PHOTOS EDITOR (TOOLBOX)

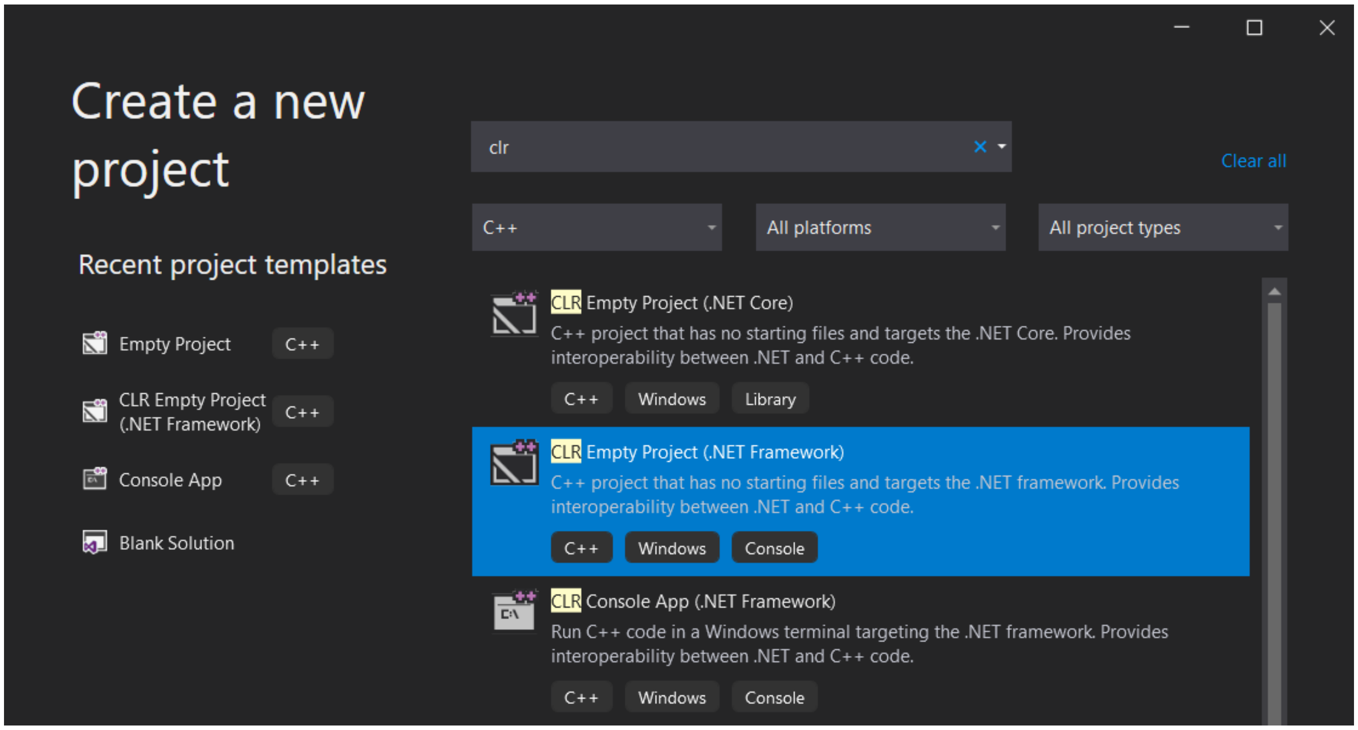
Digital Image Processing

DECEMBER 30, 2020

MAHMOUD ALI MAHMOUD
Modern academy

Steps of creating the project:

At first, we will create new CLR empty project.



Create a new project

clr

Clear all

C++ | All platforms | All project types

Recent project templates

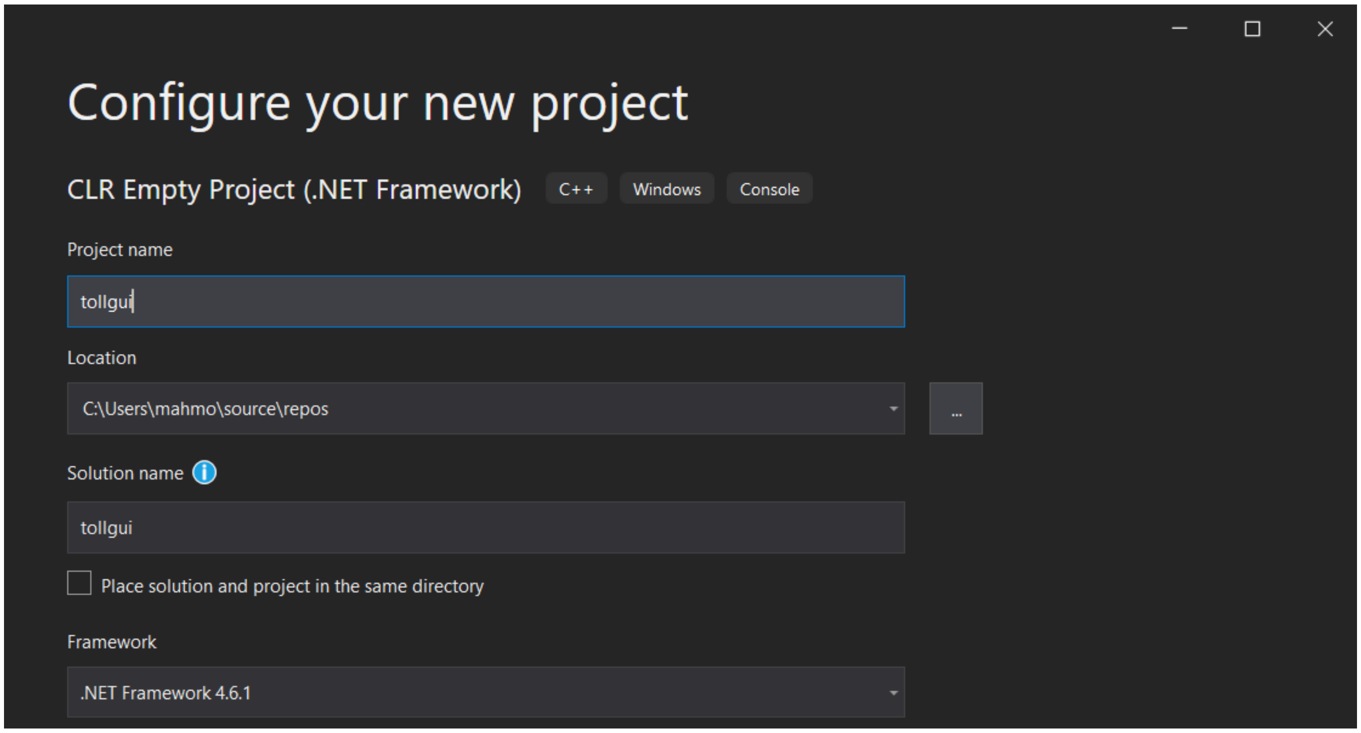
- Empty Project C++
- CLR Empty Project (.NET Framework) C++
- Console App C++
- Blank Solution

CLR Empty Project (.NET Core)
C++ project that has no starting files and targets the .NET Core. Provides interoperability between .NET and C++ code.
C++ Windows Library

CLR Empty Project (.NET Framework)
C++ project that has no starting files and targets the .NET framework. Provides interoperability between .NET and C++ code.
C++ Windows Console

CLR Console App (.NET Framework)
Run C++ code in a Windows terminal targeting the .NET framework. Provides interoperability between .NET and C++ code.
C++ Windows Console

Then name the project and select project location folder .



Configure your new project

CLR Empty Project (.NET Framework) C++ Windows Console

Project name: tollgui

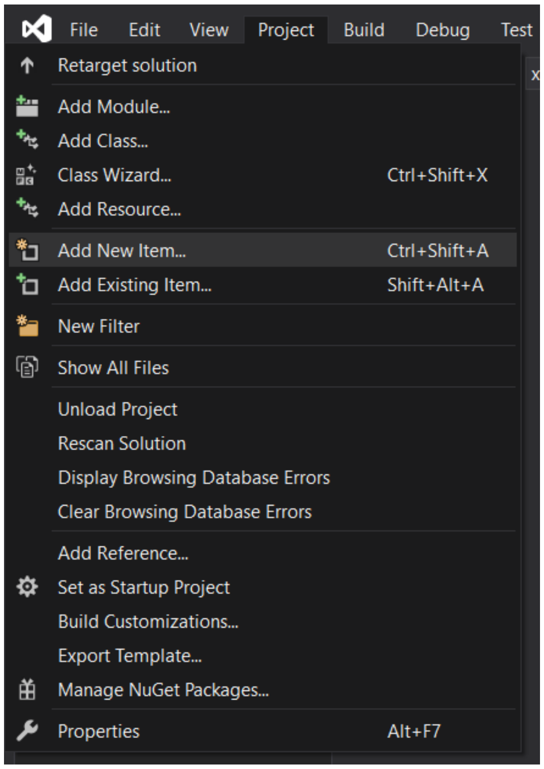
Location: C:\Users\mahmo\source\repos

Solution name: tollgui

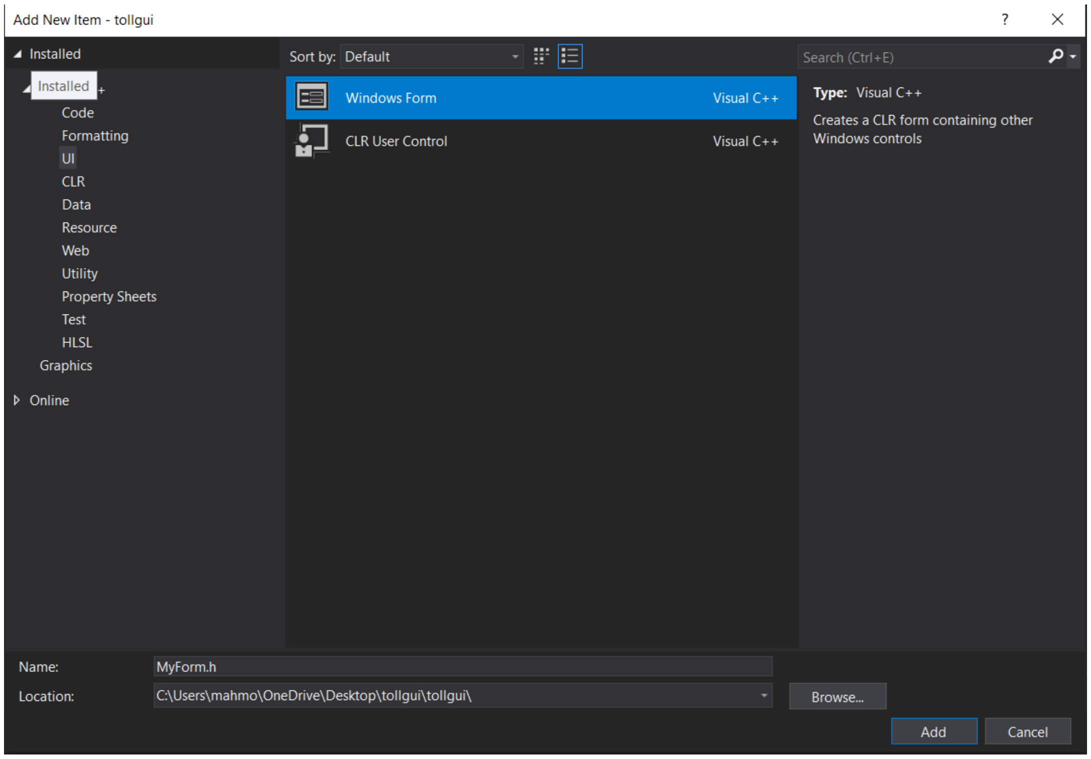
☐ Place solution and project in the same directory

Framework: .NET Framework 4.6.1

Then go to project → add new item.



In UI section select windows form and click add.



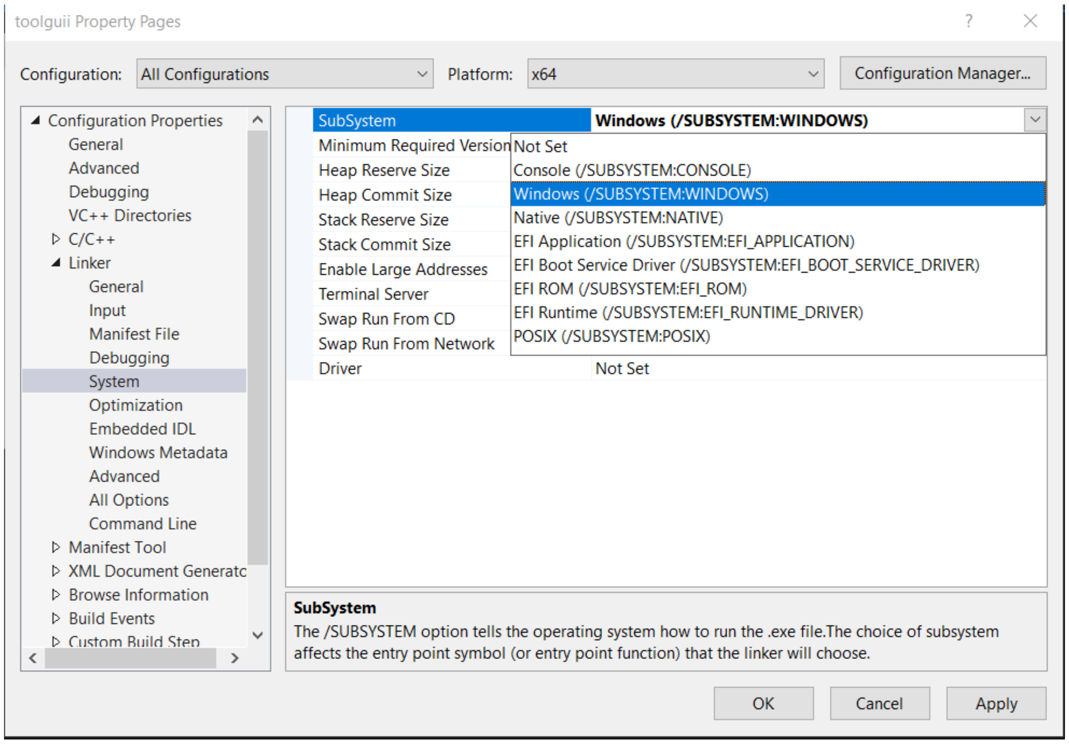
Type: Visual C++

Creates a CLR form containing other Windows controls

Name: MyForm.h

Location: C:\Users\mahmo\OneDrive\Desktop\tollgui\tollgui\

We must do some edits, go project → properties → linker → system → subsystem then choose Windows (/SUBSYSTEM:WINDOWS) and click apply.



toolguii Property Pages

Configuration: All Configurations   Platform: x64   Configuration Manager...

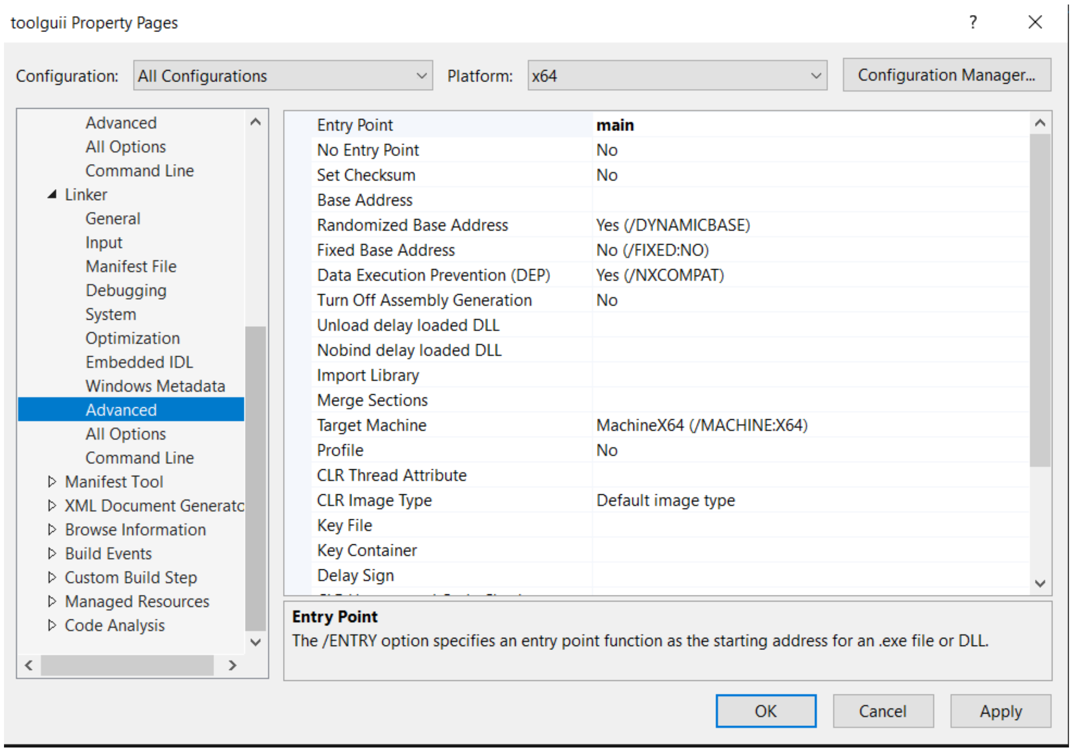
- Configuration Properties
  - General
  - Advanced
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Windows Metadata
    - Advanced
    - All Options
    - Command Line
  - Manifest Tool
  - XML Document Generatc
  - Browse Information
  - Build Events
  - Custom Build Step

SubSystem	**Windows (/SUBSYSTEM:WINDOWS)**
Minimum Required Version	Not Set
Heap Reserve Size	Console (/SUBSYSTEM:CONSOLE)
Heap Commit Size	Windows (/SUBSYSTEM:WINDOWS)
Stack Reserve Size	Native (/SUBSYSTEM:NATIVE)
Stack Commit Size	EFI Application (/SUBSYSTEM:EFI_APPLICATION)
Enable Large Addresses	EFI Boot Service Driver (/SUBSYSTEM:EFI_BOOT_SERVICE_DRIVER)
Terminal Server	EFI ROM (/SUBSYSTEM:EFI_ROM)
Swap Run From CD	EFI Runtime (/SUBSYSTEM:EFI_RUNTIME_DRIVER)
Swap Run From Network	POSIX (/SUBSYSTEM:POSIX)
Driver	Not Set

SubSystem
The /SUBSYSTEM option tells the operating system how to run the .exe file.The choice of subsystem affects the entry point symbol (or entry point function) that the linker will choose.

OK Cancel Apply

At the same window of properties go to linker → advanced →entry point and type main.



toolguii Property Pages

Configuration: All Configurations   Platform: x64   Configuration Manager...

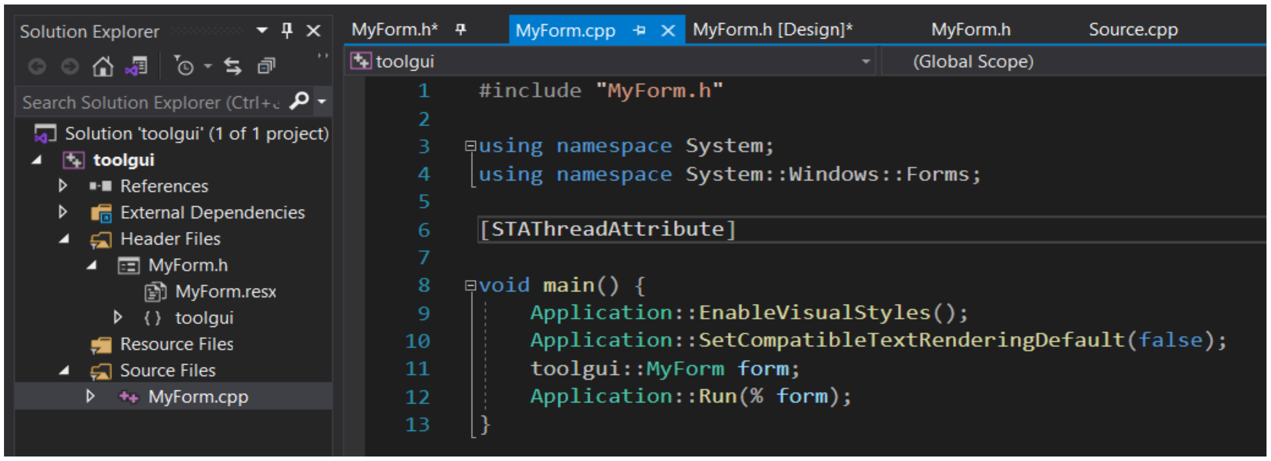
- Advanced
- All Options
- Command Line
- Linker
  - General
  - Input
  - Manifest File
  - Debugging
  - System
  - Optimization
  - Embedded IDL
  - Windows Metadata
  - Advanced
  - All Options
  - Command Line
- Manifest Tool
- XML Document Generatc
- Browse Information
- Build Events
- Custom Build Step
- Managed Resources
- Code Analysis

Entry Point	**main**
No Entry Point	No
Set Checksum	No
Base Address	
Randomized Base Address	Yes (/DYNAMICBASE)
Fixed Base Address	No (/FIXED:NO)
Data Execution Prevention (DEP)	Yes (/NXCOMPAT)
Turn Off Assembly Generation	No
Unload delay loaded DLL	
Nobind delay loaded DLL	
Import Library	
Merge Sections	
Target Machine	MachineX64 (/MACHINE:X64)
Profile	No
CLR Thread Attribute	
CLR Image Type	Default image type
Key File	
Key Container	
Delay Sign	

Entry Point
The /ENTRY option specifies an entry point function as the starting address for an .exe file or DLL.

OK Cancel Apply

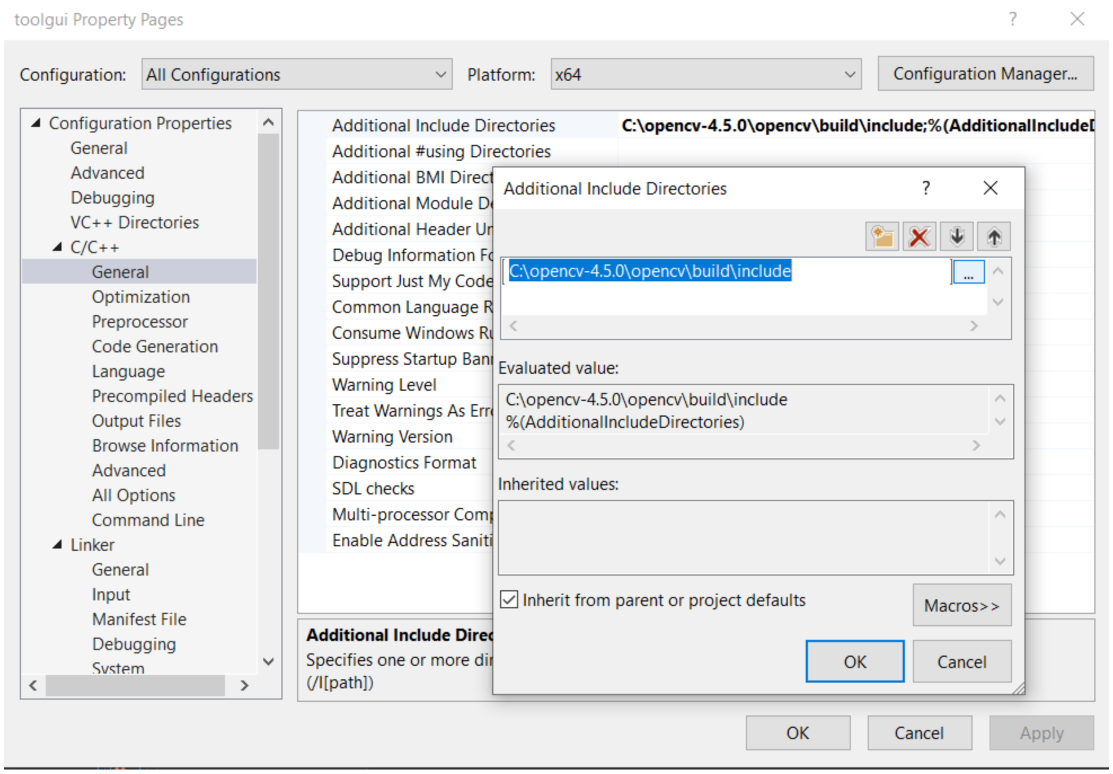
Then go to MyForm.cpp file and write this code,this code is necessary to run the project.



```
#include "MyForm.h"

using namespace System;
using namespace System::Windows::Forms;

[STAThreadAttribute]

void main() {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    toolgui::MyForm form;
    Application::Run(% form);
}
```

Because of we are using opencv libraries in our project so we must add opencv files to our project. First, you can download latest version of opencv form this link: (I used opencv-4.5.0) https://sourceforge.net/projects/opencvlibrary/files/latest/download

then go project → properties → c/c++ → general → additional include directories and select the include folder in your opencv folder.



Configuration: All Configurations
Platform: x64

Additional Include Directories C:\opencv-4.5.0\opencv\build\include;%(AdditionalIncludeD

C:\opencv-4.5.0\opencv\build\include

Evaluated value:

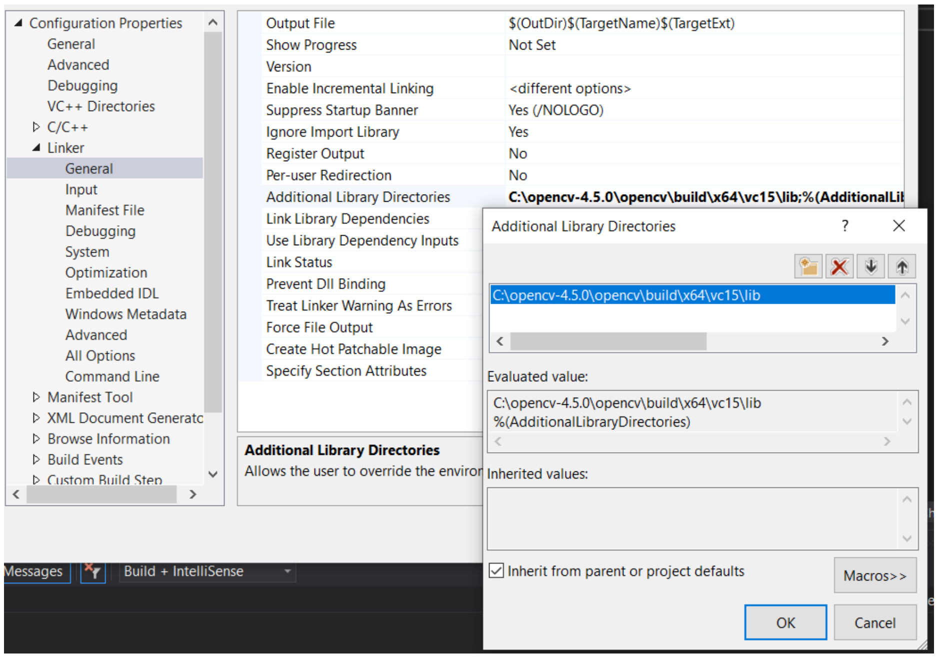
C:\opencv-4.5.0\opencv\build\include
%(AdditionalIncludeDirectories)

Inherited values:

☑ Inherit from parent or project defaults

Additional Include Direc
Specifies one or more di
(/I[path])

at the same window of project go to linker → additional library directories and select lib folder in your opencv folder then click apply.



Configuration Properties
General
Advanced
Debugging
VC++ Directories
C/C++
Linker
General
Input
Manifest File
Debugging
System
Optimization
Embedded IDL
Windows Metadata
Advanced
All Options
Command Line
Manifest Tool
XML Document Generatc
Browse Information
Build Events
Custom Build Step

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	<different options>
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	Yes
Register Output	No
Per-user Redirection	No
Additional Library Directories	**C:\opencv-4.5.0\opencv\build\x64\vc15\lib;%(AdditionalLi**
Link Library Dependencies	
Use Library Dependency Inputs	
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Additional Library Directories
Allows the user to override the enviror

Messages | Build + IntelliSense

Additional Library Directories

C:\opencv-4.5.0\opencv\build\x64\vc15\lib

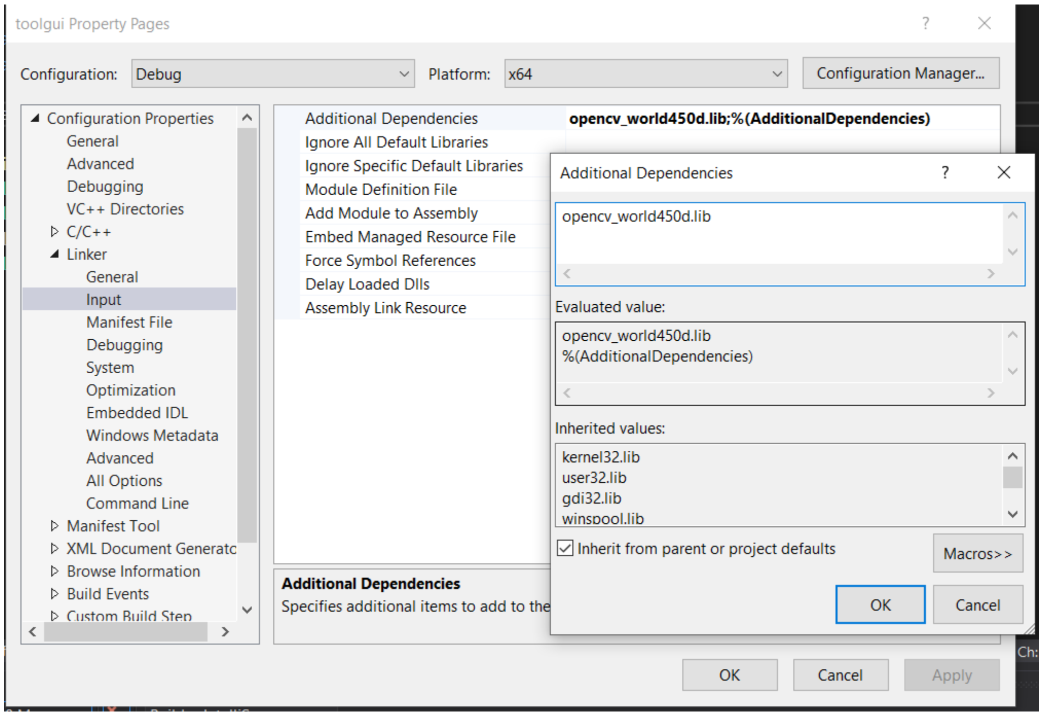
Evaluated value:
C:\opencv-4.5.0\opencv\build\x64\vc15\lib
%(AdditionalLibraryDirectories)

Inherited values:

☑ Inherit from parent or project defaults

Macros>> OK Cancel

Change Configuration→ Debug, and Change platform→ x64 then go to Linker→input → Additional dependencies and write opencv_world450d.lib (depended on your opencv version) then click apply.



toolgui Property Pages

Configuration: Debug Platform: x64 Configuration Manager...

Configuration Properties
General
Advanced
Debugging
VC++ Directories
C/C++
Linker
General
Input
Manifest File
Debugging
System
Optimization
Embedded IDL
Windows Metadata
Advanced
All Options
Command Line
Manifest Tool
XML Document Generatc
Browse Information
Build Events
Custom Build Step

Additional Dependencies	**opencv_world450d.lib;%(AdditionalDependencies)**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the

Additional Dependencies

opencv_world450d.lib

Evaluated value:
opencv_world450d.lib
%(AdditionalDependencies)

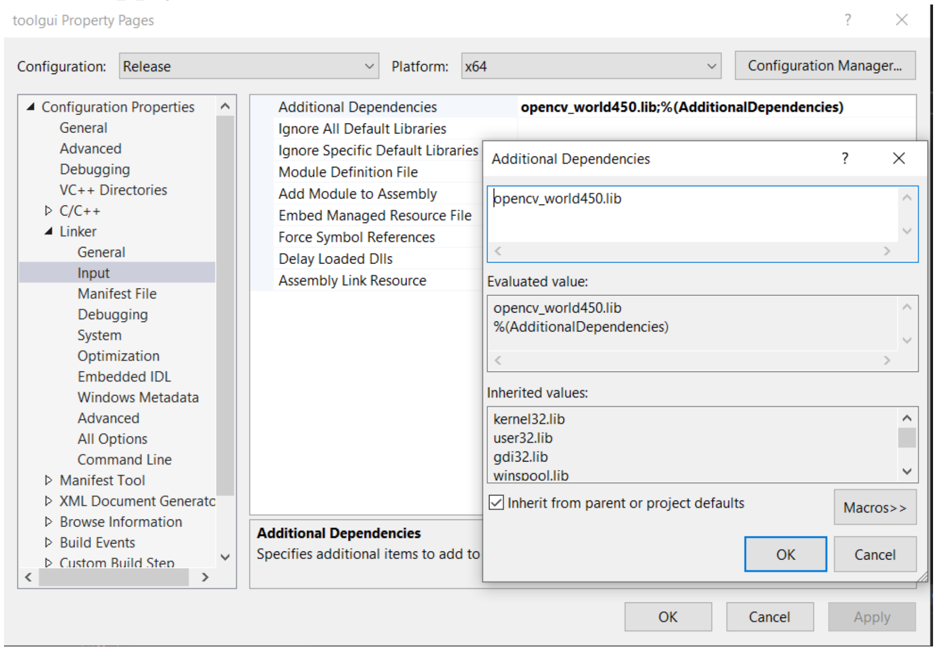
Inherited values:
kernel32.lib
user32.lib
gdi32.lib
winspool.lib

☑ Inherit from parent or project defaults

Macros>> OK Cancel

OK Cancel Apply

Change Configuration→ Release and platform→ x64 then go to Linker→input → Additional dependencies and write opencv_world450.lib (depended on your opencv version) then click apply.



toolgui Property Pages

Configuration: Release   Platform: x64   Configuration Manager...

- Configuration Properties
  - General
  - Advanced
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Windows Metadata
    - Advanced
    - All Options
    - Command Line
  - Manifest Tool
  - XML Document Generatc
  - Browse Information
  - Build Events
  - Custom Build Step

Additional Dependencies	**opencv_world450.lib;%(AdditionalDependencies)**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to

Additional Dependencies

opencv_world450.lib

Evaluated value:

opencv_world450.lib
%(AdditionalDependencies)

Inherited values:

kernel32.lib
user32.lib
gdi32.lib
winspool.lib

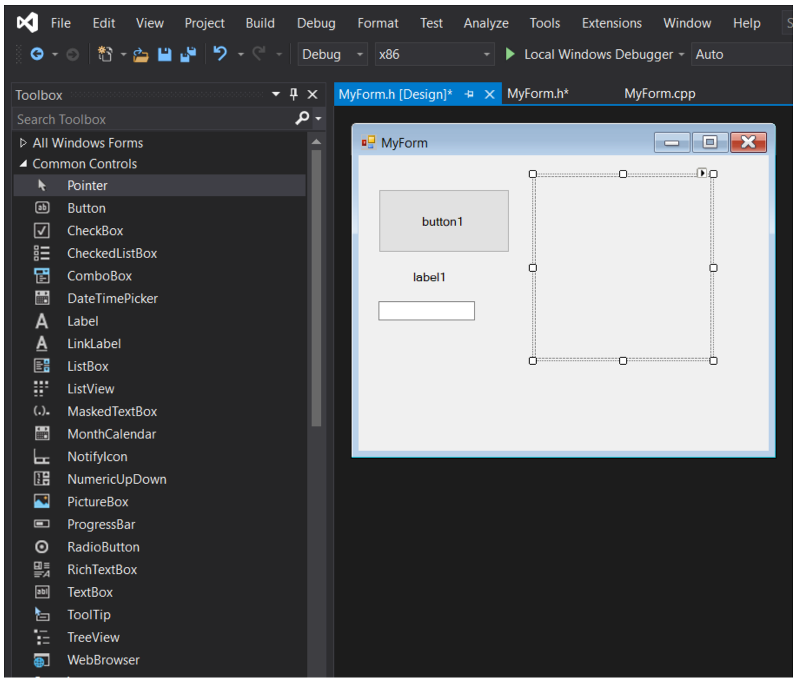
☑ Inherit from parent or project defaults   Macros>>

OK   Cancel

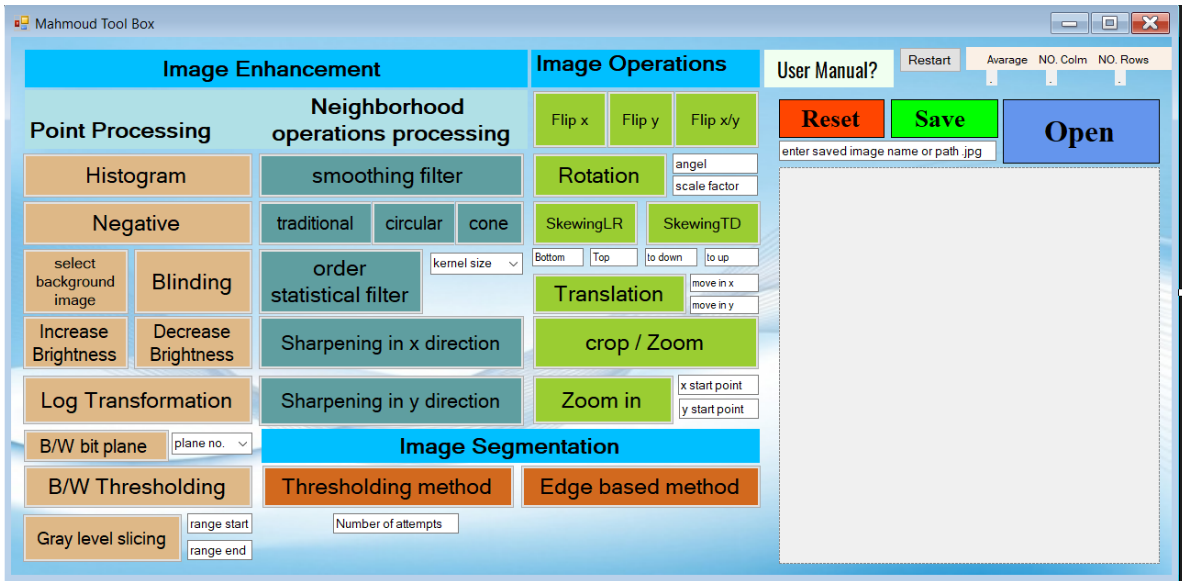
OK   Cancel   Apply

Now we added all opencv libraries we need.

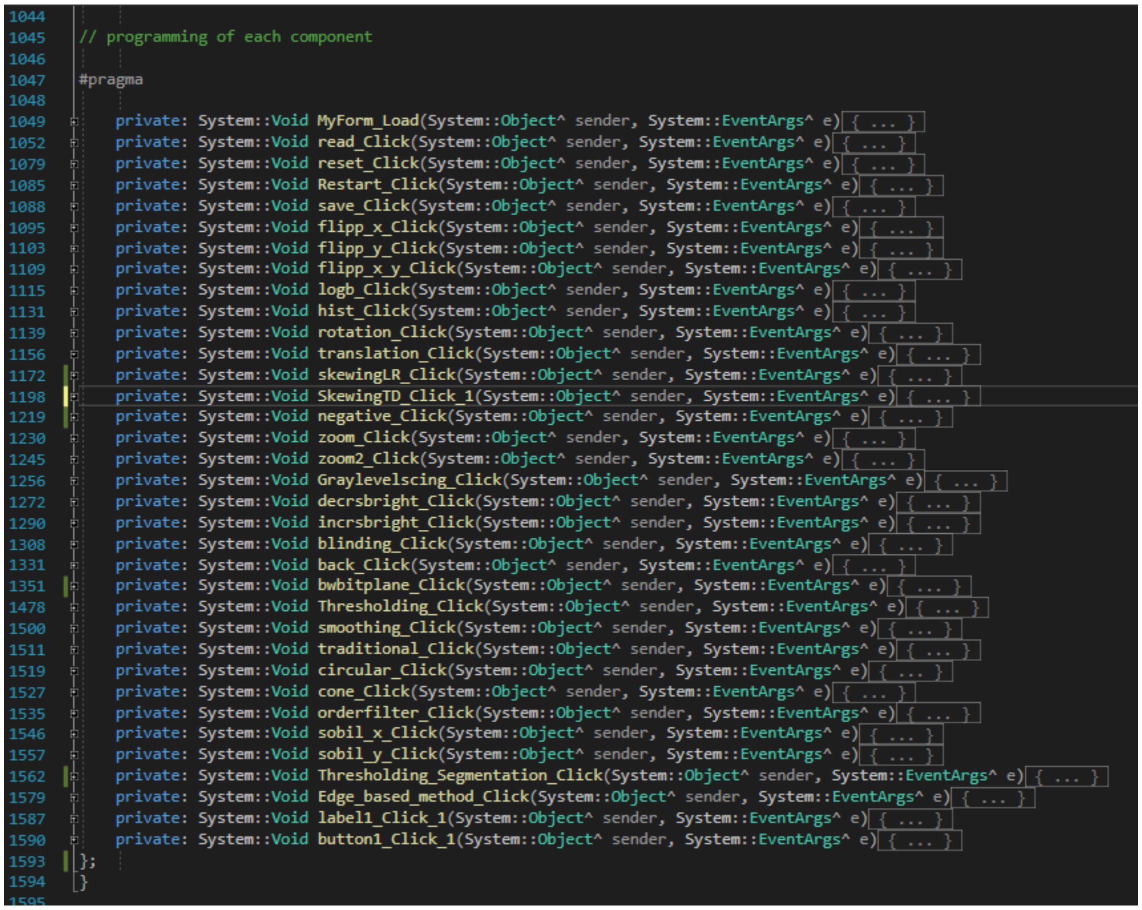
Now we can design our GUI by using (MyForm.h[design]) header file, by drag and drop each component we will use in this toolbox.



then rearrangement these components in a suitable design for user.



then we will program each component to do specific function in MyForm.h header file by double click on each component then type function code for each button.


```
// programming of each component

#pragma

    private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void read_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void reset_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void Restart_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void save_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void flipp_x_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void flipp_y_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void flipp_x_y_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void logb_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void hist_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void rotation_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void translation_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void skewingLR_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void SkewingTD_Click_1(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void negative_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void zoom_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void zoom2_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void Graylevelscing_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void decrsbright_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void incrsbright_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void blinding_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void back_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void bwbitplane_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void Thresholding_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void smoothing_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void traditional_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void circular_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void cone_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void orderfilter_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void sobil_x_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void sobil_y_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void Thresholding_Segmentation_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void Edge_based_method_Click(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void label1_Click_1(System::Object^ sender, System::EventArgs^ e){ ... }
    private: System::Void button1_Click_1(System::Object^ sender, System::EventArgs^ e){ ... }
};
}
```

How the program operates:

You can run the program by Open .exe file that called toolgui.exe or by open toolgui.sln in visual studio program then run the program. Once you run the program, main gui will appear which contain a number of button, each one do specific function.

Open button: First, we should choose the target photo that we want to apply edit on it by clicking on open button then choose the photo. Target photo will appear in the picture box that located below the open button. now we can apply any effect we want in the selected photo. and the path of selected photo will appear in the textbox below the save button.

Save button: After we apply the wanted effect and get the required edit in photo, we can save our edited photo by type path or name of the image that we want to save the photo as and her extension in the textbox below the save button then clicking in save button.

When apply an edit in the photo then apply another edit, the second edit will affect in the last change in the photo. in another meaning last edit output will used as input in the next edit.

Reset button: by clicking on Reset button the selected photo will return to its original case.

Restart button: by clicking on restart button program will restart and you can choose new photo to edit it.

Flip buttons (image mirror): user can flip the photo around x or y axis or both.

Rotation button: user can rotate the photo by any angle he wants and by any scale factor, dependent on his input. (note: rotation angle operate anticlockwise).

Skewing buttons: all parallel lines in the original image will still be parallel in the output image. This transform is obtained from the relation between three or four points. if we want to change the upper position of the photo, we will input the effect value in the (top) textbox or if we want to change the lower position, we will input the effect value in the (bottom) textbox.

Translation button: user can move the image by any value he wants in both x and y axis dependent on his input.

Crop / Zoom button: user can crop part of image and zoom the image 2x times by enter the start point that he want to make zoom from (x start point, y start point).

Zoom in button: user can zoom the image 1.1x times every click by enter the start point that he want to make zoom from (x start point, y start point).

Histogram button: It Is a method which increases the dynamic range of the gray levels in a low contrast image to cover the full range of gray levels in order to improve image contrast.

Negative button: It’s used for enhancing white or grey detail embedded in dark regions of an image.

Select background image button: lets user to choose background image that used for blinding.

Blinding button: used to merge two images or to hide some information in the image.

Increase Brightness button: we will use Power law (Gamma) transformation to increase image Brightness by using value of gamma <1 (0.9),this will map a narrow range of dark input values into a wider range of output values.

Decrease Brightness button: we will use Power law (Gamma) transformation to Decrease image Brightness by using value of gamma >1 (1.1), this will map a narrow range of bright input values into a wider range of output values.

Log transformation button: It is used to expand values of dark pixels, while compressing higher-level values (increase brightness).

B/W bit plane: we will use Bit plane slicing method that representing an image with one or more bits of the byte used for each pixel. Where M.S.B usually contain most of the significant visual information. User can choose plane 1:7 from combobox and he can use plane 7(AND with 128) to convert the image to black and white image and also to Compression the Image (Representing an image with fewer bits)

B/W Thresholding button: It's the simplest segmentation method It is used to differentiate the pixels we are interested; we perform a comparison of each pixel intensity value with respect to a threshold value (determined according to the problem to solve).so, we can get perfect black and white image by set average pixels value as Thresholding value. User can enter number of attempts he want to try.

Gray level slicing button: It's a way to highlight gray range of interest to a viewer and It's used when we need to make an object more clear. User can enter gray range start point and gray range end point.

Smoothing buttons: we will use Linear low pass filters, it's the average of the pixels contained in the neighborhood of the filter kernel. his Advantage Remove spark noise (reduce irrelevant detail in image) but his Disadvantages is bullring edges. there is Different kernels, we will use pyramidal filter Because this filter produces highest smoothing. user can use also traditional, circular or cone filters to get different smoothing levels.

Order Statistical Filter button: it is replacing the value of the center pixel of kerenl with the value determined by the ranking result. his Advantages is removing salt and pepper noise while preserved the Edges. from its types; Min, Max or Median filter, we will use Median filter. User can enter choose kernel size from combobox.

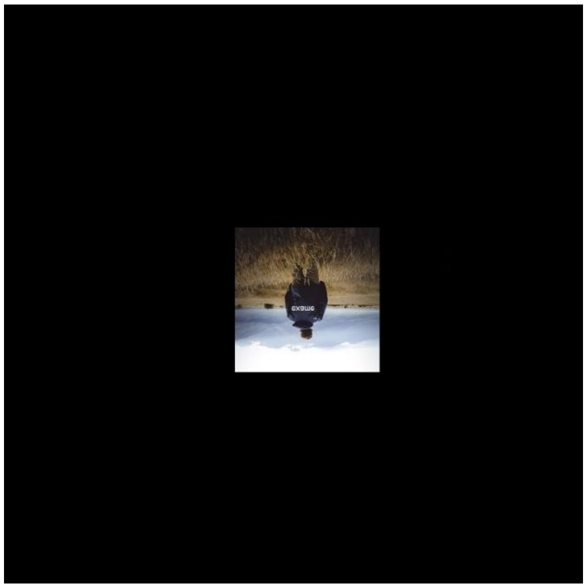
Sharpening buttons: we will use Sharpening Spatial filters It’s called Derivative as it produces values only if there is a change, so all masks coefficients sum to zero. It try to find edge by finding sharp changes in intensities. but his Disadvantages is it sensitive to the noise (increase spark noise) So usually it needs a smooth filter before detecting edges. we will use sobel operators to detect the edges in x-direction and in y-direction.

Thresholding method button: used to divide the image into two regions (object and background), we will use track bar to get different threshold values. User can enter number of attempts he want to try, and he must press any keyboard button to show the new result.

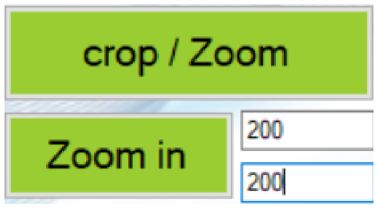
Edge Based method button: We can use this method to define a boundary of the object because There is always an edge between two adjacent regions with different grayscale values. because of edge detection is sensitive to spark noise, we need to remove it by using smoothing filter first before detecting edges (Gaussian filter). then we can detect these edges By making filters and convolutions, By using Laplace operator we can detect both horizontal and vertical edges at the same time.

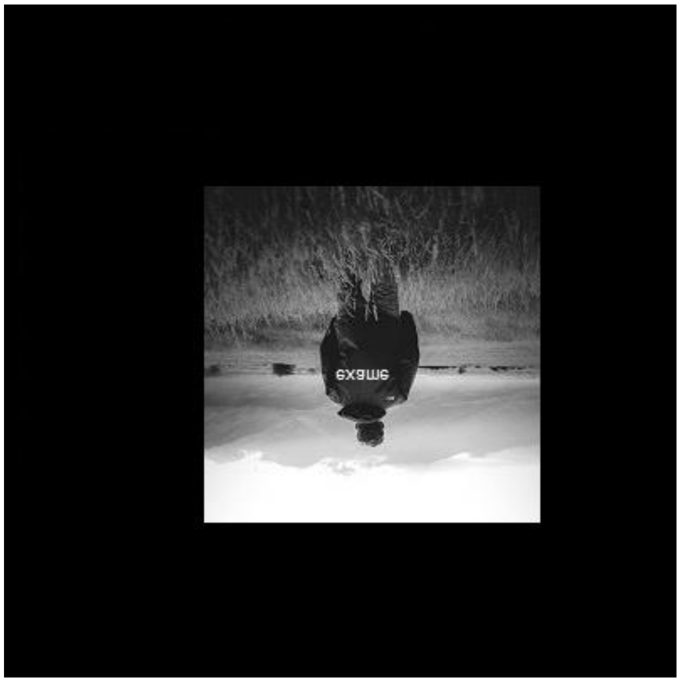
results of running the program

first, we will test the program to solve the problems in image x1, first we will read the original image (x1.jpg).

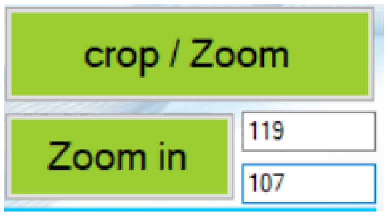

original image

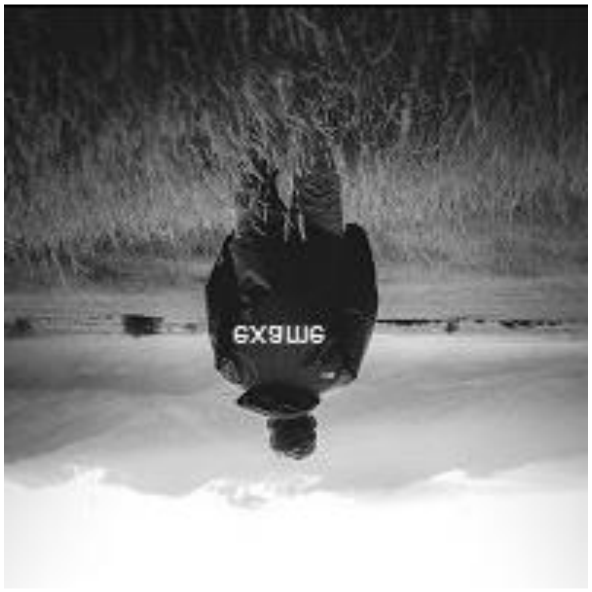
1. to enhance this image will click on crop / zoom button but first enter x start point = 200 & y start point = 200.




First output image.

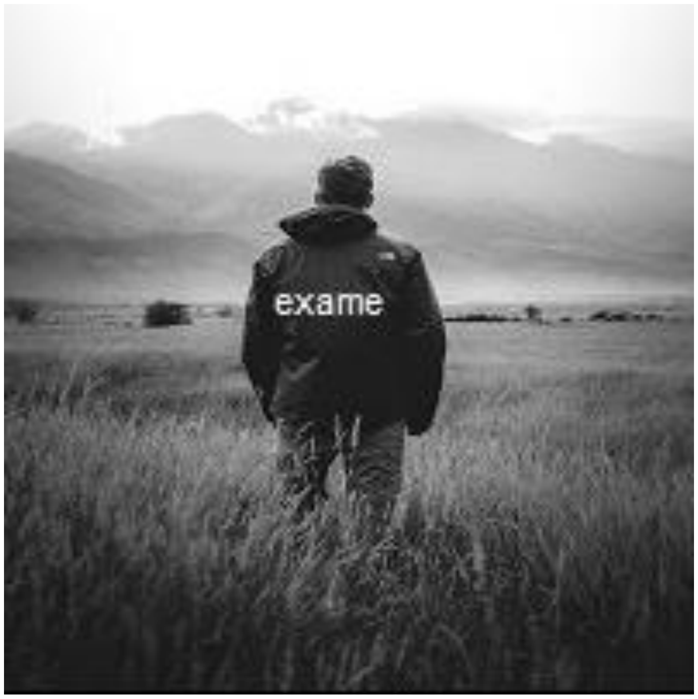
2. Then we will use same first output image and apply crop / zoom again, but this time we will change x start point to 119 and y start point to 107 (this values obtained from several trials).





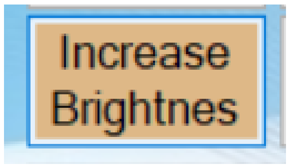
second output image.

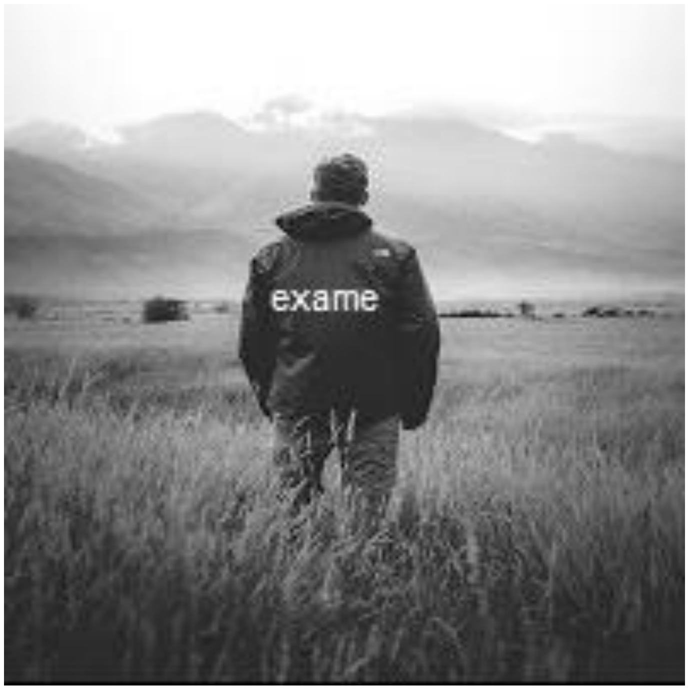
3. Then we will click flip x button to flip the image around x axis and keep (exam) word at right direction. This can be the final edit (enhanced image).



Third output image

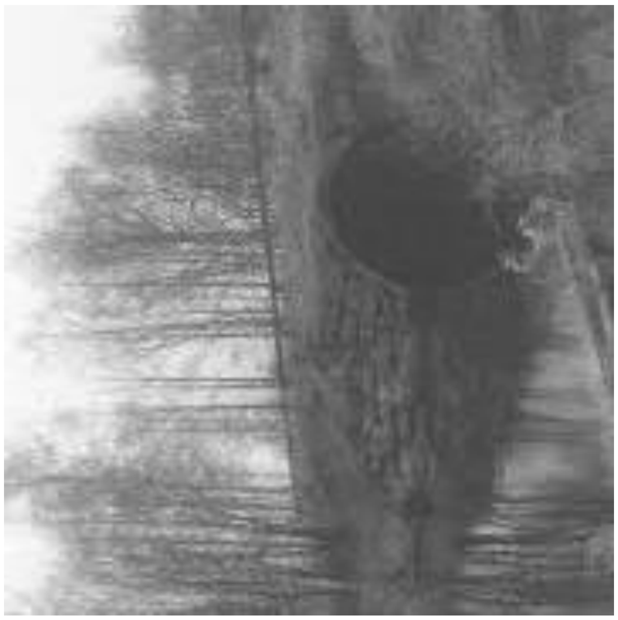
4. we can add additional option (not necessary) to increase image brightness by double click on increase brightness button twice to make the image more clear and show some hidden details.



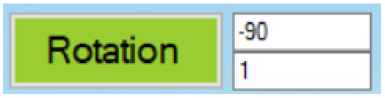


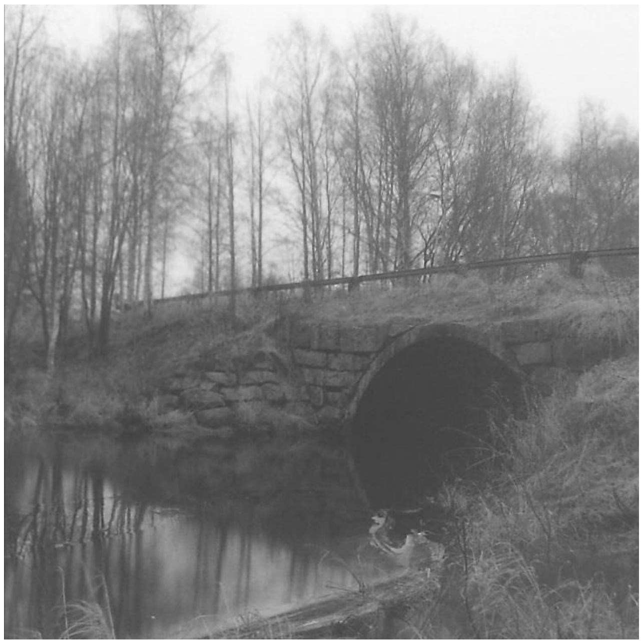
Third output image

we will test the program to solve the problems in image x1, first we will read the original image (x1.jpg).



1. to enhance this image will click rotation button but first enter angle = -90 & scale factor = 1.





Then we will click histogram button to increases the dynamic range of the gray levels in a low contrast image to cover the full range of gray levels in order to improve image contrast.