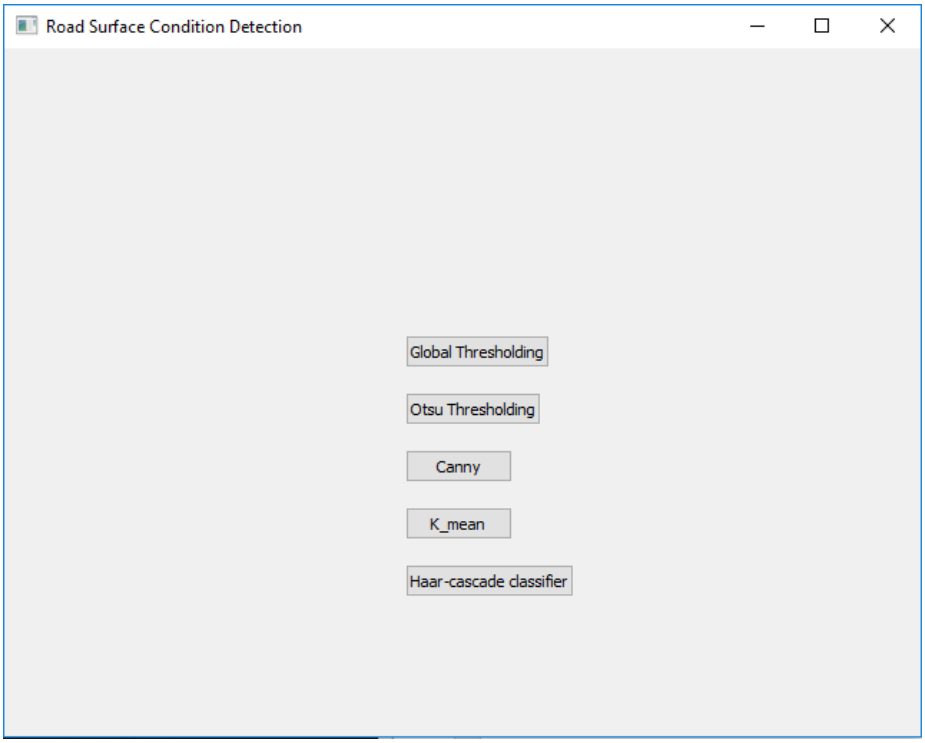

Step 2: Go to python 3.6 and type the following command, if the command does not show any error, it means works



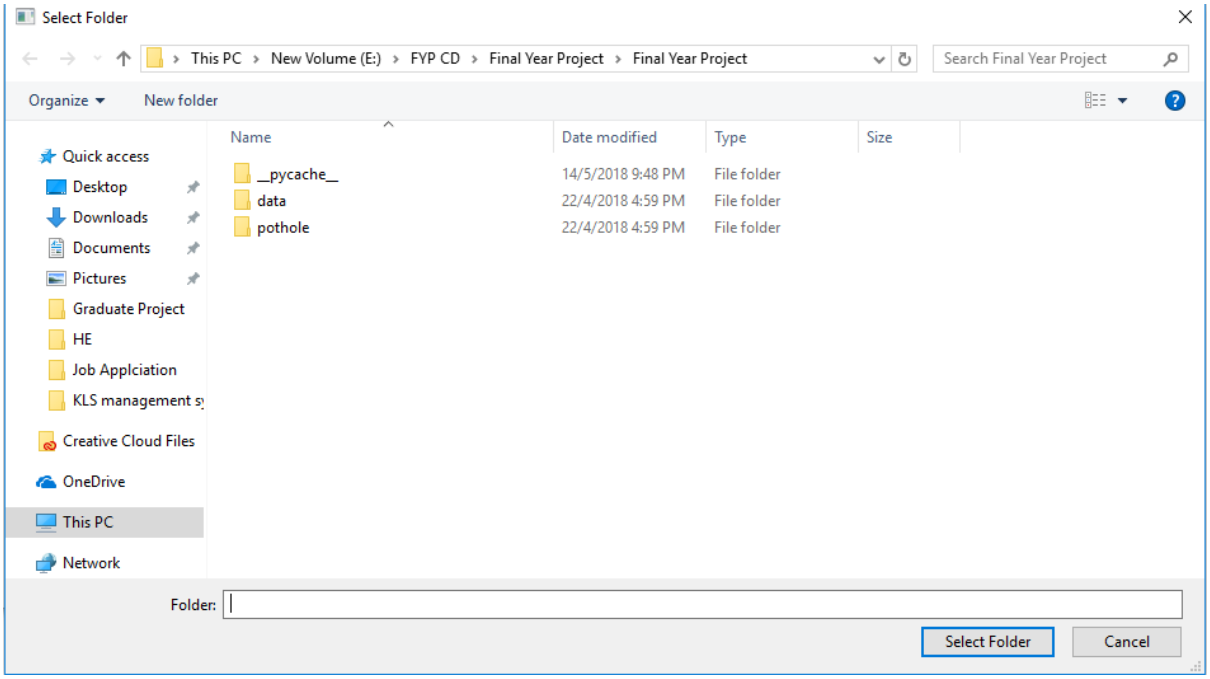

### 2.4 Run the program

Step 1: Double click “Final Year Project.py” file


Step 2: You will see the interface as shown, click any option you wish to know. For example: Global Thresholding




You will see the interface as shown below and select “pothole” folder




Step 3: You will see the result as shown


Press “esc” or spacebar to proceed until it finish the program