README

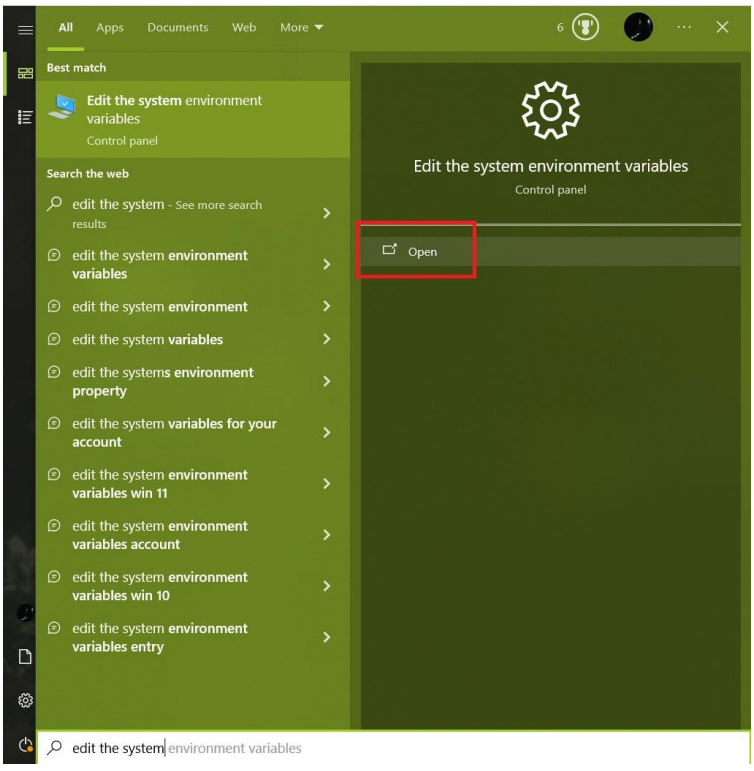
This is the Client module of the Image Processing Application. This application communicates with the Server module to send an image or multiple images and filter/s to apply on it, and receive the image/s with the filter/s applied.

Requirements

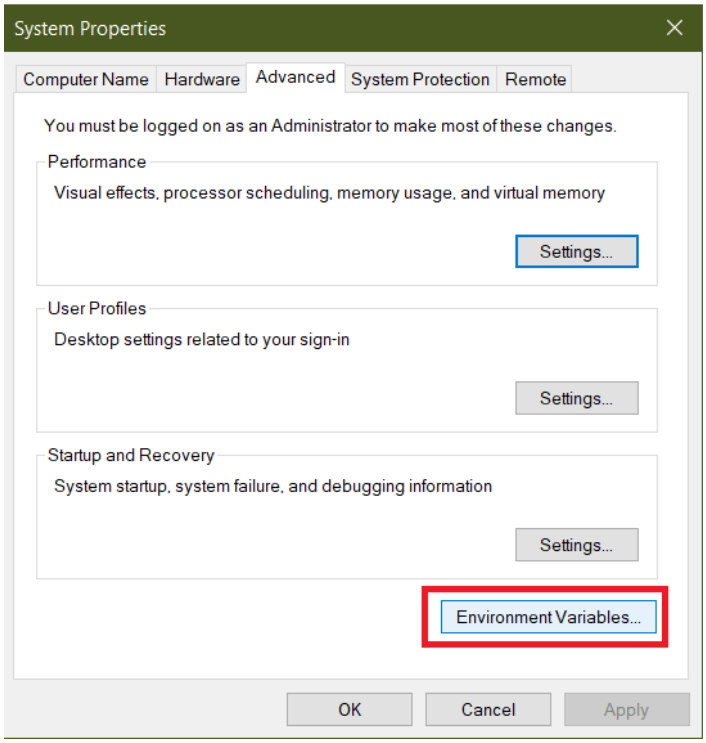
- 64-bit Windows operating system
- OpenCV 4.8.0
- Server companion module
- Visual Studio (2022 preferred)

Configuring OpenCV 4.8.0

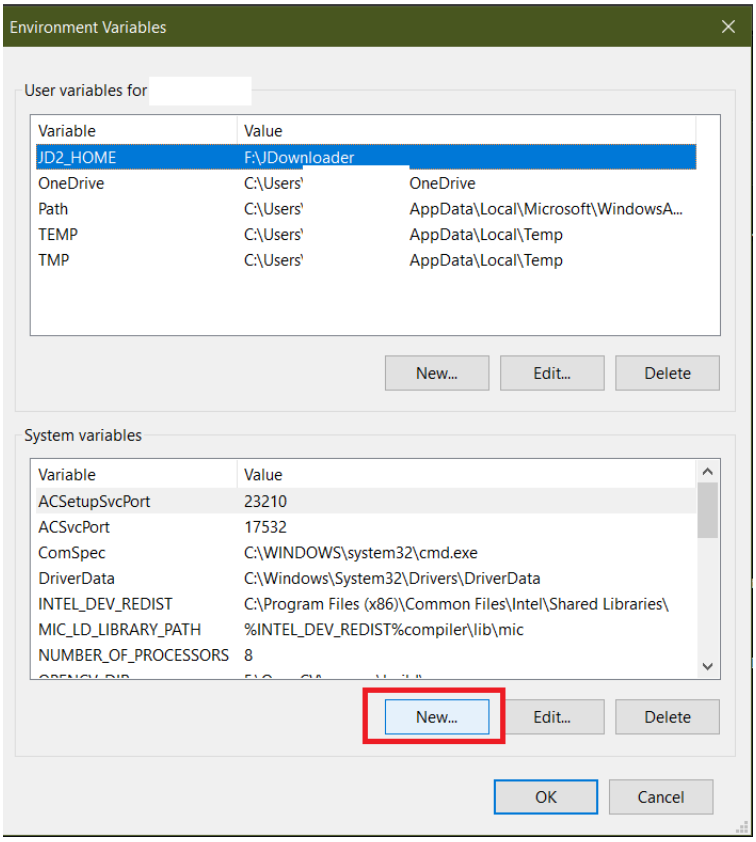
1. Download 4.8.0 from https://opencv.org/releases/. Alternatively, use the opencv-4.8.0-windows.exe file provided with the submission.
2. Install OpenCV to the desired location.
3. Add the OpenCV directory environment variable. Below are the steps to do this on Windows:
   - Open Start menu, search for 'Edit the system environment variables' and click on 'Open' as shown:



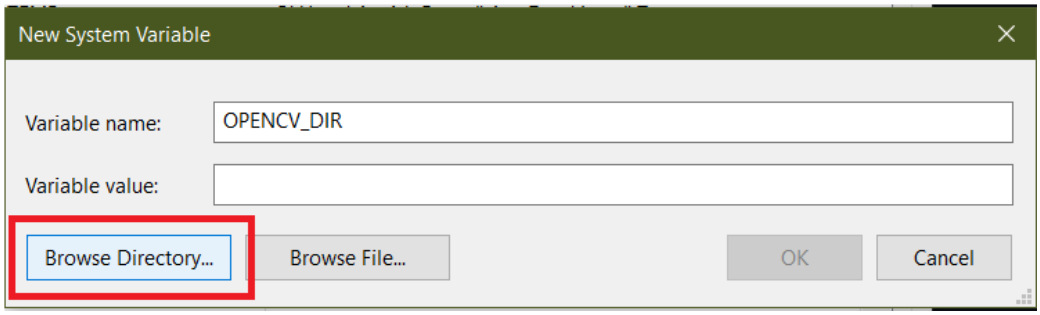
- In the 'System Properties' dialog box, click the 'Environment Variables...' button:



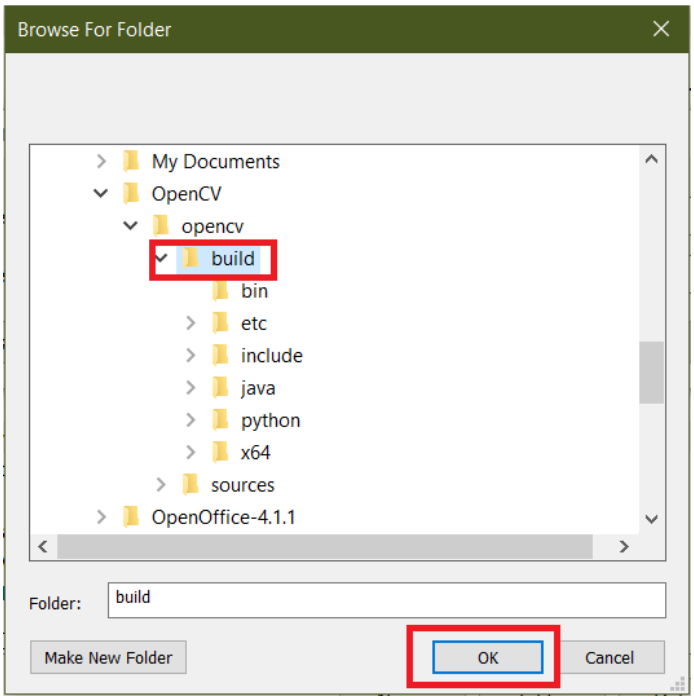
- Click 'New' at the bottom:



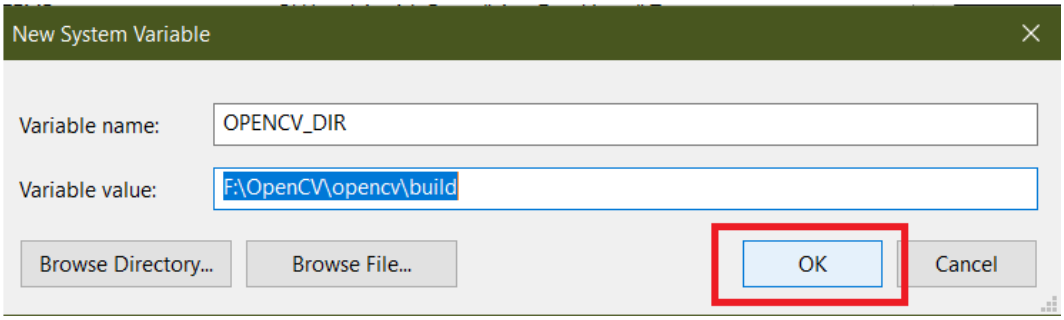
- Type in 'OPENCV_DIR' (without the quotes) in the 'Variable name' field as shown, and click on the 'Browse Directory' button:



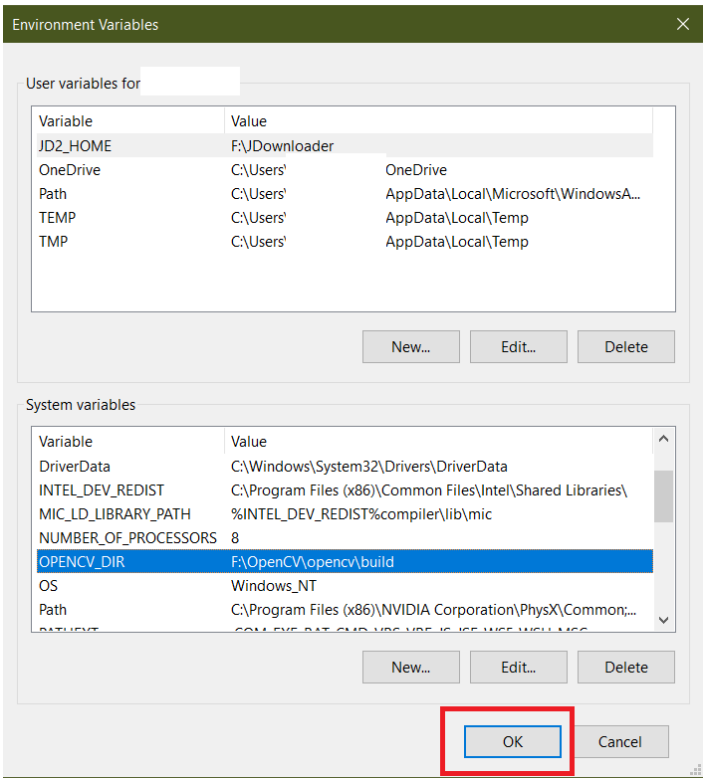
- Browse to the 'build' folder of your OpenCV installation, as shown below. For example, if you installed OpenCV in the 'Documents' folder, browse to 'C:\Users\abc\Documents\opencv\build\'. Select the 'build' folder and click 'OK':



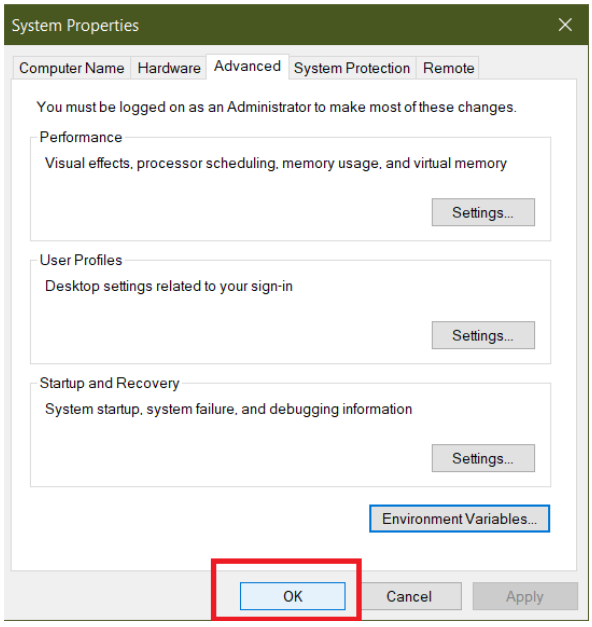
- The 'Variable value' field in the 'New System Value' dialog box should now contain the address of the OpenCV build folder, as shown below. Click OK:



- The OPENCV_DIR variable will now be added to the 'System Variables' list in the 'Environment Variables' dialog box. Verify the value and click 'OK':



- Finally, click OK in the 'System Properties' dialog box:



- More information on how to set up environment variable for OpenCV can be found here (https://www.opencv-srf.com/2017/11/install-opencv-with-visual-studio.html).

Preparing command-line arguments

The application takes user input in the form of command-line arguments, which should be in the following format:

<Server IP:Port> <Original image path> <Filter name> <Filter values>

The block

<Original image path> <Filter name> <Filter values>

can be repeated any number of times depending on the number of images to submit to the server.

Examples

- 127.0.0.1:8080 F:/MyImages/Hollywood/ClubOfFighting.jpg Flip Vertical
- 127.0.0.1:8080 F:/MyImages/Hollywood/AnIslandCalledShutter.jpg Rotate Clockwise 1 F:/MoreImages/Gaming/GTA6.jpg Resize 320 240

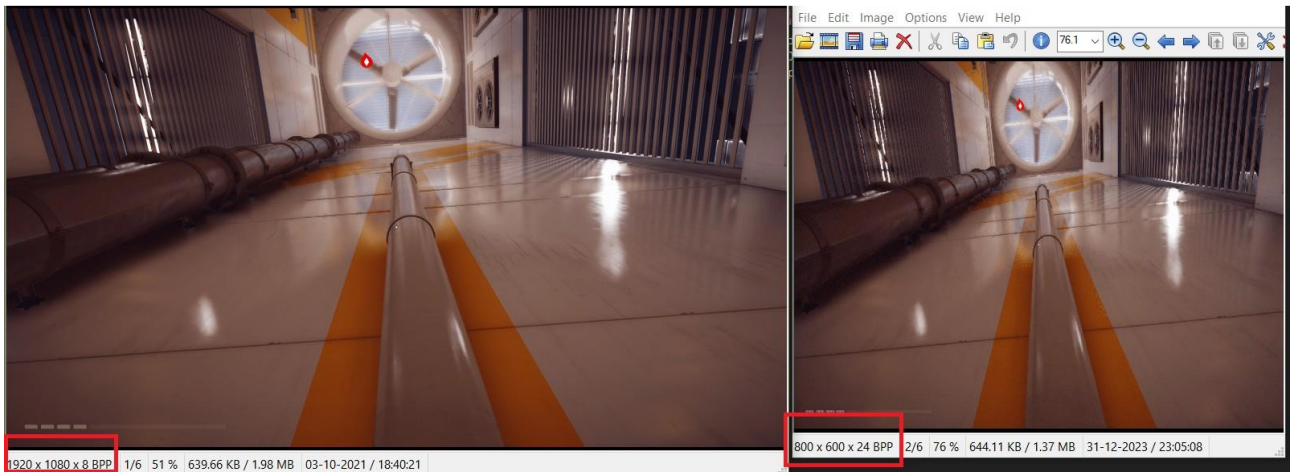
Supported filters

Below is a list of the filters currently supported by the application and the parameters to send along with each filter:

1. Resize - Changes the size of the original image.

Syntax:
Resize <Target width> <Target height>

Example:
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Resize 800 600



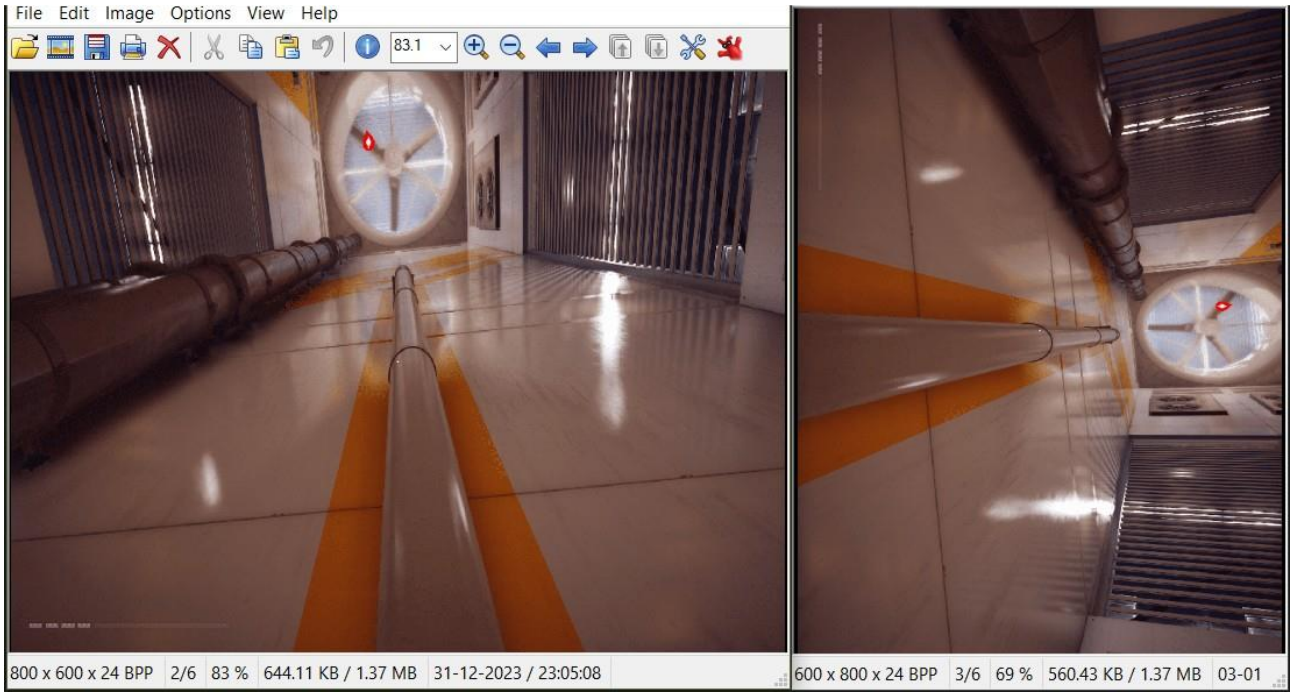
2. Rotate - Rotates the image in the specified direction, turning it the specified number of times.

Syntax:
Rotate <Rotation direction> <Number of turns>

'Rotation direction' can be either 'Clockwise' or 'Anti-clockwise' (case-sensitive).
'Number of turns' can be any positive whole number.

Examples:
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Rotate Anti-clockwise 2

127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Rotate Clockwise 1



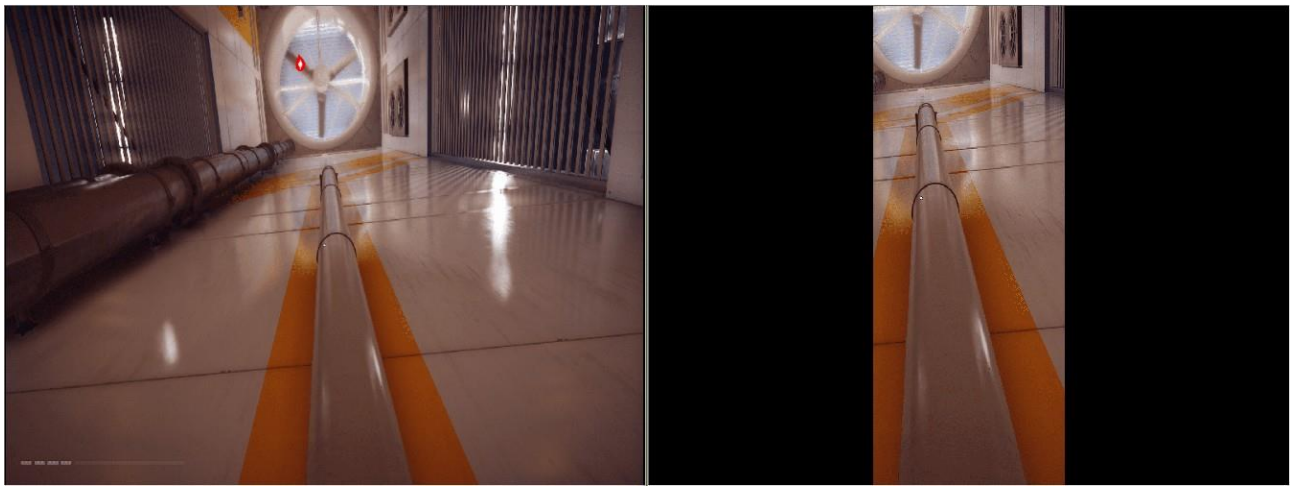
3. Crop - Crops the image according to the specified parameters.

Syntax:
Crop <X-coordinate of top left corner of crop area> <Y-coordinate of top left corner of crop area>

<Width of crop area> <Height of crop area>

Example:
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Crop 350 100 200 500

The above arguments will crop the image to an area 200 pixels wide and 500 pixels high, whose top left corner is at the point (350,100) with the origin (0,0) being on the top left corner of the original image.



4. Flip - Flips the image in the specified direction.

Syntax:
Flip <Flip direction>

'Flip direction' can be 'Vertical' or 'Horizontal' (case-sensitive).

Examples:
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Flip Horizontal

127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Flip Vertical

5. Grayscale - Converts the image to a grayscale image.

Syntax:
Grayscale

Example:
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Grayscale

6. Brightness - Changes the brightness of the original image.

Syntax:
Brightness <Brightness adjustment factor>

'Brightness adjustment factor' can be any positive decimal value.
A brightness adjustment factor of '0' gives a completely dark image, while '1' returns the same image. '2' gives an image having twice the brightness of the original.

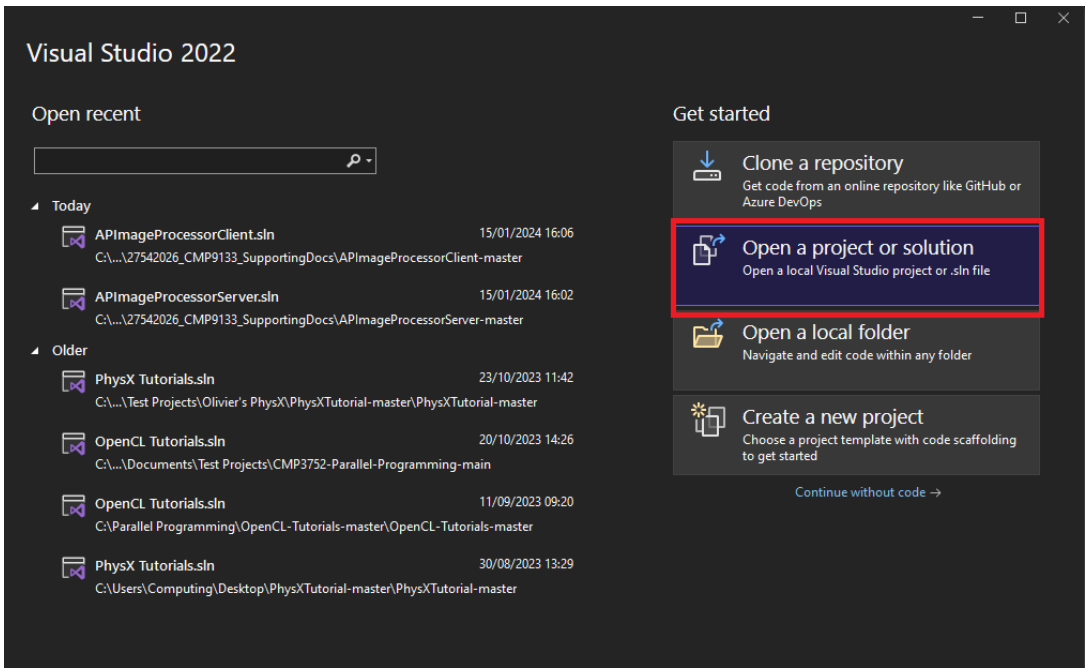
Example
127.0.0.1:8080 F:/MyImages/Gaming/Catalyst.png Brightness 1.8

Points to note

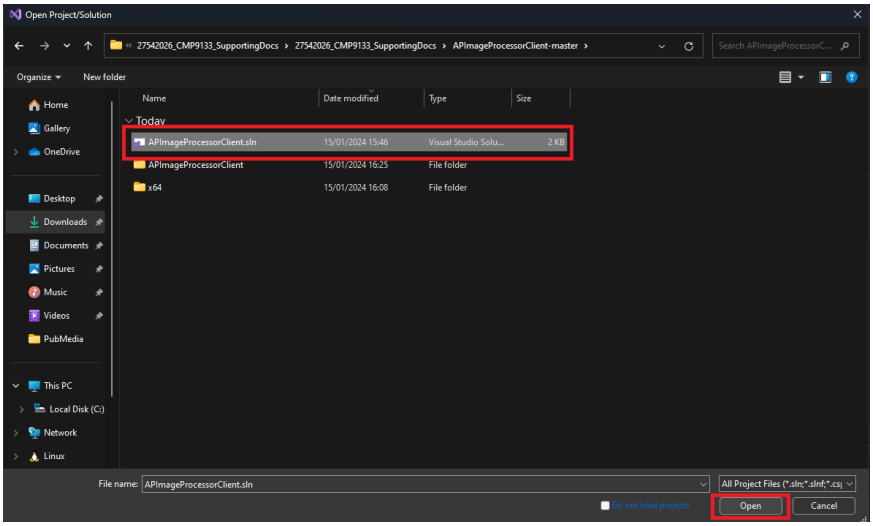
- Image path should not contain any empty characters (whitespace, newline, return, tab etc.).
- Filter names (Resize, Grayscale etc.) are case-sensitive.
- Since the server runs on port 8080 of its respective machine by default, please modify the server URL accordingly.

Running the application

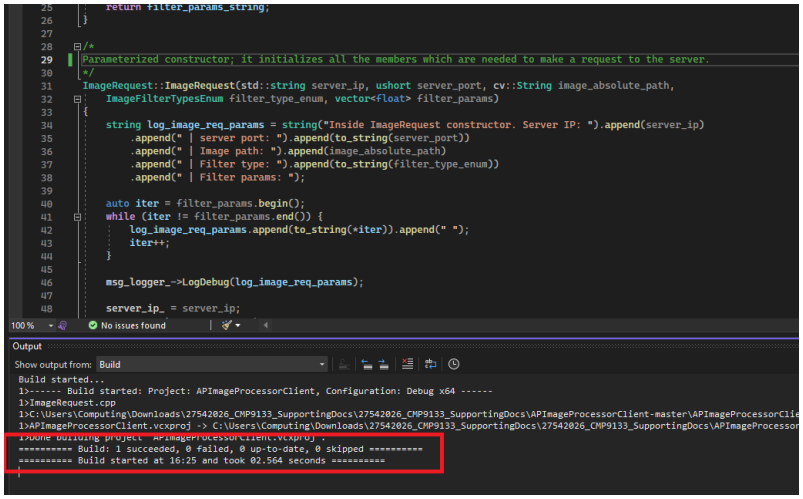
1. Start the companion server module of the application. Please refer to the README file of the server for steps to start it up.
2. Start the client module of the application i.e. this module. To do this, open Visual Studio. Visual Studio 2022 is recommended.
3. Click on ‘Open a project or solution’.



4. Browse to the 'APImageProcessorClient.sln' file and click ‘Open’.



5. Build the project in Visual Studio using Ctrl + B. Below message will be displayed on a successful build:



6. Execution starts in APImageProcessorClient.cpp. Press Ctrl + F5 in Visual Studio to run the program. Take care to provide the appropriate command line arguments.
3. On successful image processing, a console and two windows - the original and the modified images - should appear, as shown: