

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA: KỸ THUẬT ĐIỆN TỬ 2



TIÊU LUẬN MÔN HỌC

Đề Tài: **CONTOUR OPENCV**

Sinh Viên Thực Hiện:

Mai Bảo Quốc

Đỗ Đức Huy

Lê Nguyễn Vũ Hoàng Long

Dương Đình Khánh

TP. Hồ Chí Minh, Tháng 10 Năm 2022

Lời Nói Đầu:

Ngày nay ở thời đại công nghệ 4.0, mọi công nghệ đều kết nối thông qua Internet, việc học tập và làm việc để nắm bắt với xu thế hiện tại là tất yếu và cần thiết, ở mỗi sinh viên Kỹ Thuật việc chúng ta làm chủ được kiến thức và ứng dụng vào thực tiễn hầu như là điều ưu tiên hàng đầu, Cách đây 10 năm chúng ta chưa thể tin rằng các công nghệ đang có mặt ở hiện tại: Điện Thoại SmartPhone, Robot dẫn đường, Robot làm nhiệm vụ, Công nghệ giám sát qua InterFace,... và để theo kịp những Công Nghệ đó nhóm chúng em - nhóm 1 xin trình bày về một vấn đề rất thực tiễn với việc giám sát ngày nay: Chủ đề - OPENCV + TOPIC: CONTOUR. Đối tượng nguyên cứu nhóm muốn hướng đến là các vật thể sống ở trong đời sống của chúng ta, tìm hiểu hình dáng và chuyển động của vật thể. Công nghệ để sử dụng là ngôn ngữ lập trình C++. Và sau khi làm đề tài, nhóm cũng mong muốn được ứng dụng đề tài vào thực tiễn để góp phần kiến thức không chỉ ở bản thân mà còn cho đất nước, có những con người tài năng giúp Việt Nam vươn mình trở thành một đất nước Công Nghệ.

Phân Công

Phân Công:

Lê Nguyễn Vũ Hoàng Long: OverView + Introduction To OpenCV + Project 3

Mai Bảo Quốc: Định Nghĩa Contour + project 2

Đỗ Đức Huy: Project 1 + Support Bạn Khánh

Khánh: Getting To Know OpenCV + Graphcial User Interface

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. OverView

1.1.1 What is OpenCV ?

- OpenCV là viết tắt của Open Source Computer Vision library, viết bởi ngôn ngữ C, C++ chạy trong các hệ điều hành : Windows, IOS, Linux, Android, ... Lập trình trên các nền tảng ngôn ngữ lập trình như: Python, Java, Ruby, Matlab, ...
- OpenCV được sinh ra để phục vụ mục đích các tính toán hiệu quả và tập trung chủ yếu vào tạo ra các thiết bị xử lí trong thời gian thực.
- Mục tiêu của OpenCV hướng tới là tạo ra một cơ sở hạ tầng tối giản nhất giúp người dùng có thể tạo ra các phần mềm ứng dụng về vison.

1.1.2 Who uses OpenCV ?

- OpenCV được tạo ra để các kĩ sư máy tính, lập trình viên trí tuệ nhân tạo, Các công ty, người dùng có sử dụng ứng dụng về Vison.

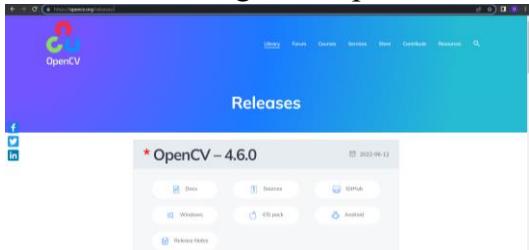
1.1.3 What is Computer Vison ?

- Thị giác máy tính là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng đưa vào máy tính.
- Được phát triển bởi Intel, trải qua nhiều phiên bản từ Beta đến Ver2.0, hiện nay mới nhất là 4.6.0, bài tiểu luận sẽ code trên nền tảng ver 4.5.5.

1.1.4 Intalling and Using OpenCV

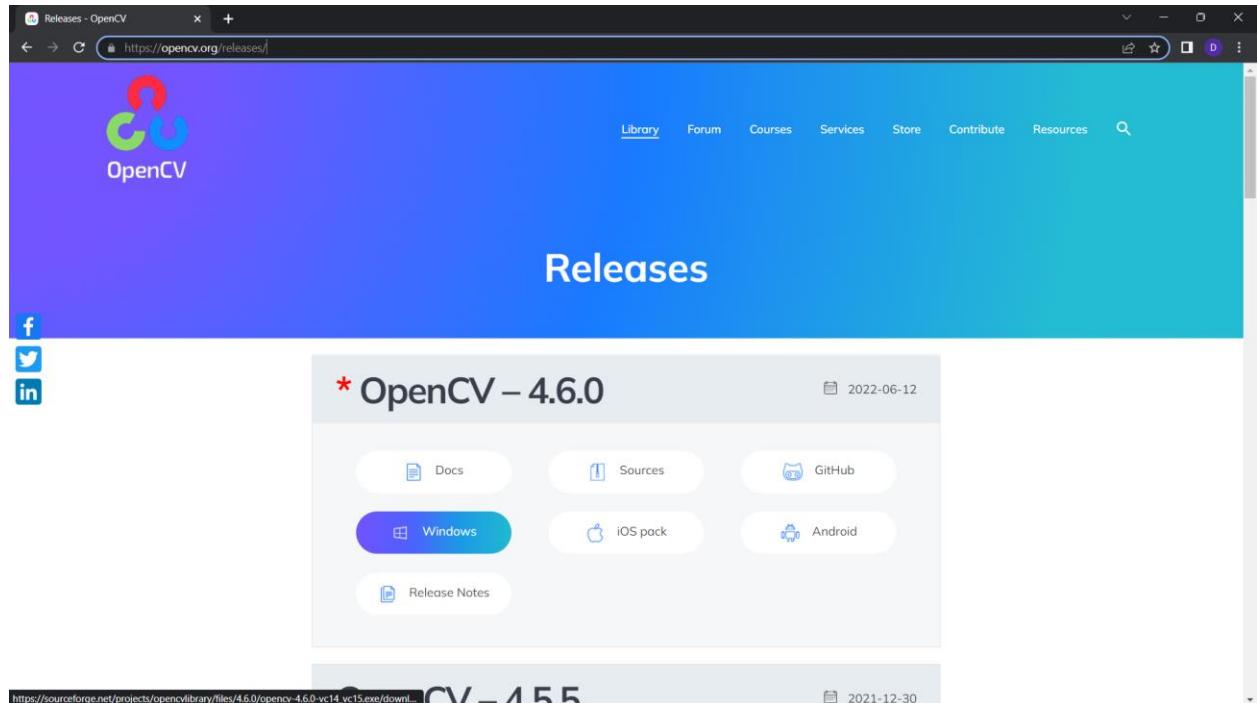
1.1.1. Tải và cài đặt OpenCV

Bước1: Vào trang web OpenCV theo đường link sau <https://opencv.org/releases/>

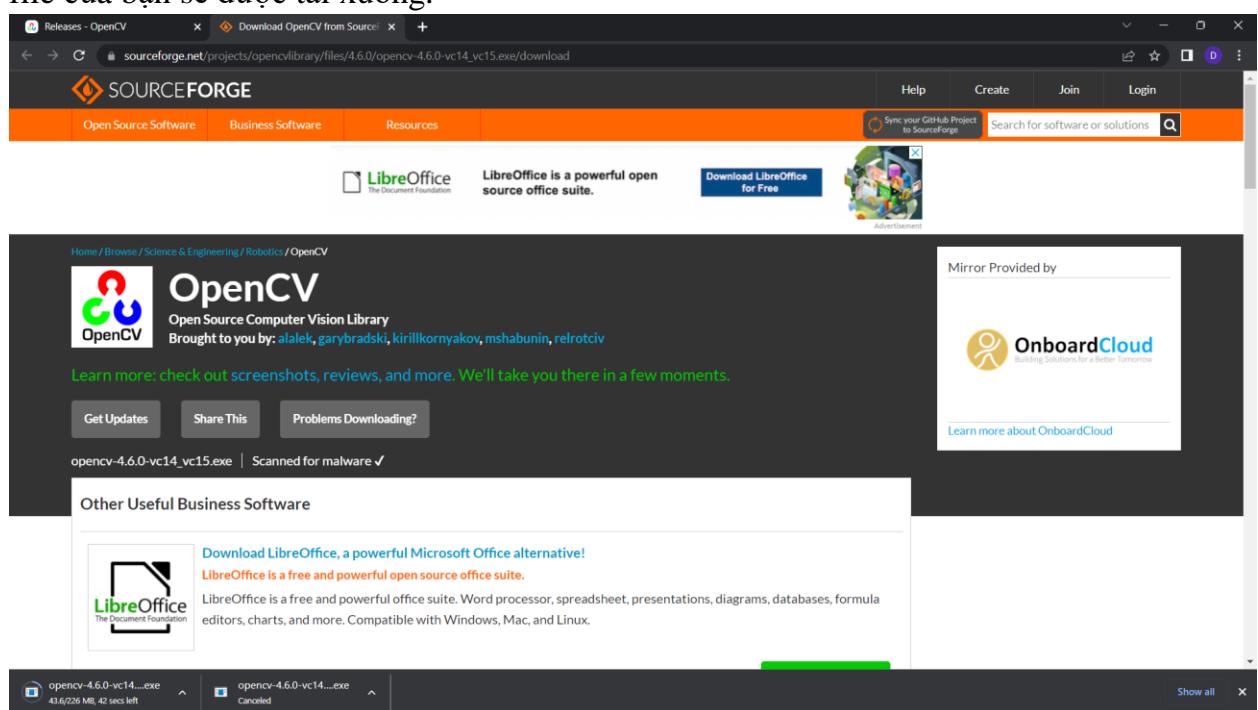


Chương 1: OverView

Bước 2: Tùy theo máy của bạn mà chọn hệ điều hành và phiên bản của OpenCV. Tại vì máy chạy hệ điều hành Windows và muốn trải nghiệm phiên bản mới nhất nên mình click vào phần Windows của phiên bản 4.6.0



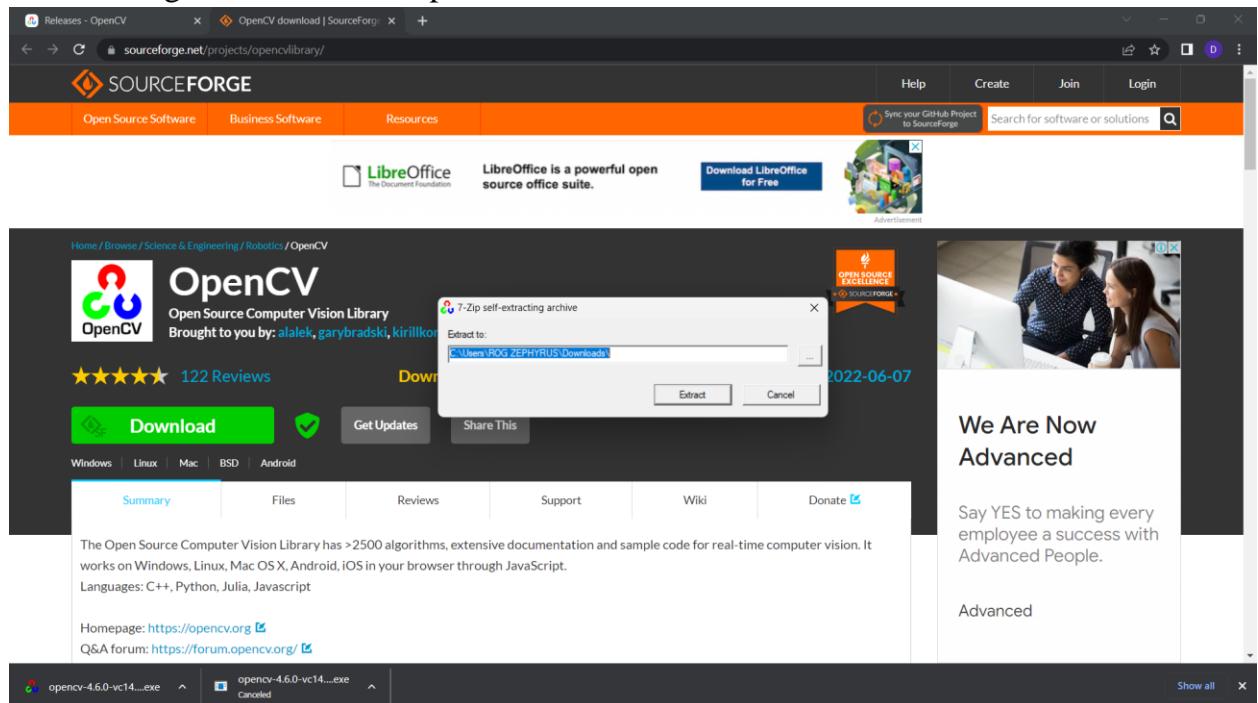
Sau khi bấm vào Windows sẽ chuyển sang trang tải khi đó bạn chỉ cần chờ đợi thì file của bạn sẽ được tải xuống.



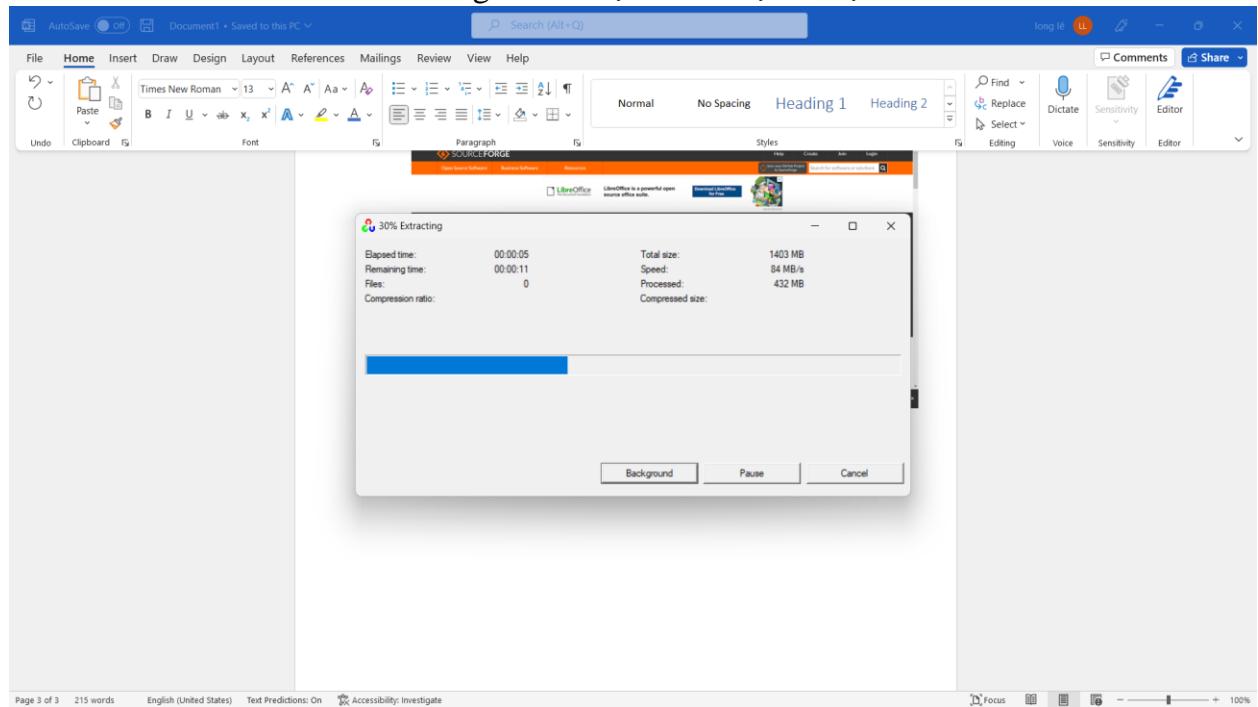
Chương 1: OverView

Bước 3: Sau khi tải xuống thành công thì chúng ta ấn vào file tải về và tiến hành cài đặt

Chọn đường dẫn để cài file tiếp theo nhấn vào nút Extract

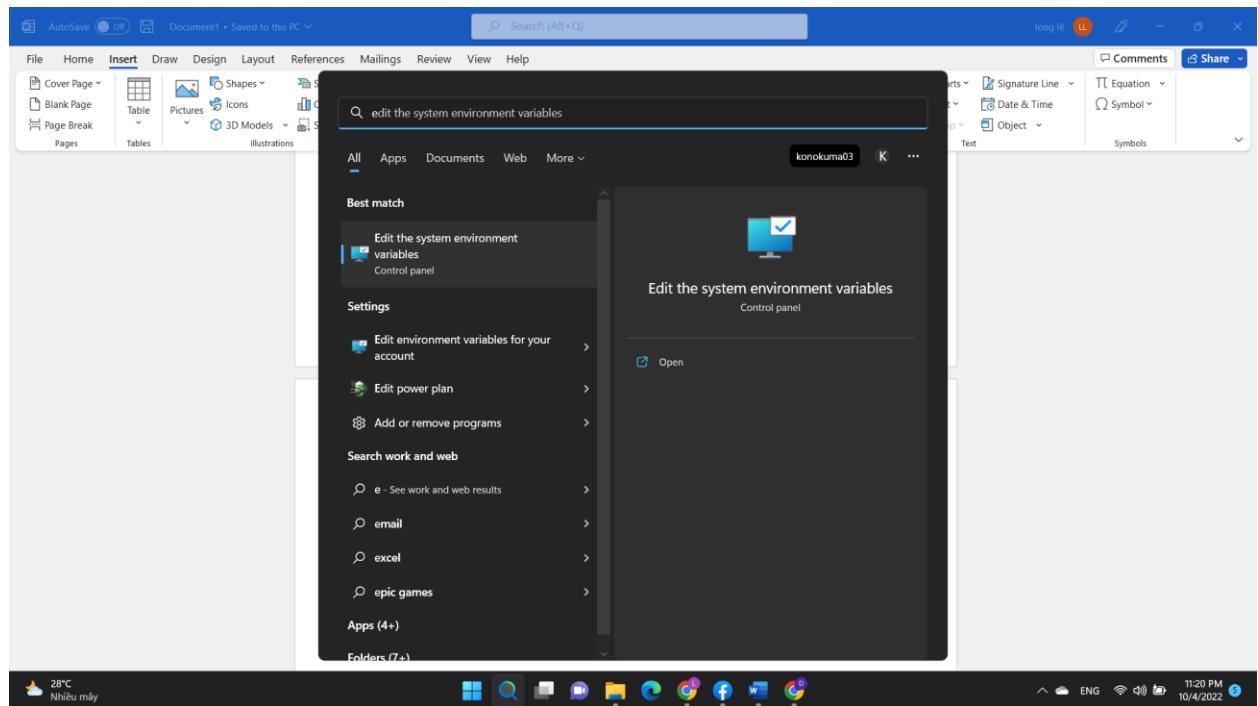


Sau khi nhấn Extract thì chúng ta chờ đợi để file tự cài đặt.

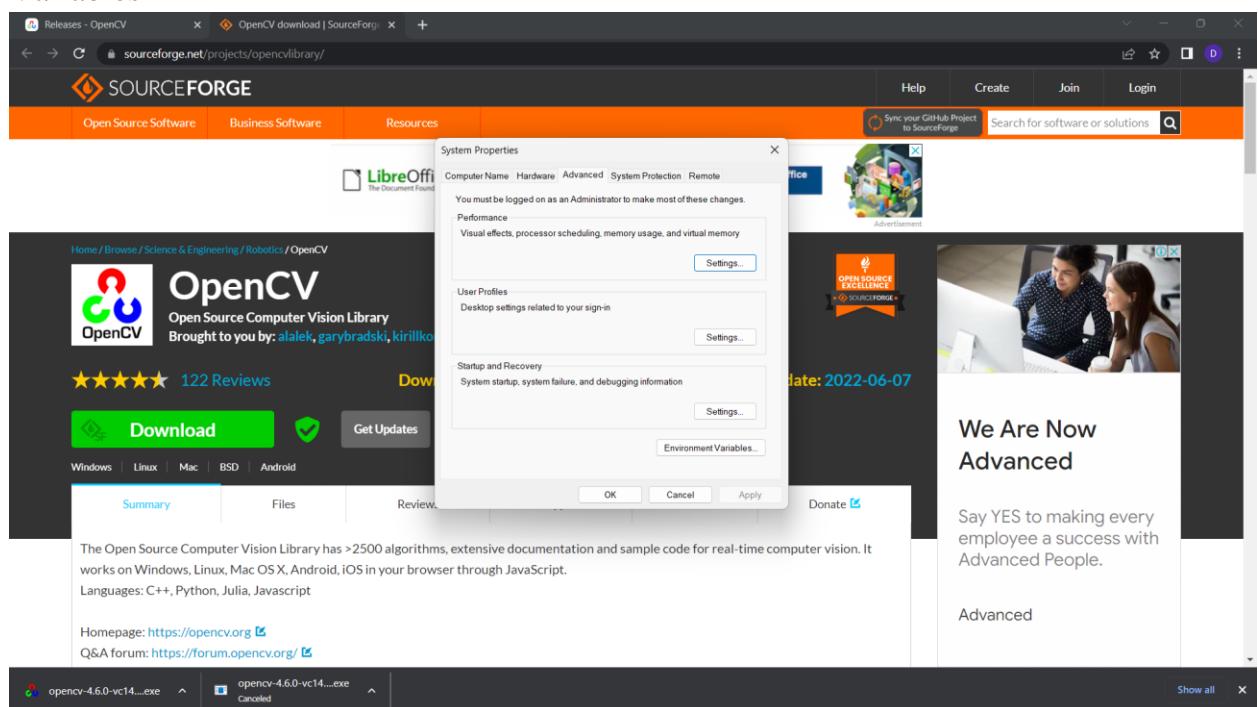


Chương 1: OverView

Bước 4: Cài môi trường OpenCV cho Visual Studio
Đầu tiên trong phần windows các bạn search từ “Edit the system environment variables”

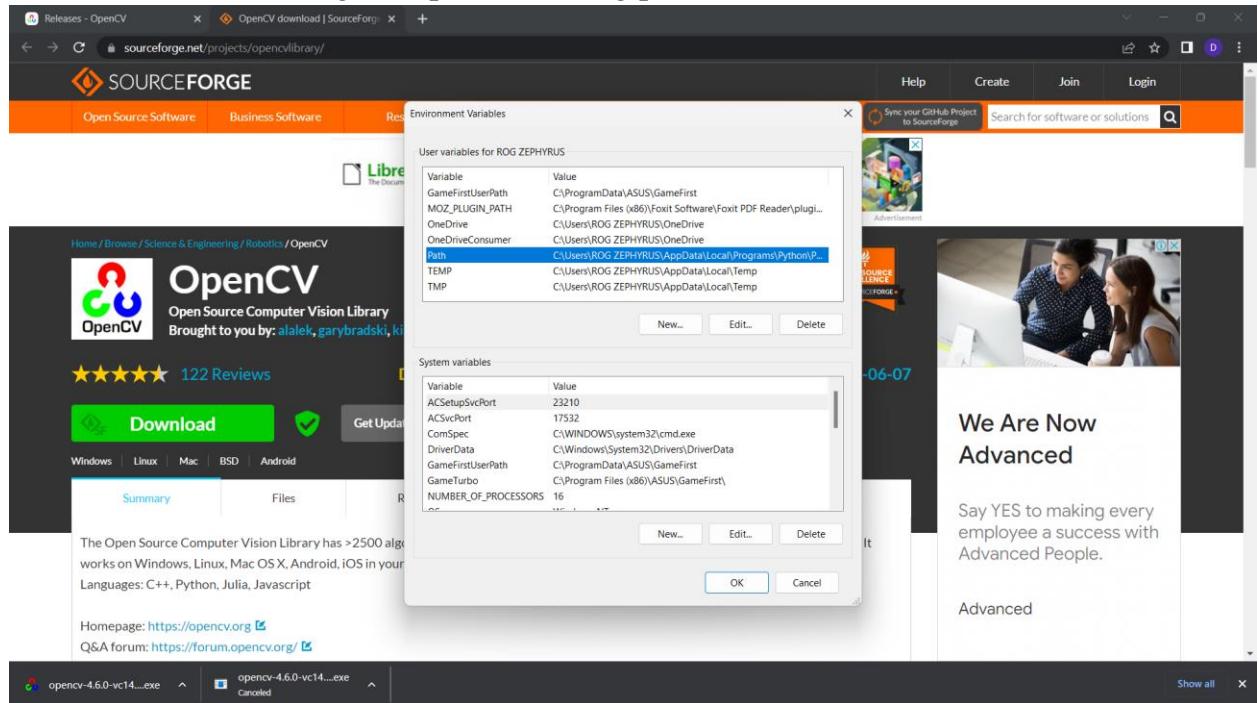


Sau khi nhấn vào thì hiện cửa sổ như hình lúc đó bạn nhấn vào phần Environment Variables

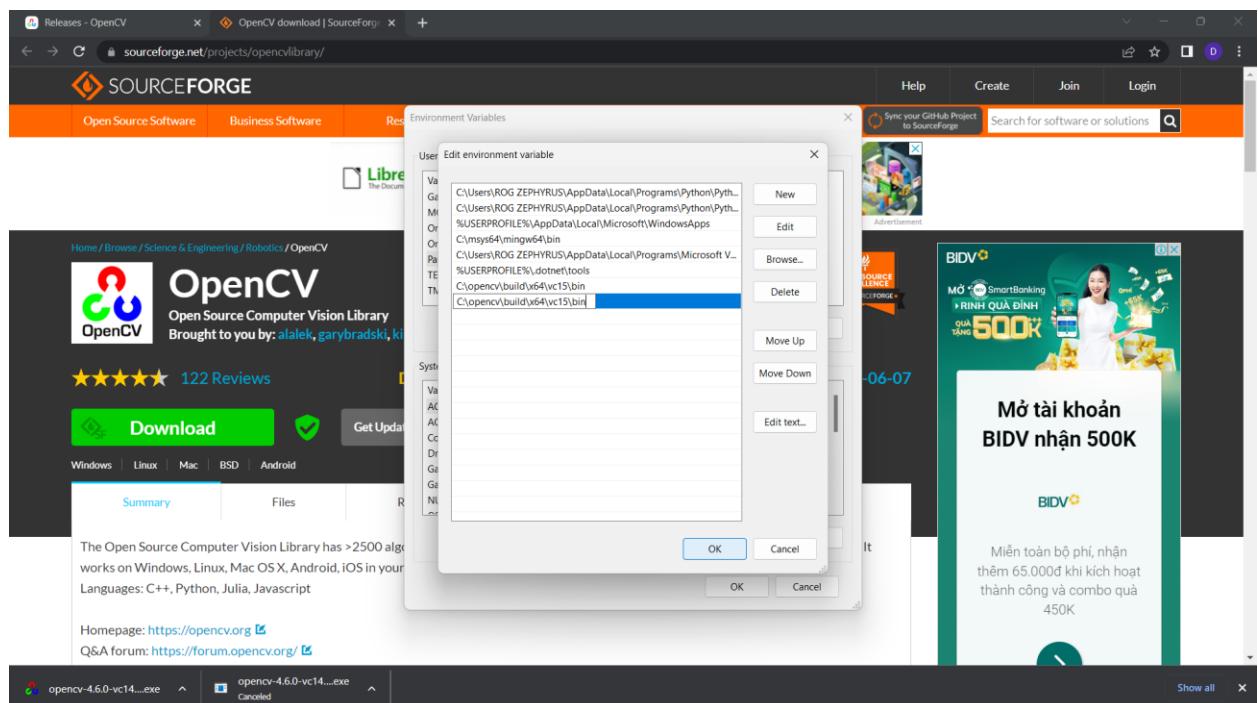


Chương 1: OverView

Sau khi ấn vào thì chúng ta tiếp tục tìm dòng path rồi nhấn vào chữ edit

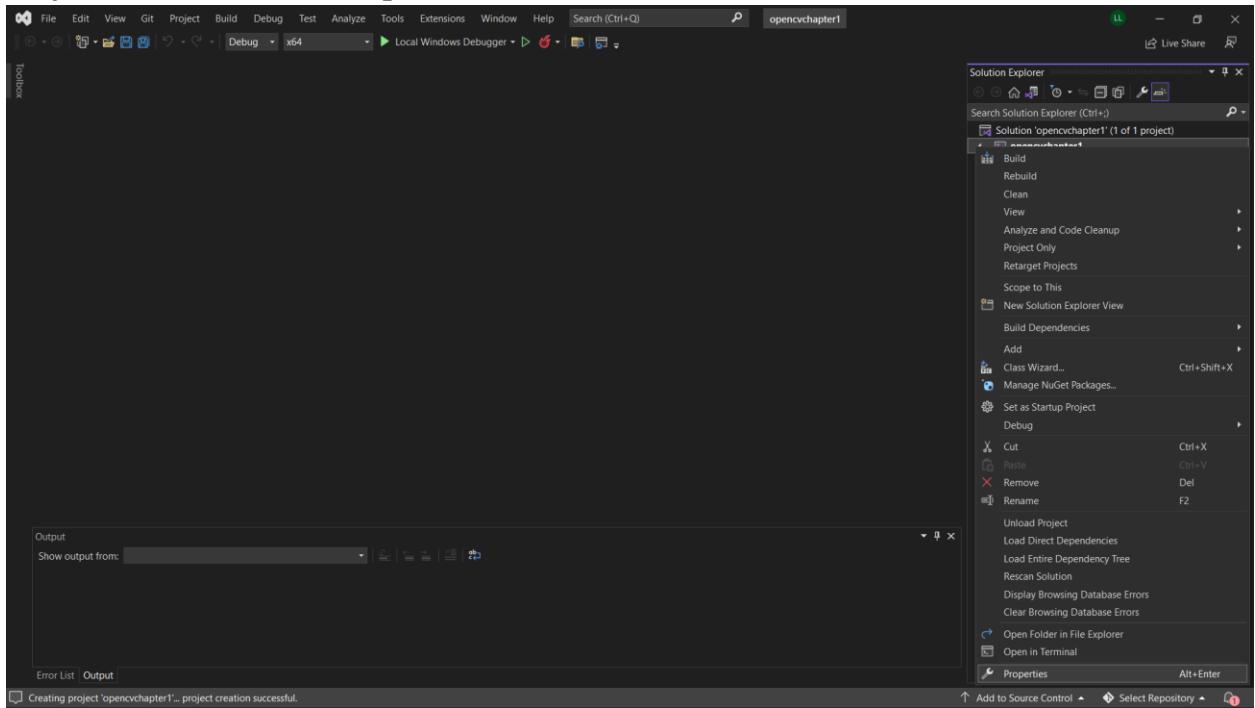


Sau khi ấn vào phần edit sẽ xuất hiện thêm cửa sổ khi đó chúng ta ấn vào new và tìm đường dẫn paste vào phần new (trong hình lúc cài đặt mình chọn ổ C là đường dẫn nên nó sẽ có đường dẫn như hình là "C:\opencv\build\x64\vc15\bin") rồi sau đó nhấn ok.



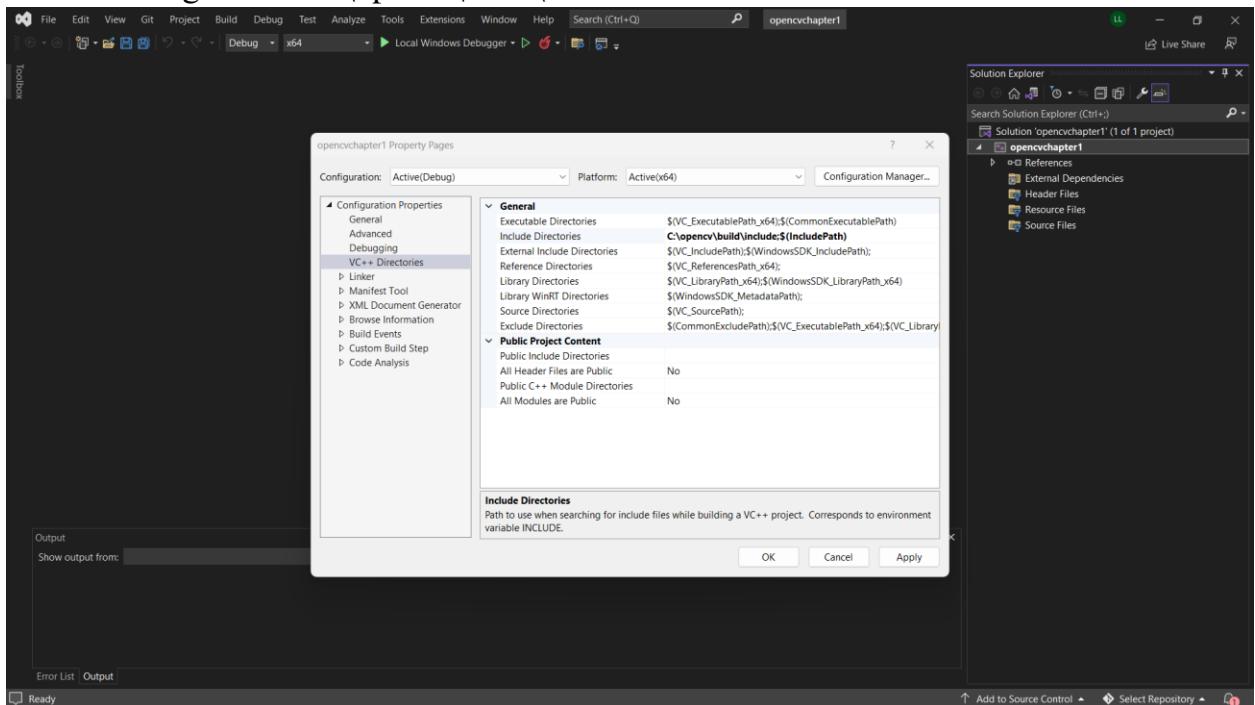
Chương 1: OverView

Bước 7: Sau khi cài xong môi trường thì chúng ta bắt đầu tạo một cái Project mới
Sau khi tạo một Project mới xong thì chúng ta bắt đầu click chuột phải vào phần Project rồi sau đó chọn Properties



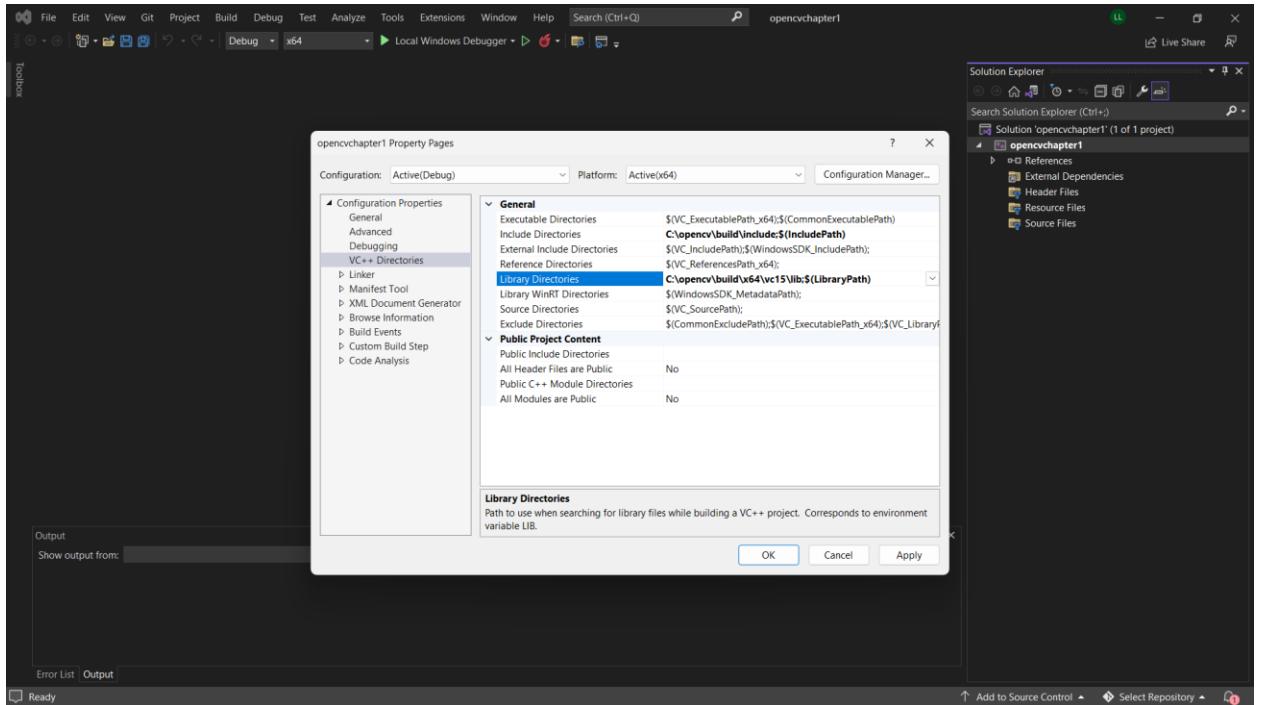
Sau khi ánh vào phần Properties thì xuất hiện cửa sổ

Trong dòng Include Directories chúng ta chọn đường đường dẫn phần đối với máy mình chọn ổ đĩa C và phần này cần đường dẫn đến Include nên mình tìm đường dẫn như trong hình là C:\opencv\build\include

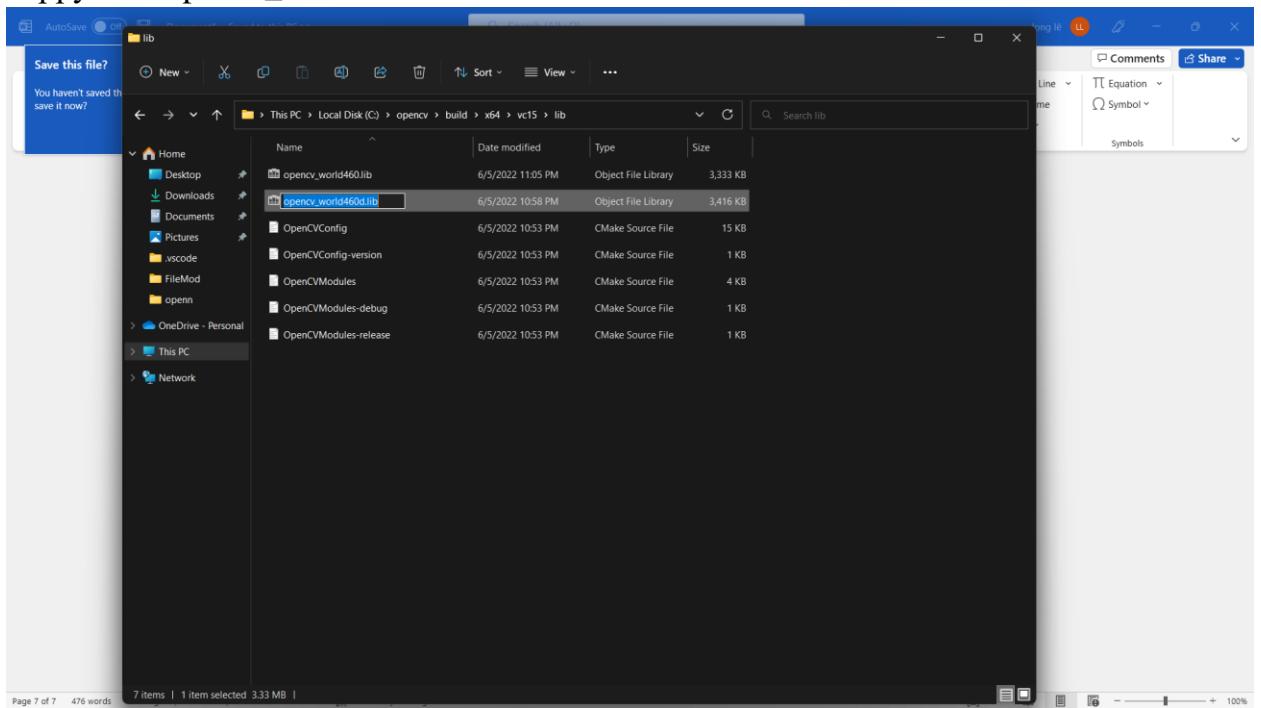


Chương 1: OverView

Cũng như trên trong phần Library thì mình chọn đường dẫn C:\opencv\build\x64\vc15\lib cho phần này

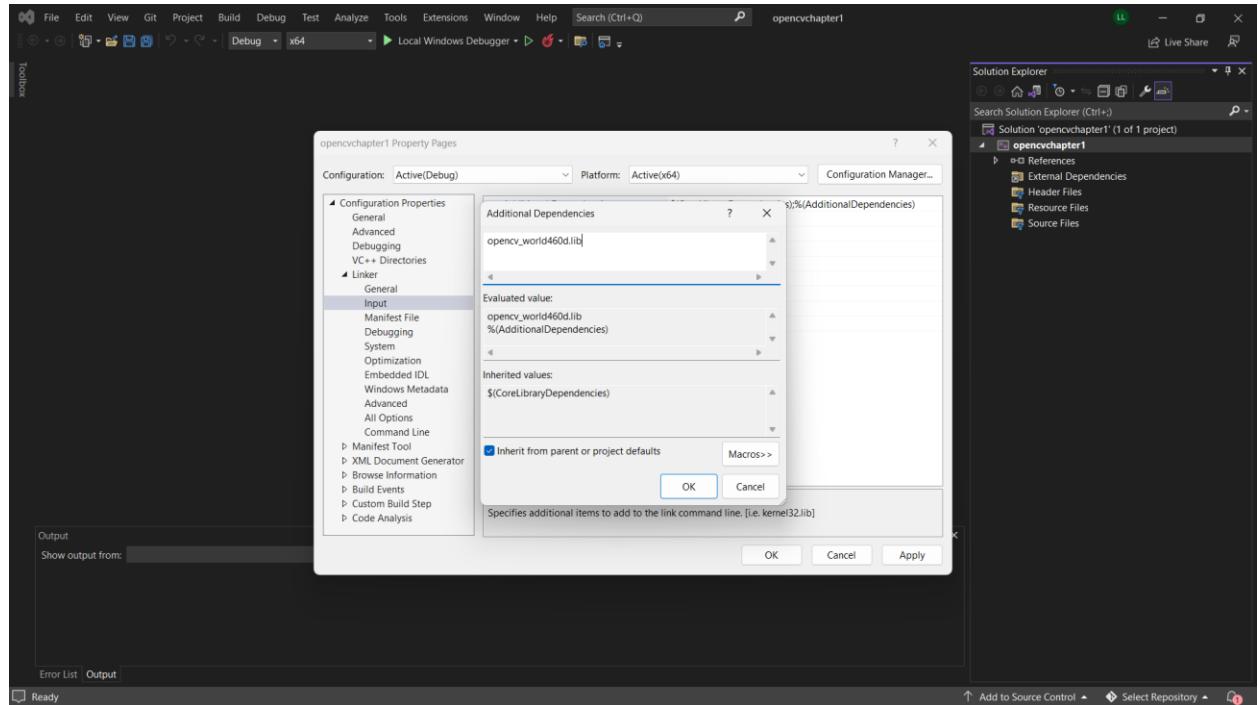


Sau đó mình tìm đường dẫn đến phần như trong hình theo đường dẫn C:\opencv\build\x64\vc15\lib trong máy mình sẽ xuất hiện cửa sổ như sau sau đó copy tên “opencv_world460d.lib”

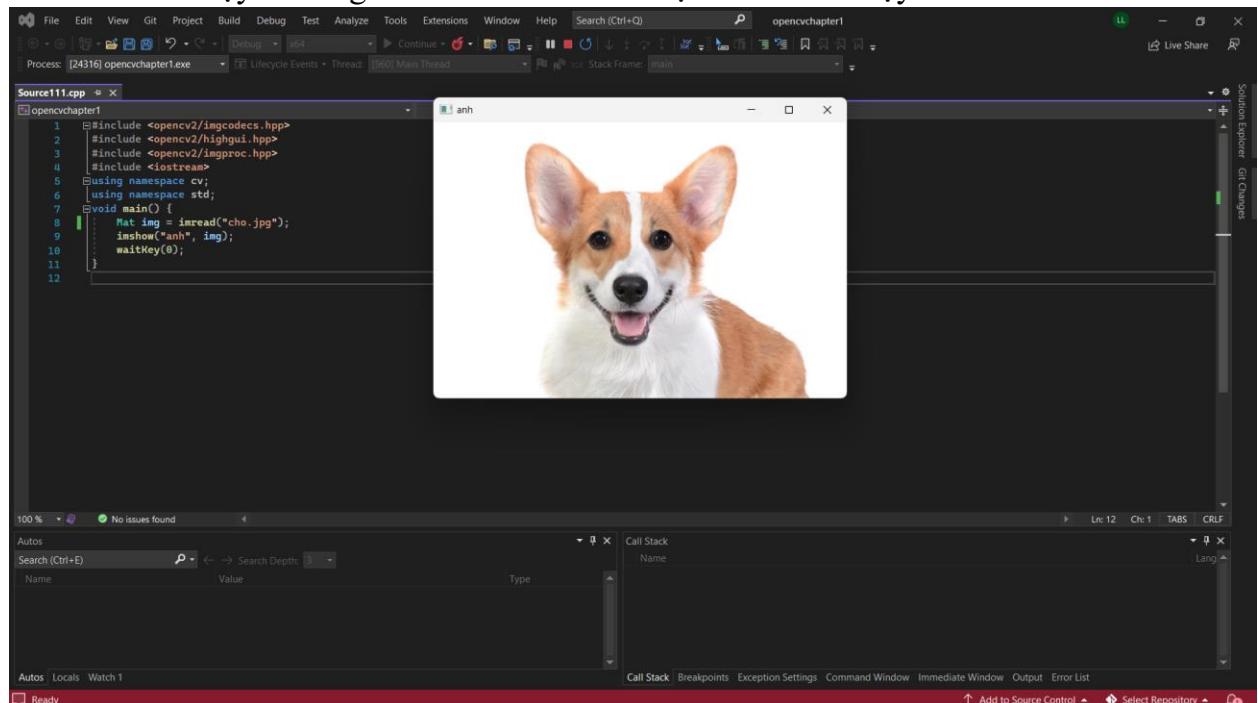


Chương 1: OverView

Copy nó đến phần vào phần Linker -> Input -> Additional Dependencies rồi paste vào như trong hình rồi sau đó nhấn oke



Bước 8 Test chạy chương trình. Minh sẽ đưa đoạn code vào chạy thử



Chạy thành công

2. Getting to Know OpenCV

2.1 Các loại dữ liệu opencv

- Các kiểu dữ liệu như : **int, float ,v.v....**
- + **Các kiểu dữ liệu nâng cao như :**
 - lớp và con trỏ
 - Mảng

2.2 Các loại cơ bản trong opencv

1. Point

- Đây là kiểu dữ liệu đơn giản nhất
- Lớp Point (cũng như các kiểu dữ liệu khác trong opencv) là một lớp template, tức là ta có thể khai báo kiểu point của kiểu integer, floating-point
- Lớp point có thể là point 2 chiều (có 2 giá trị như tọa độ Oxy) hoặc point 3 chiều (có 3 giá trị như tọa độ Oxyz)
- Ưu điểm của lớp point là nó tốn ít chi phí trong lưu trữ và xử lý. Tuy nhiên, có khá ít toán tử được định nghĩa trong lớp này nhưng từ nó có thể xây dựng nên các lớp tổng quát hơn như lớp vector, maxtrix ...
- Toán tử :

Operation	Examples
Default constructors	<code>cv::Point2i p();</code> <code>cv::Point3f p();</code>
Copy constructor	<code>cv::Point3f p2(p1);</code>
Value constructors	<code>cv::Point2i p(x0, x1);</code> <code>cv::Point3d p(x0, x1, x2);</code>
Cast to the fixed vector classes	<code>(cv::Vec3f) p;</code>
Member access	<code>p.x; p.y; // and for three-dimensional</code> <code>// point classes: p.z</code>

Dot product	float x = p1.dot(p2)
Double-precision dot product	double x = p1.ddot(p2)
Cross product	p1.cross(p2) // (for three-dimensional point // classes only)
Query if point p is inside of rectangle r	p.inside(r) // (for two-dimensional point // classes only)

2. Class Scalar

- Lớp scalar là một lớp point với 4 chiều

- Nó cũng là một lớp template, tuy nhiên, mặc định trả về của lớp này là 4 thành phần đều ở dạng double-precision floating-point

- Không giống như lớp point, lớp scalar không thể "cast" sang lớp vector vì nó là kế thừa trực tiếp từ lớp vector này. Do đó, nó cũng kế thừa các toán tử, các hàm truy cập các thành phần (ví dụ như operator[]), và các thuộc tính khác.

- Toán tử :

Operation	Examples
Default constructors	cv::Scalar s();
Copy constructor	cv::Scalar s2(s1);
Value constructors	cv::Scalar s(x0); cv::Scalar s(x0, x1, x2, x3);
Element-wise multiplication	s1.mul(s2);

3. size classes

- Chứa thông tin để lưu lại kích thước

VD: cv::Size(int w=0, int h=0);

4.Rect classes

- Chứa thông tin hình chữ nhật

VD: cv::Rect(x,y, width, height);

5.class cv:: RotatedRect

- Lớp đại diện cho các hình chữ nhật xoay

Operation	Example
Default constructors	cv::RotatedRect rr();
Copy constructor	cv::RotatedRect rr2(rr1);
Construct from two corners	cv::RotatedRect(p1, p2);
Value constructors; takes a point, a size, and an angle	cv::RotatedRect rr(p, sz, theta);
Member access	rr.center; rr.size; rr.angle;

6.Kiểu dữ liệu Mat

- Mat cơ bản là một lớp với hai phần dữ liệu: phần đầu ma trận (có chứa thông tin như kích thước của ma trận) và một con trỏ tới ma trận có chứa giá trị pixel.
- Lớp Mat trong opencv nằm trong modules core của bộ thư viện opencv và nằm trong namespace cv của bộ thư viện

2.3 Array Operators

- Các toán tử ma trận và hình ảnh cơ bản

Cv::abs() giá trị tuyệt đối của các phần tử trong một mảng

Cv::absdiff() giá trị tuyệt đối của hai mảng

Cv::add() thêm phần tử vào hai mảng

Cv::addWeighted() phép cộng trọng hai mảng

Cv::Compare toán tử so sánh cho các phần tử trong hai mảng

Cv::determinant tính định thức của ma trận vuông

Chương 2: Getting to Know OpenCV

Cv::dct tính toán biến đổi cosin rời rạc của mảng

Cv::InRange Kiểm tra xem các phần tử của mảng có nằm trong các giá trị của mảng khác không

Cv::magnitude tính toán độ lớn vecto hai chiều

Cv::Lut chuyển đổi mảng thành các chỉ số của bảng tra cứu

3 Graphical User Interface

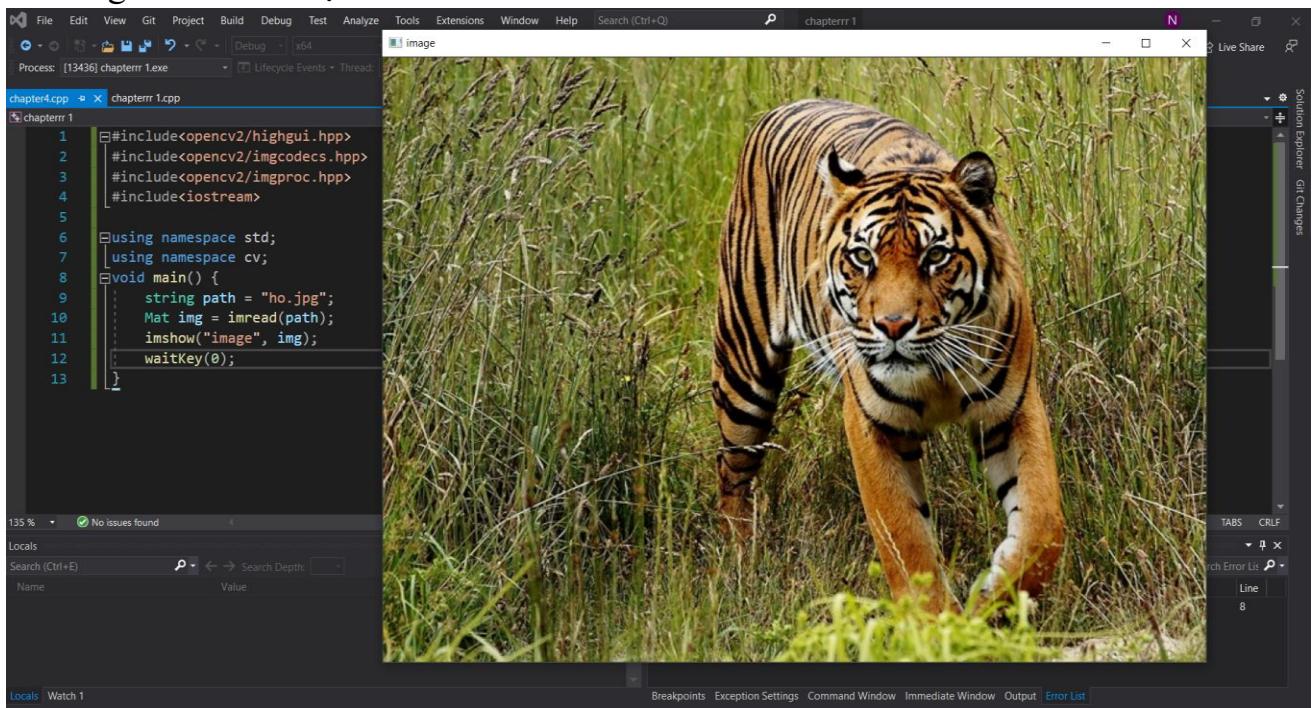
-HighGui : cho phép chúng ta mở cửa sổ , hiện thị hình ảnh , đọc và ghi các tệp liên quan về hình ảnh và video

3.1 Các module

- **Core functionality (core)** – module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.
- **Image Processing (imgproc)** – module xử lý hình ảnh gồm cả lọc hình ảnh , phép biến đổi hình học, chuyển đổi không gian màu, biểu đồ .
- **High-level GUI (highgui)** – dùng để thực hiện việc giao tiếp
- **Video** – module cho phép phân tích các video
- **Core functionality (core)** – module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.
- **Object Detection (objdetect)** – phát hiện các đối tượng và mô phỏng của các hàm được định nghĩa sẵn (vd: khuôn mặt, mắt, con người).

3.2 Loading and Saving Images

Chương trình minh họa



The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the code editor displays a C++ file named chapterrr 1.cpp with the following content:

```

1 #include<opencv2/highgui.hpp>
2 #include<opencv2/imgcodecs.hpp>
3 #include<opencv2/imgproc.hpp>
4 #include<iostream>
5
6 using namespace std;
7 using namespace cv;
8 void main() {
9     string path = "ho.jpg";
10    Mat img = imread(path);
11    imshow("image", img);
12    waitKey(0);
13 }

```

On the right, a large image of a tiger walking through tall grass is displayed. The Visual Studio interface includes various toolbars, menus, and status bars at the bottom.

- Highgui, imgproc, imgcodecs : xử lý và hiển thị hình ảnh video
- String path : đường dẫn file ảnh
- Mat : Các kiểu dữ liệu ma trận ảnh được lưu dưới dạng Mat
- Imread : đọc ảnh đầu vào và lưu vào biến img
- Imshow : Hiện thị ảnh ra cửa sổ được tạo ra trước đó
- WaitKey : dừng màn hình lại trong một khoảng thời gian bằng tham số delay. Nếu muốn dừng màn hình mãi ta cho tham số delay là 0.

3.3 Làm việc với video

- Mở webcam

Chương trình minh họa :

```
#include<opencv2/highgui.hpp>
#include<opencv2/imgcodecs.hpp>
#include<opencv2/imgproc.hpp>
#include<iostream>

using namespace std;
using namespace cv;

void main() {
    VideoCapture cap(0);
    Mat img;
    while (true) {
        cap.read(img);
        imshow("Camera", img);
        waitKey(1);
    }
}
```

Locals
Search (Ctrl+E)
Breakpoints Exception Settings Command Window Immediate Window Output Error List

Error List
Entire Solution 0 Errors 18 Warnings 0 Messages
Code Description Project File Line Suppression State
C26812 The enum type 'cvNormTypes' is unscoped. chapterrr 1 base.hpp 384
Prefer 'enum class' over 'enum' (Enum.3).
The enum type 'cvAccessFlag' is unscoped.
Breakpoints Exception Settings Command Window Immediate Window Output Error List

- **Videocapture** : mở tệp video hoặc thiết bị chụp ,tạo đối tượng quay video, phát trực tuyến hoặc hiển thị video.

Load và hiện thị video:

Chương trình minh họa



The screenshot shows a video player window displaying a scene from a factory. Multiple orange ABB robotic arms are visible, working on a production line. The background shows industrial equipment and lighting.

```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window
Process: [17968] chapterrrr 1.exe Lifecycle Events + Thread: [10756] Main Thread
chapter2.cpp chapterrrr 1.cpp (Global Scope)
1 #include<opencv2/highgui.hpp>
2 #include<opencv2/imgcodecs.hpp>
3 #include<opencv2/imgproc.hpp>
4 #include<iostream>
5
6 using namespace std;
7 using namespace cv;
8 void main() {
9     string path = "videos.mp4";
10    VideoCapture cap(path);
11    Mat img;
12    while (true) {
13        cap.read(img);
14        imshow("video", img);
15        waitKey(1);
16    }
17 }
18
19
135 % No issues found
Locals
Search (Ctrl+E) <- > Search Depth: 3
Name Value Type
Locals Watch 1
Ready

```

3.4 Không gian màu , chuyển đổi giữa các không gian màu

- Các không gian màu được được chuyển đổi qua lại nhờ hàm cvtColor

Chương trình mẫu :



The screenshot shows a video player window displaying a tiger walking through tall, dry grass. The tiger's stripes are clearly visible against the brown background.

```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Process: [7440] chapterrrr 1.exe Lifecycle Events + Thread: [10756] Main Thread
chapter4.cpp chapter3.cpp chapterrrr 1.cpp (Global Scope)
1 #include<opencv2/highgui.hpp>
2 #include<opencv2/imgcodecs.hpp>
3 #include<opencv2/imgproc.hpp>
4 #include<iostream>
5
6 using namespace std;
7 using namespace cv;
8 void main() {
9     string path = "ho.jpg";
10    Mat img = imread(path);
11    Mat imgGray;
12    cvtColor(img, imgGray, COLOR_BGR2GRAY);
13    imshow("Gray", imgGray);
14    waitKey(0);
15
16
17
135 % No issues found
Locals
Search (Ctrl+E) <- > Search Depth: 3
Name Value Type
Locals Watch 1
Breakpoints Exception Settings Command Window Immediate Window Output Error List

```

3.5 Bộ lọc Blur

- Bộ lọc Blur bộ lọc này có tác dụng làm mờ, trơn ảnh.
- GaussianBlur làm mờ ảnh

Chương trình mẫu



The screenshot shows the Visual Studio IDE interface. On the left, the code editor displays a C++ program named chapterrr 1.cpp. The code reads an image named 'ho.jpg', applies a Gaussian blur with a kernel size of 9x9, and then displays the result. On the right, the main window shows a tiger walking towards the camera in a grassy field. The blurred version of the image is displayed in the center of the screen, showing the effect of the Gaussian blur.

```

1 #include<opencv2/highgui.hpp>
2 #include<opencv2/imgcodecs.hpp>
3 #include<opencv2/imgproc.hpp>
4 #include<iostream>
5
6 using namespace std;
7 using namespace cv;
8 void main() {
9     string path = "ho.jpg";
10    Mat img = imread(path);
11    Mat imgBlur;
12    GaussianBlur(img, imgBlur, Size(9,9), 5, 0);
13    imshow("Blur", imgBlur);
14    waitKey(0);
15
16 }
17

```

3.6 Bộ lọc canny

- bộ lọc canny để tìm và tối ưu đường biên

Chương trình minh họa :



The screenshot shows the Visual Studio IDE interface. On the left, the code editor displays a C++ program named chapterrr 1.cpp. The code reads an image named 'ho.jpg', applies a Gaussian blur with a kernel size of 9x9, and then uses the Canny algorithm to detect edges. On the right, the main window shows the original tiger image. The detected edges are overlaid on the image, appearing as a dense network of white lines against a black background.

```

1 #include<opencv2/highgui.hpp>
2 #include<opencv2/imgcodecs.hpp>
3 #include<opencv2/imgproc.hpp>
4 #include<iostream>
5
6 using namespace std;
7 using namespace cv;
8 void main() {
9     string path = "ho.jpg";
10    Mat img = imread(path);
11    Mat imgBlur, imgCanny;
12    GaussianBlur(img, imgBlur, Size(9, 9), 5, 0);
13    Canny(imgBlur, imgCanny, 26, 26);
14    imshow("Canny", imgCanny);
15    waitKey(0);
16
17 }
18

```

4. Contour

4.1 Định nghĩa:

Các bạn có thể hiểu contour là “tập các điểm-liên-tục tạo thành một đường cong(curve) (boundary), và không có khoảng hở trong đường cong đó, đặc điểm chung trong một contour là các các điểm có cùng /gần xâú với một giá trị màu, hoặc cùng mật độ. Contour là một công cụ hữu ích được dùng để phân tích hình dạng đối tượng, phát hiện đối tượng và nhận dạng đối tượng”. Để tìm contour chính xác, chúng ta cần phải *nhi phân hóa bức ảnh* (nhớ là ảnh nhị phân nha các bạn, không phải ảnh grayscale đâu). Các kỹ thuật nhị phân hóa ảnh ở xử lý ảnh cơ bản có thể liệt kê đến là đặt ngưỡng, hoặc candy edge detection. Chúng ta sẽ không bàn kỹ về các cách đặt ngưỡng (mặc dù có khá nhiều cách đặt ngưỡng, và trong opencv cũng có implement một vài phương pháp, nhưng nó không phải là mục tiêu của bài này, nên mình không đề cập ở đây) hoặc edge detection ở bài viết này, mà chúng ta sẽ đi vào các tìm contours bằng các sử dụng opencv luôn.

Project của nhóm

Project 1: Nhận Diện Ánh

```
#include<iostream>
#include<opencv2/imgcodecs.hpp>
#include<opencv2/highgui.hpp>
#include<opencv2/imgproc.hpp>
#include <opencv2/core.hpp>

using namespace cv;
using namespace std;

Mat srcgray, srcblur, srccil, srccero, srccanny, dt;
void getContours(Mat srccil, Mat src) {
    vector<vector<Point>> contours;
    vector<Vec4i> hierarchy;
    string objectType;

    findContours(srccil, contours, hierarchy, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);
    //drawContours(src, contours, -1, Scalar(255, 0, 255), 10);

    vector<vector<Point>>conPoly(contours.size());
    vector<Rect>boundRect(contours.size());

    for (int i = 0; i < contours.size(); i++) {
        int area = contourArea(contours[i]);
        cout << area << endl;
        if (area > 0) {
```

```
vector<vector<Point>>conPoly(contours.size());
vector<Rect>boundRect(contours.size());

for (int i = 0; i < contours.size(); i++) {
    int area = contourArea(contours[i]);
    cout << area << endl;
    if (area > 0) {
        float peri = arcLength(contours[i], true);
        approxPolyDP(contours[i], conPoly[i], 0.02 * peri, true);
        drawContours(src, conPoly, i, Scalar(255, 0, 255), 2);
        cout << conPoly[i].size() << "\n";
        boundRect[i] = boundingRect(conPoly[i]);
        rectangle(src, boundRect[i].tl(), boundRect[i].br(), Scalar(256, 0, 256), 2);

        int objcor = (int)conPoly[i].size();

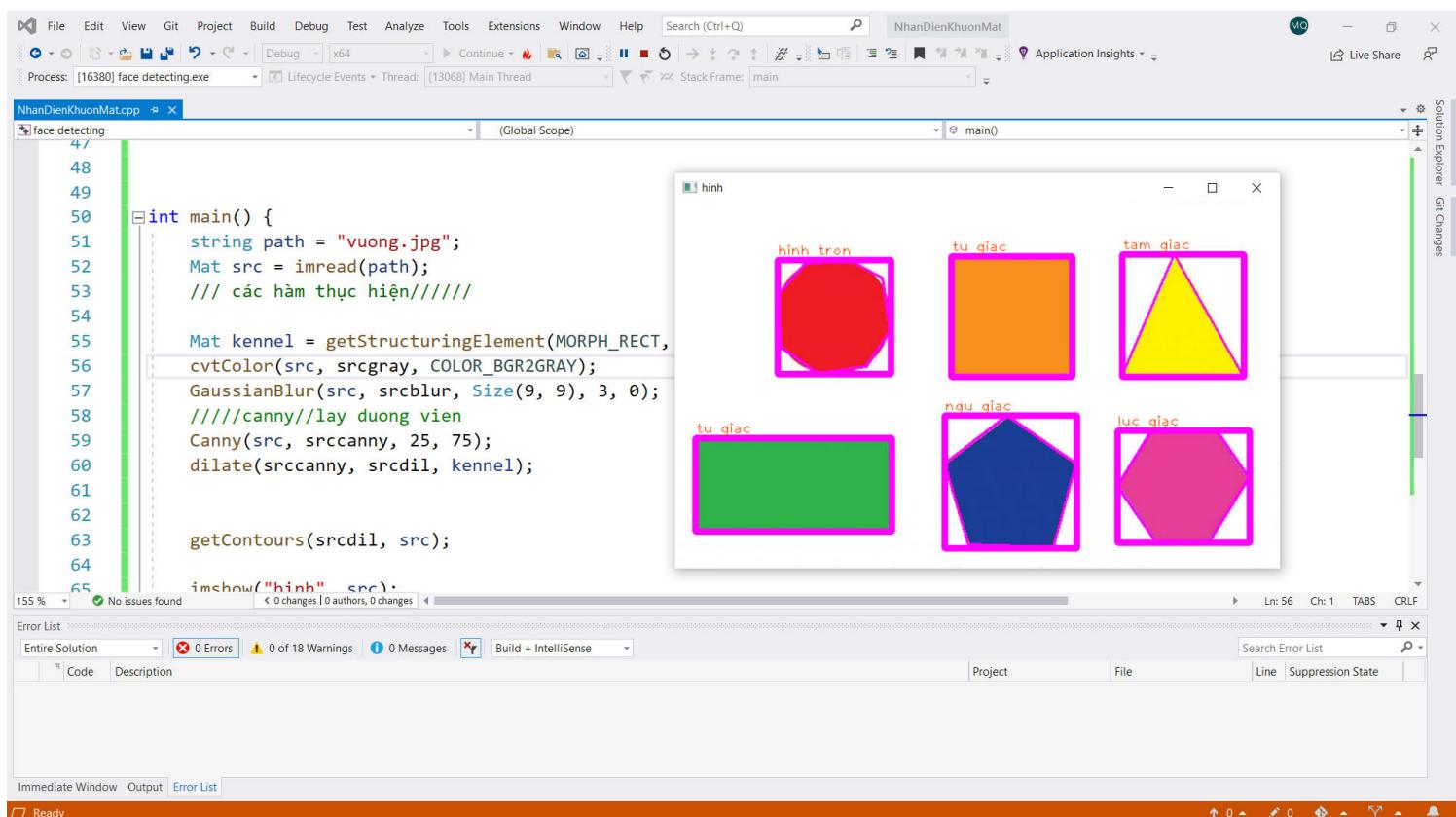
        if (objcor == 3) { objectType = "tam giac"; }
        if (objcor == 4) { objectType = "tu giac"; }
        if (objcor == 5) { objectType = "ngu giac"; }
        if (objcor == 6) { objectType = "luc giac"; }
        if (objcor > 6) { objectType = "hinh tron"; }

        putText(src, objectType, { boundRect[i].x, boundRect[i].y - 5 }, FONT_HERSHEY_SIMPLEX, 1, Scalar(0, 0, 255), 2);
    }
}
```

Screenshot of Microsoft Visual Studio showing the code editor and Solution Explorer. The code editor displays the file `NhanDienVienAnh.cpp` with the following content:

```
49
50 int main() {
51     string path = "vuong.jpg";
52     Mat src = imread(path);
53     /// các hàm thực hiện/////
54
55     Mat kernel = getStructuringElement(MORPH_RECT, Size(3, 3));
56     cvtColor(src, srcgray, COLOR_BGR2GRAY);
57     GaussianBlur(src, srcblur, Size(9, 9), 3, 0);
58     ////canny//lay duong vien
59     Canny(src, srccanny, 25, 75);
60     dilate(srccanny, srcdil, kernel);
61
62
63     getContours(srcdil, src);
64
65     imshow("hinh", src);
66
67
68     waitKey(0);
69
70     return 0;
71 }
72 }
```

The Solution Explorer shows a single project named `nhandienkhuanmat` with files `x64`, `daphuongtien.jpg`, and `NhanDienVienAnh.cpp`.



Project 2: Nhận Diện Viền Vật qua Camera

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Solution Explorer:** Shows the solution 'NhanDienVienCamera' (1 of 1 project) with a single item 'face detecting' under it.
- Toolbox:** Standard Visual Studio toolbox.
- Code Editor:** The file 'NhanDienVienCamera.cpp' is open, containing the following code:

```
#include <opencv2/imgcodecs.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#include <iostream>

using namespace std;
using namespace cv;

void ContourTool(Mat imgDil, Mat img)
{
    vector<vector<Point>> contours; // vector điểm trên hình
    vector<Vec4i> hierarchy;
    findContours(imgDil, contours, hierarchy, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);
    drawContours(img, contours, -1, Scalar(255, 0, 255), 2);
}

void main()
{
    VideoCapture cap(0);
    Mat img;
    Mat imgCVT, imgBlur, imgCanny, imgDilate;

    while (true)
    {
        cap.read(img);
```

The code implements a basic face detection loop using OpenCV's contour tools.

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Solution Explorer:** Shows the solution 'NhanDienVienCamera' (1 of 1 project) with a single item 'face detecting' under it.
- Toolbox:** Standard Visual Studio toolbox.
- Code Editor:** The file 'NhanDienVienCamera.cpp' is open, containing the following code:

```
void main()
{
    VideoCapture cap(0);
    Mat img;
    Mat imgCVT, imgBlur, imgCanny, imgDilate;

    while (true)
    {
        cap.read(img);
        cvtColor(img, imgCVT, COLOR_BGR2GRAY);
        GaussianBlur(img, imgBlur, Size(7, 7), 4, 0);
        Canny(imgBlur, imgCanny, 70, 7);
        Mat kernel = getStructuringElement(MORPH_RECT, Size(2, 2));
        dilate(imgCanny, imgDilate, kernel);

        ContourTool(imgDilate, img);

        imshow("Image", img);
        waitKey(20);
    }
}
```

The code has been modified to include image processing steps like grayscale conversion, Gaussian blur, and Canny edge detection before performing the contour tool operation.

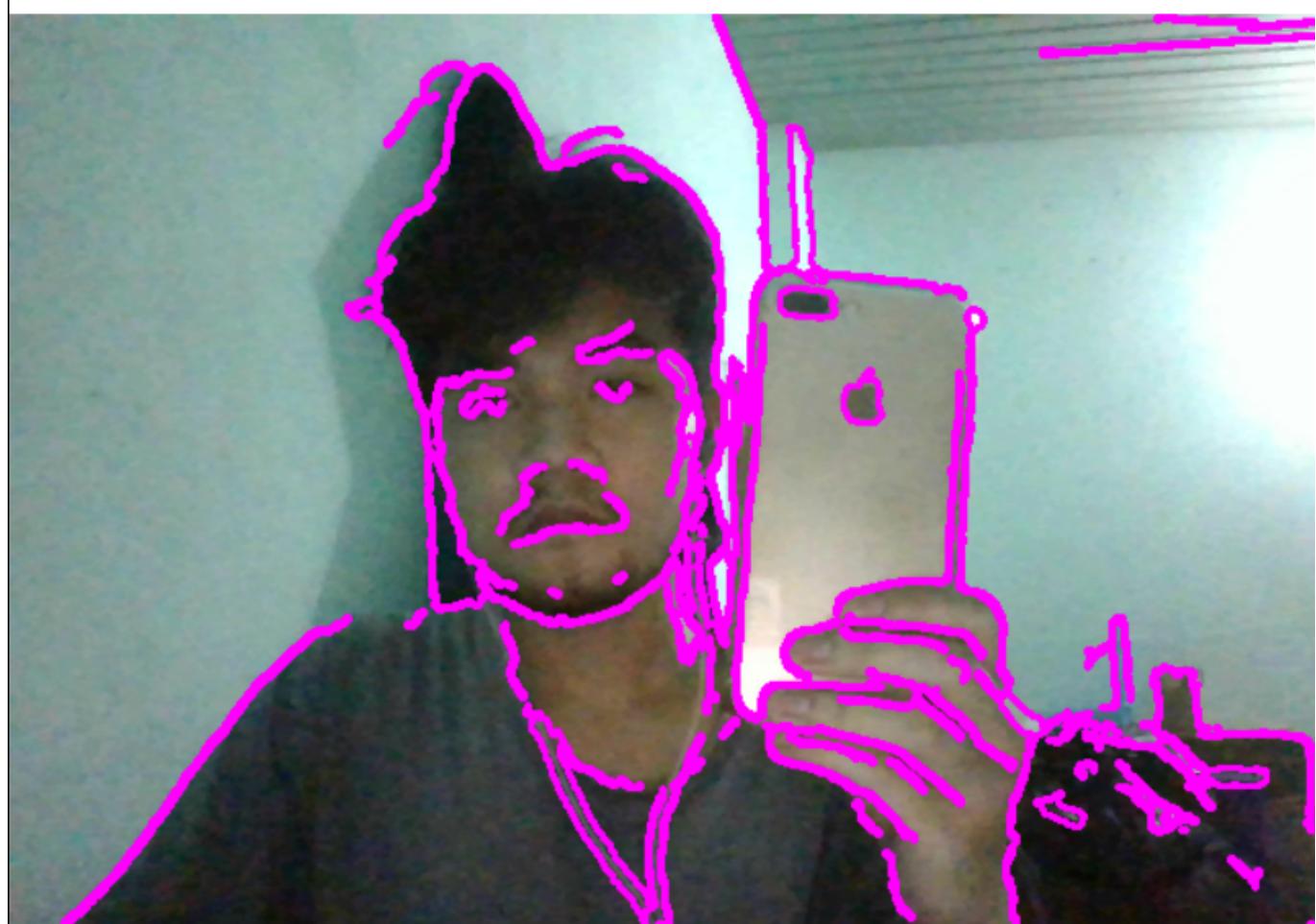
Image

- □ ×



Image

- □ ×



Project 3: Nhận Diện Khuôn Mặt

The screenshot shows the Visual Studio IDE interface with the following details:

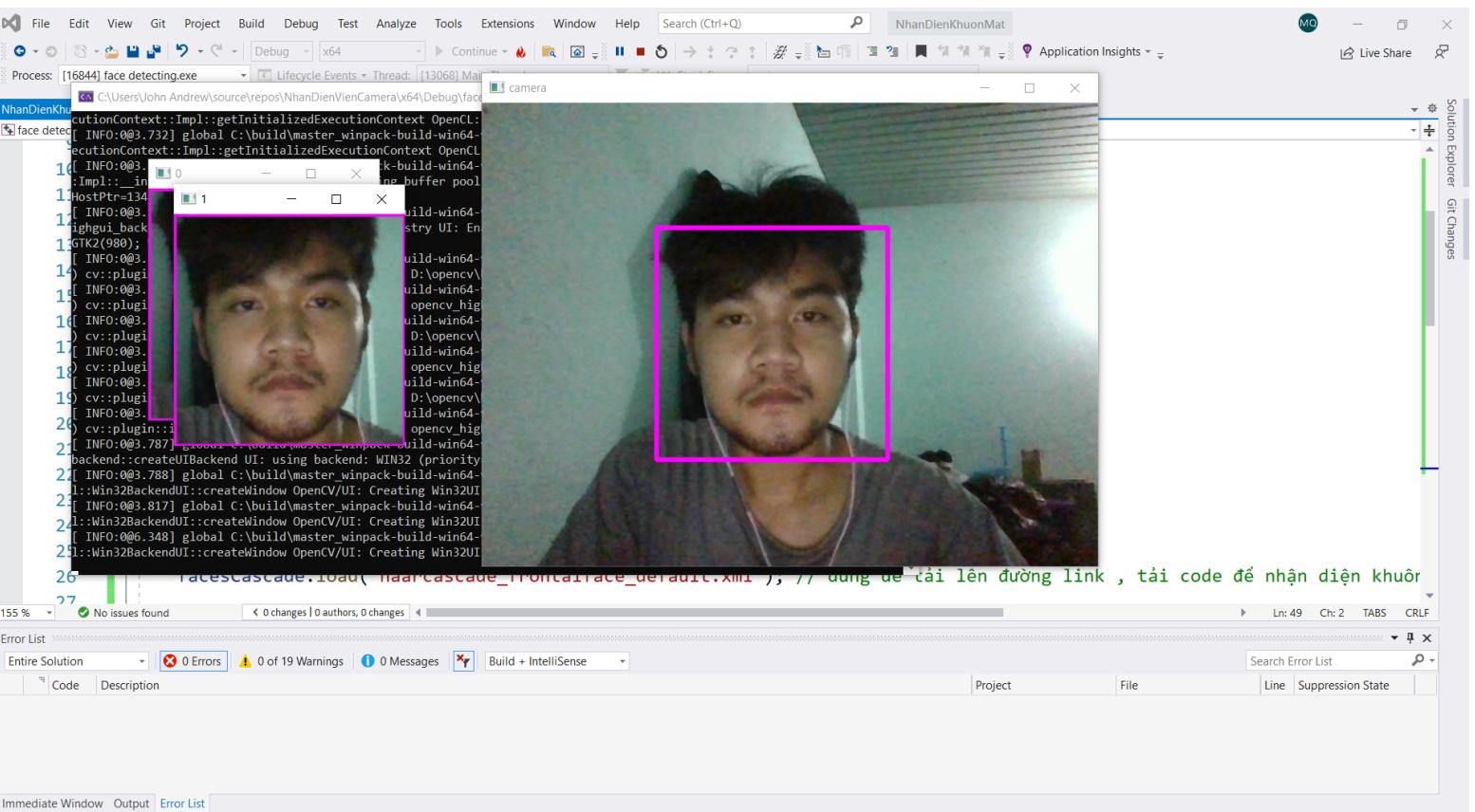
- Solution Explorer:** Shows the project structure for "OpenCVProject1". It includes a solution, an external dependencies folder, an x64 folder containing .gitattributes, .gitignore, and bai.jpg, and a chapter1, 2, 3, chapter4, and contours folder. The contours folder contains chapter1, 2, 3.cpp, chapter4.cpp, chapter5.cpp, cr.jpg, haarcascade_frontalface_default.xml, hinh.jpg, huy.jpg, huy.mp4, khannh.jpg, main.cpp, opencv_haarcascade_russian_plate_number.xml, project1.cpp, test1.cpp, and vuong.jpg.
- Properties:** The properties for project1.cpp are shown, with "Included In Project" set to True.
- Code Editor:** The main.cpp file is open, displaying C++ code for face detection using OpenCV. The code includes imports for iostream, opencv2/imgcodecs.h, opencv2/highgui.h, opencv2/imgproc.h, opencv2/core.h, and opencv2/objdetect.h. It defines a getContours function that finds contours in an image and a main function that captures video from a camera, detects faces using a cascade classifier, and displays the results.
- Error List:** Shows 17 errors, mostly related to conversion from double to int or float, and one undeclared identifier error.
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File Manager.

The screenshot shows the Visual Studio IDE interface with the following details:

- Solution Explorer:** Shows the project structure for "OpenCVProject1". It includes a solution, an external dependencies folder, an x64 folder containing .gitattributes, .gitignore, and bai.jpg, and a chapter1, 2, 3, chapter4, chapter5, and contours folder. The contours folder contains chapter1, 2, 3.cpp, chapter4.cpp, chapter5.cpp, cr.jpg, haarcascade_frontalface_default.xml, hinh.jpg, huy.jpg, huy.mp4, khannh.jpg, main.cpp, opencv_haarcascade_russian_plate_number.xml, project1.cpp, test1.cpp, and vuong.jpg.
- Properties:** The properties for project1.cpp are shown, with "Included In Project" set to True.
- Code Editor:** The main.cpp file is open, displaying the completed C++ code for face detection. The code includes imports for iostream, opencv2/imgcodecs.h, opencv2/highgui.h, opencv2/imgproc.h, opencv2/core.h, and opencv2/objdetect.h. It defines a getContours function that finds contours in an image and a main function that captures video from a camera, detects faces using a cascade classifier, and displays the results. The code also includes a loop to continuously process frames.
- Error List:** Shows 16 errors, mostly related to conversion from double to int or float, and one undeclared identifier error.
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File Manager.

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OpenProject1
- Solution Explorer:** Shows the project structure for "OpenCVproject1".
 - OpenCVcontours
 - x64
 - External Dependencies
 - gitattributes
 - gitignore
 - bau.jpg
 - chapter 1,2,3.cpp
 - chapter4.cpp
 - chapter5.cpp
 - contours.cpp
 - cri.jpg
 - haarcascade_frontalface.default.xml
 - hinh.jpg
 - hoay.mp4
 - khanh.jpg
 - main.cpp
 - opencv_haarcascade_russian_plate_number.xml
 - project1.cpp
 - test1.cpp
 - vuong.jpg
- Properties:** Shows the properties for selected files.
- Code Editor:** Displays the "project1.cpp" file content. The code uses OpenCV's Haar Cascade classifier to detect faces in an image. It includes comments in Vietnamese explaining the code logic.
- Error List:** Shows 16 errors, 3 warnings, and 0 messages.
 - MS8802: The intermediate directory 'x64\Debug' contains files shared from another project (OpenCVproject1.vcxproj). This can lead to incorrect clean and rebuild behavior.
 - C4244: 'initializing': conversion from 'double' to 'int', possible loss of data
 - C4244: 'initializing': conversion from 'double' to 'float', possible loss of data
 - C2065: 'conPoly': undeclared identifier
 - C2660: 'cv::approxPolyDP': function does not take 3 arguments
- Status Bar:** Ready, Type here to search, ENG, 10:46 PM, 9/20/2023, master



Kết luận: Qua các project trên nhóm đã hoàn thành các contour cơ bản, ứng dụng cơ bản xong kiến thức là vô tận nhưng qua đó cũng tích lũy thêm các kinh nghiệm trong việc học và ứng dụng OPENCV

Tài Liệu Tham Khảo:

+) Youtube

+) doc.Opencv.org

Danh Mục Từ Viết Tắt:

.....