Shape, square

Description automatically generated**l**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA ĐIỆN - ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN HỌC**

\*\*\*\*\*\*\*\*\*\*\*



**BÁO CÁO BÀI TẬP LỚN**

**Môn: Xử lý ảnh trong công nghiệp**

**Đề tài:Thiết kế phần mềm nhận diện biển số xe**

Sinh viên thực hiện:Nguyễn Tùng Lâm

Lớp:Kỹ thuật điều khiển và Tự động hóa 2-K62

Giảng viên hướng dẫn: Nguyễn Hoàng Vân

***Hà Nội***

**Mục lục**

**Phần I:Mở đầu**

1.Giới thiệu đề tài……………………………………………………………….1

2.Mục đích nghiên cứu………………………………………………………….1

3.Mục tiêu đề tài………………………………………………………………...2

**Phần II:Cơ sở lý thuyết**

**Chương 1:Ứng dụng xử lý ảnh vào công nghệ nhận dạng biển số xe**………3

**Chương 2:Phần mềm Pycharm và các thư viện đi kèm**……………………..8

2.1.Giới thiệu ngôn ngữ Python và phần mềm Pycharm………………………..8

2.2.Thư viện OpenCV………………………………………………………….14

2.3.Thư viện Tkinter…………………………………………………………...17

**Chương 3:Trình bày và thực hiện thuật toán chương trình**

3.1.Sơ đồ thuật toán……………………………………………………………18

3.2.Ý tưởng và hình ảnh……………………………………………………….19

3.3.Giao diện chương trình và hình ảnh………………………………………..23

**Phần III:Kết Luận**

1.Kết quả đạt được……………………………………………………………..25

2.Một số hạn chế của phần mềm……………………………………………….25

2.Định hướng đề tài trong tương lai……………………………………………25

4.Chương trình…………………………………………………………………26

**Phần I:Mở đầu**

1. **Giới thiệu đề tài:**

Cùng với sự phát triển khoa học kỹ thuật,nhu cầu đi lại của con người ngày càng tăng,lưu lượng giao thông ngày càng lớn.Với số lượng phương tiện giao thông ngày càng lớn và không ngừng tăng thì việc quản lý các phương tiện giao thông gặp rất nhiều khó khăn do đó cần một hệ thống tự động.Một trong những hệ thống đó là hệ thống nhận diện biển số xe.Đó là một hệ thống có khả năng “đọc” và hiểu các biển số xe một cách tự động

Các vấn đề của các bãi đỗ xe truyền thống hiện nay vẫn đang mắc phải đó chính là:

* Tốn nhiều nhân công
* An toàn chưa cao,vẫn còn xảy ra hiện tượng mất xe
* Vé xe bằng giấy,dễ bị nhàu

1. **Mục đích nghiên cứu**

Tại Việt Nam, nghiên cứu phát triển xử lí ảnh đã có những bước tiến đáng kể trong những năm vừa qua. Nhiều đơn vị trên toàn quốc thực hiện các nghiên cứu cơ bản và nghiên cứu ứng dụng về xử lí ảnh, ngoài ra các doanh nghiệp thiết kế và chế tạo cũng đã có nhiều sản phẩm ấn tượng trên trường quốc tế về xử lí ảnh

Trước sự ra đời mạnh mẽ của xử lý ảnh và đã có nhiều ứng dụng trong thực tế. Như trong cuộc sống hằng ngày, xử lý ảnh có mặt hầu hết ở các đồ dùng thông dụng như: Tivi, điện thoại, máy ảnh… hoặc trong lĩnh vực khoa học kỹ thuật, xử lý ảnh đã và đang có những đóng góp quan trọng,đặc biệt là lĩnh vực nhận dạng

Và để hiểu hơn và ứng dụng công nghệ xử lý ảnh em quyết định chọn đề tài “Thiết kế phần mềm nhận diện biển số xe ứng dụng công nghệ xử lý ảnh”

1. **Mục tiêu đề tài**

Thiết kế và khởi chạy phần mềm nhận diện biển số xe.Lấy ảnh từ camera trước của máy tính sau đó tách và hiện biển số xe trên cửa sổ giao diện

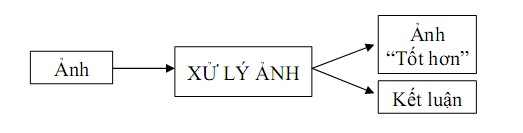
**Phần II:Cơ sở lý thuyết**

**Chương 1:Ứng dụng xử lý ảnh vào công nghệ nhận dạng biển số**

#### **a. Khái niệm xử lý ảnh**

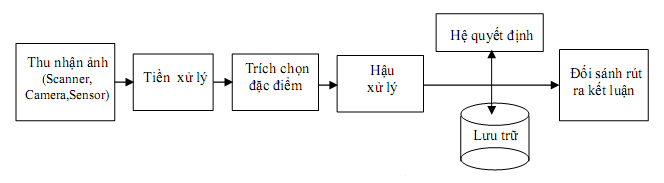
Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy.

Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1.5. *Quá trình xử lý ảnh.*

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P(c1, c2,…,cn). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

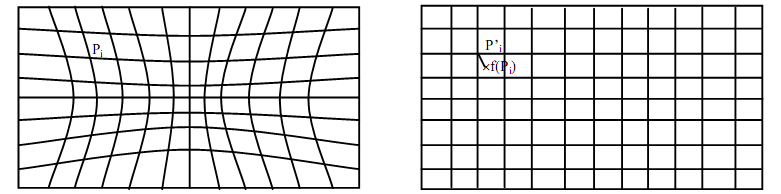


Hình 1.6. *Các bước cơ bản trong một hệ thống xử lý ảnh.*

#### **b. Các vấn đề cơ bản trong xử lý ảnh**

* Một số khái niệm cơ bản
* Ảnh và điểm ảnh: Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại một tọa độ trong không gian của đối tượng và ảnh được xem như là một tập hợp các điểm ảnh.
* Mức xám, màu: Là số các giá trị có thể có của các điểm ảnh của ảnh.
* Nắn chỉnh biến dạng

Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử.



Ảnh thu nhận Ảnh mong muốn

Hình 1.7. *Ảnh thu nhận và ảnh mong muốn.*

Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển.

Giả sử (Pi, Pi’) i =  có n các tập điều khiển.

Tìm hàm f: Pi 🡪 f (Pi) sao cho 

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm f có dạng:



Ta có:

Để cho 



Giải hệ phương trình tuyến tính tìm được a1, b1, c1

Tương tự tìm được a2, b2, c2.

⇒Xác định được hàm f

* Khử nhiễu

Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh

* Nhiễu hệ thống: là nhiễu có quy luật có thể khử bằng các phép biến đổi.
* Nhiễu ngẫu nhiên: vết bẩn không rõ nguyên nhân → khắc phục bằng các phép lọc.
* Chỉnh mức xám

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông thường có hai hướng tiếp cận:

* Giảm số mức xám: Thực hiện bằng tách nhóm các mức xám gần nhau thành một bó. Trường hợp chỉ có hai mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng: In ảnh màu ra máy in đen trắng.
* Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh.
* Trích chọn đặc điểm

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

* Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn…
* Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn...)
* Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích trọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử la bàn, toán tử “chéo không” (zero crossing) v.v…

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

* Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanable, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn, mẫu là một thực thể, được xác định một cách gần đúng và có thể gán cho nó một tên gọi nào đó”. Ví dụ mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một tín hiệu tiếng nói. Khi biết một mẫu nào đó, có thể nhận dạng hoặc phân loại mẫu đó:

Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analyis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.

Hoặc phân loại không có mẫu (unsupervised classfication hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau đây:

* Thu nhận dữ liệu và tiền xử lý.
* Biểu diễn dữ liệu.
* Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng là:

1. Đối chiếu, so sánh mẫu dựa trên các đặc trưng được trích chọn.
2. Phân loại thống kê.
3. Đối chiếu, so sánh cấu trúc.
4. Phân loại dựa trên mạng noron nhân tạo.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

**Chương 2:Phần mềm Pycharm và các thư viện đi kèm**

***2.1.Giới thiệu ngôn ngữ Python và phần mềm Pycharm***

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Pycharm là một nền tảng kết kết hợp được JetBrains phát triển như một IDE (Môi trường phát triển tích hợp) để phát triển các ứng dụng cho[**lập trình**](http://t3h.edu.vn/)trong Python. Một số ứng dụng lớn như Tweeter, Facebook, Amazon và Pinterest sử dụng Pycharm để làm IDE Python của họ

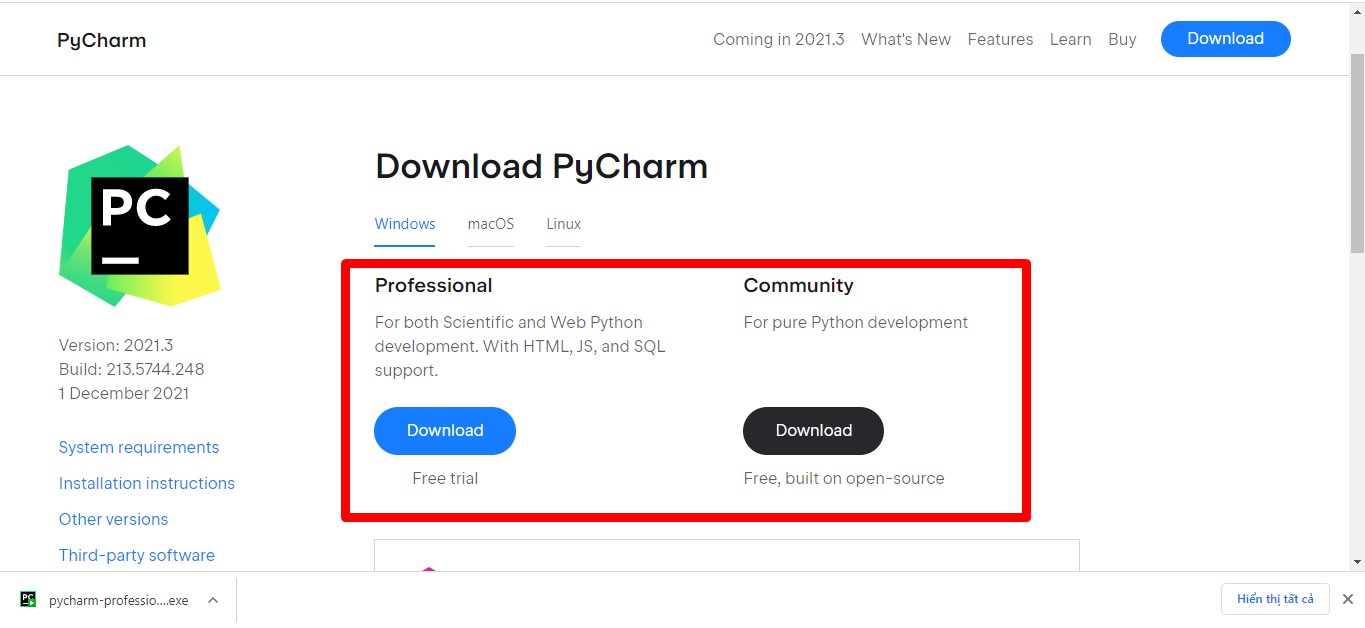
Cài đặt phần mềm Pycharm

**Bước 1:** Truy cập vào trang chủ của Pycharm tại https://www.jetbrains.com/pycharm/ và **nhấn Download** để tải phần mềm về máy tính.

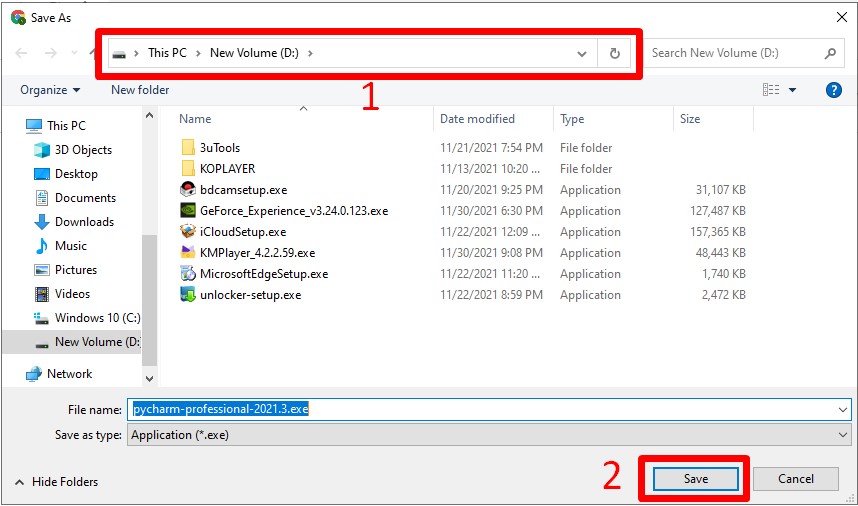


Lúc này, một cửa sổ mới xuất hiện hiển thị 2 lựa chọn để bạn tải về.

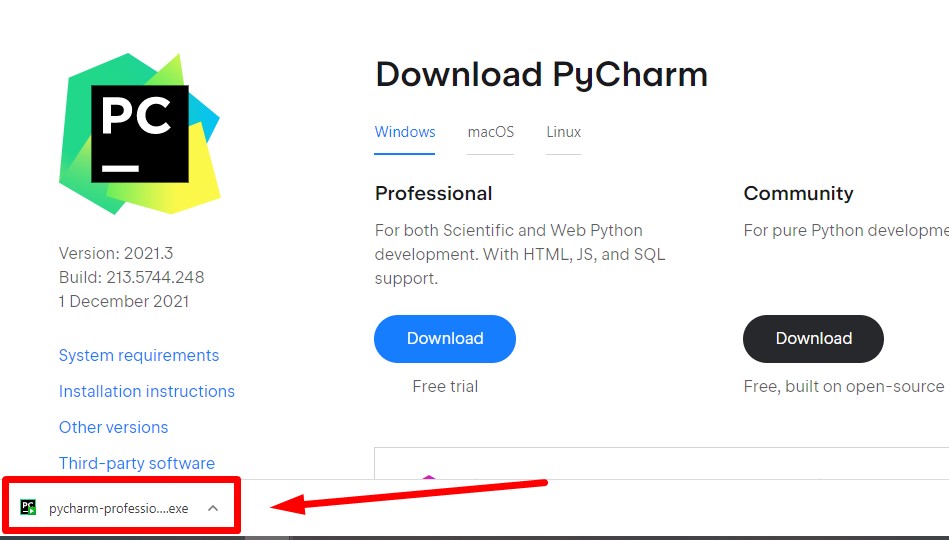
* **Professional:** Tải phiên bản chuyên nghiệp có thể dùng để phát triển Khoa học và Web Python. Được hỗ trợ HTML, JS và SQL. Bạn có thể tải bản dùng thử miễn phí trước khi trải nghiệm phiên bản trả phí bản quyền.
* **Community:** Dùng để phát triển Python thuần túy, bản dùng miễn phí.



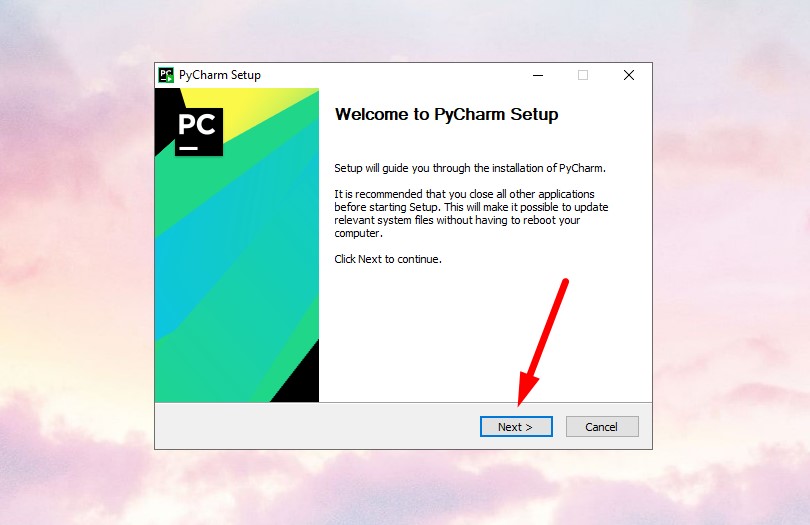
**Bước 2:** Bấm chọn thư mục lưu trữ sau khi tải về và **bấm Save** để tải.



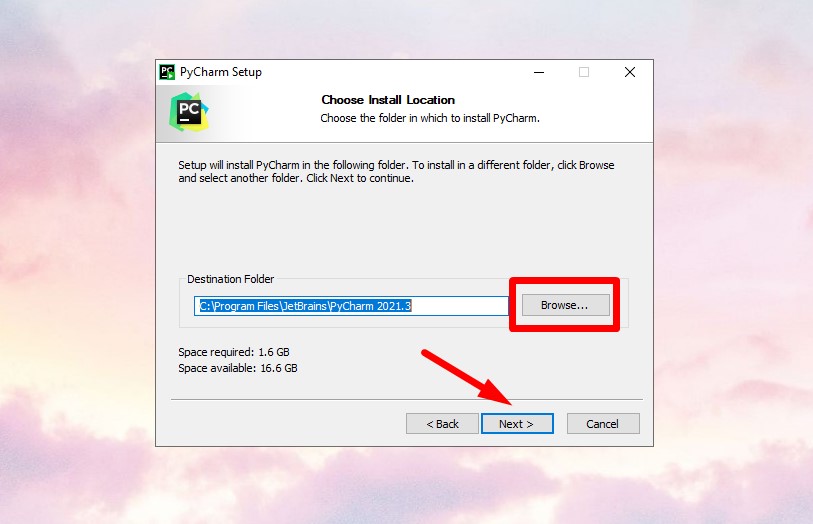
**Bước 3:** Sau khi phần mềm được tải xong, **ấn file .exe** ở góc trái [màn hình](https://cellphones.com.vn/man-hinh.html) để bắt đầu cài đặt.



**Bước 4:** Giao diện cài đặt xuất hiện >**Nhấn Next** để tiếp tục.



**Bước 5:** **Chọn Browse** nếu muốn thay đổi thư mục lưu trữ sau khi cài đặt > Sau đó,**click Next** để chuyển sang bước kế tiếp.



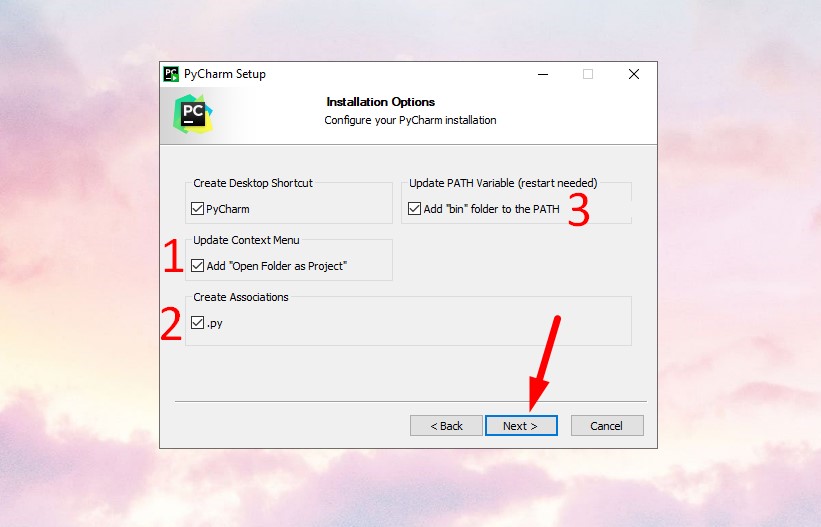
**Bước 6:** Tiếp theo, tích vào các tùy chọn cấu hình khi cài đặt.

1. Mở folder như là một Project.

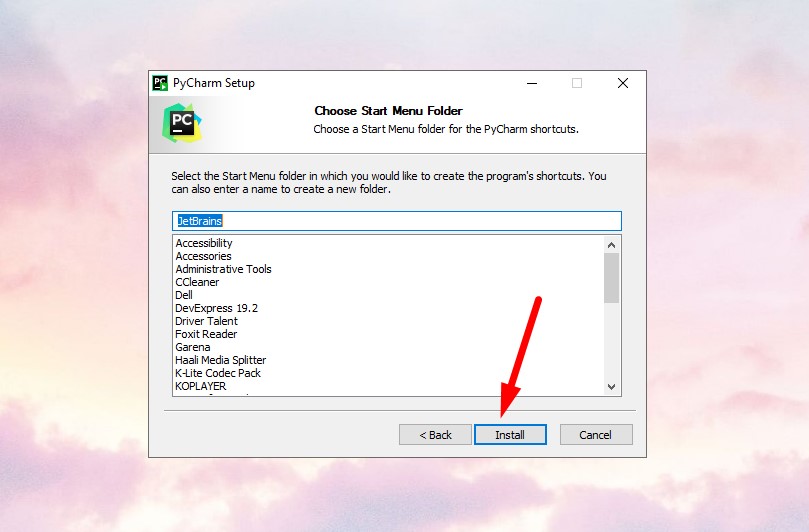
2. Tạo Associations

3. Thêm và cập nhật các biến môi trường

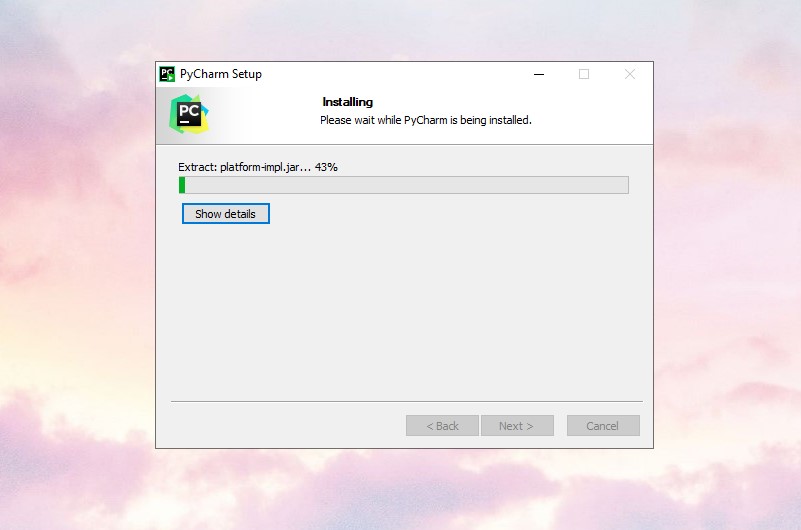
Bạn **tích vào ô Pycharm** để tạo Shortcut tại màn hình chính > **Bấm Next** để tiếp tục.



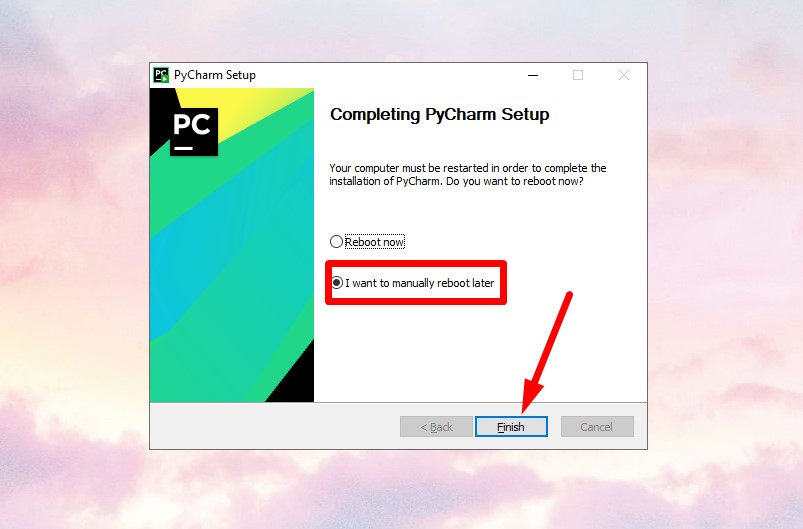
**Bước 7: Nhấn Install** để bắt đầu quá trình cài đặt.



**Bước 8:** Đợi quá trình cài đặt diễn ra khoảng 1 phút, tùy thuộc vào cấu hình máy của mỗi người.



**Bước 9:** Sau khi cài đặt xong, tích vào ô **Reboot now** để khởi động lại ngay hoặc tích vào ô **I want to manually reboot later** nếu bạn muốn khởi động lại sau > Chọn xong**, nhấn Finish** để hoàn tất quá trình cài đặt.



***2.2.Thư viện OpenCV***

OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

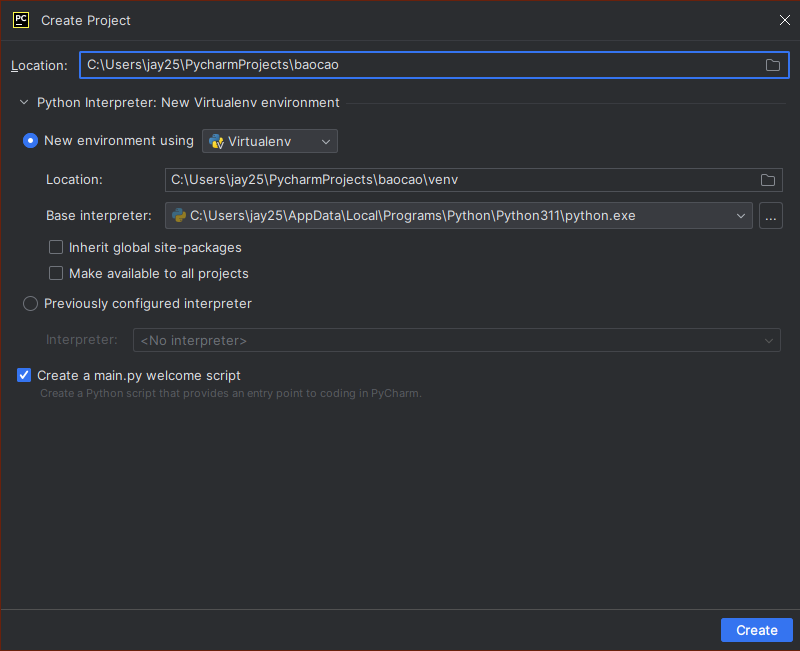
OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác.

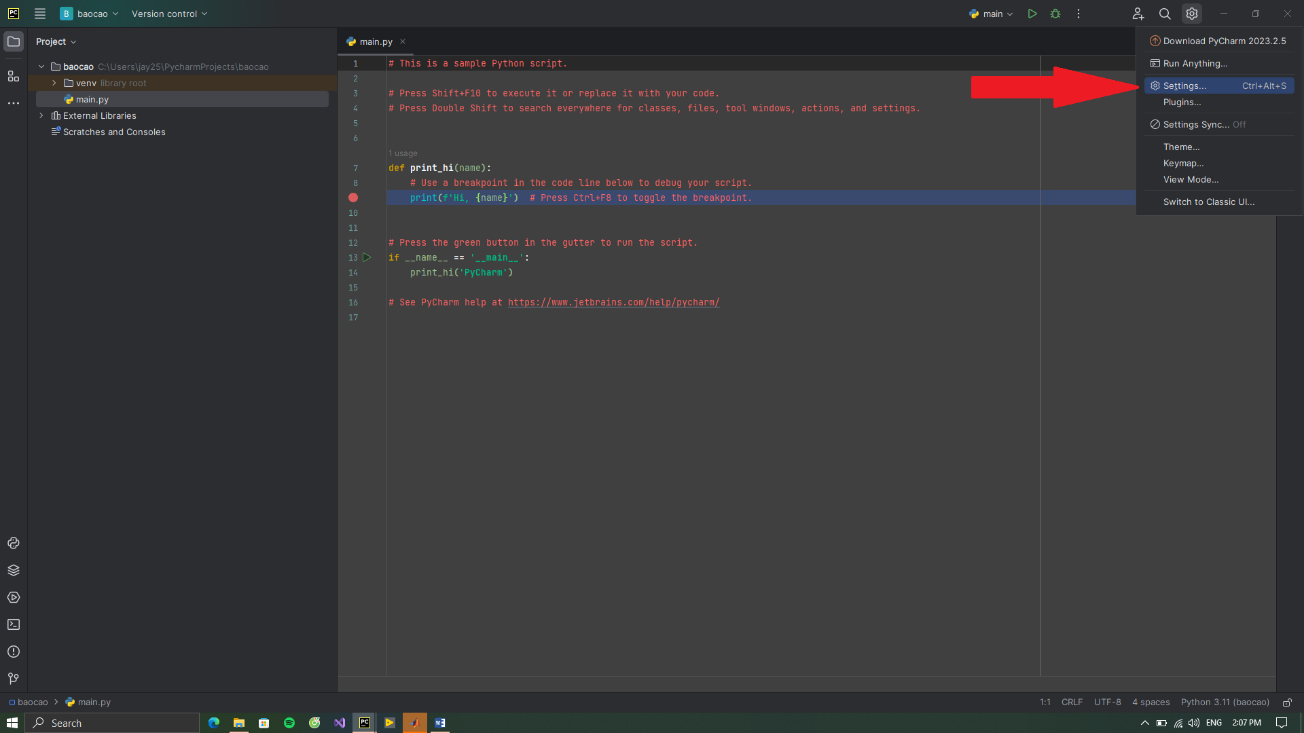
OpenCV được cho là một phần mềm đa nhiệm. Nó được ứng dụng trong rất nhiều trường hợp khác nhau. Ví dụ, ta sẽ nói về các phần mềm định vị, bản đồ nói chung. Hẳn rằng trong chúng ta ai cũng đã có ít nhất một lần cần sử dụng đến các map online đúng không. Bạn sử dụng các map để tìm đường, tra cứu tình hình giao thông hoặc đơn giản là xem xét các hình ảnh thực tế của địa điểm cần đến. Những lúc như vậy, OpenCV đóng vai trò là nhà cung cấp dữ liệu hình ảnh cho các app về Map này. OpenCV sẽ đem đến cho người dùng hình ảnh về đường phố hay các căn nhà, con người xung quanh địa điểm được chỉ định.

OpenCV còn được dùng để khởi tạo ra những hình ảnh 3 chiều phức tạp. Hoạt động này rất được yêu thích, nhất là trong thời đại trí tuệ nhân tạo AI phát triển như thế này.

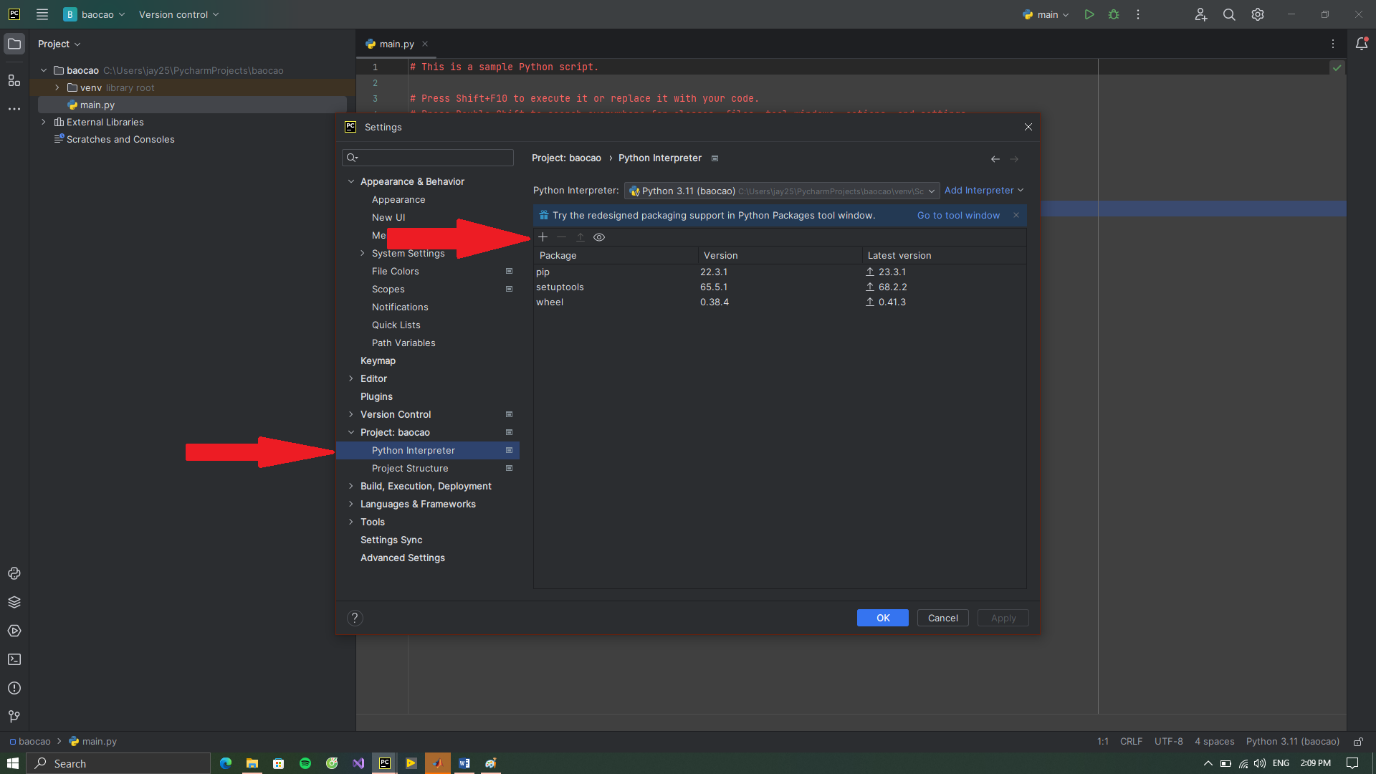
Đối với các công nghệ hiện đại, OpenCV cũng là một yếu tố không thể thiếu. Tất cả những ứng dụng công nghệ như robot, xe tự lái, bảng cảm ứng thông minh… đều có sự góp mặt của OpenCV trong khâu xử lý hình ảnh. Ví dụ gần gũi nhất trong cuộc sống có thể kể đến hệ thống mở khóa điện thoại bằng cách nhận diện khuôn mặt người dung

Cài đặt Thư viện Opencv trên Pycharm:

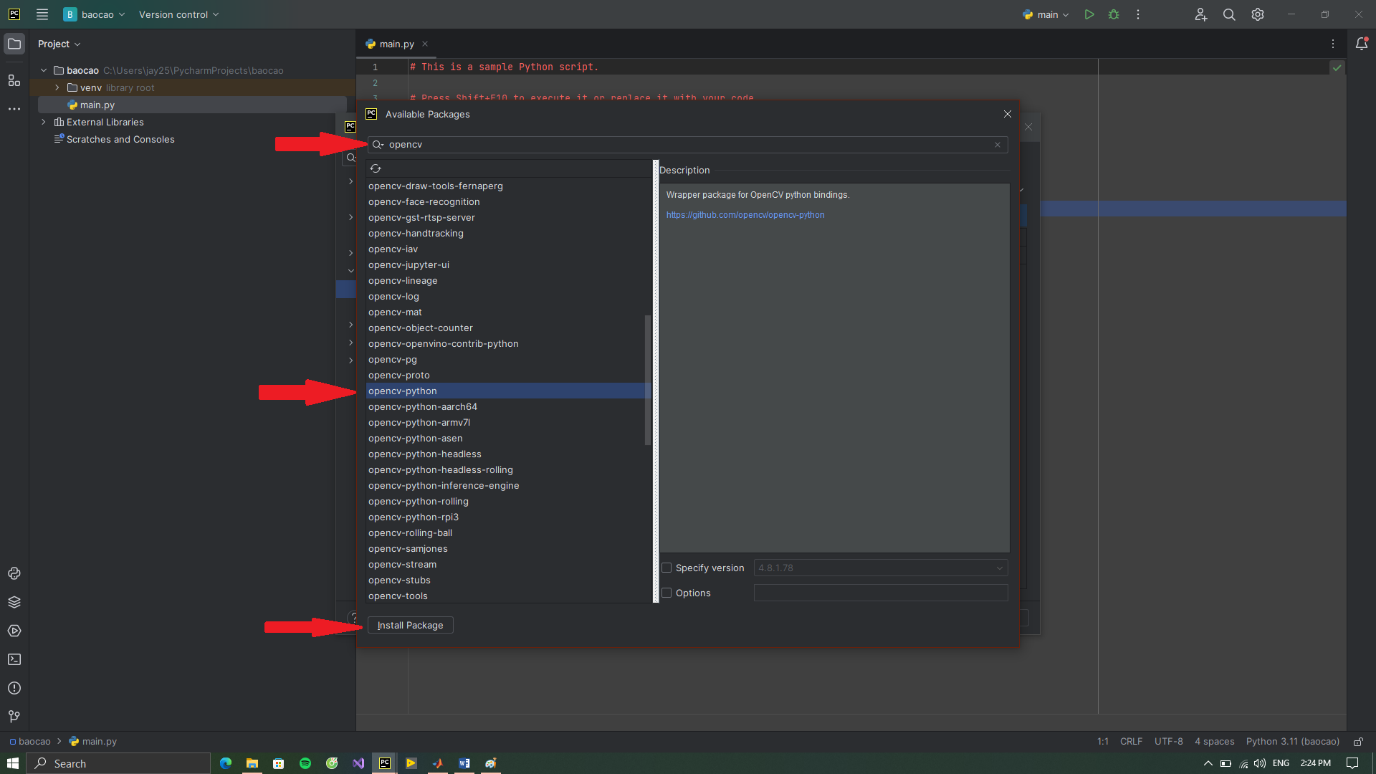
***Bước 1:Ấn create để tạo 1 project mới trên Pycharm***



***Bước 2:Sau khi vào project ấn vào setting trên góc phải màn hình***

****

***Bước 3:Ấn vào Python Interpreter rồi ấn vào dấu cộng***

******

***Bước 4:Tìm kiếm OpenCV rồi chọn opencv-python rồi chọn install package***

***2.3.Thư viện Tkinter***

Tkinter là thư viện GUI tiêu chuẩn cho Python. Khi kết hợp với Tkinter, Python sẽ được cung cấp các công cụ một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng mạnh mẽ đến các bộ công cụ Tk GUI.

Việc tạo một ứng dụng sử dụng Tkinter là một công việc vô cùng đơn giản. Tất cả những gì bạn cần làm là làm theo các bước sau:

- Nhập mô đun Tkinter

- Tạo cửa sổ ứng dụng chính của GUI

- Thêm một vài widgets vào ứng dụng GUI

- Nhập vòng lặp event chính để thực hiện hành động với từng sự kiện do người dùng kích hoạt

Ví dụ:

 #!/usr/bin/python

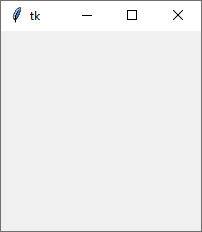
import Tkinter

top = Tkinter.Tk()

# Code to add widgets will go here...

top.mainloop()

Điều này sẽ tạo ra một cửa sổ sau:



**Chương 3:Trình bày và thực hiện thuật toán chương trình**

***3.1.Sơ đồ thuật toán***

***3.2.Ý tưởng và hình ảnh đạt được***

***3.2.1 Ý tưởng:***

**-**Bật camera và lấy ảnh từ camera rồi lưu dưới tên “Plate.png” rồi chuyển đổi ảnh thành ảnh xám

**-**Sử dụng phương pháp tách ngưỡng trong bài này ta kết hợp 2 phương pháp đó chính là:THRESH\_BINARY và THRESH\_OTSU

\_**,** thresh = cv2.threshold(imggray**, 200, 255,** cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

+THRESH\_BINARY:Ngưỡng nhị phân với ảnh đầu vào là imggray(với điều kiện imggray phải là ảnh xám),các giá trị điểm ảnh có giá trị lớn hơn hoặc bằng 200 sẽ gán giá trị các điểm ảnh đó là 255 còn nhỏ hơn thì mặc địch sẽ thành 0

+THRESH\_OTSU:Sử dụng thuật toán OTSU để tìm ngưỡng cố định của ảnh dựa vào tần suất xuất hiện của các điểm ảnh

-Với câu lệnh trên chúng ta có thể lấy ảnh tiền xử lý dưới tên ***thresh***

-Sau khi xử lý ảnh bằng phương pháp tách ngưỡng,ta tìm biên(các vùng điểm ảnh kết nối với nhau) rồi sắp xếp lại các diện tích đường biên và loại bỏ các đường biên có diện tích nhỏ hơn 30.Điều này giúp cho vùng nhận dạng trở nên rõ ràng hơn và loại bỏ các đường biên ảnh hưởng đến biển số

contours**,** \_ = cv2.findContours(thresh**,** cv2.RETR\_TREE**,** cv2.CHAIN\_APPROX\_SIMPLE)  
cv2.drawContours(phanvung1**,** contours**,** -**1,** (**255, 0, 0,**)**, 2**)  
contours = sorted(contours**,** key=cv2.contourArea**,** reverse=True)[:**30**]

-Vẽ các đường biên sau khi phát hiện được.Lấy ra ảnh đã phát hiện biên

-Ta sử dụng phương pháp xấp xỉ đường biên.Với mỗi đường biên xấp xỉ là 1 hình đa giác lồi,vì biển số là 1 hình đa giác có 4 cạnh nên ta có

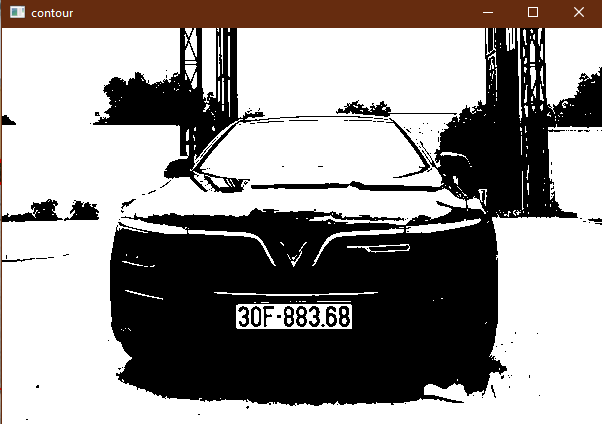
for c in contours:  
 perimeter = cv2.arcLength(c**,** True)  
 approx = cv2.approxPolyDP(c**, 0.018** \* perimeter**,** True)  
 if len(approx) == **4**:  
 screencontour = approx  
 x**,** y**,** w**,** h = cv2.boundingRect(c)  
 break

-Với mỗi đa giác lồi có 4 cạnh ta tìm các Boundingbox,Boundingbox là một đường biên đóng bao quanh vật thể,với câu lệnh boundingRect ta tìm ra các hình chữ nhật đứng bao quanh các đa giác lồi có 4 cạnh(có khả năng là biển số)

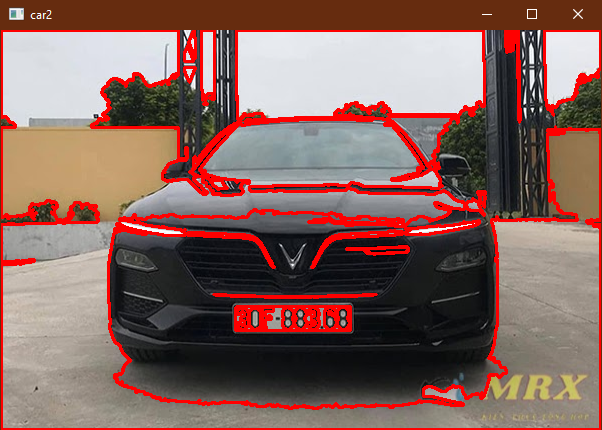
-Sau khi đã xác định được vùng có khả năng là biển số ta cắt vùng có biển số ra và hiển thị

cv2.rectangle(result**,** (x**,** y)**,** (x + w**,** y + h)**,** (**0, 255, 0**)**, 3**)  
crop\_img = img[y:y + h**,** x:x + w]

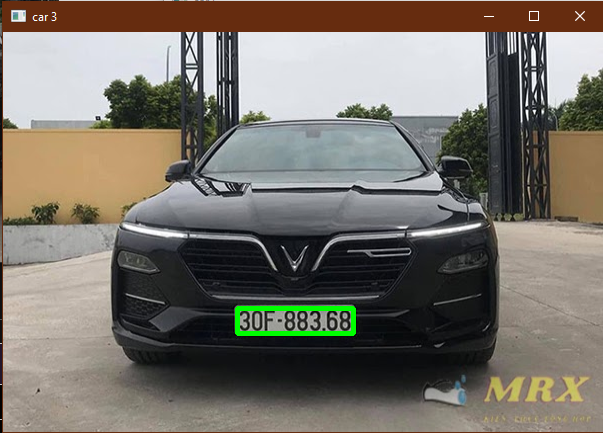
***3.2.2 Hình ảnh đạt được***



*Ảnh tiền xử lý sử dụng phương pháp tách ngưỡng*



*Ảnh phát hiện biên sau khi sắp xếp diện tích các đường biên*



*Ảnh phát hiện vùng có biển số xe sau khi dùng phương pháp xấp xỉ đường biên*



*Biển số sau khi được cắt ra*

***3.3.Giao diện chương trình và hình ảnh***

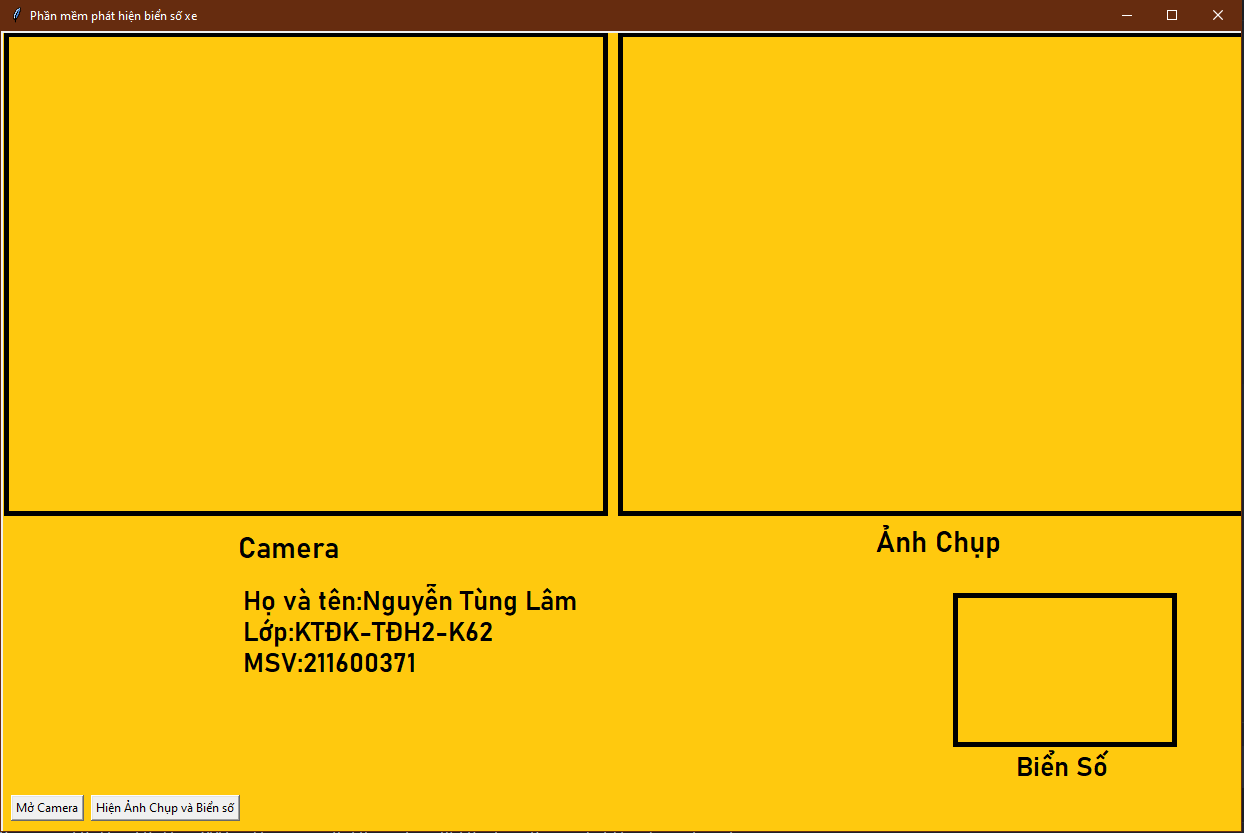
- Cửa sổ chương trình bao gồm 2 nút:

+Nút 1(Button1):Để mở camera

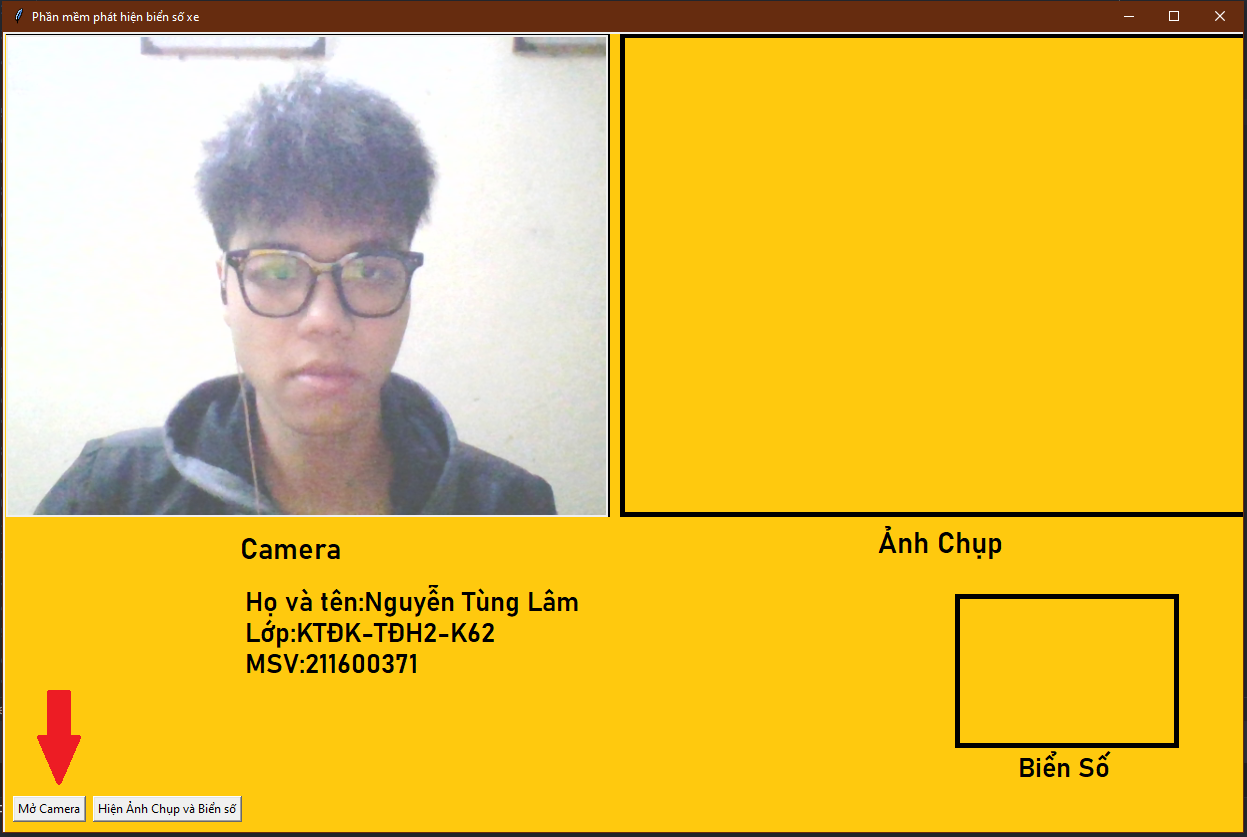
+Nút 2(Button2):Hiện ảnh đã chụp và hiện biển số lên khung trong cửa sổ giao diện

-Đầu tiên nhấn nút “Mở camera”,giơ ảnh có chứa biển số

-Nhấn nút “Hiện ảnh đã chụp và biển số”



*Giao diện chương trình*

**

*Ấn nút “Mở camera” để mở camera trước của laptop*

**

*Ấn nút “Hiện Ảnh chụp và Biển số” để hiện Ảnh chụp và Biển số*

**Phần III:Kết Luận**

**1.Kết quả đạt được**

**-**Đã tách được biển số từ hình ảnh chiếu trên camera

**-**Phần mềm chạy ổn định

**2.Một số hạn chế**

-Chưa tách được các kí tự trên biển số để hiện lên cửa sổ giao diện

-Hình ảnh cần phải có đủ ánh sáng và độ nét thì mới có thể tách được biển số

**3.Định hướng đề tài trong tương lai**

**-**Tách kí tự từ biển số lên cửa sổ giao diện

**-**Kết hợp với các loại vi xử lý và thẻ từ rfid để tự động chụp và nhận dạng hạn chế việc bấm nút trên cửa sổ

**4.Chương trình**

*import* tkinter  
*import* cv2  
*from* tkinter *import* \*  
*from* PIL *import* Image  
*from* PIL *import* ImageTk  
root= Tk()  
cap=cv2.VideoCapture(0,cv2.CAP\_DSHOW)  
cap.set(600,800)  
root.title("Phần mềm phát hiện biển số xe")  
root.geometry("1240x800")  
background=PhotoImage(file="Anh/background.png")  
background1=Label(root,image=background)  
background1.place(x=0,y=0)  
*def* opencamera():  
 *global* A,imgcv  
 \_,frame=cap.read()  
 frame=cv2.resize(frame,(598,478))  
 imgcv=cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)  
 img=Image.fromarray(imgcv)  
 imgtk=ImageTk.PhotoImage(image=img)  
 A=Label(image=imgtk)  
 A.image=imgtk  
 A.place(x=3,y=3)  
 A.after(10,opencamera)  
*def* Hienanhdachupvabienso():  
 *global* B,C  
 cv2.imwrite(f"Anh/plate.png", imgcv)  
 anh1=cv2.imread("Anh/plate.png")  
 anh1=cv2.resize(anh1,(620,477))  
 imggray = cv2.cvtColor(anh1, cv2.COLOR\_BGRA2GRAY)  
 \_, thresh = cv2.threshold(imggray, 200, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)  
 contours, \_ = cv2.findContours(thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  
 contours = sorted(contours, key=cv2.contourArea, reverse=*True*)[:30]  
 *for* c *in* contours:  
 perimeter = cv2.arcLength(c, *True*)  
 approx = cv2.approxPolyDP(c, 0.018 \* perimeter, *True*)  
 *if* len(approx) == 4:  
 x, y, w, h = cv2.boundingRect(c)  
 *break* cv2.rectangle(anh1, (x, y), (x + w, y + h), (0, 255, 0), 3)  
 crop\_img = anh1[y:y + h, x:x + w]  
 crop\_img=cv2.resize(crop\_img,(210,148))  
 crop\_img = cv2.cvtColor(crop\_img, cv2.COLOR\_BGR2RGB)  
 img1 = Image.fromarray(anh1)  
 imgtk1 = ImageTk.PhotoImage(image=img1)  
 img2 = Image.fromarray(crop\_img)  
 imgtk2 = ImageTk.PhotoImage(image=img2)  
  
 B= Label(image=imgtk1)  
 B.image = imgtk1  
 B.place(x=620, y=3)  
  
 C= Label(image=imgtk2)  
 C.image = imgtk2  
 C.place(x=955, y=565)  
  
button1=Button(root,text="Mở Camera",command=opencamera)  
button1.place(x=10,y=764)  
button2=Button(root,text="Hiện Ảnh Chụp và Biển số",command=Hienanhdachupvabienso)  
button2.place(x=90,y=764)  
root.mainloop()