

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
THÀNH PHỐ HỒ CHÍ MINH



BÀI TẬP 0: CÀI ĐẶT OPENCV

I. THÔNG TIN SINH VIÊN

Họ và tên: **TRẦN NHẬT HUY**

Mssv: **1612272**

Email: nhathuy13598@gmail.com

Sdt: **0354 878 677**

II. BẢNG BÁO CÁO CÔNG VIỆC

STT	Yêu cầu	Ghi chú
1	Sinh viên cài đặt OpenCV, chạy thử đoạn code mẫu	100%
2	Chuyển sang chương trình chạy command line như đoạn code bên dưới. Cho biết công dụng của các dòng code (1), (2), (3), (4)	100%
3	Tìm hiểu các kiểu lưu trữ dữ liệu cơ bản của OpenCV	100%
4	Tìm hiểu công dụng chính của các module: core, improc, highgui.	100%
5	<p>Viết chương trình đơn giản cho phép người dùng mở một hình ảnh (màu hoặc trắng đen) và thực hiện các thao tác sau (lưu ý chương trình được chạy bằng command line, sinh viên cần tuân thủ quy định về tham số):</p> <p>Biến đổi ảnh màu thành trắng đen hoặc ngược lại:</p> <p><tenchuongtrinh> <duongdantaptinanh> <malenh></p> <p>Thay đổi độ sáng của ảnh: <tenchuongtrinh> <duongdantaptinanh> <malenh> <thamso> -</p> <p>Thay đổi độ tương phản của ảnh: <tenchuongtrinh> <duongdantaptinanh> <malenh> <thamso></p>	100%

III. HƯỚNG DẪN CHI TIẾT

1. Cài đặt OpenCV và chạy code mẫu

B1: Tạo Empty Project

B2: Cấu hình project như hình

1612272_BT00 Property Pages

? X

Configuration: All Configurations Platform: All Platforms Configuration Manager...

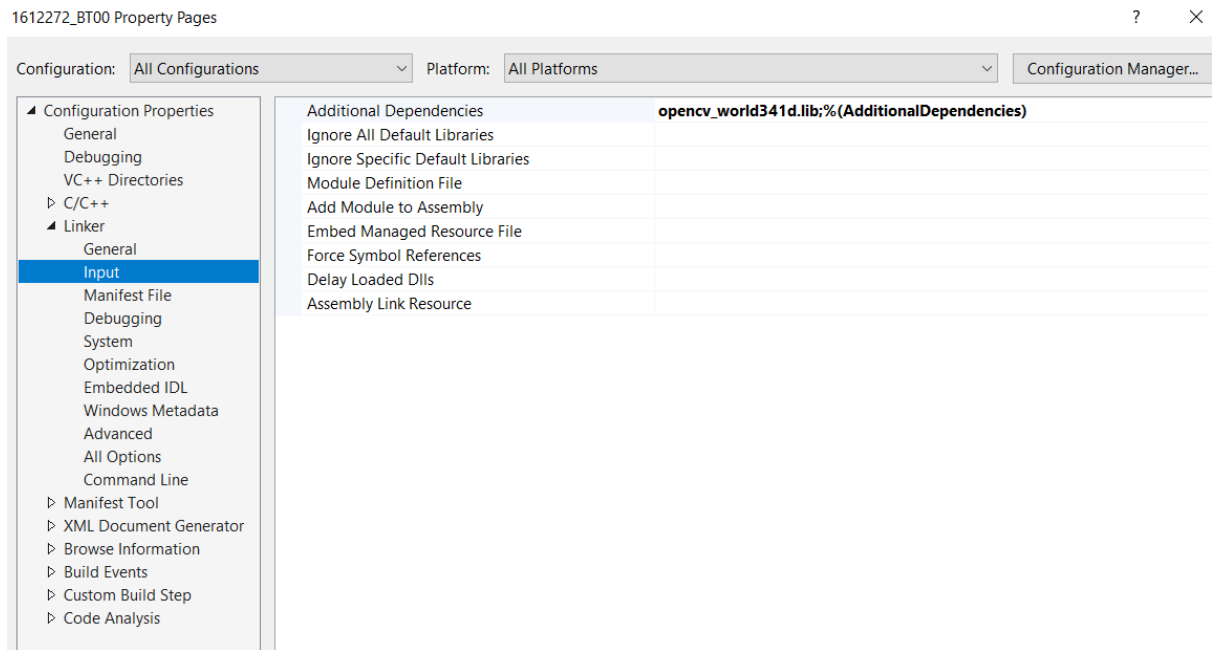
<ul style="list-style-type: none"> Configuration Properties <ul style="list-style-type: none"> General Debugging VC++ Directories C/C++ <ul style="list-style-type: none"> General Optimization Preprocessor Code Generation Language Precompiled Headers Output Files Browse Information Advanced All Options Command Line Linker Manifest Tool XML Document Generator Browse Information Build Events Custom Build Step Code Analysis 	<table border="1"> <tr> <td>Additional Include Directories</td> <td>D:\OpenCV3\opencv\build\include;%(AdditionalIncludeDirectories)</td> </tr> <tr> <td>Additional #using Directories</td> <td></td> </tr> <tr> <td>Debug Information Format</td> <td><different options></td> </tr> <tr> <td>Support Just My Code Debugging</td> <td><different options></td> </tr> <tr> <td>Common Language RunTime Support</td> <td></td> </tr> <tr> <td>Consume Windows Runtime Extension</td> <td></td> </tr> <tr> <td>Suppress Startup Banner</td> <td>Yes (/nologo)</td> </tr> <tr> <td>Warning Level</td> <td>Level3 (/W3)</td> </tr> <tr> <td>Treat Warnings As Errors</td> <td>No (/WX-)</td> </tr> <tr> <td>Warning Version</td> <td></td> </tr> <tr> <td>Diagnostics Format</td> <td>Classic (/diagnostics:classic)</td> </tr> <tr> <td>SDL checks</td> <td>Yes (/sdl)</td> </tr> <tr> <td>Multi-processor Compilation</td> <td></td> </tr> </table>	Additional Include Directories	D:\OpenCV3\opencv\build\include;%(AdditionalIncludeDirectories)	Additional #using Directories		Debug Information Format	<different options>	Support Just My Code Debugging	<different options>	Common Language RunTime Support		Consume Windows Runtime Extension		Suppress Startup Banner	Yes (/nologo)	Warning Level	Level3 (/W3)	Treat Warnings As Errors	No (/WX-)	Warning Version		Diagnostics Format	Classic (/diagnostics:classic)	SDL checks	Yes (/sdl)	Multi-processor Compilation	
Additional Include Directories	D:\OpenCV3\opencv\build\include;%(AdditionalIncludeDirectories)																										
Additional #using Directories																											
Debug Information Format	<different options>																										
Support Just My Code Debugging	<different options>																										
Common Language RunTime Support																											
Consume Windows Runtime Extension																											
Suppress Startup Banner	Yes (/nologo)																										
Warning Level	Level3 (/W3)																										
Treat Warnings As Errors	No (/WX-)																										
Warning Version																											
Diagnostics Format	Classic (/diagnostics:classic)																										
SDL checks	Yes (/sdl)																										
Multi-processor Compilation																											

1612272_BT00 Property Pages

? X

Configuration: All Configurations Platform: All Platforms Configuration Manager...

<ul style="list-style-type: none"> Configuration Properties <ul style="list-style-type: none"> General Debugging VC++ Directories C/C++ Linker <ul style="list-style-type: none"> General Input Manifest File Debugging System Optimization Embedded IDL Windows Metadata Advanced All Options Command Line Manifest Tool XML Document Generator Browse Information Build Events Custom Build Step Code Analysis 	<table border="1"> <tr> <td>Output File</td> <td>\$(OutDir)\$(TargetName)\$(TargetExt)</td> </tr> <tr> <td>Show Progress</td> <td>Not Set</td> </tr> <tr> <td>Version</td> <td></td> </tr> <tr> <td>Enable Incremental Linking</td> <td><different options></td> </tr> <tr> <td>Suppress Startup Banner</td> <td>Yes (/NOLOGO)</td> </tr> <tr> <td>Ignore Import Library</td> <td>No</td> </tr> <tr> <td>Register Output</td> <td>No</td> </tr> <tr> <td>Per-user Redirection</td> <td>No</td> </tr> <tr> <td>Additional Library Directories</td> <td>D:\OpenCV3\opencv\build\x64\vc14\lib;%(AdditionalLibraryDirectories)</td> </tr> <tr> <td>Link Library Dependencies</td> <td>Yes</td> </tr> <tr> <td>Use Library Dependency Inputs</td> <td>No</td> </tr> <tr> <td>Link Status</td> <td></td> </tr> <tr> <td>Prevent Dll Binding</td> <td></td> </tr> <tr> <td>Treat Linker Warning As Errors</td> <td></td> </tr> <tr> <td>Force File Output</td> <td></td> </tr> <tr> <td>Create Hot Patchable Image</td> <td></td> </tr> <tr> <td>Specify Section Attributes</td> <td></td> </tr> </table>	Output File	\$(OutDir)\$(TargetName)\$(TargetExt)	Show Progress	Not Set	Version		Enable Incremental Linking	<different options>	Suppress Startup Banner	Yes (/NOLOGO)	Ignore Import Library	No	Register Output	No	Per-user Redirection	No	Additional Library Directories	D:\OpenCV3\opencv\build\x64\vc14\lib;%(AdditionalLibraryDirectories)	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	\$(OutDir)\$(TargetName)\$(TargetExt)																																		
Show Progress	Not Set																																		
Version																																			
Enable Incremental Linking	<different options>																																		
Suppress Startup Banner	Yes (/NOLOGO)																																		
Ignore Import Library	No																																		
Register Output	No																																		
Per-user Redirection	No																																		
Additional Library Directories	D:\OpenCV3\opencv\build\x64\vc14\lib;%(AdditionalLibraryDirectories)																																		
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																																			

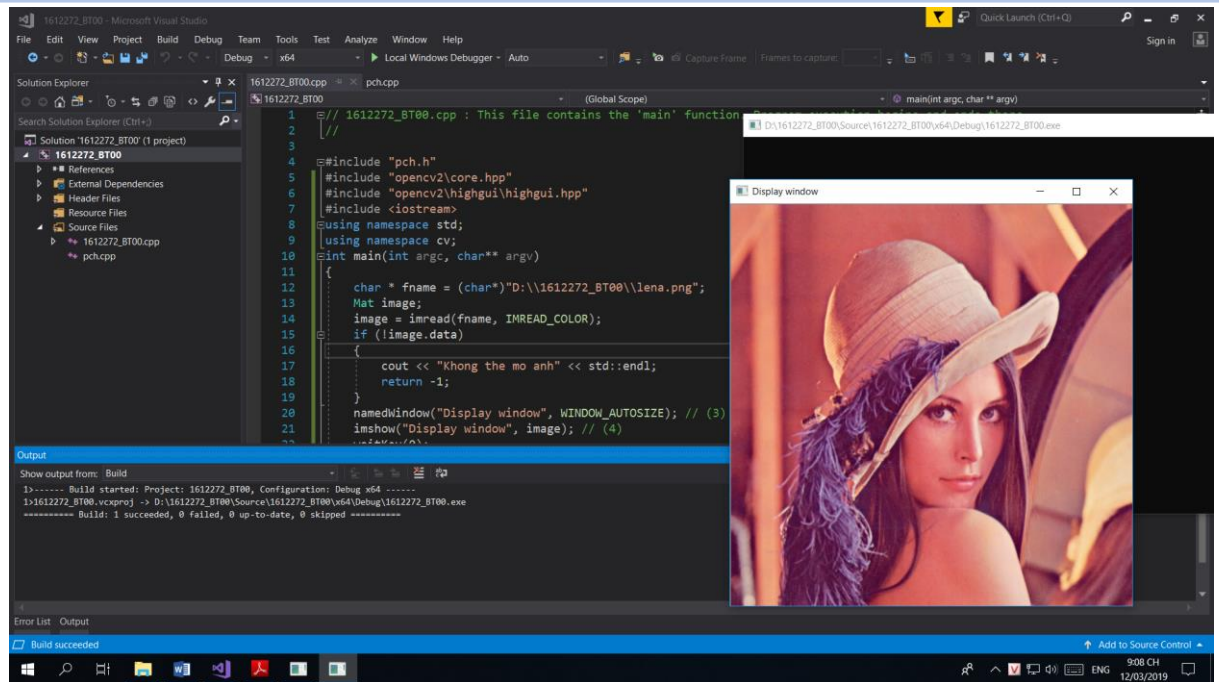


B3: Chạy thử code mẫu (Sửa lại địa chỉ ảnh cho phù hợp)

// 1612272_BT00.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

```
#include "pch.h"
#include "opencv2\core.hpp"
#include "opencv2\highgui\highgui.hpp"
#include <iostream>
using namespace std;
using namespace cv;
int main(int argc, char** argv)
{
    char * fname = (char*)"D:\\1612272_BT00\\lena.png";
    Mat image;
    image = imread(fname, IMREAD_COLOR);
    if (!image.data)
    {
        cout << "Khong the mo anh" << std::endl;
        return -1;
    }
    namedWindow("Display window", WINDOW_AUTOSIZE); // (3)
    imshow("Display window", image); // (4)
    waitKey(0);
    return 0;
}
```



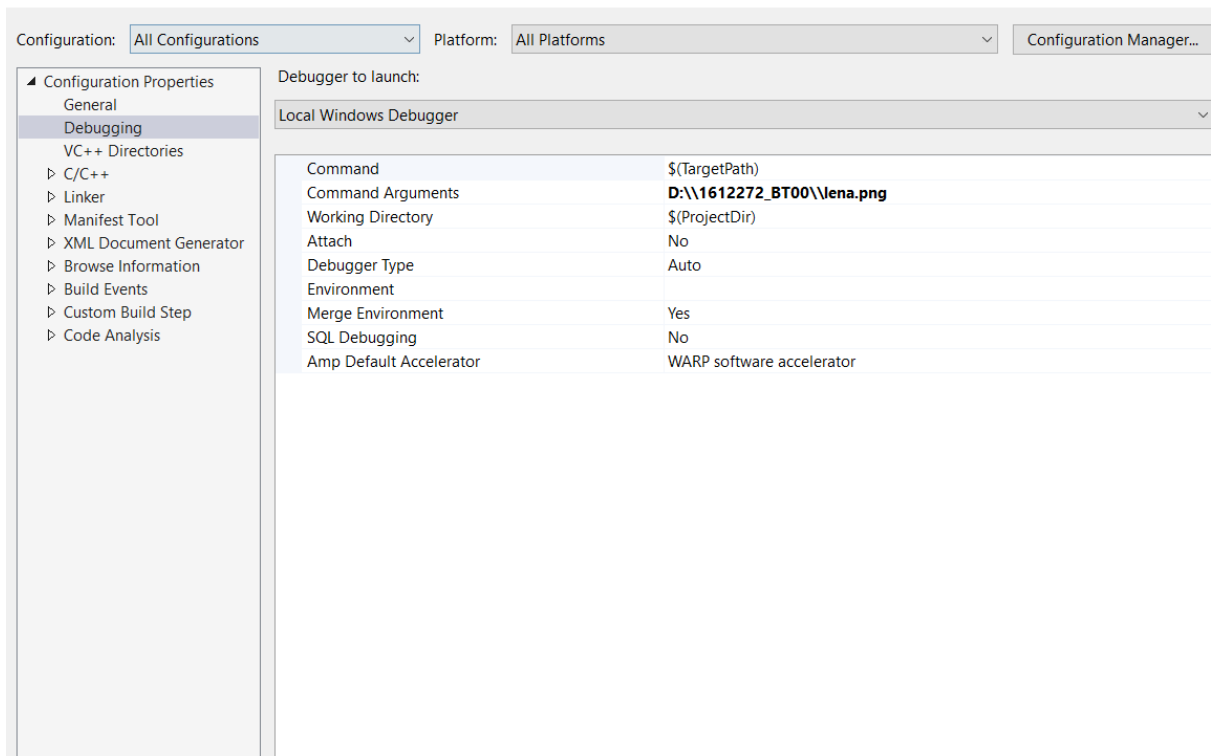
2. Command line và công dụng của các dòng code

B1: Chỉnh sửa lại code để chạy command line

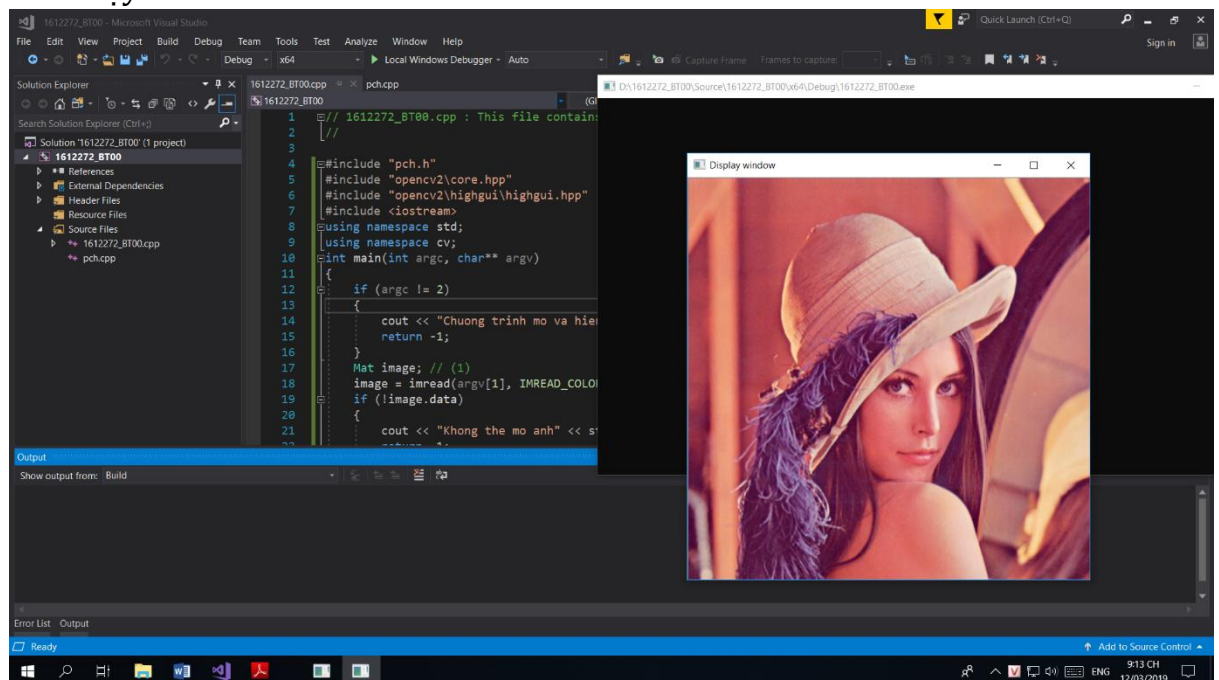
// 1612272_BT00.cpp : This file contains the 'main' function. Program execution begins and ends there.
//

```
#include "pch.h"
#include "opencv2\core.hpp"
#include "opencv2\highgui\highgui.hpp"
#include <iostream>
using namespace std;
using namespace cv;
int main(int argc, char** argv)
{
    if (argc != 2)
    {
        cout << "Chương trình mo va hien thi anh" << endl;
        return -1;
    }
    Mat image; // (1)
    image = imread(argv[1], IMREAD_COLOR); // (2)
    if (!image.data)
    {
        cout << "Khong the mo anh" << std::endl;
        return -1;
    }
    namedWindow("Display window", WINDOW_AUTOSIZE); // (3)
    imshow("Display window", image); // (4)
    waitKey(0);
    return 0;
}
```

B2: Cấu hình để chạy thử command line



B3: Chạy thử



Dòng code	Ý nghĩa
<code>Mat image; // (1)</code>	Khởi tạo đối tượng Mat có tên là image để lưu trữ ảnh
<code>image = imread(argv[1], IMREAD_COLOR); // (2)</code>	Đọc hình ảnh với địa chỉ chứa trong biến argv[1] Hằng số

	<i>IMREAD_COLOR</i> nói rằng sẽ đọc lên ảnh màu RGB
<code>namedWindow("Display window", WINDOW_AUTOSIZE); // (3)</code>	Tạo cửa sổ để hiển thị hình với tên là <i>Display window</i> Hằng số <i>WINDOW_AUTOSIZE</i> là để kích thước cửa sổ tự động vừa với hình cần hiển thị
<code>imshow("Display window", image); // (4)</code>	Hiển thị hình ảnh trên cửa sổ vừa tạo

3. Kiểu dữ liệu cơ bản của OpenCV

a. Cấu trúc Point_

```
template<typename _Tp> class CV_EXPORTS Point_
{
public:
    typedef _Tp value_type;

    // various constructors
    Point_();
    Point_(_Tp _x, _Tp _y);
    Point_(const Point_& pt);
    Point_(const CvPoint& pt);
    Point_(const CvPoint2D32f& pt);
    Point_(const Size_<_Tp>& sz);
    Point_(const Vec_<_Tp, 2>& v);

    Point_& operator = (const Point_& pt);
    /// conversion to another data type
    template<typename _Tp2> operator Point_<_Tp2>() const;

    /// conversion to the old-style C structures
    operator CvPoint() const;
    operator CvPoint2D32f() const;
    operator Vec_<_Tp, 2>() const;

    /// dot product
    _Tp dot(const Point_& pt) const;
    /// dot product computed in double-precision arithmetics
    double ddot(const Point_& pt) const;
    /// cross-product
    double cross(const Point_& pt) const;
    /// checks whether the point is inside the specified rectangle
    bool inside(const Rect_<_Tp>& r) const;

    _Tp x, y; ///< the point coordinates
};
```

Đây là Template Class của điểm 2D được định nghĩa bằng tọa độ của x và y . Một đối tượng của class này có thể hoán đổi với cấu trúc của C là *CvPoint* và *CvPoint2D32f*.

Chúng ta có thể chuyển tọa độ số chấm động (floating-point) sang tọa độ nguyên bằng phép toán ép kiểu.

Đối tượng của class *Point_* có thể thực hiện các phép toán $+$, $-$, $*$, $+=$, $-=$, $*=$, $==$, $!=$

Để thuận tiện thì OpenCV có tạo ra các Type Alias sau:

```
typedef Point_<int> Point2i;
typedef Point2i Point;
typedef Point_<float> Point2f;
typedef Point_<double> Point2d;
```

Các phương thức của lớp *Point_* gồm có:

- Tích vô hướng:

```
///! dot product
_Tp dot(const Point_& pt) const;
///! dot product computed in double-precision arithmetics
double ddot(const Point_& pt) const;
```
- Tích hữu hướng:

```
///! cross-product
double cross(const Point_& pt) const;
```
- Kiểm tra một điểm có thuộc một hình chữ nhật:

```
///! checks whether the point is inside the specified rectangle
bool inside(const Rect_<_Tp>& r) const;
```

b. Cấu trúc *Point3_*


```

template<typename _Tp> class CV_EXPORTS Point3_
{
public:
    typedef _Tp value_type;

    // various constructors
    Point3_();
    Point3_(_Tp _x, _Tp _y, _Tp _z);
    Point3_(const Point3_& pt);
    explicit Point3_(const Point<_Tp>& pt);
    Point3_(const CvPoint3D32f& pt);
    Point3_(const Vec<_Tp, 3>& v);

    Point3_& operator = (const Point3_& pt);
    /// conversion to another data type
    template<typename _Tp2> operator Point3_<_Tp2>() const;
    /// conversion to the old-style CvPoint...
    operator CvPoint3D32f() const;
    /// conversion to cv::Vec<>
    operator Vec<_Tp, 3>() const;

    /// dot product
    _Tp dot(const Point3_& pt) const;
    /// dot product computed in double-precision arithmetics
    double ddot(const Point3_& pt) const;
    /// cross product of the 2 3D points
    Point3_ cross(const Point3_& pt) const;

    _Tp x, y, z; ///< the point coordinates
};

```

Đây là Template Class của điểm 2D được định nghĩa bằng tạo độ của x, y và z . Một đối tượng của class này có thể hoán đổi với cấu trúc của C là *CvPoint3D32f*.

Lớp *Point3_* tương tự như lớp *Point_*

c. Cấu trúc Mat

```

class CV_EXPORTS Mat
{
public:
    // ... a lot of methods ...
    ...

    /*! includes several bit-fields:
        - the magic signature
        - continuity flag
        - depth
        - number of channels
    */
    int flags;
    /*! the array dimensionality, >= 2
    int dims;
    /*! the number of rows and columns or (-1, -1) when the array has more than 2 dimensions
    int rows, cols;
    /*! pointer to the data
    uchar* data;

    /*! pointer to the reference counter;
    // when array points to user-allocated data, the pointer is NULL
    int* refcount;

    // other members
    ...
};

```

Lớp Mat biểu diễn ma trận số n chiều với 1 kênh hoặc nhiều kênh. Nó có thể dùng để lưu trữ vector thực và ảo, ma trận thực và ảo, ảnh màu và xám, ...

Một số attribute của Mat :

- Số chiều của Mat (lớn hơn hoặc bằng 2) :

```

/*! the array dimensionality, >= 2
int dims;

```

- Số lượng dòng và cột (là (-1, -1) nếu số chiều lớn hơn 2) :

```

/*! the number of rows and columns or (-1, -1) when the array has more than 2 dimensions
int rows, cols;

```

```

/*! pointer to the data
uchar* data;

```

- Con trỏ đến dữ liệu :

Một số cách tạo đối tượng Mat :

- Sử dụng phương thức `create(nrows, ncols, type)` :

```

M.create(100,60,CV_8UC(15));

```

- Sử dụng phương thức khởi tạo mặc định

```

Mat(nrows, ncols, type[, fillValue]) ;

```

```

// make a 7x7 complex matrix filled with 1+3j.
Mat M(7,7,CV_32FC2,Scalar(1,3));

```

- Sử dụng phương thức khởi tạo sao chép hoặc gán. Tuy nhiên, chúng chỉ sao chép địa chỉ và tăng số lượng biến reference. Nếu muốn thực hiện deep copy thì dùng phương thức `Mat::clone()`

4. Các module

a. Core

Module Core functionality cung cấp đầy đủ các cốt lõi của OpenCV gồm một số modules nhỏ sau:

- Các cấu dữ liệu cơ bản (Có thể xem phần ở trên)
- Các cấu trúc và phép toán C (*C structures and operations*)
- Các phép toán trên mảng (*Operations on arrays*)
- Và một số các modules khác

Core functionality

Modules

Basic structures
C structures and operations
Operations on arrays
XML/YAML Persistence
Clustering
Utility and system functions and macros
OpenGL interoperability
Intel IPP Asynchronous C/C++ Converters
Optimization Algorithms
DirectX interoperability
Eigen support
OpenCL support
Intel VA-API/OpenCL (CL-VA) interoperability
Hardware Acceleration Layer

b. Improc

Module này cung cấp các hàm xử lý ảnh trong OpenCV:

- Làm trơn ảnh (*Smoothing Images*)
Làm trơn ảnh hay còn gọi là làm mờ, là phương pháp thường được sử dụng trong xử lý ảnh. Các phương pháp được cung cấp bởi OpenCV gồm có: *medianBlur*, *GaussianBlur*
- Phân đoạn ảnh (*Basic Thresholding Operations*)
Phân đoạn ảnh được dùng để tách các vùng của ảnh mà chứa object mà ta muốn phân tích.

OpenCV cung cấp các phương pháp phân đoạn ảnh như sau: ***Threshold Binary, Threshold Binary Inverted, Truncate, Threshold to Zero, Threshold to Zero Inverted.***

```
threshold( src_gray, dst, threshold_value, max_BINARY_value, threshold_type );
```

- Xác định biên cạnh theo Sobel (***Sobel Derivatives***)

OpenCV cung cấp hàm xác định biên cạnh bằng phương pháp Sobel

```
/// Gradient X
//Scharr( src_gray, grad_x, ddepth, 1, 0, scale, delta, BORDER_DEFAULT );
Sobel( src_gray, grad_x, ddepth, 1, 0, 3, scale, delta, BORDER_DEFAULT );
convertScaleAbs( grad_x, abs_grad_x );
```

```
/// Gradient Y
//Scharr( src_gray, grad_y, ddepth, 0, 1, scale, delta, BORDER_DEFAULT );
Sobel( src_gray, grad_y, ddepth, 0, 1, 3, scale, delta, BORDER_DEFAULT );
convertScaleAbs( grad_y, abs_grad_y );
```

```
/// Total Gradient (approximate)
addWeighted( abs_grad_x, 0.5, abs_grad_y, 0.5, 0, grad );
```

- Toán tử Laplace (***Laplace Operator***)

OpenCV cũng cung cấp hàm để xác định biên cạnh bằng phương pháp Laplace

```
Laplacian( src_gray, dst, ddepth, kernel_size, scale, delta, BORDER_DEFAULT );
```

- Xác định biên cạnh bằng Canny (***Canny Edge Detector***)

OpenCV cũng cung cấp hàm xác định biên cạnh bằng phương pháp Canny

```
Canny( detected_edges, detected_edges, lowThreshold, lowThreshold*ratio, kernel_size );
```

- Và OpenCV còn cung cấp nhiều hàm xử lý ảnh khác

c. Highgui

Module này cung cấp các giao diện người dùng được build sẵn có trong OpenCV

Với module này chúng ta có thể:

- Thêm Trackbar vào ứng dụng của chúng ta

```
createTrackbar( TrackbarName, "Linear Blend", &alpha_slider, alpha_slider_max, on_trackbar );
```

- Trackbar có tên là TrackbarName
- Trackbar được gắn vào cửa sổ có tên là Linear Blend
- Giá trị Trackbar từ 0 đến alpha_slider_max
- Giá trị của Trackbar lưu trong alpha_slider
- Khi người dùng di chuyển Trackbar thì sẽ gọi hàm on_trackbar
- Mở/đọc video streams và kiểm tra độ tương đồng giữa 2 hình ảnh (thường được sử dụng để kiểm tra phương pháp nén) bằng 2 phương pháp là PSNR và SSIM.

- Tạo video

IV. KẾT QUẢ 3 BÀI THỰC NGHIỆM

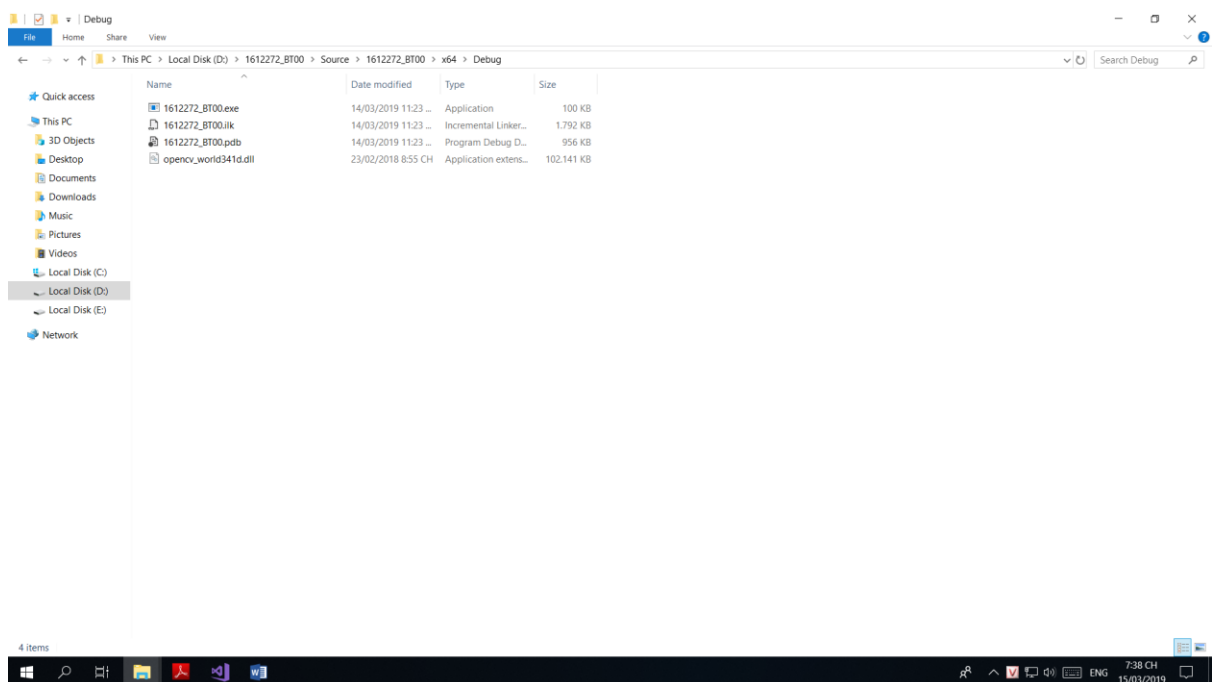
1. Thông tin các hàm

STT	TÊN HÀM	CHỨC NĂNG	GIẢI THÍCH THÊM
1	Chỉ cần đưa vào link file ảnh	Mở ảnh	
2	RGB2Grey	Chuyển ảnh màu thành ảnh xám	
3	Grey2RGB	Chuyển ảnh xám thành ảnh màu	
4	Brightness [Tham số]	Chỉnh độ sáng của ảnh	Tham số: là độ sáng tối đa có thể tăng
5	Contrast [Tham số]	Chỉnh độ tương phản	Tham số: là độ tương phản tối đa có thể tăng

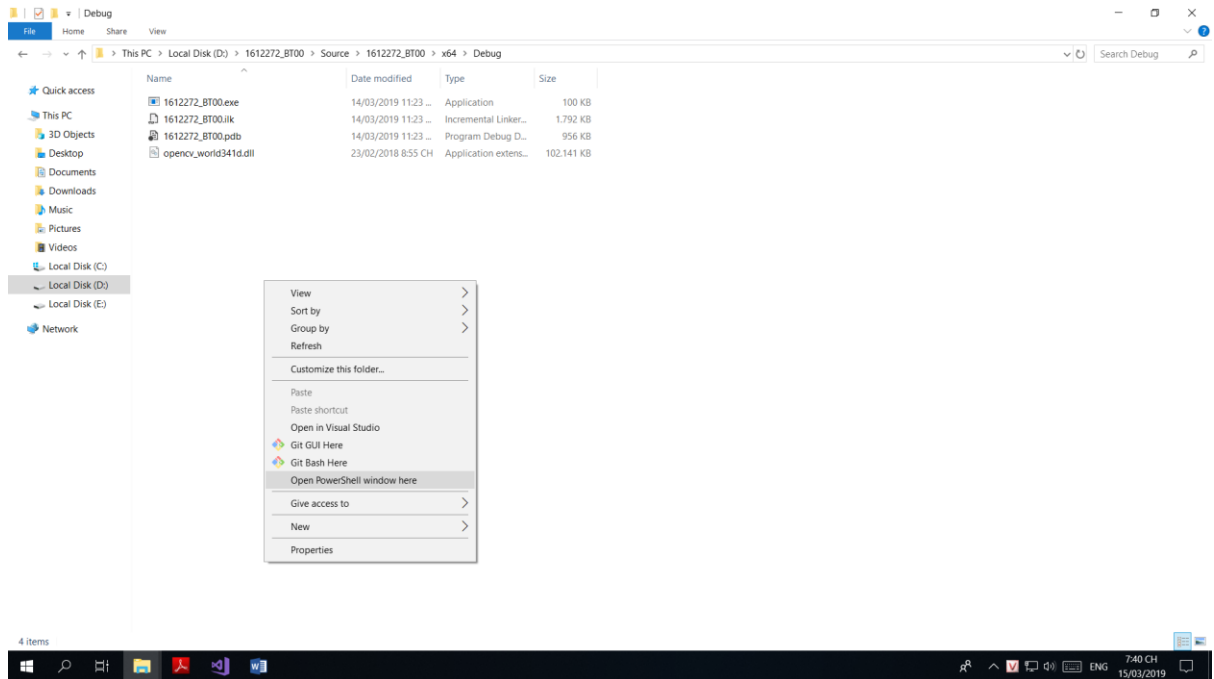
2. Hướng dẫn sử dụng

Cách để chạy chương trình:

B1: Vào thư mục chứa file .exe



B2 : Bấm Shift + Chuột phải → Chọn Open PowerShell window here

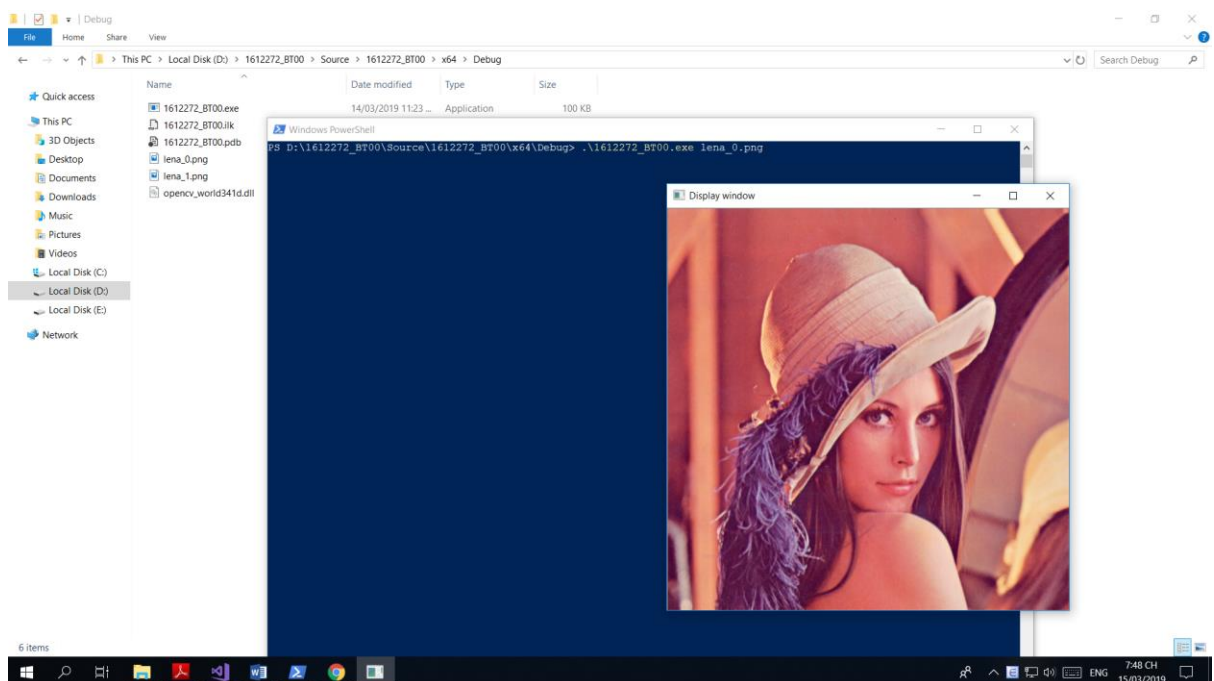


B3 : Gõ câu lệnh theo format sau <.\Tên chương trình> <Link file ảnh> <Mã lệnh> [<Tham số>] (tùy vào mã lệnh mà có tham số)

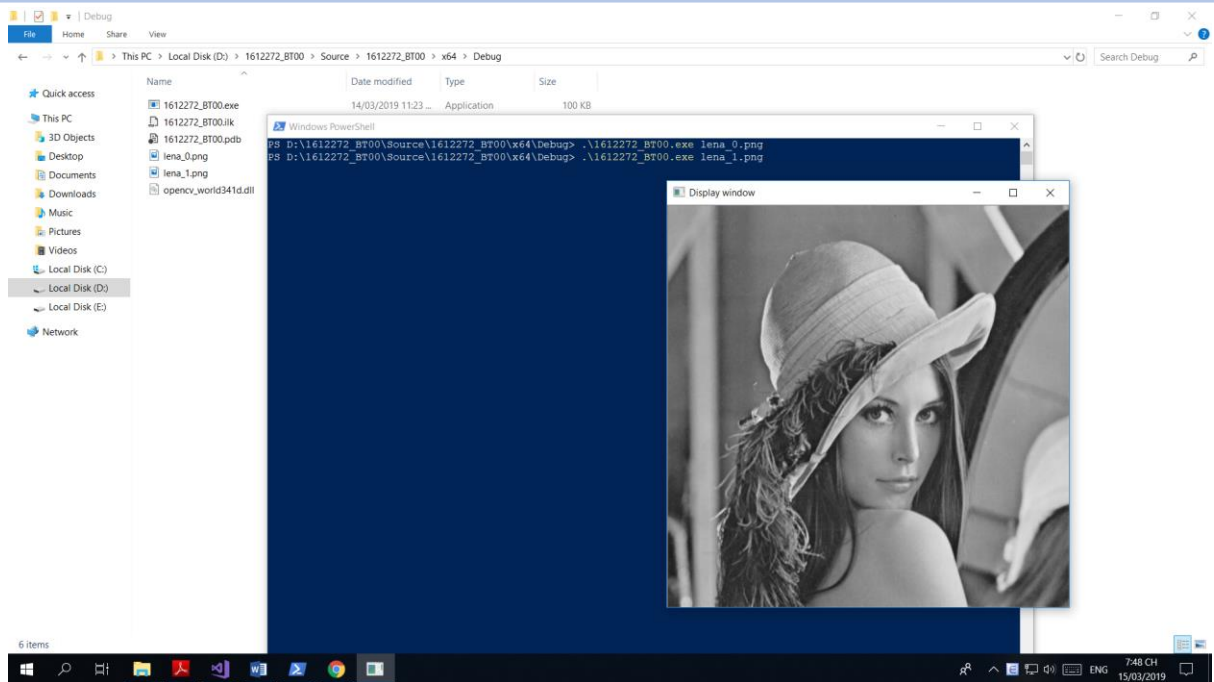
3. Kết quả chạy

a. Mở ảnh

```
.\1612272_BT00.exe lena_0.png
```

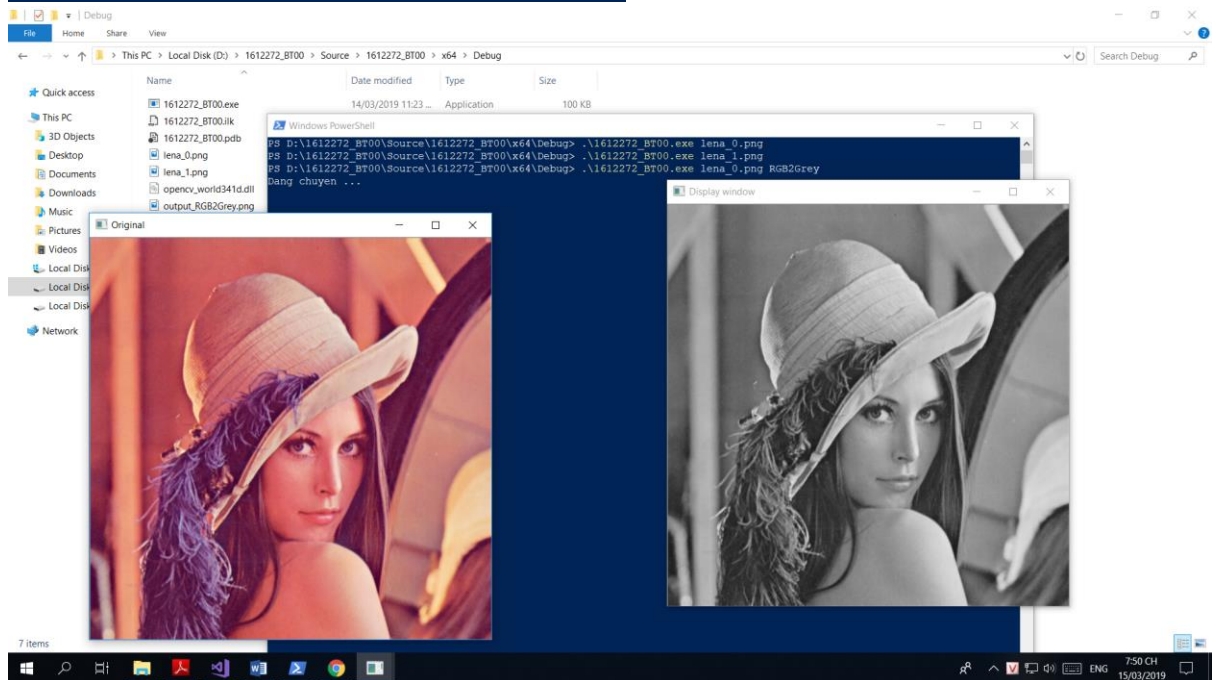


```
.\1612272_BT00.exe lena_1.png
```

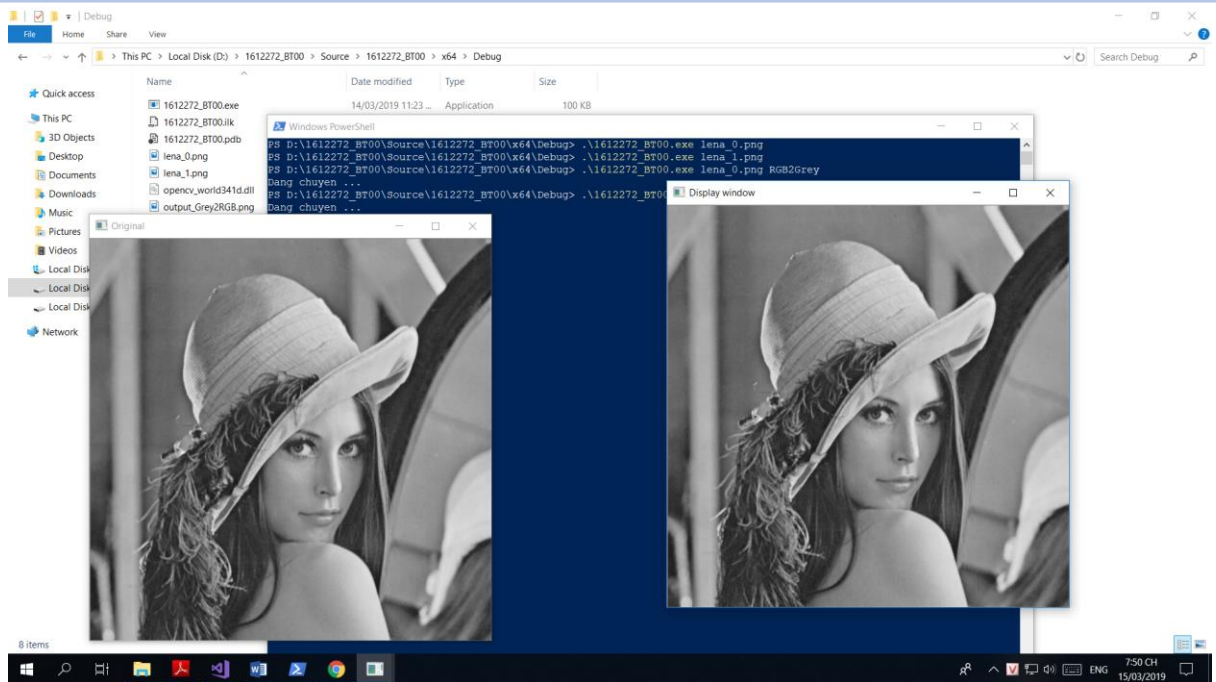


b. Chuyển ảnh màu thành trắng đen và ngược lại

`.\1612272_BT00.exe lena_0.png RGB2Grey`

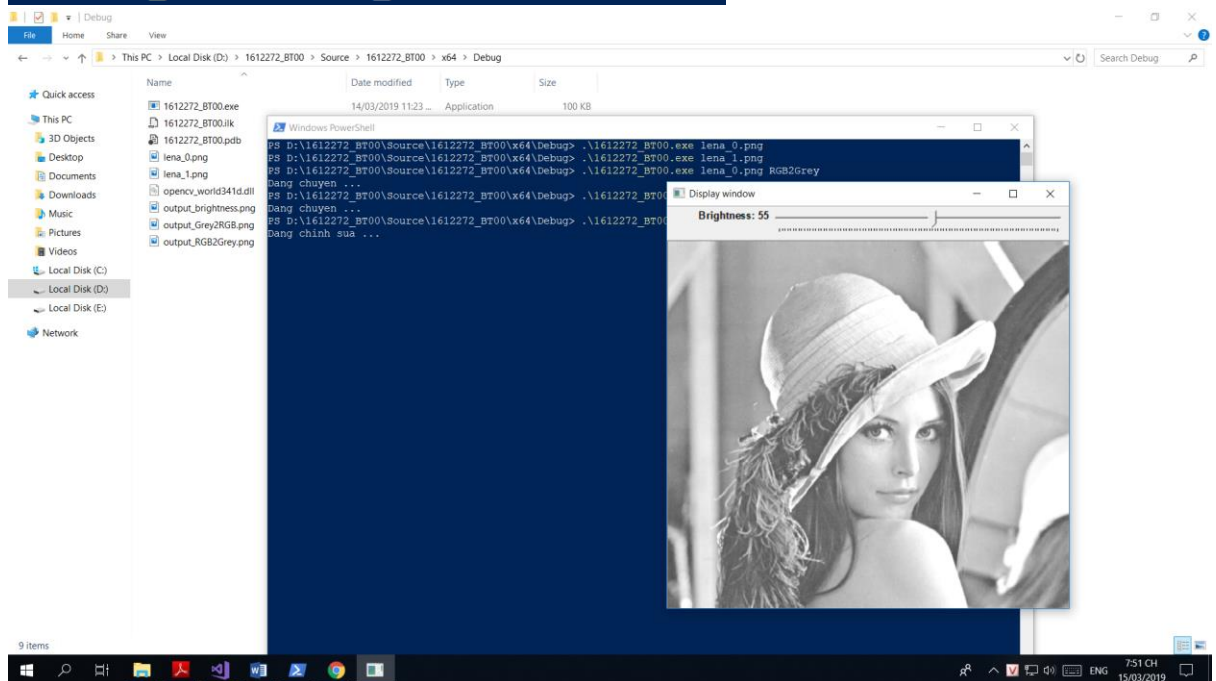


`.\1612272_BT00.exe lena_1.png Grey2RGB`



c. Thay đổi độ sáng

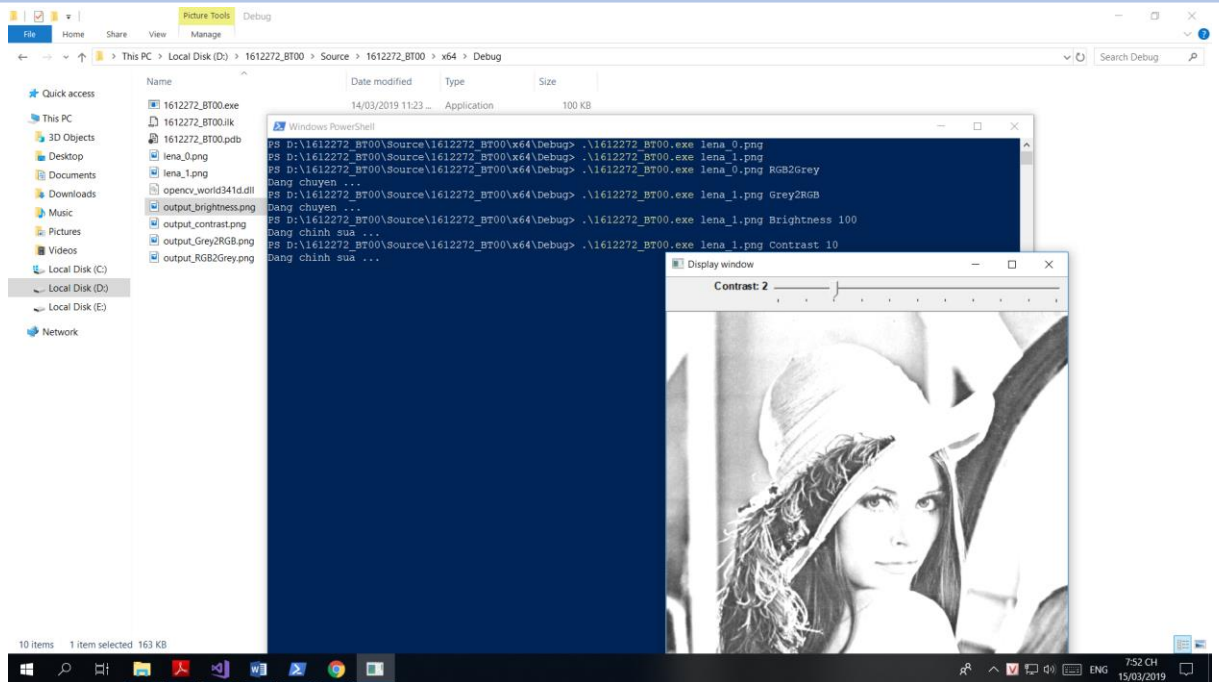
`.\1612272_BT00.exe Lena_1.png Brightness 100`



Kéo thanh Brightness để điều chỉnh độ sáng

d. Thay đổi độ tương phản

`.\1612272_BT00.exe Lena_1.png Contrast 10`



Kéo thanh Contrast để thay đổi độ tương phản