Tool Box Using Digital Image processing

☐ ❖ Steps to create the project :

Needed environment.

- Hardware

→ CPU (Intel i5/ i7/ Xeon recommended).

→ 8 GB RAM, 10 GB HDD Free Space.

- Software

→Windows 8, 10, 64 bits

→OpenCV

→ latest release

→ Visual studio latest release

OpenCV

→ OpenCV stands for open-source library for image processing and computer vision.

→OpenCV supports Windows, Linux, Android, and Mac OS.

→ OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc.

→Primary interface of OpenCV is in C++

→OpenCV 4.3.0 Download

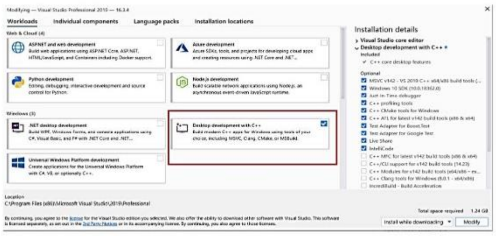
Link→ https://sourceforge.net/projects/opencvlibrary/files/4.3.0/opencv-4.3.0-vc14_vc15.exe/download

visual studio

→visual studio 2019 community free download

link→https://visualstudio.microsoft.com/downloads/

→ There are different kinds of components. What we will need is desktop development with C++ check it and goes with the install process



Modifying — Visual Studio Professional 2019 — 16.3.4

Web & Cloud (4)

ASP.NET and web development ☐
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.

Azure development ☐
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET...

Python development ☐
Editing, debugging, interactive development and source control for Python.

Node.js development ☐
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Windows (3)

.NET desktop development ☐
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET...

Desktop development with C++ ☑
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

Universal Windows Platform development ☐
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Installation details

> Visual Studio core editor
⌄ Desktop development with C++ *
Included
✓ C++ core desktop features
Optional
☑ MSVC v142 - VS 2019 C++ x64/x86 build tools (...
☑ Windows 10 SDK (10.0.18362.0)
☑ Just-In-Time debugger
☑ C++ profiling tools
☑ C++ CMake tools for Windows
☑ C++ ATL for latest v142 build tools (x86 & x64)
☑ Test Adapter for Boost.Test
☑ Test Adapter for Google Test
☑ Live Share
☑ IntelliCode
☐ C++ MFC for latest v142 build tools (x86 & x64)
☐ C++/CLI support for v142 build tools (14.23)
☐ C++ Modules for v142 build tools (x64/x86 – ex...
☐ C++ Clang tools for Windows (8.0.1 - x64/x86)
☐ IncrediBuild - Build Acceleration

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional

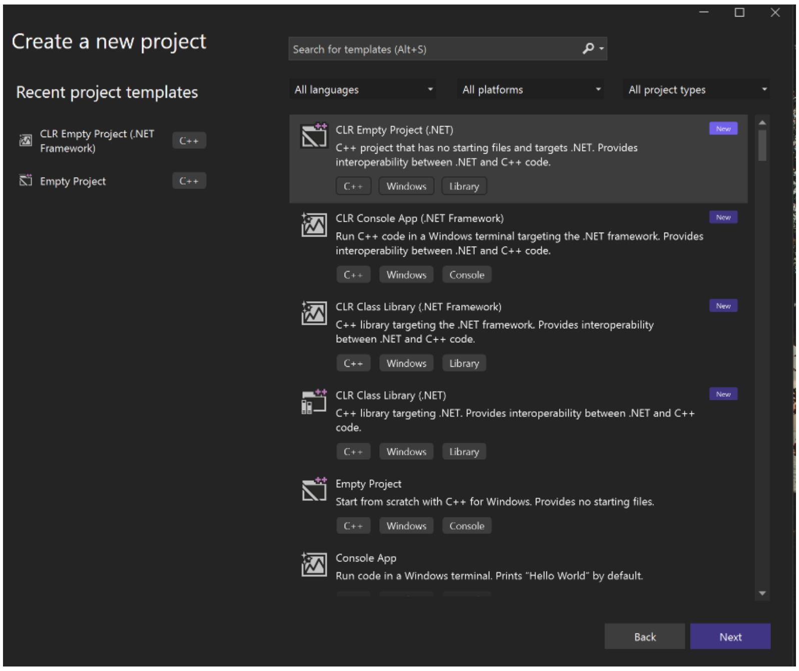
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 1.24 GB

- Create a new CLR empty project

→Choose CLR Empty Project(.NET)

→Click on Next



Create a new project

Search for templates (Alt+S)

All languages | All platforms | All project types

Recent project templates

CLR Empty Project (.NET Framework) C++

Empty Project C++

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

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

CLR Class Library (.NET Framework) New
C++ library targeting the .NET framework. Provides interoperability between .NET and C++ code.
C++ Windows Library

CLR Class Library (.NET) New
C++ library targeting .NET. Provides interoperability between .NET and C++ code.
C++ Windows Library

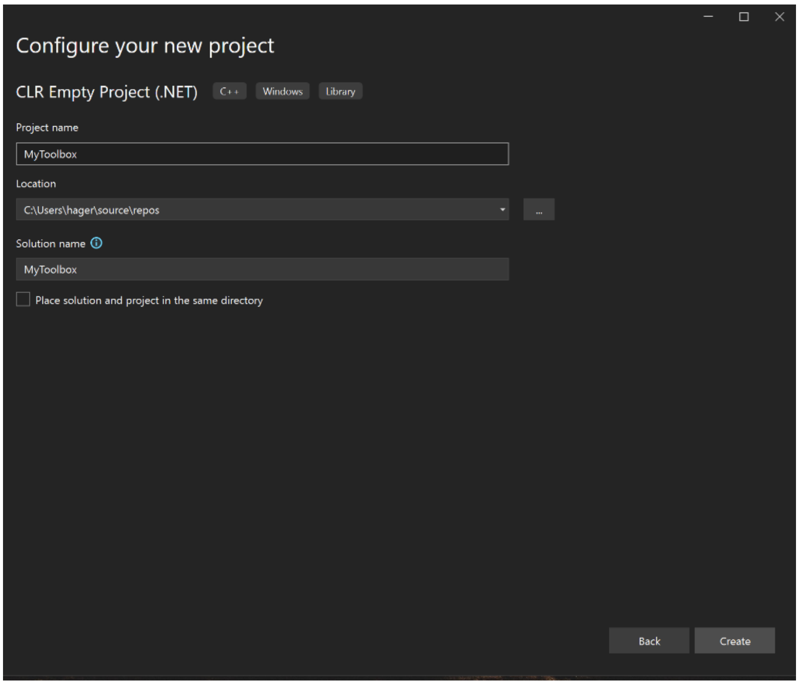
Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.

- Configure your new project

→Project name(MyToolbox)

→Click on Create



Configure your new project

CLR Empty Project (.NET) C++ Windows Library

Project name

MyToolbox

Location

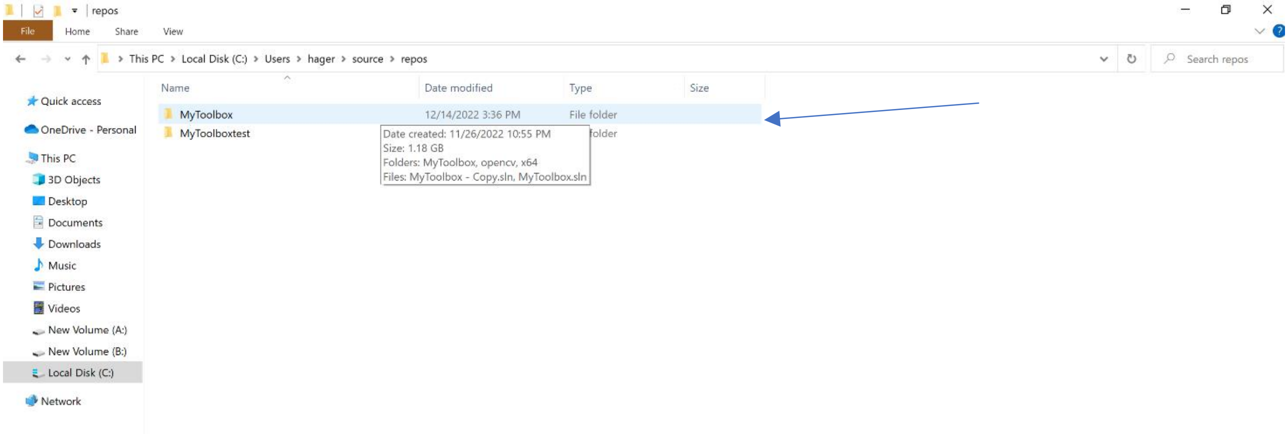
C:\Users\hager\source\repos

Solution name

MyToolbox

□ Place solution and project in the same directory

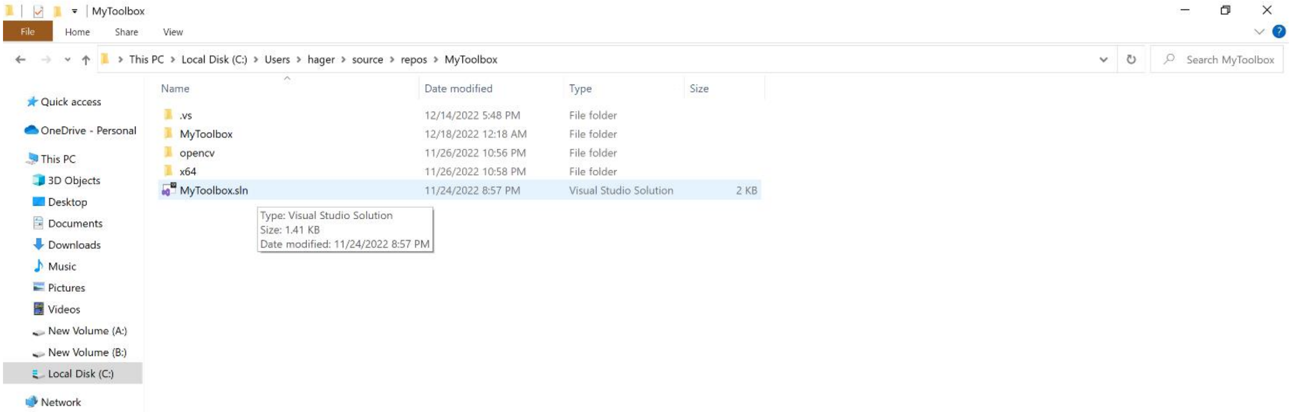
- Project created (MyToolbox)



Name	Date modified	Type	Size
MyToolbox	12/14/2022 3:36 PM	File folder	
MyToolboxtest		folder	

Date created: 11/26/2022 10:55 PM
Size: 1.18 GB
Folders: MyToolbox, opencv, x64
Files: MyToolbox - Copy.sln, MyToolbox.sln

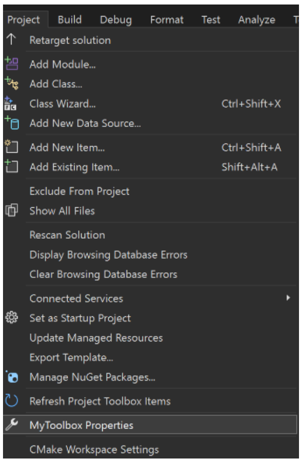
- Open MyToolbox.sln work space



Name	Date modified	Type	Size
.vs	12/14/2022 5:48 PM	File folder	
MyToolbox	12/18/2022 12:18 AM	File folder	
opencv	11/26/2022 10:56 PM	File folder	
x64	11/26/2022 10:58 PM	File folder	
MyToolbox.sln	11/24/2022 8:57 PM	Visual Studio Solution	2 KB

Type: Visual Studio Solution
Size: 1.41 KB
Date modified: 11/24/2022 8:57 PM

1. To create the project interface(UI):

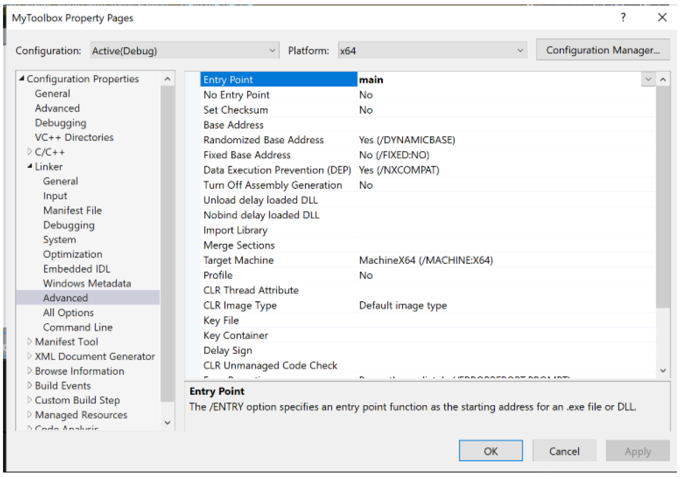


- To adjust the settings

→Project→MyToolbox Properties

→Linker→Advanced→Entry Point →main

→Click on Apply

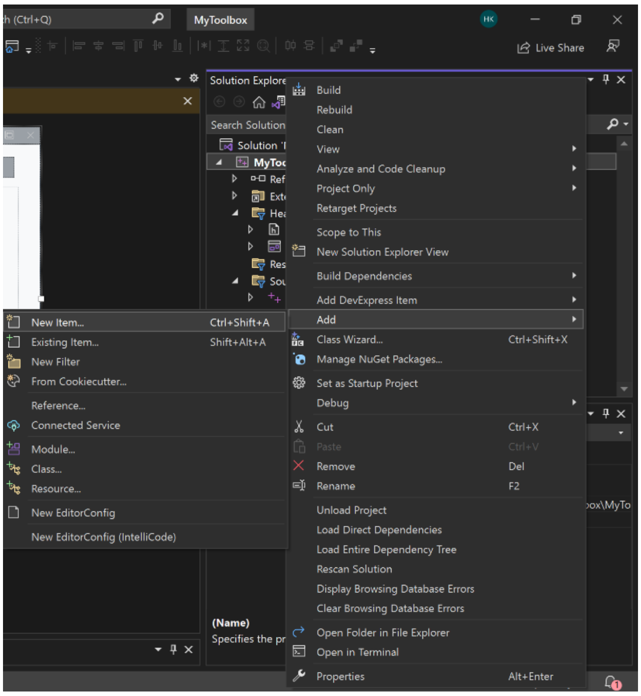


Configuration: Active(Debug)   Platform: x64

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	
CLR Unmanaged Code Check	

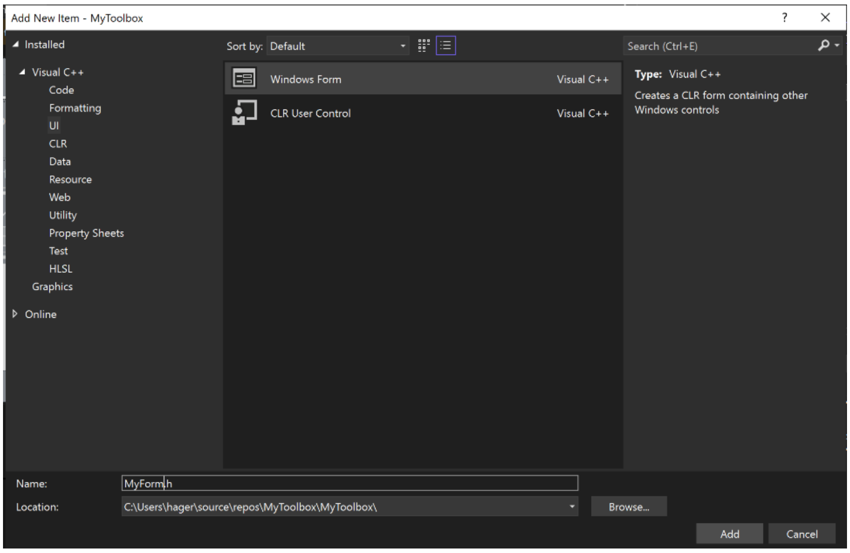
Entry Point
The /ENTRY option specifies an entry point function as the starting address for an .exe file or DLL.

→Solution Explore→Right Click→MyToolbox→Add→New item



→Add New Item→UI→Windows Form→Name→MyForm.h

→Click on Add



Type: Visual C++

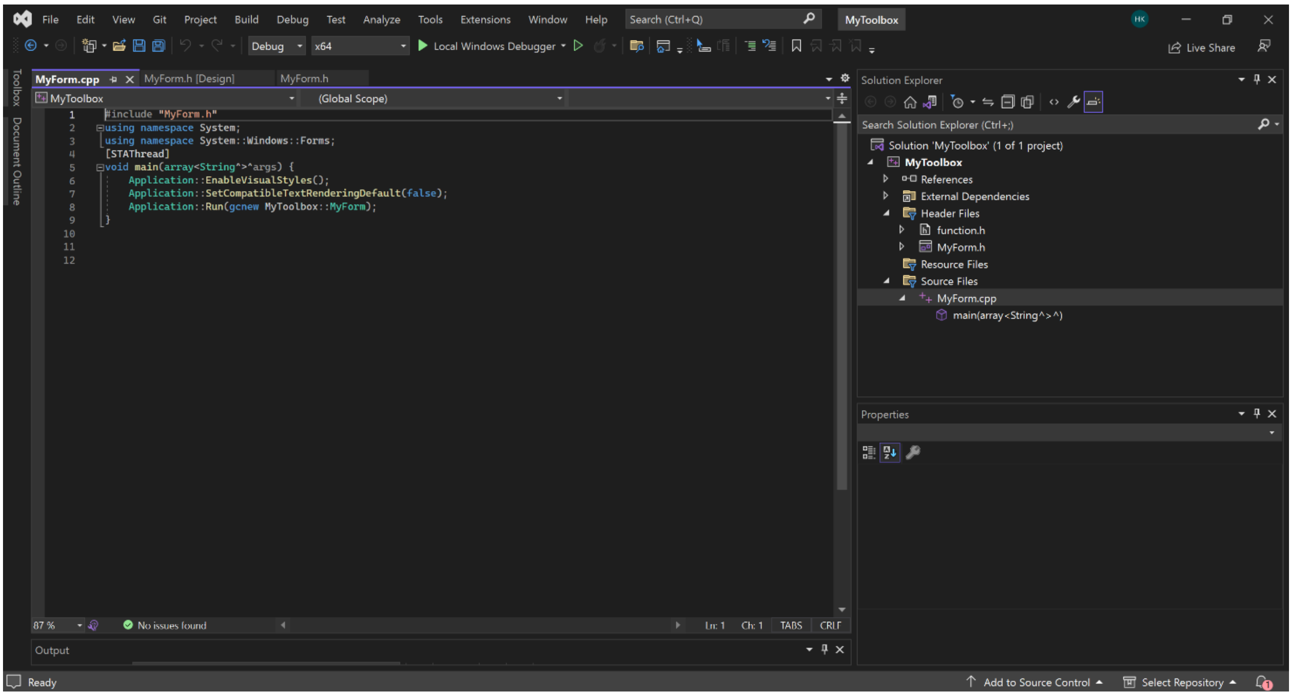
Creates a CLR form containing other Windows controls

Name: MyForm.h

Location: C:\Users\hager\source\repos\MyToolbox\MyToolbox\

→Source Files→MyForm.cpp

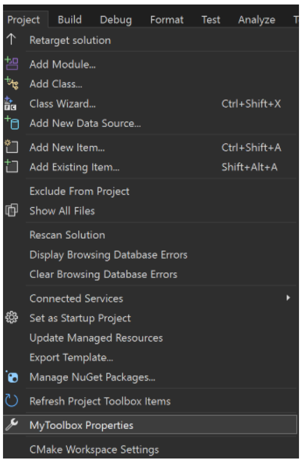
→Define Libraries and main



```
#include "MyForm.h"
using namespace System;
using namespace System::Windows::Forms;
[STAThread]
void main(array<String^>^args) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew MyToolbox::MyForm);
}
```

2. To Add opencv files to your project:

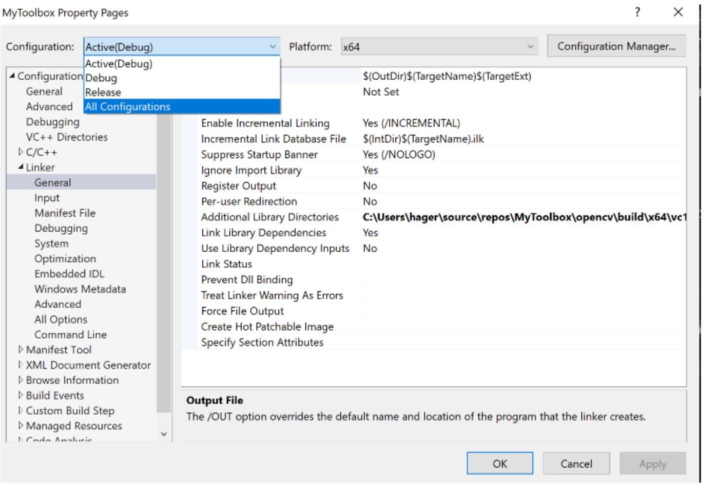
- To adjust the settings



→Project→MyToolbox Properties

→ Configuration→ all configuration

→platform→x64



MyToolbox Property Pages

Configuration: Active(Debug)

Platform: x64

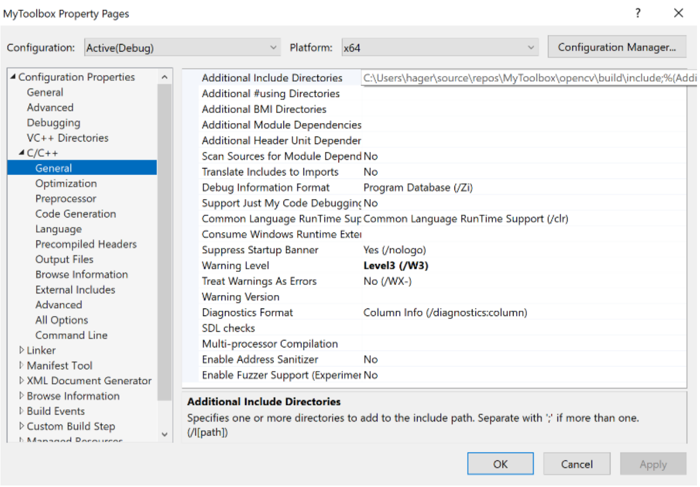
	$(OutDir)$(TargetName)$(TargetExt)
	Not Set
Enable Incremental Linking	Yes (/INCREMENTAL)
Incremental Link Database File	$(IntDir)$(TargetName).ilk
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	Yes
Register Output	No
Per-user Redirection	No
Additional Library Directories	**C:\Users\hager\source\repos\MyToolbox\opencv\build\x64\vc1**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Output File
The /OUT option overrides the default name and location of the program that the linker creates.

→C/C++→General → Additional include directory.

→Write the path of the include folder

C:\User\hager\source\MyToolbox\opencv\build\include.



Configuration: Active(Debug) Platform: x64

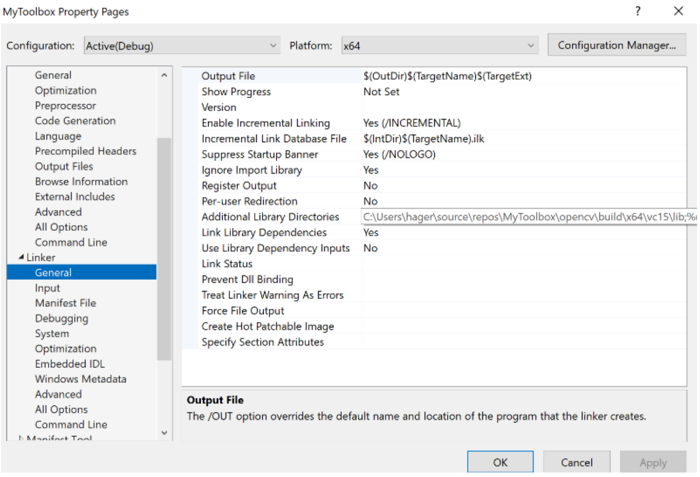
Additional Include Directories	C:\Users\hager\source\repos\MyToolbox\opencv\build\include;%(Addi
Additional #using Directories	
Additional BMI Directories	
Additional Module Dependencies	
Additional Header Unit Depender	
Scan Sources for Module Depend	No
Translate Includes to Imports	No
Debug Information Format	Program Database (/Zi)
Support Just My Code Debugging	No
Common Language RunTime Sup	Common Language RunTime Support (/clr)
Consume Windows Runtime Exte	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Warning Version	
Diagnostics Format	Column Info (/diagnostics:column)
SDL checks	
Multi-processor Compilation	
Enable Address Sanitizer	No
Enable Fuzzer Support (Experimer	No

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one. (/I[path])

→Linker→General → Additional library directories

→ Write the path of the lib folder.

C:\User\hager\source\MyToolbox \OpenCV\build\x64\vc15\lib



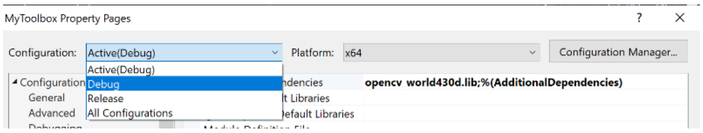
Configuration: Active(Debug) Platform: x64

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	Yes (/INCREMENTAL)
Incremental Link Database File	$(IntDir)$(TargetName).ilk
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	Yes
Register Output	No
Per-user Redirection	No
Additional Library Directories	C:\Users\hager\source\repos\MyToolbox\opencv\build\x64\vc15\lib;%(
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Output File
The /OUT option overrides the default name and location of the program that the linker creates.

→Configuration→ Debug

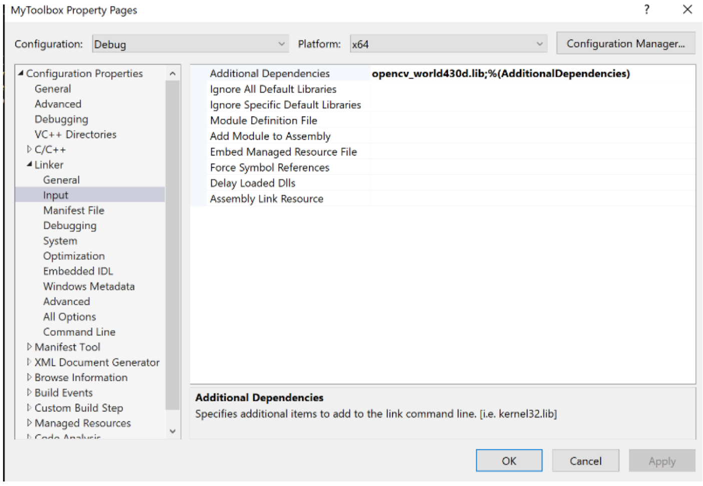
→ platform→x64



Configuration: Active(Debug) Platform: x64

ndencies	**opencv_world430d.lib;%(AdditionalDependencies)**
t Libraries	
efault Libraries	

→ Linker→input → Additional dependencies→opencv_world430d.lib



MyToolbox Property Pages

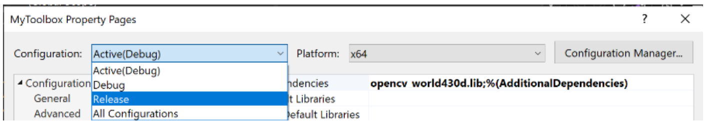
Configuration: Debug Platform: x64

Additional Dependencies opencv_world430d.lib;%(AdditionalDependencies)

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

→Configuration→Release

→ platform→x64

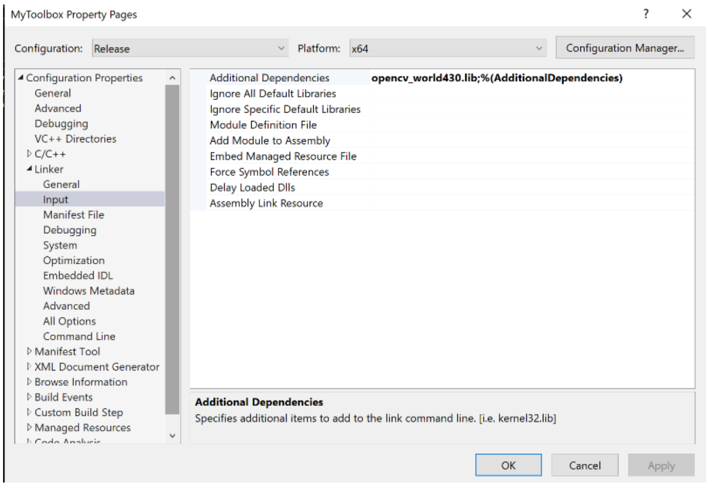


MyToolbox Property Pages

Configuration: Active(Debug) Platform: x64

opencv_world430d.lib;%(AdditionalDependencies)

→ Linker→input → Additional dependencies→opencv_world430.lib



MyToolbox Property Pages

Configuration: Release Platform: x64

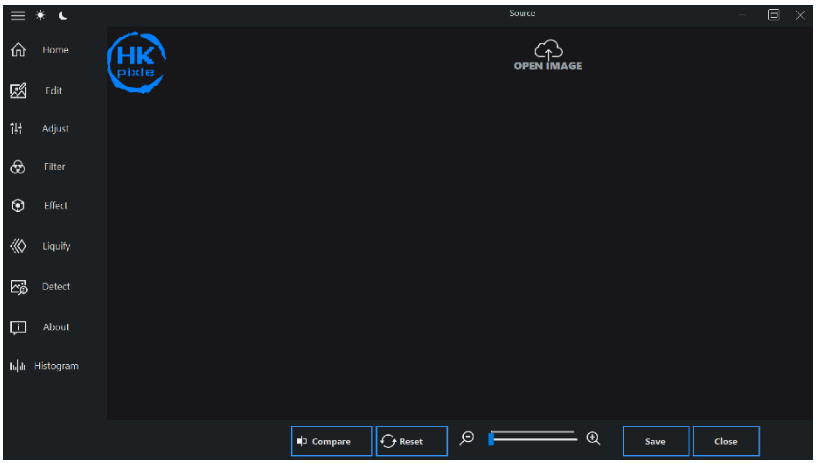
Additional Dependencies opencv_world430.lib;%(AdditionalDependencies)

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

❖ Steps to make User InterFace (UI):

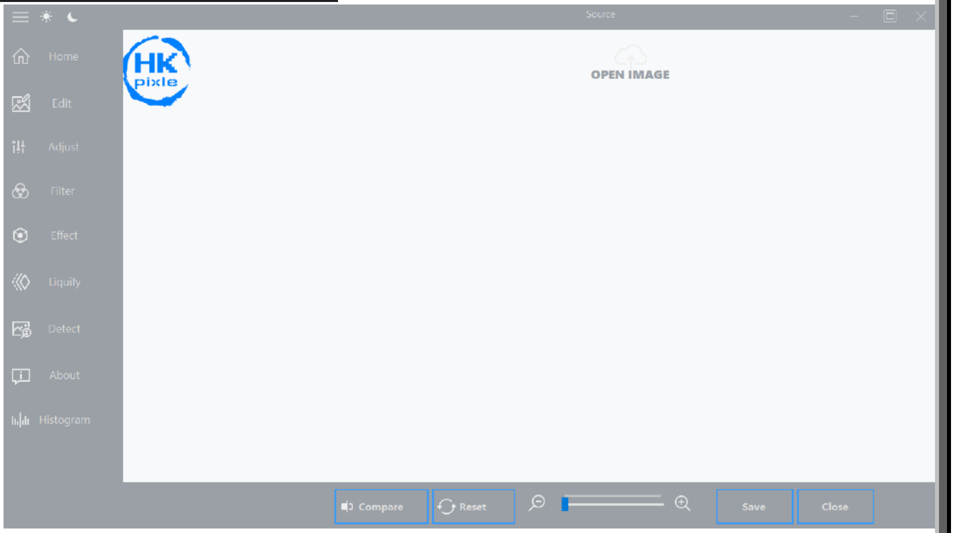
✓ Create an app in two modes(light and dark)

→To be comfortable for the users



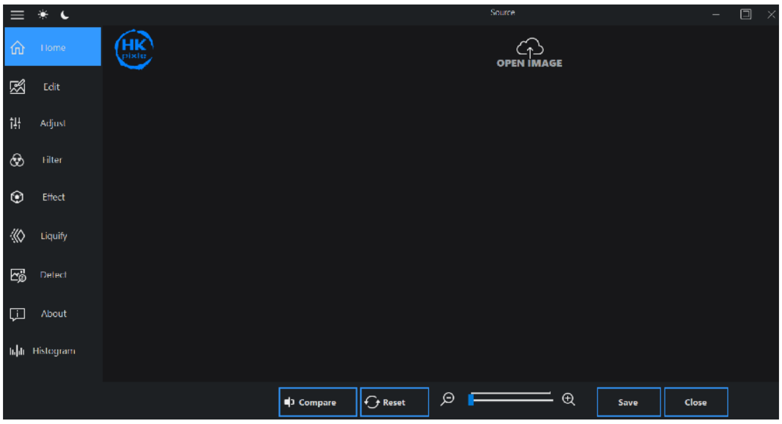
←DARK MOOD

LIGHT MOOD→

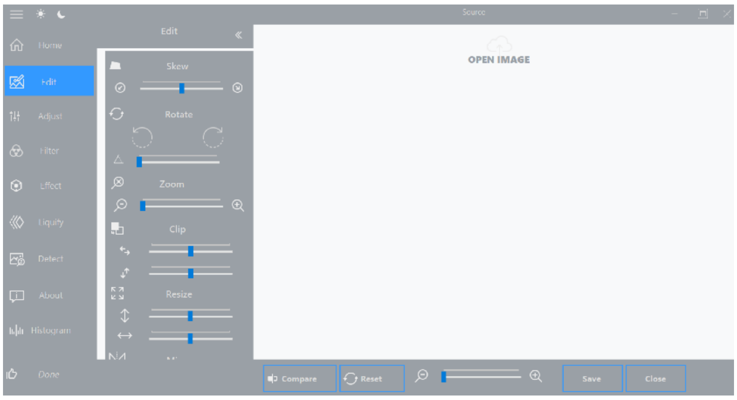


✓ The application has been divided into several sections
→ For the convenience of the users

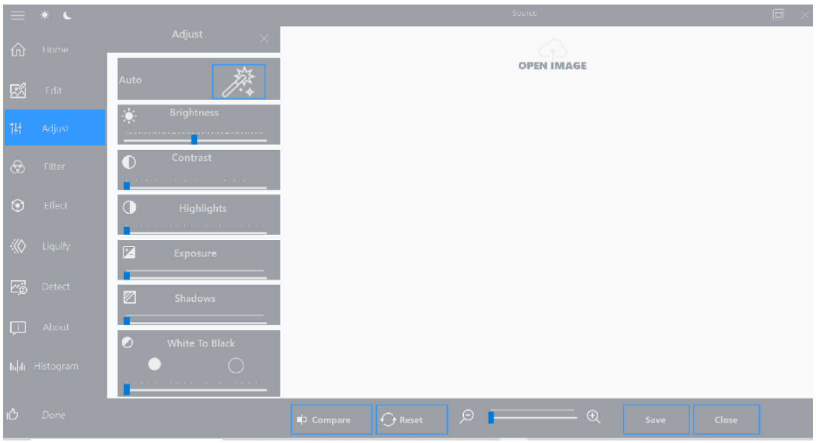
HOME SECTION→



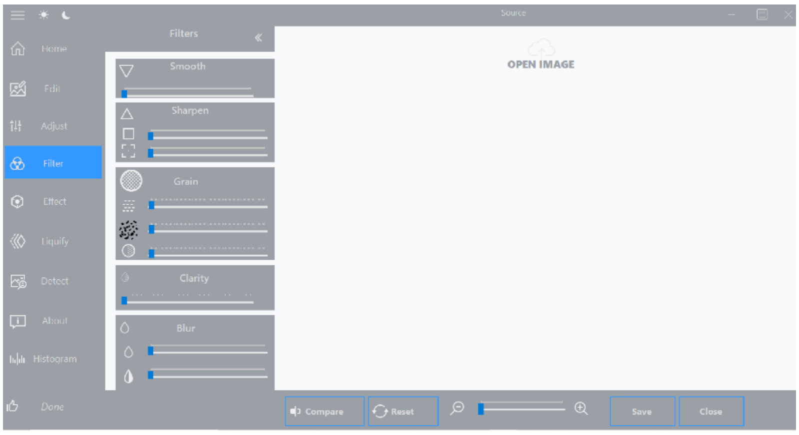
EDIT SECTION →



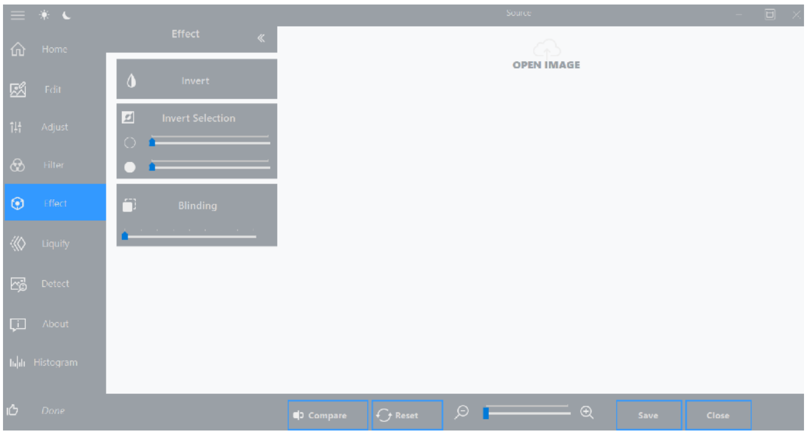
←ADJUST SECTION

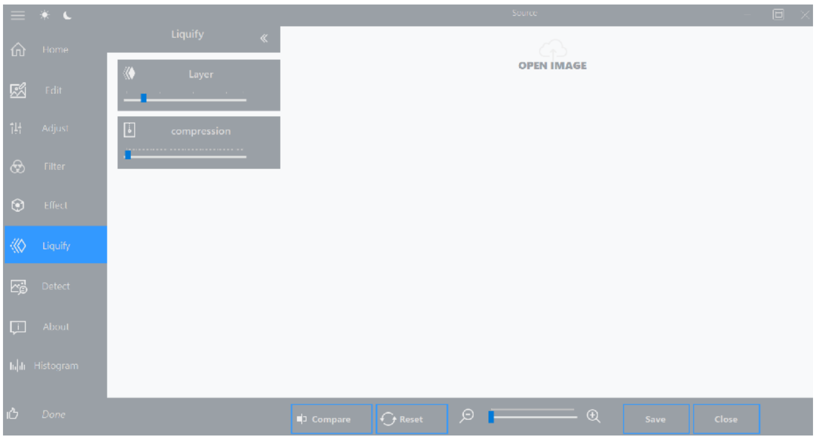


FILTER SECTION →



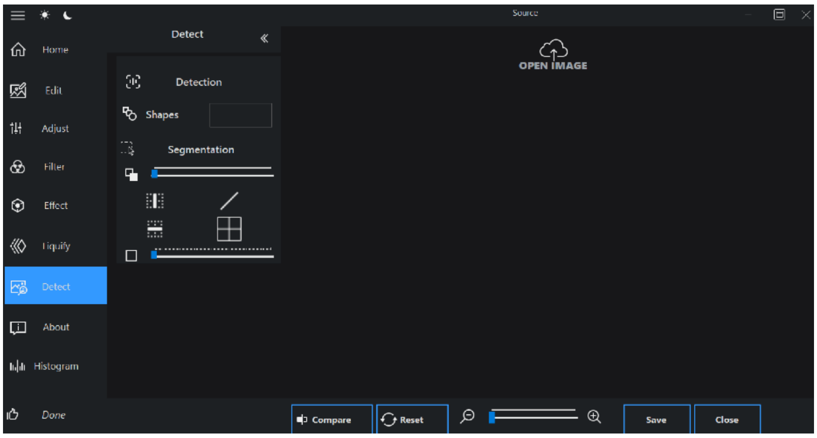
EFFECT SECTION →





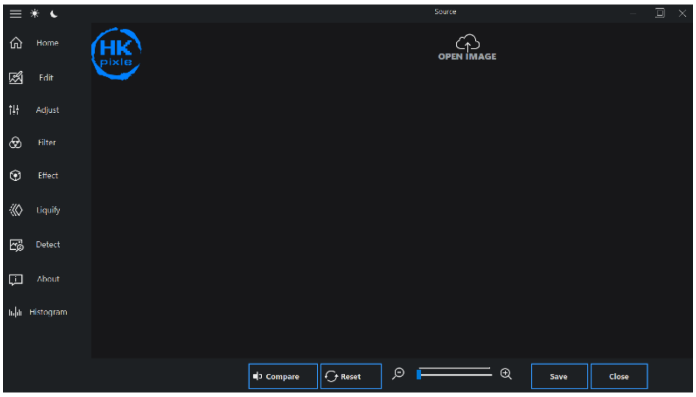
←LIQUIFIY SECTION

DETECT SECTION →



Button Functions

→Main Buttons



× ←Using to close App

☐ ←Using to Maxmize App

– ←Using to Minimize App

☀ ←Use to convert App Dark mode to Light mode

☾ ← Use to convert App mode Light to Dark mode

☰ ←Use to minmize Menu

Compare ←Open Original Image

Reset ←Restart App

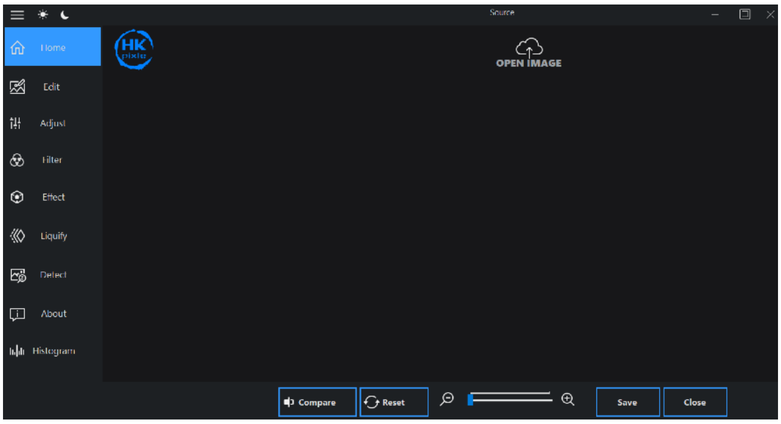
⊖ ⊕ ←Zooming in and out

Save ←Save enhancement Image and open window to choose location wanted to save

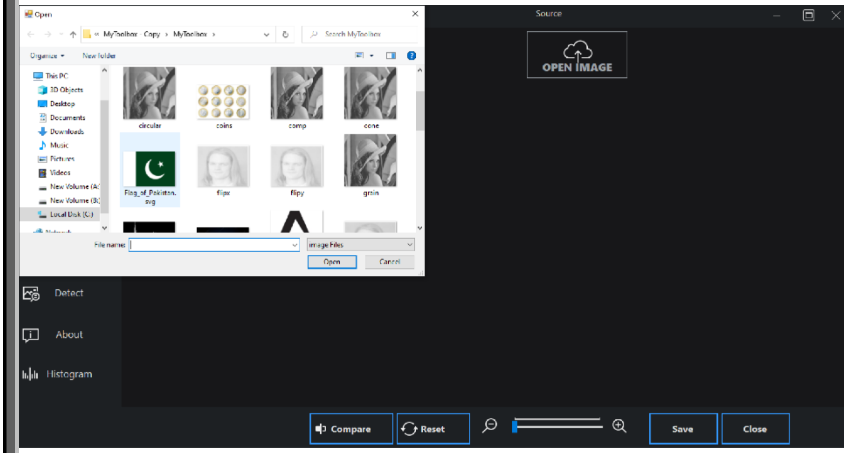
Close ←Close Current Image

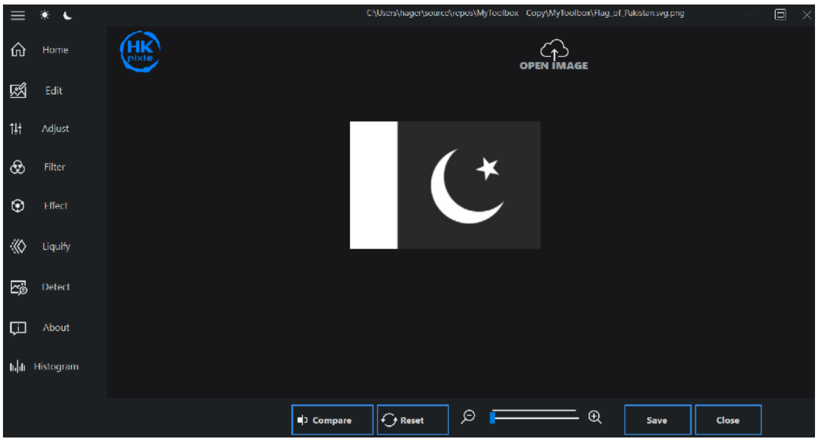
Done ←To Save and update enhance on Image.

→HOME SECTION BUTTONS

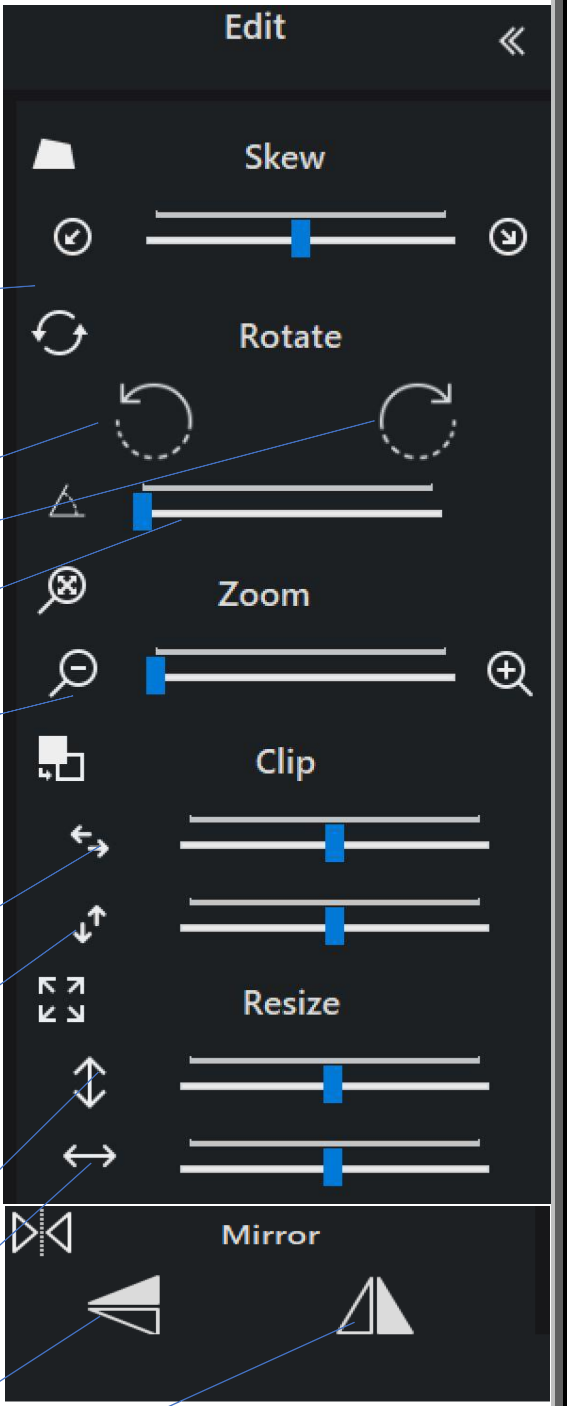


←Using to uploade Image and Convert image to gray Scale





→ EDIT SECTION BUTTONS



This Section uses if image has been scanned or photographed crookedly.

→ Skew

It's the button used to straighten an image.

→ Rotate

- It's the button using to Rotate Left image.
- It's the button using to Rotate Right image.
- It's the button using to Rotate image with angle value.

→Zoom

It's the button using to Zooming image in and out depending on trackbar values.

→Clip

- Translation (shifting) of an object's location Horizontal depending on trackbar values.
- Translation (shifting) of an object's location Vertical depending on trackbar values.

→Resize

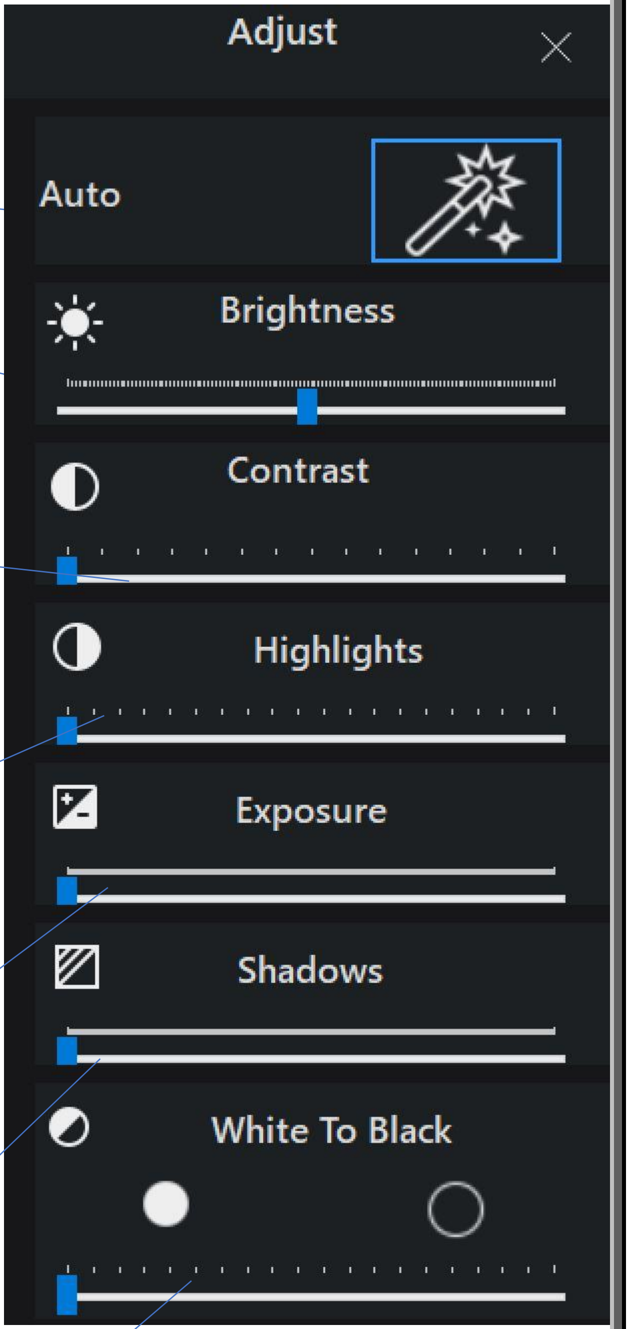
- Edit Horizontal Scaling depending on trackbar values.
- Edit Vertical Scaling depending on trackbar values.

→Mirror

- Flip(reflecting) image Horizontal.
- Flip(reflecting) image Vertical.

→ ADJUST SECTION BUTTONS

This Section is used to enhance (brightness and darkness) image.



→Auto

- Using Histogram Equalization techniques to auto adjust on image.

→Brightness

- Using gamma correction to increase or decrease the brightness of image depending on gamma values.

→Contrast

- Using gamma correction techniques to Contrast image depending on trackbar values.
  Note:
  *gamma>1 & trackbar value start at 1.

→Highlights

- Using gamma correction techniques to Highlights image depending on trackbar values.
  Note:
  *gamma<1 & trackbar value start at 1.

→Exposure

- Using Natural Log techniques to brights darker zones in image (Like contrast but using bright) depending on trackbar values.

→Shadows

- Using Mean Noise techniques to make shadows on image depending on trackbar values.

→White To Black

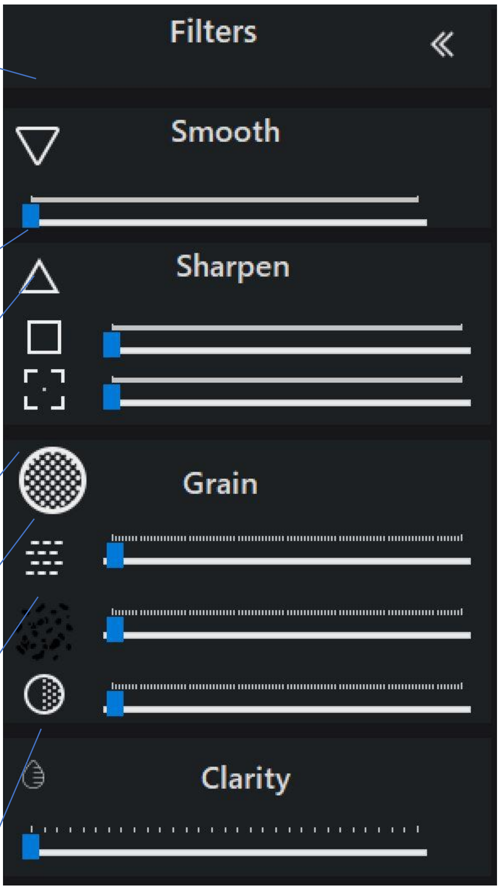
- Using Log inverse techniques to decrease the brightness of image depending on trackbar values.

→ FILTERS SECTION BUTTONS

This Section is used to enhance (Edges) and add or remove noise from image.

→Smooth

- Using Traditional filter techniques to Remove spark noise (reduce irrelevant detail in image) on image.
  Note:
  *using blur function in OpenCV.
  *Kernel Size depending on trackbar values.



→Sharpen

- Using Laplacian filter techniques to Sharp edges (increase detail in image) on image in Frequency domain.
- Using Laplacian filter techniques to Sharp edges (increase detail in image) on image in Spatial domain. Using Laplacian filter techniques to Sharp edges (increase detail in image) on image in Spatial domain.
  Note:
  *Sharpen degree depending on trackbar values.

→Grain

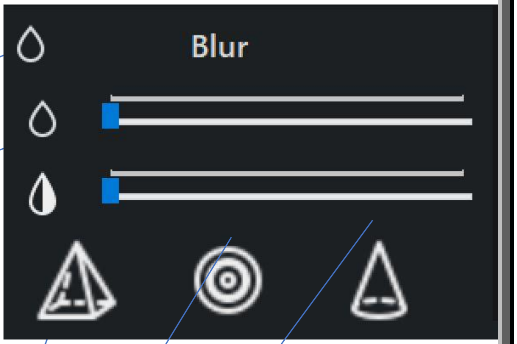
- Using to add salt noise on image.
- Using to add pepper noise on image.
- Using to add salt pepper noise on image.
  Note:

  *Noise degree depending on trackbar values.

→Clarity

- Using Median filter techniques to Salt and pepper noise removal while preserving edges
  Note:
  *Clarity degree depending on trackbar values.

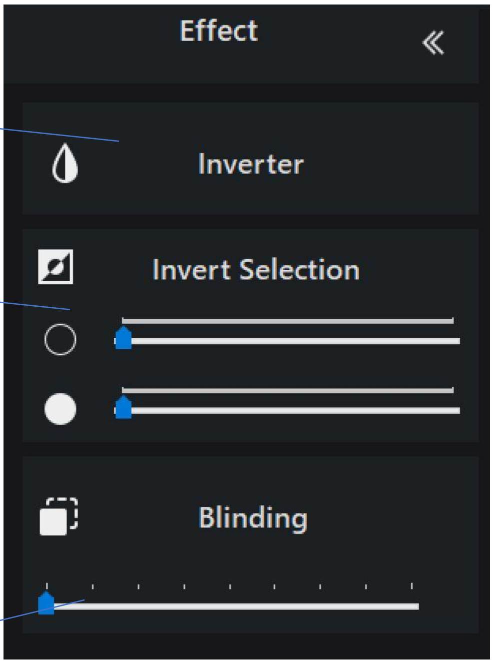
→Blur



- Using Gaussian filter techniques to Remove spark noise (reduce irrelevant detail in image) on image in Frequency domain.
- Using Gaussian filter techniques to Remove spark noise (reduce irrelevant detail in image) on image in Spatial domain.
  Note:
  *Blurred degree depending on trackbar values.
- Using Pyramidal filter techniques to Remove spark noise (reduce irrelevant detail in image) (Kernel value like Pyramidal else edge not blurred)
- Using Circle filter techniques to Remove spark noise (reduce irrelevant detail in image) (Kernel value like Circle else edge not blurred)
- Using Cone filter techniques to Remove spark noise (reduce irrelevant detail in image) (Kernel value like Cone else edge not blurred)

→ EFFECT SECTION BUTTONS

This Section is used to enhance convert color images or blind to image.



→Inverter

- Using Negative techniques to invert image.

→Invert Selection

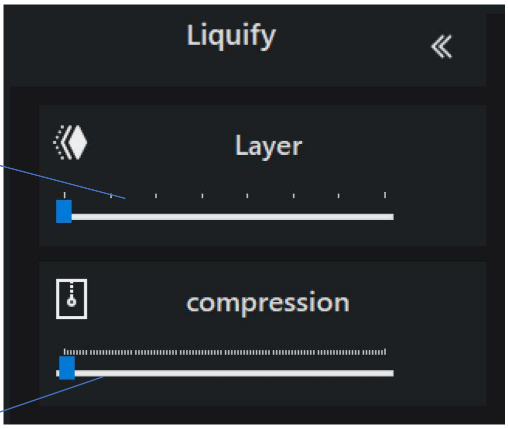
- Using Gray level slicing techniques to invert image.
  By select black degree depending on trackbar values
  By select white degree depending on trackbar values

→Blinding

- Using Blinding techniques to be maxed to images
  And toggles between us using trackbar values.

→ LIQUIFY SECTION BUTTONS

This Section is used to compress images size and quality.



→Layer

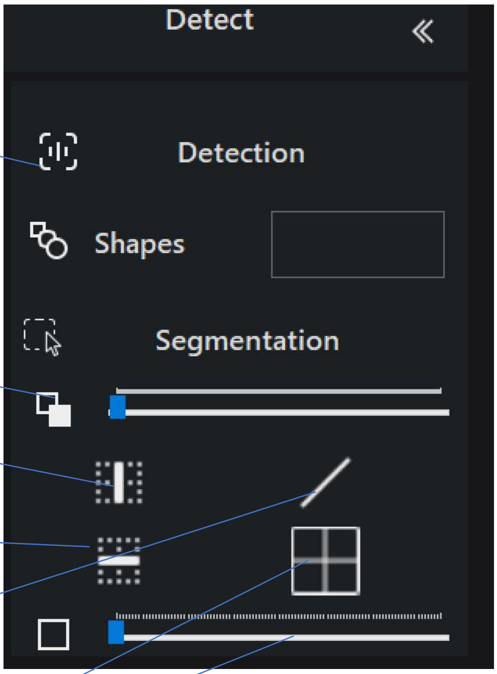
- Using Bit Plane Slicing techniques to Convert Binary Scale and remove layers from image.
  Note:
  *Layer Controller depending on trackbar values.

→Compression

- Using Huffman Encoding techniques to Compressed quality of image or increased Controller by trackbar values.

→ DETECT SECTION BUTTONS

This Section is used to Detect edges and understand shapes in images.



→Detection

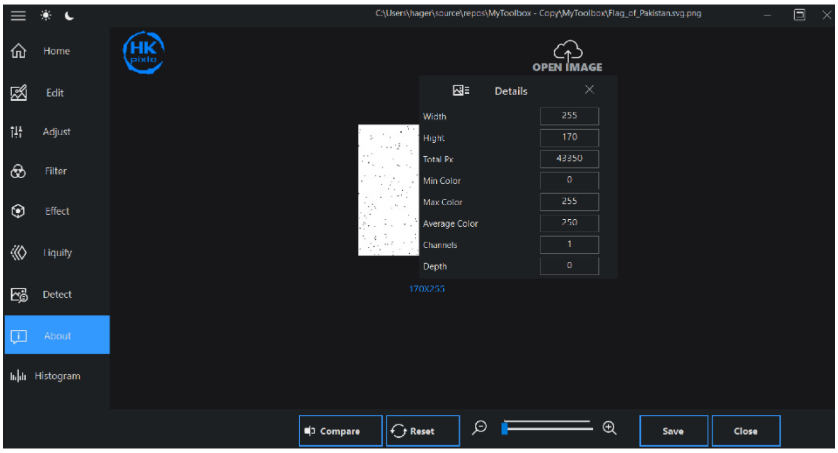
- Using Gaussian and canny filters to prepare image and make probability to shapes to be understood What is the shape into image.

→Segmentation

- Using Threshold techniques to detect edges in image.
- Using Sobel Vertical techniques to detect Vertical edges in image.
- Using Sobel Horizontal techniques to detect horizontal edges in image.
- Using Sobel Diagonal techniques to detect Diagonal edges in image.
- Using Laplacian techniques to detect edges in image.
- Using Canny techniques to detect edges in image.

→ ABOUT SECTION BUTTONS

- Show details of image after enhancing.



C:\Users\hager\source\repos\MyToolbox - Copy\MyToolbox\Flag_of_Pakistan.svg.png

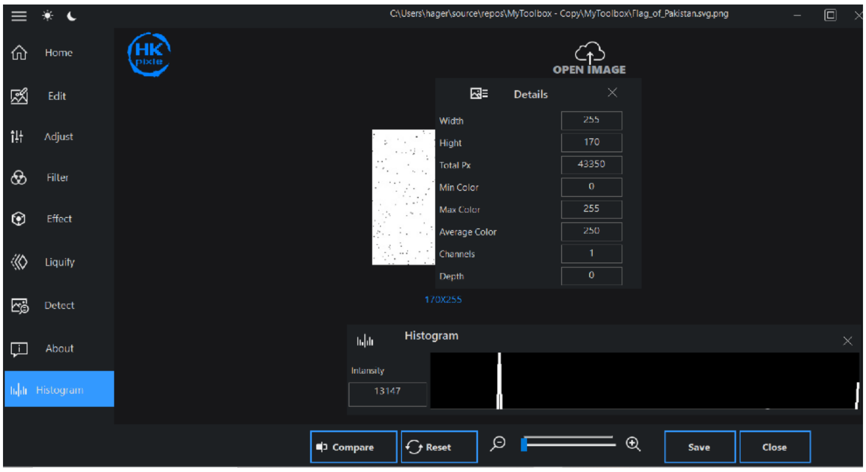
Details

Width	255
Hight	170
Total Px	43350
Min Color	0
Max Color	255
Average Color	250
Channels	1
Depth	0

170X255

→ HISTOGRAM SECTION BUTTONS

- Show histogram and max intensity in image after enhancing.



C:\Users\hager\source\repos\MyToolbox - Copy\MyToolbox\Flag_of_Pakistan.svg.png

Details

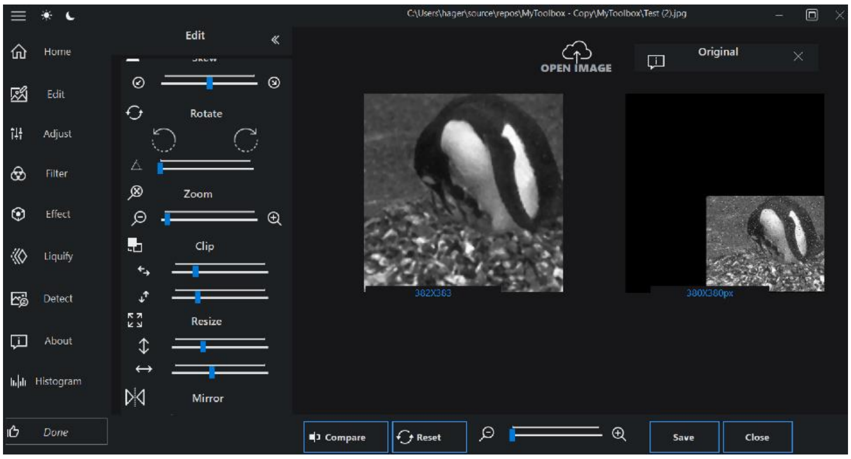
Width	255
Hight	170
Total Px	43350
Min Color	0
Max Color	255
Average Color	250
Channels	1
Depth	0

170X255

Histogram

Intensity: 13147

- Using Toolbox For enhance the follow image:

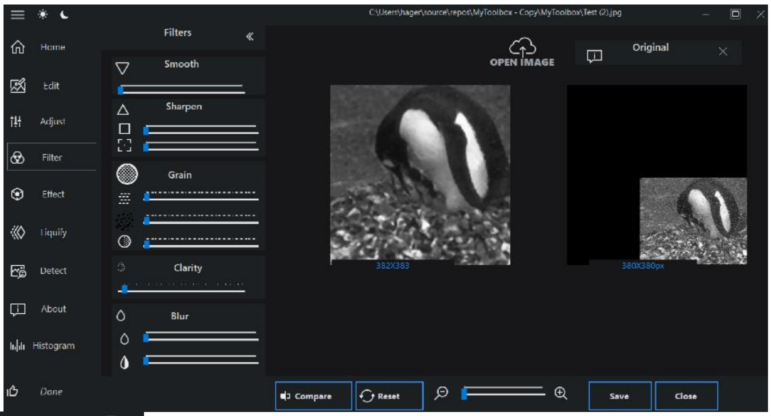


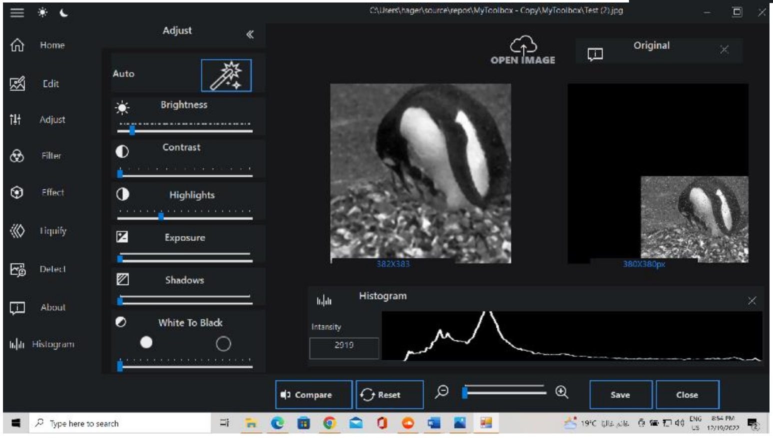
1.Using Clip to Shifting image on X, Y to Put image into center.

2.Zoom in image.

3.Resize image.

4.Using Clarity to remove noise

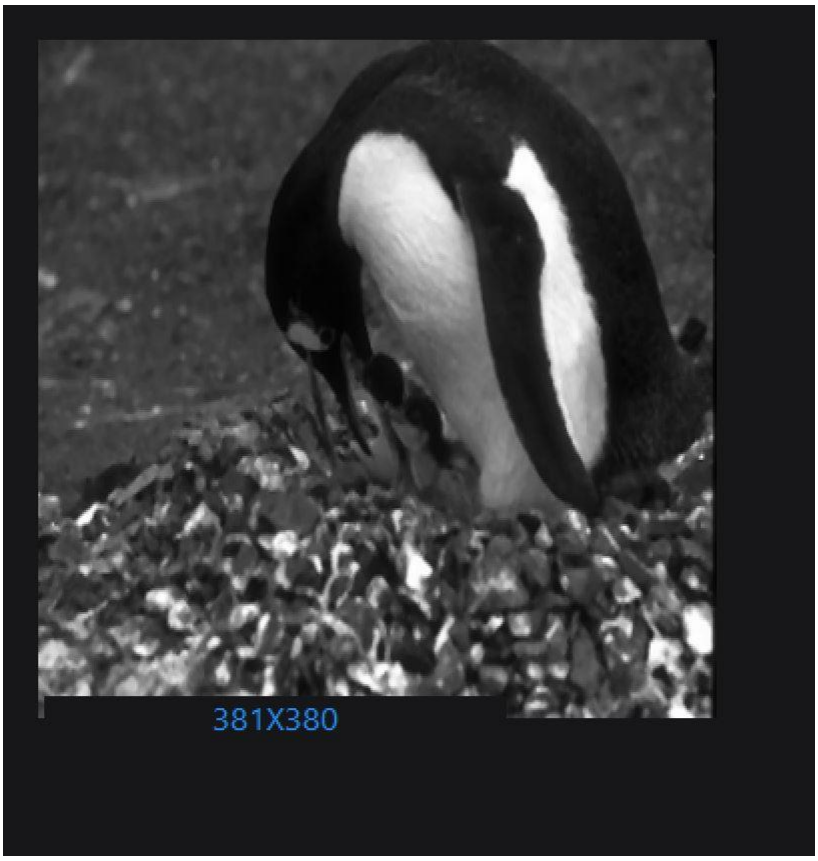


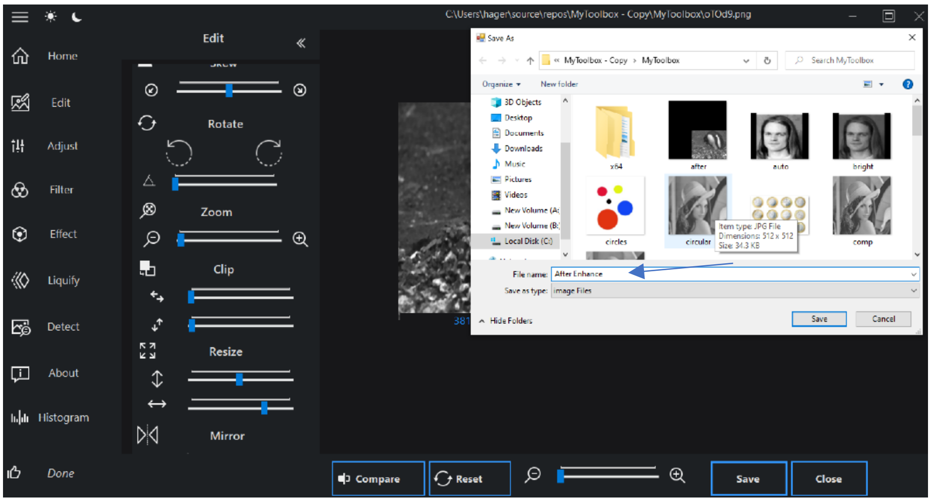


5.Increase brightness

6. Increase highlights

7.Save image Finale output



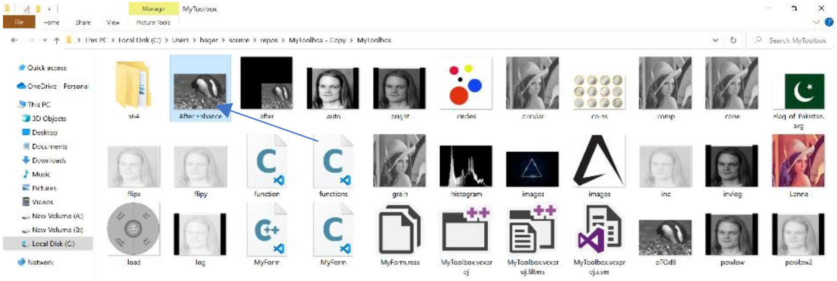

C:\Users\hager\source\repos\MyToolbox - Copy\MyToolbox\oTOd9.png

Save As

« MyToolbox - Copy › MyToolbox

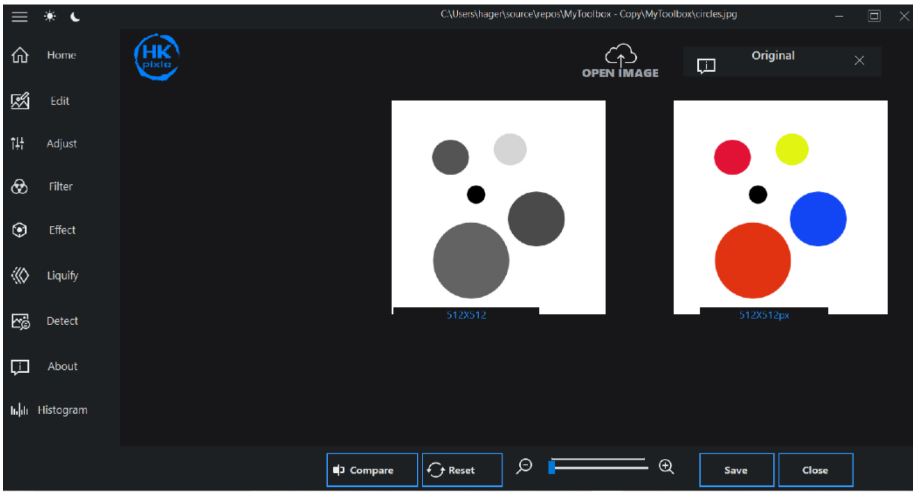
File name: After Enhance

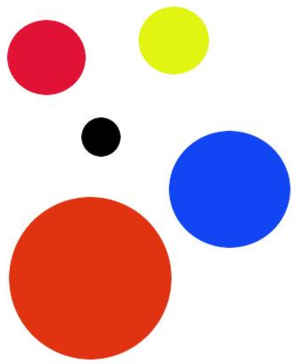
Save as type: image Files


This PC › Local Disk (C:) › Users › hager › source › repos › MyToolbox - Copy › MyToolbox

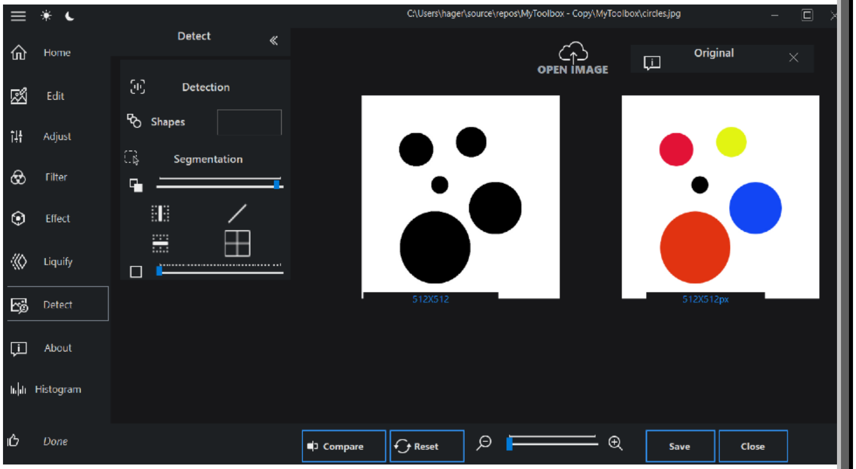
- Using Toolbox for Detection the follow image:

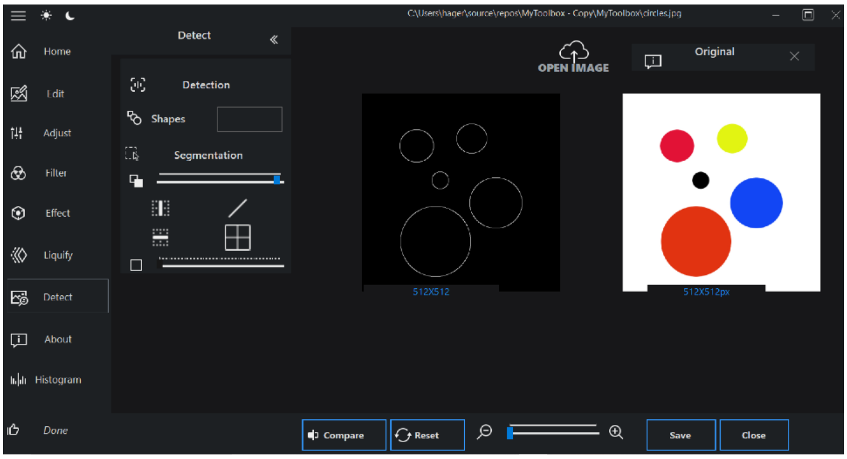
1.Open image.





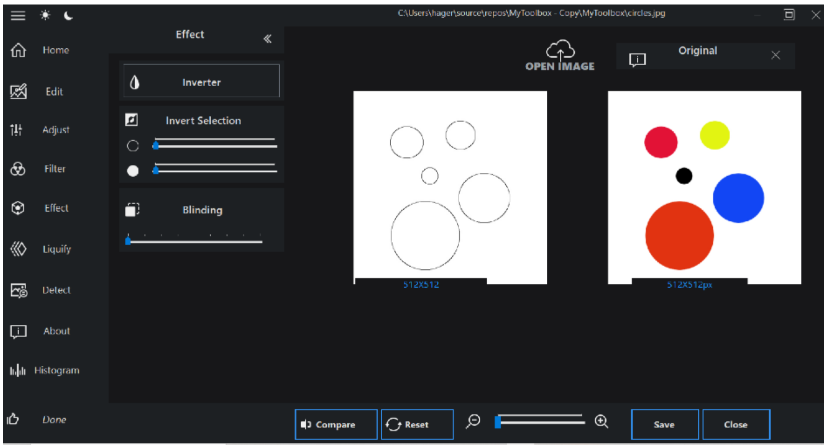
2.Apply Threshold on image to detect edge.

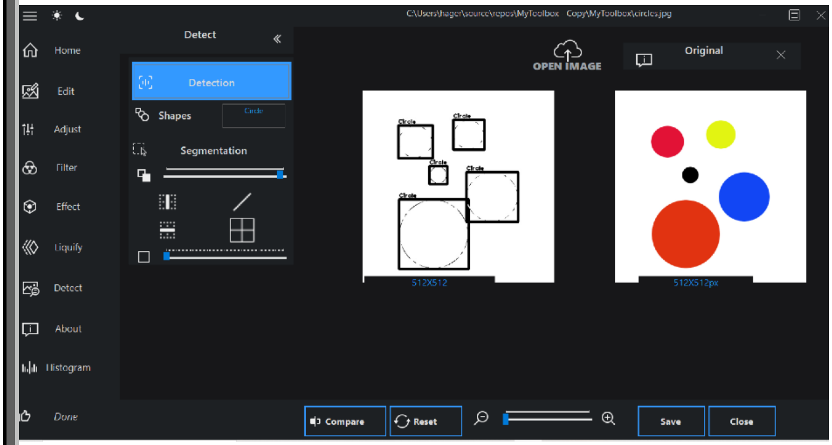




3.Apply Canny Filter on image to sharped edges.

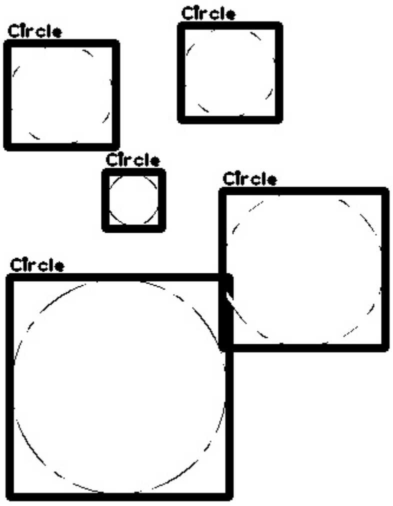
4.Apply negative on image to Show edges only.





5.Apply Detection on image to Understand Shapes into image.

Finale output


Circle
Circle
Circle
Circle
Circle