

Date: 12.10.14

IMPORTANT

Supported Image Specification:

Input image: Reconstructed oral CT scan (.raw)

Image dimension: 800x800x(400~500: around 400-500 slices for a single patient)

16-bit signed integer (raw)

One folder should contain all image slices of a patient.

Environment

Visual Studio 2010

OpenCV 2.4.9

These two methods are described below:

In this package, there are two ways to test the algorithm developed for CT images:

- A. By running the stand-alone executable file.
- B. By running the code in visual studio environment.

A. Testing by running the stand-alone executable file:

In the package, “executable” folder contains a setup file. Run the setup file and a shortcut named “**CAD-CT**” will be created in the desktop. From that shortcut folder, run the **CT_CAD.exe** file. A folder browsing window will pop up and you can select a folder containing CT images from there. The program will run and will show detected images (if any). To exit from the program, close the console window.

If you want, you can uninstall the software anytime from “add/remove program” section in control panel.

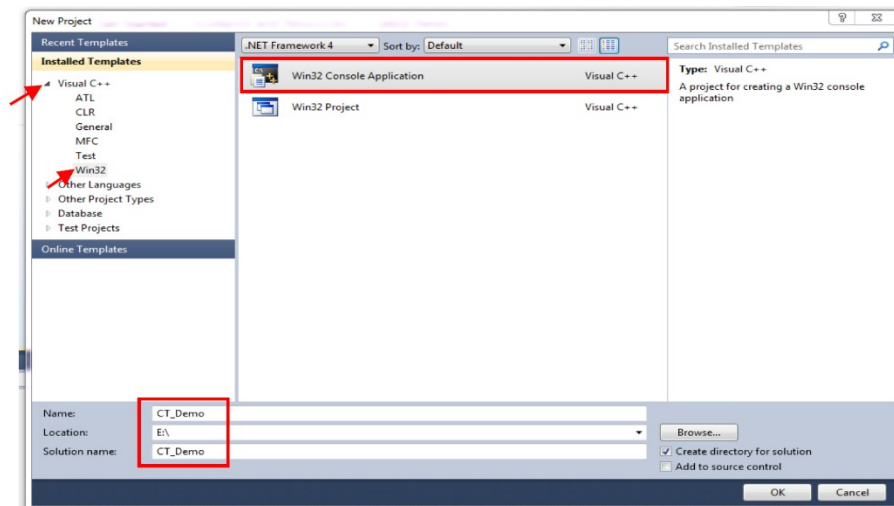
B. Testing the code in Visual Studio environment:

The project was tested using Visual Studio 2010 (Service Pack 1). Therefore, it may not run properly in other versions of visual studio.

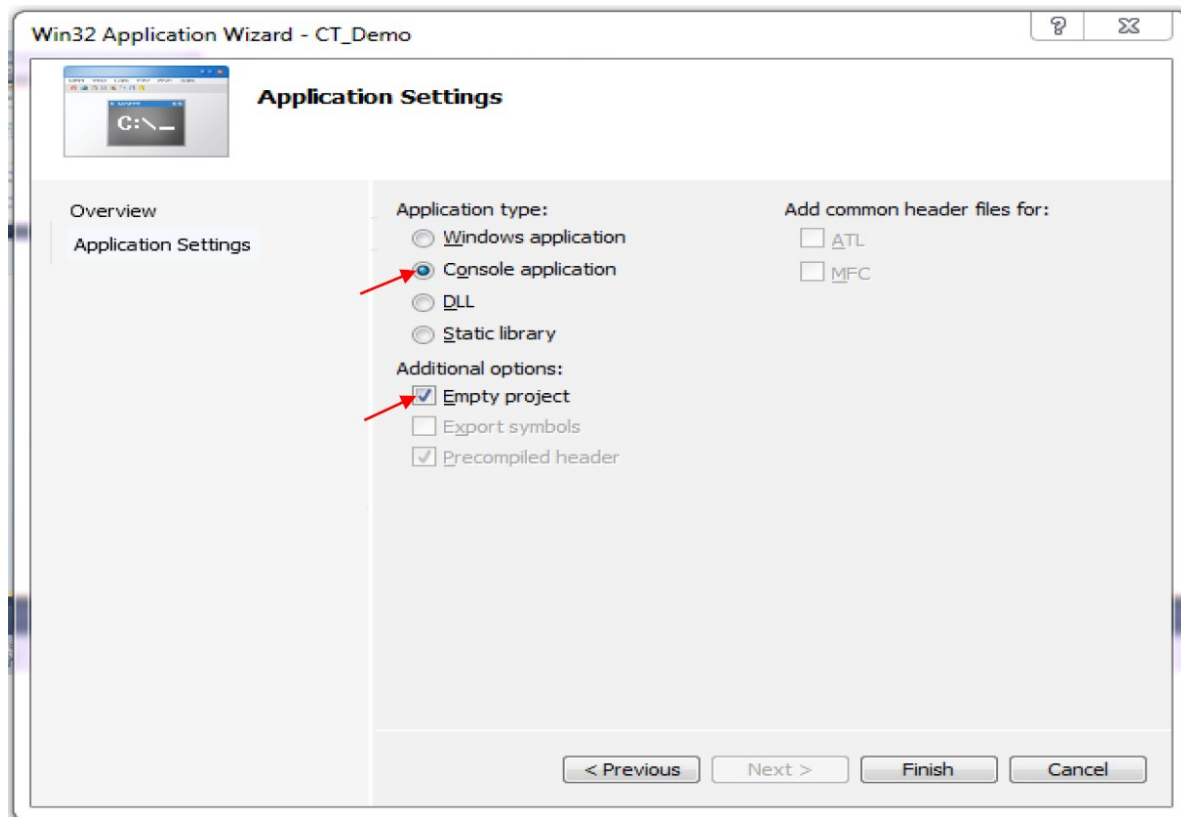
“OpenCV” was used as a third party library for this project. Only free for commercial use modules were used.

Below are the instructions on how to use the code and set up a project in visual studio 2010:

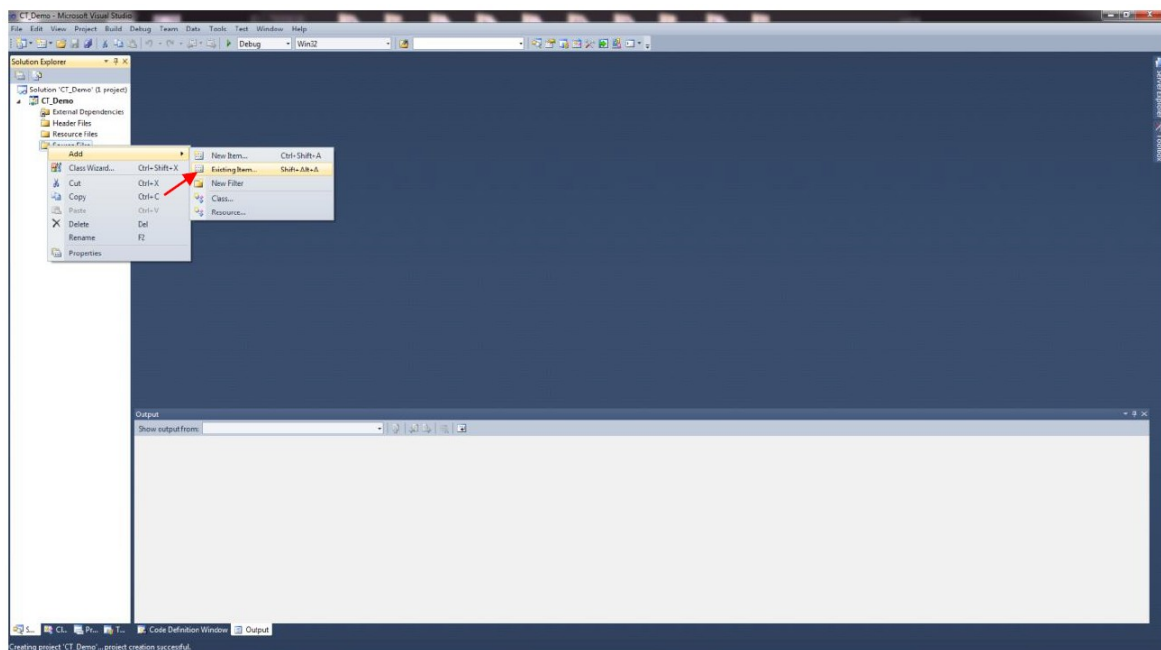
1. Create a New project in visual studio. Suppose the project name is “CT_Demo”.



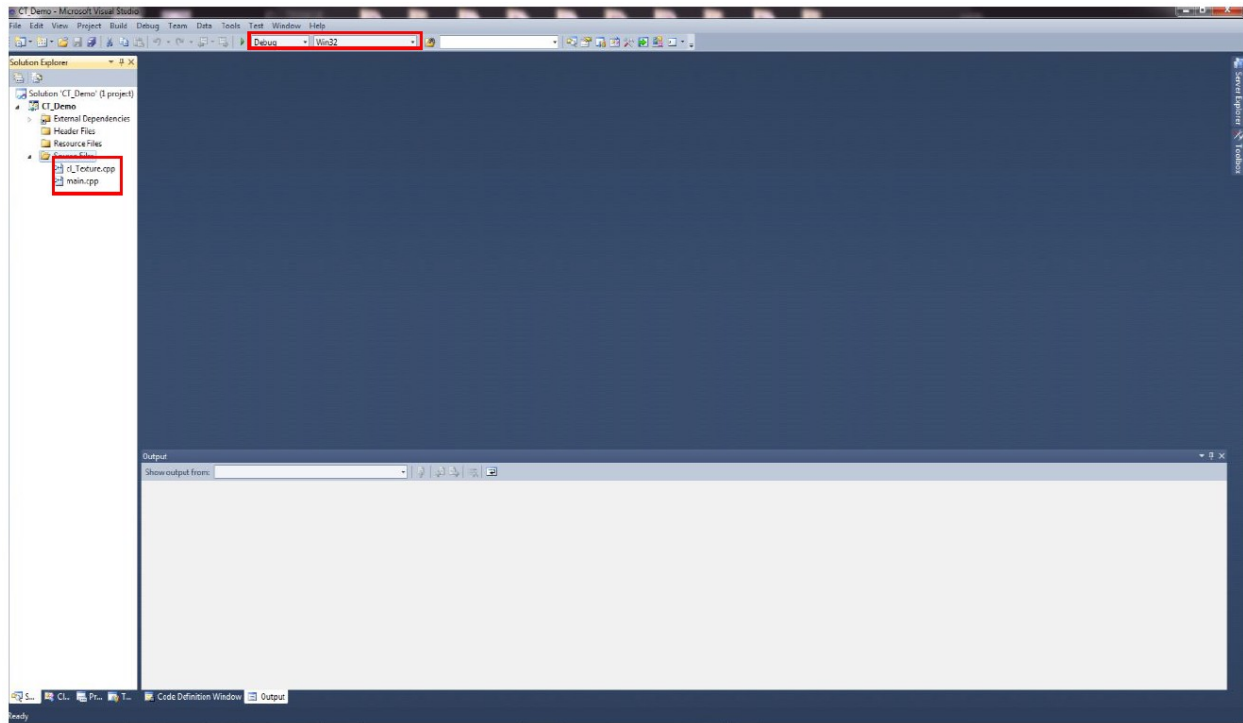
2. Select “Console application” and “Empty project”. New project will be created.



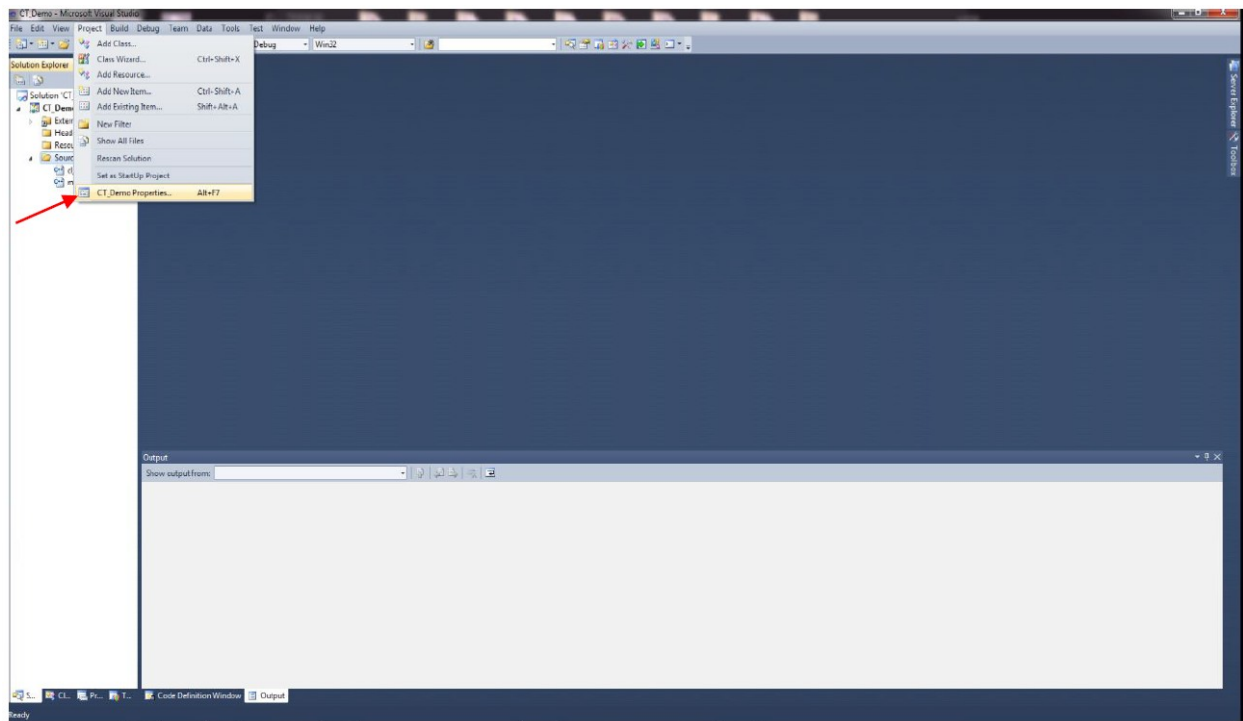
3. Add source files in the project. There are two source files provided in the “src” folder. Add them by right clicking on the “source files” folder in the solution explorer and then use add existing item option.



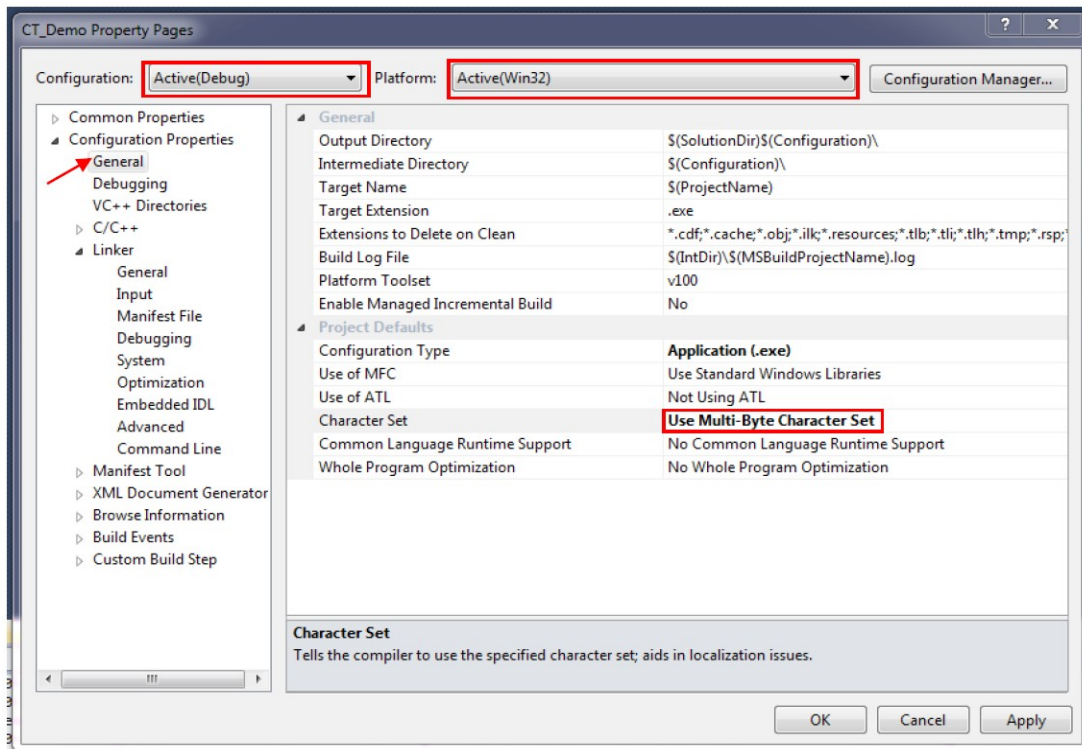
4. After adding source files, confirm the configuration setting: either Debug or Release mode is ok. Then select WIN32 platform.



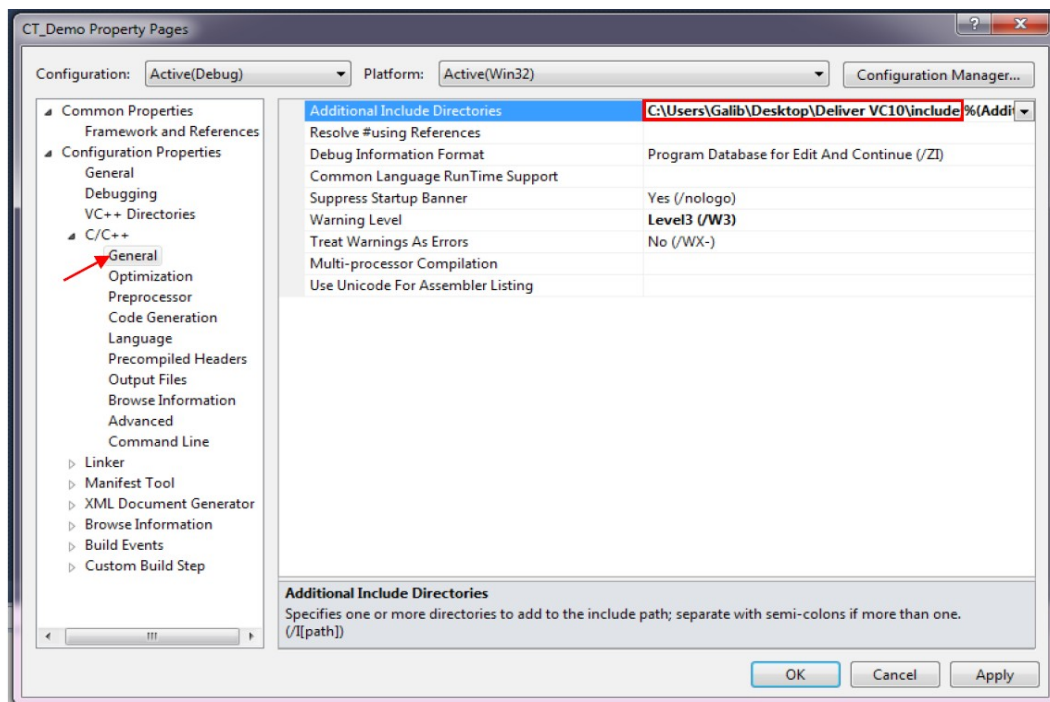
5. From the menu bar, open project property pages.



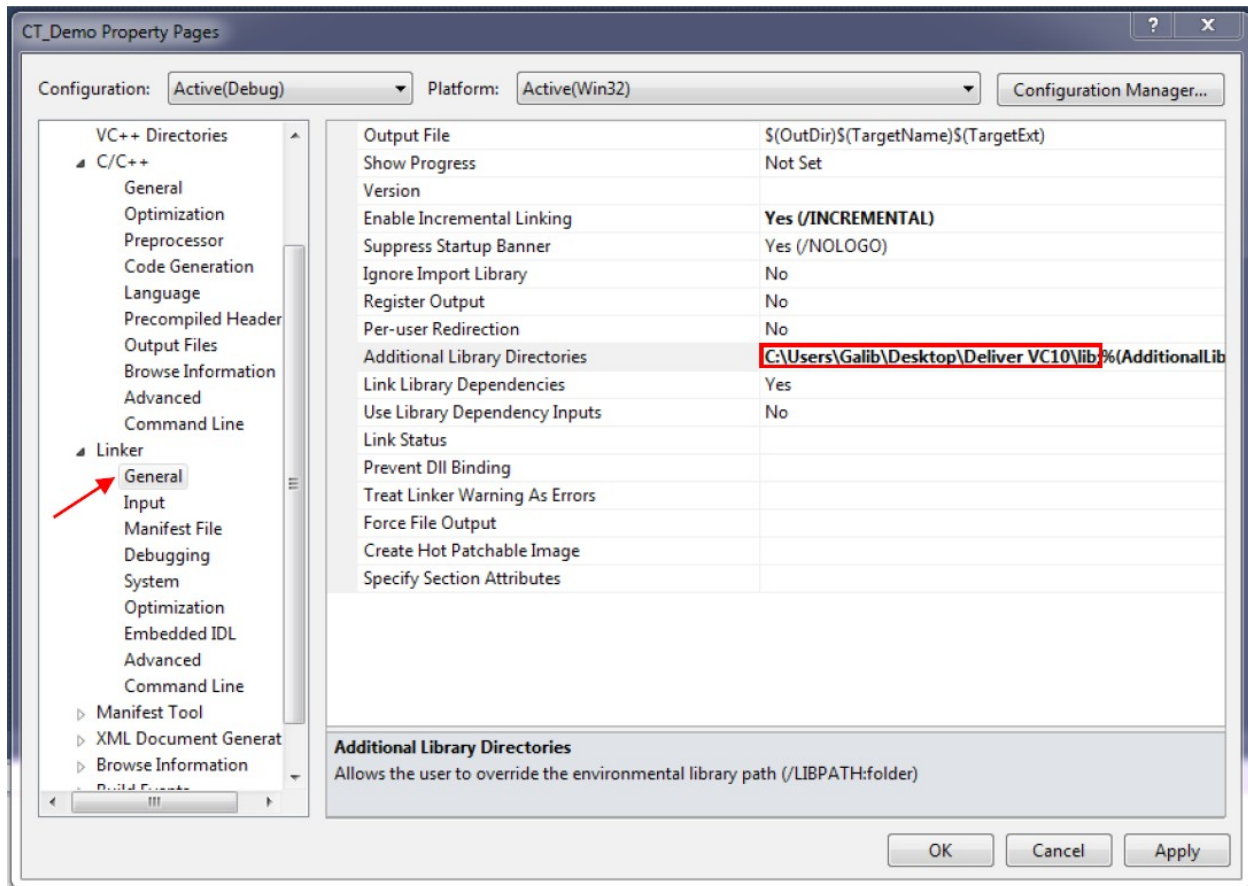
6. In project property page, configure the following: Use platform WIN32 and edit character set to “use multi-byte character set”. You can run the code in both Debug or Release mode.



7. In additional include directories, add the path of “include” folder provided with the package.



8. In additional library directories, specify the path of library files provided in the “lib” folder.



9. In additional dependencies, write the library file names. (See the figure below)

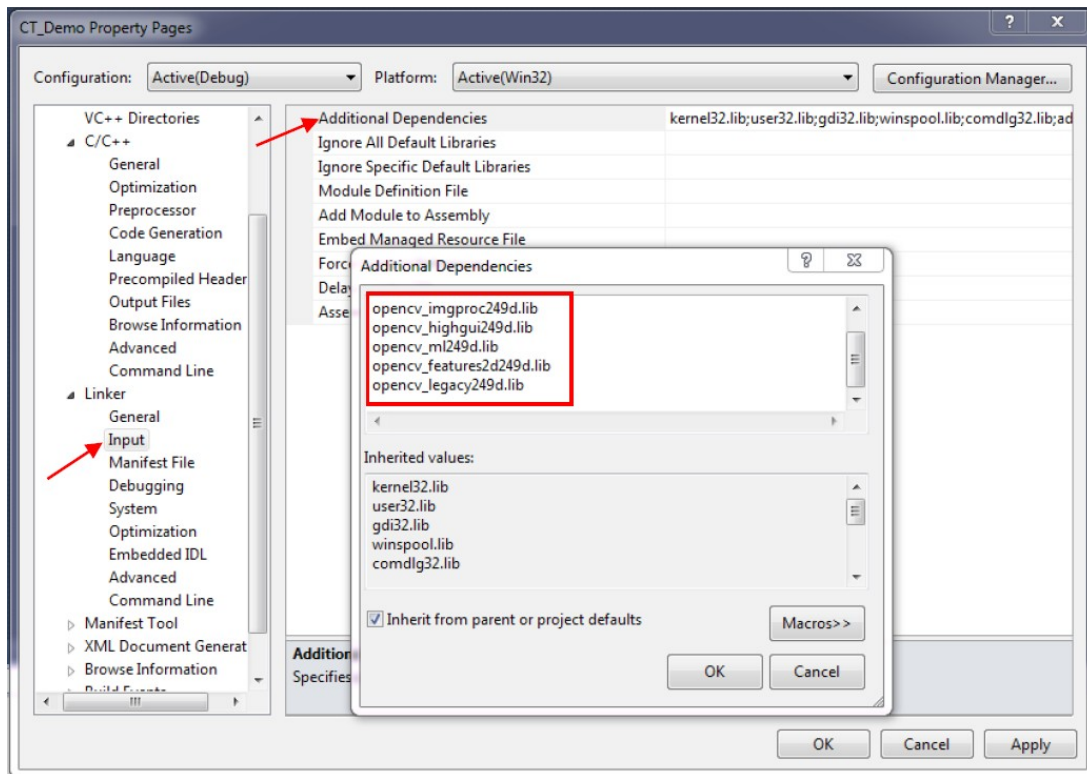
For **debug version**, lib file names are:

opencv_core249d.lib
opencv_imgproc249d.lib
opencv_highgui249d.lib
opencv_ml249d.lib
opencv_features2d249d.lib
opencv_legacy249d.lib

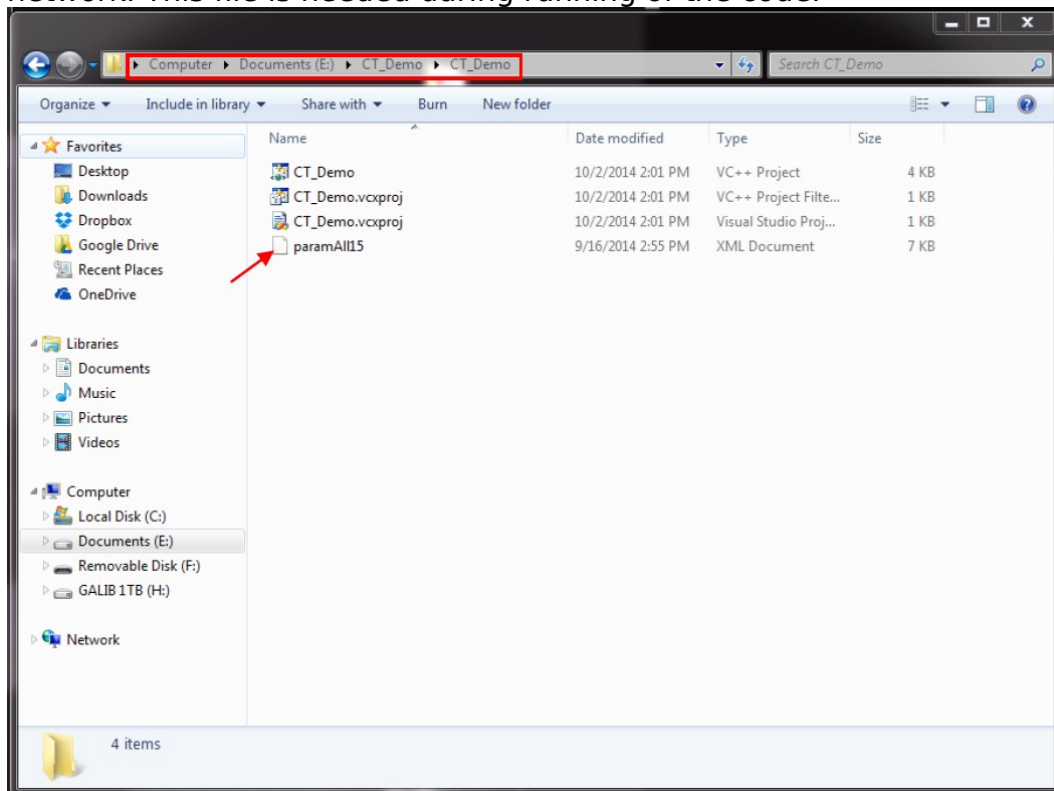
For **Release version**, lib file names are:

opencv_core249.lib
opencv_imgproc249.lib
opencv_highgui249.lib
opencv_ml249.lib
opencv_features2d249.lib
opencv_legacy249.lib

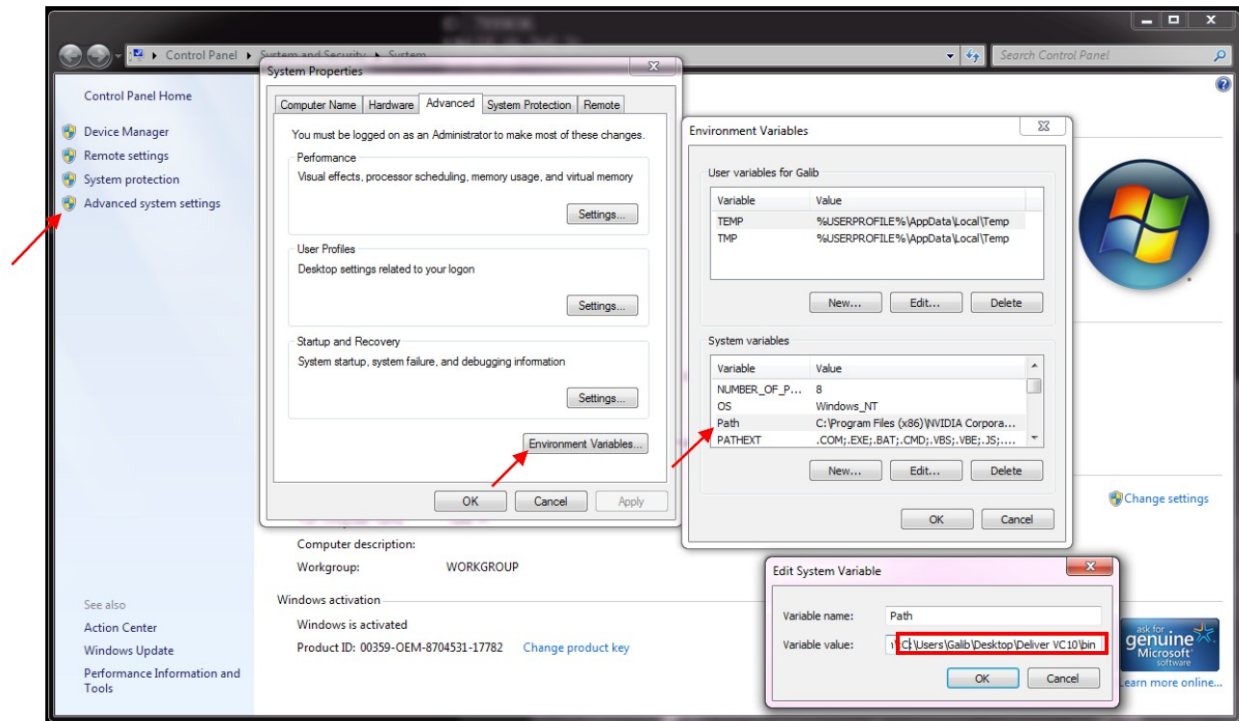
Property page configuration is now finished.



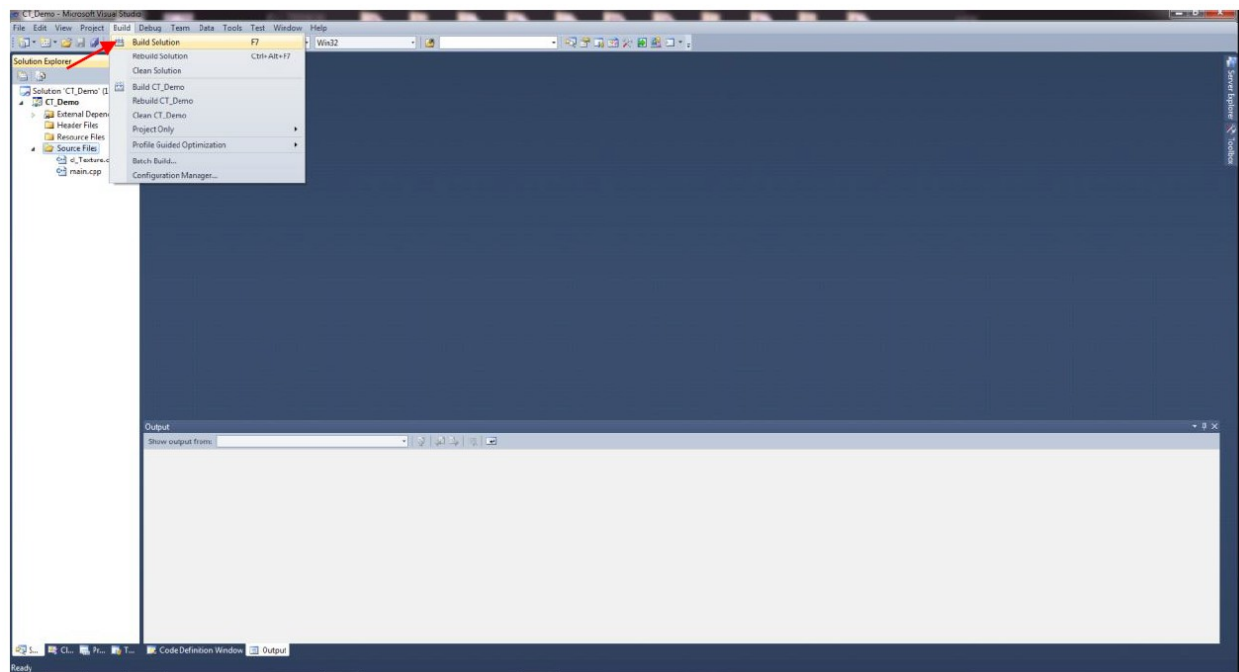
10. Now copy the “paramAll15” file from “ANN file” folder and paste it in your project folder. “paramAll15” is a XML file that contains weights of a trained neural network. This file is needed during running of the code.



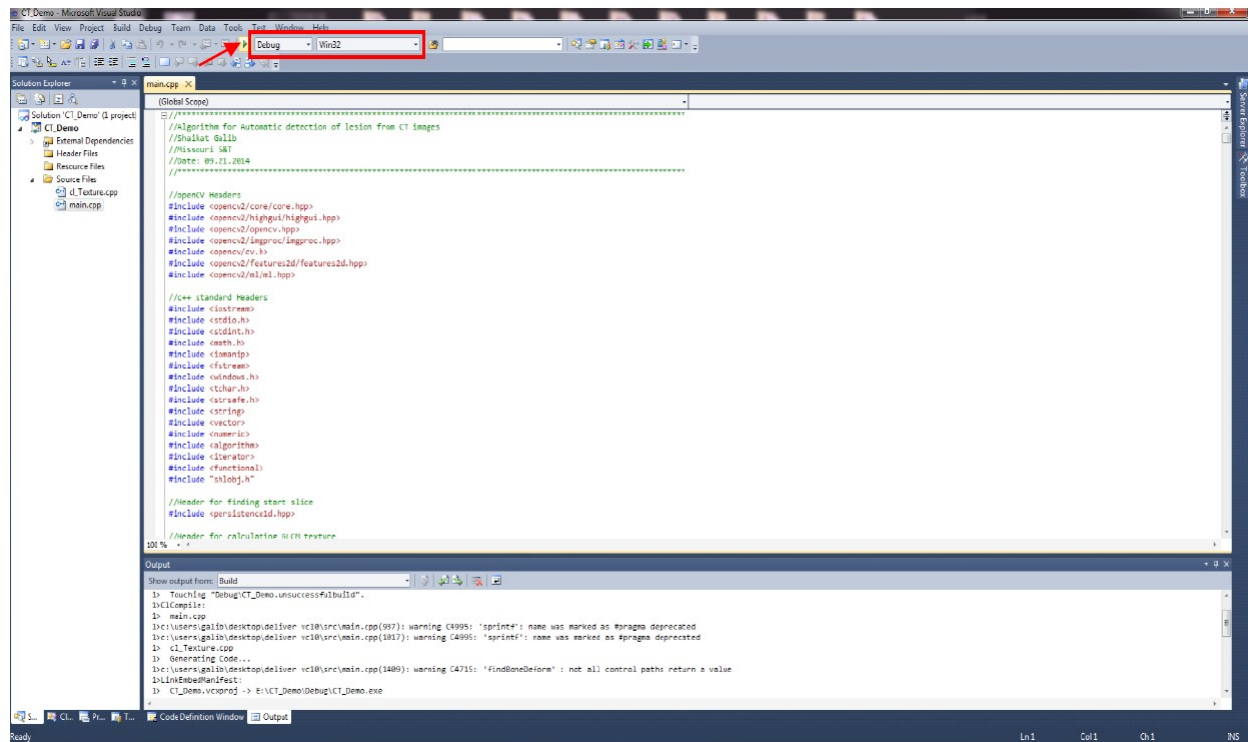
11. Finally add the file path of the “Bin” folder in the System Environment Variable Path.



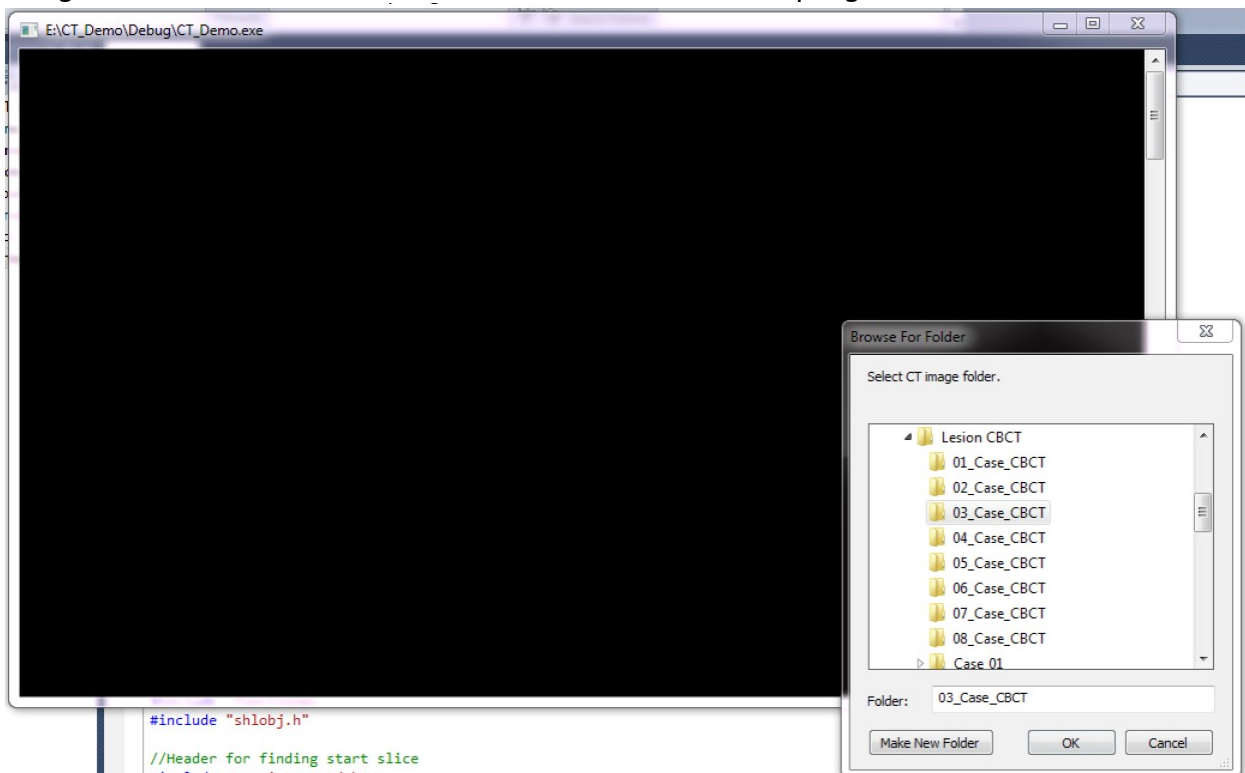
12. Build the solution.



13. Now you can run the code from “Local Windows Debugger”



14. Graphic User Interface should look like the image below. You can select CT image folder from the folder browser window and the program will run.



15. The software will display results like below.

