

**CÔNG TY TNHH CÔNG NGHỆ A.I**

**HƯỚNG DẪN SỬ DỤNG**  
**AI.ANPR V5 SDK**

# MỤC LỤC

<b>1. Giới thiệu.....</b>	<b>1</b>
<b>2. Các lưu ý quan trọng.....</b>	<b>2</b>
2.1 Điểm ảnh (pixel):.....	2
2.2 Vùng nhận dạng:.....	2
2.3 Camera nhận dạng biển số chuyên dụng.....	3
2.4 Đường link RTSP của camera.....	4
2.5 Tài nguyên máy tính khi giải mã video từ camera.....	5
2.6 Mã hóa Base64.....	5
<b>3. Cài đặt.....</b>	<b>6</b>
<b>4. License.....</b>	<b>8</b>
4.1 License.....	8
4.2 Activate.....	8
<b>5. Tích hợp AIANPR SDK vào phần mềm.....</b>	<b>10</b>
5.1 Quy trình khởi tạo.....	11
5.2 Quy trình kết nối camera IP.....	13
5.3 Quy trình xử lý nhận dạng biển số xe.....	14
5.4 Quy trình xử lý hình ảnh.....	15
<b>6. Các “Phương thức” sử dụng.....</b>	<b>15</b>
6.1 setDifficultyLevel.....	15
6.2 initEvent.....	16
6.3 setIndexRtspUrl.....	16
6.4 setIndexControlDisplay.....	16
6.5 setDisplay.....	17
6.6 play.....	17
6.7 setCameraEventMethod.....	17
6.8 cameraCallback.....	17
6.9 setAnprEventMethod.....	18
6.10 anprCallback.....	18
6.11 setAnprContinues.....	18
6.12 setAreaAnpr.....	19
6.13 setAreaDraw.....	19
6.14 setHeightChar.....	20
6.15 setRepeatPlateChecktime.....	20
6.16 getNumberPlate.....	20
6.17 getNumberPlateWithArea.....	21
6.18 saveBitmap.....	21
6.19 saveJpg.....	21
6.20 displayImage.....	22
6.21 loadImageToImgBase64.....	22
6.22 getPlayStatus.....	22
6.23 getAnprStatus.....	23
6.24 getImageBase64.....	23
6.25 getNumberPlateFromCamera.....	23
6.26 Dispose.....	24
<b>7. Thông tin công ty TNHH Công nghệ A.I.....</b>	<b>24</b>

## 1. Giới thiệu

AI.ANPR là một SDK về nhận dạng biển số xe ô tô và xe máy, là một công cụ hiệu quả giúp tách ký tự biển số xe từ hình ảnh hoặc từ luồng dữ liệu RTSP (dữ liệu camera), được thiết kế để nhà phát triển có thể tích hợp cho các dự án như trạm thu phí, trạm cân, bãi giữ xe...

Sử dụng các giải thuật và mô hình học máy trong trí tuệ nhân tạo, AI.ANPR giúp tối đa hóa tính năng tính toán và cải thiện hiệu suất xử lý, đặc biệt là với phiên bản mới nhất - phiên bản V5. Chức năng chính của Module bao gồm:

- Kết nối camera IP hiển thị tín hiệu video lên control của phần mềm chính.
- Chụp ảnh xe từ camera IP, tải hình ảnh từ đĩa, lưu hình ảnh.
- Tách biển số thành dạng ký tự.

Phiên bản V5 đã được phát triển để đáp ứng yêu cầu của khách hàng trong các dự án đòi hỏi tốc độ xử lý cao, như xe chạy với tốc độ trung bình dưới 120km/h. AI.ANPR V5 được chia thành ba phiên bản chính để phù hợp với các yêu cầu khác nhau của khách hàng:

- Phiên bản đầu tiên chỉ sử dụng CPU
- Phiên bản thứ hai sử dụng GPU Intel từ thế hệ 6 trở lên.
- Phiên bản thứ ba sử dụng GPU Nvidia hỗ trợ "Compute Capability" phiên bản 6.1 trở lên.

Với mỗi phiên bản, chúng tôi đã tối ưu hóa tính năng tính toán để đạt được hiệu suất tối đa.

AI.ANPR V5 chỉ hỗ trợ ứng dụng 64 bit để đảm bảo sự tương thích với các thế hệ CPU mới nhất. Để đảm bảo tương thích máy tính xử lý phải cài sẵn trình điều khiển mới nhất và cài đặt sẵn các thành phần cần thiết như NET framework 4.8 và C++ 2015-2019.

AI.ANPR V5 hỗ trợ đa dạng các ngôn ngữ lập trình như c#, vb.net, c++, powerbuilder, java, python ... giúp dễ dàng tích hợp vào các ứng dụng khác.

AI.ANPR SDK được thiết kế để tích hợp vào ứng dụng của nhà phát triển, nhằm cung cấp cho ứng dụng khả năng nhận dạng và trích xuất thông tin từ các hình ảnh chứa biển số xe.

Tuy nhiên, SDK này không được thiết kế để sử dụng cho người dùng cuối, mà chỉ dành cho các nhà phát triển và các doanh nghiệp muốn tích hợp tính năng này vào ứng dụng của mình. Nếu có yêu cầu đặc biệt khác, như yêu cầu tùy chỉnh hoặc tích hợp vào phần mềm sẵn có, người dùng có thể liên hệ trực tiếp với chúng tôi để được tư vấn và hỗ trợ

## 2. Các lưu ý quan trọng

### 2.1 Điểm ảnh (pixel):

Pixel (viết tắt của picture element) là đơn vị nhỏ nhất của hình ảnh kỹ thuật số, chúng là các điểm ảnh nhỏ nhất được sắp xếp theo một mạng lưới trên màn hình hoặc trên các thiết bị lưu trữ hình ảnh. Pixel có kích thước nhỏ hơn 1/1000 inch, và được sử dụng để tạo ra các hình ảnh số, bao gồm các bức ảnh, biểu đồ, logo, hoặc các hình ảnh khác.

Mỗi pixel được mô tả bằng một số, đại diện cho màu sắc và độ sáng của điểm ảnh đó. Số lượng pixel càng nhiều, hình ảnh sẽ càng chi tiết và sắc nét hơn.

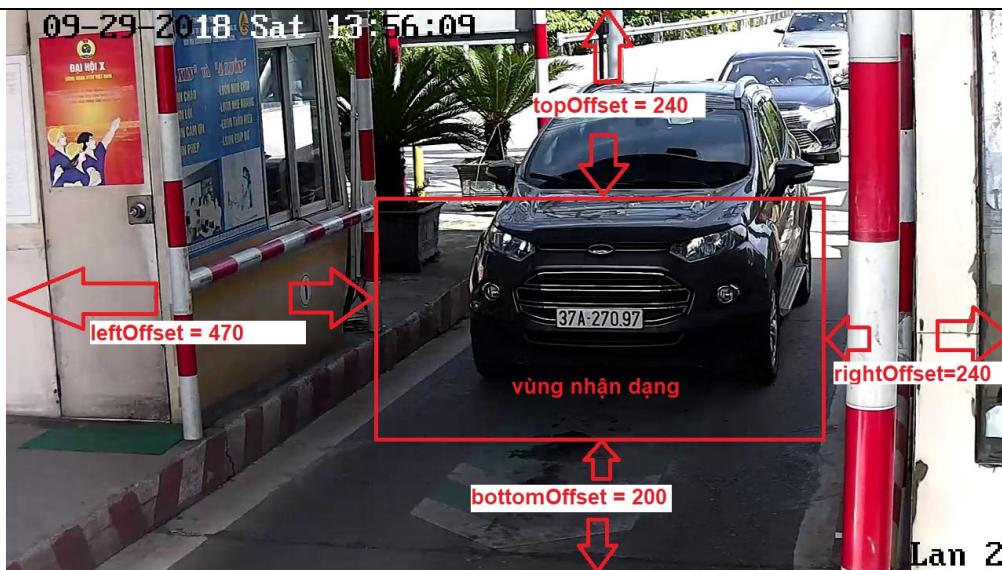
Ví dụ, một bức ảnh với độ phân giải 1024x768 có nghĩa là chiều rộng của nó là 1024 pixel và chiều cao là 768 pixel. Tổng số pixel của ảnh này là  $1024 \times 768 = 786,432$  pixel. Mỗi pixel trong bức ảnh sẽ được mô tả bằng một giá trị RGB (Red-Green-Blue) cho biết màu sắc của điểm ảnh đó..

Cùng kích thước vật lý, độ phân giải cao tức là số lượng pixel nhiều và kích thước từng pixel sẽ nhỏ hơn là độ phân giải thấp tức là có ít pixel hơn và chúng lại có kích thước lớn hơn.

Độ phân giải của hình ảnh càng lớn thì nhận dạng chậm và thiếu chính xác hơn, độ phân giải hình ảnh để **nhận dạng tốt nhất là 1280 x 720 (HD)** chiều cao của **mỗi ký tự trong biển số từ 25 đến 50 điểm ảnh.**

### 2.2 Vùng nhận dạng:

Vùng nhận dạng: là một vùng trên tấm ảnh có khả năng xuất hiện biển số xe, ngoài ra các vùng khác sẽ không bao giờ xuất hiện biển số xe. Khi đó để tăng tốc độ và tránh nhiễu trong khi nhận dạng biển số thì chúng ta khoanh vùng này khi nhận dạng, module có tham số để khoanh vùng nhận dạng này.



- topOffset: Số điểm ảnh loại bỏ không nhận dạng từ lề **trên** đến vùng nhận dạng
- bottomOffset: Số điểm ảnh loại bỏ không nhận dạng từ lề **dưới** đến vùng nhận dạng
- leftOffset: Số điểm ảnh loại bỏ không nhận dạng từ lề **trái** đến vùng nhận dạng
- rightOffset: Số điểm ảnh loại bỏ không nhận dạng từ lề **phải** đến vùng nhận dạng

### 2.3 Camera nhận dạng biển số chuyên dụng.

Để đảm bảo tính chính xác và tốc độ xử lý cao trong việc nhận dạng biển số xe, việc sử dụng các loại camera nhận dạng biển số xe chuyên dụng là cực kỳ quan trọng. Các loại camera này được thiết kế để làm nổi bật biển số và các ký tự trên biển số, giúp cho chương trình nhận dạng có thể hoạt động chính xác và nhanh chóng hơn rất nhiều. Bên cạnh đó, các camera nhận dạng biển số còn được trang bị các tính năng đặc biệt như hỗ trợ hồng ngoại, giảm nhiễu, tăng độ phân giải ảnh, giúp cho quá trình nhận dạng diễn ra thuận lợi hơn và đạt được độ chính xác cao nhất.

Tuy nhiên, việc sử dụng các loại camera này cũng đòi hỏi một chi phí đầu tư khá lớn. Để tiết kiệm chi phí, có thể sử dụng các camera IP thông thường để nhận dạng biển số xe. Tuy nhiên, việc sử dụng các camera IP thông thường sẽ ảnh hưởng đến tính chính xác và tốc độ xử lý của chương trình. Do đó, nếu điều kiện tài chính cho phép, nên sử dụng các loại camera nhận dạng biển số xe chuyên dụng để đảm bảo tính chính xác và tốc độ xử lý cao nhất trong việc nhận dạng biển số xe.

Một số hình ảnh của camera nhận dạng biển số như sau:



## 2.4 Đường link RTSP của camera

RTSP là viết tắt của Real-Time Streaming Protocol, một giao thức truyền phát video trực tiếp trên mạng. RTSP được sử dụng trong nhiều ứng dụng, bao gồm cả trong camera giám sát, phần lớn camera IP trên thị trường đều hỗ trợ giao thức RTSP.

Lấy đường link RTSP của camera bằng cách đọc tài liệu hướng dẫn sử dụng của nhà sản xuất hoặc sử dụng phần mềm ONVIF Device Manager.

Đường link RTSP tiêu chuẩn thường có định dạng như sau:

**rtsp://<username>:<password>@<camera\_IP\_address>:<port>/<stream\_name>**

Trong đó:

**<username>**: Tên đăng nhập vào camera.

**<password>**: Mật khẩu để đăng nhập vào camera.

**<camera\_IP>**:\_address>: Địa chỉ IP của camera trên mạng.

**<port>**: Cổng được sử dụng để truyền tải video.

**<stream\_name>**: Tên của luồng video được truyền tải từ camera.

Ví dụ, nếu tên đăng nhập của camera là "admin", mật khẩu là "12345", địa chỉ IP của camera là "192.168.1.10", cổng RTSP là 554 và tên của luồng video là "live", thì đường link RTSP sẽ có dạng như sau:

**rtsp://admin:12345@192.168.1.10:554/live**

## 2.5 Tài nguyên máy tính khi giải mã video từ camera

Các thông số của camera IP có thể ảnh hưởng đến tài nguyên (CPU, RAM, Băng thông,...) của máy tính khi giải mã video của camera, cụ thể như sau:

- Độ phân giải: Độ phân giải của video càng cao thì càng cần nhiều tài nguyên của máy tính để giải mã và xử lý. Nếu máy tính không đủ mạnh, thì video có thể bị giật hoặc chậm.
- Tốc độ khung hình (Frame rate): Nếu tốc độ khung hình cao, thì cần nhiều tài nguyên để xử lý.
- Chuẩn nén: Các định dạng nén video khác nhau có độ phức tạp khác nhau và yêu cầu nhiều hoặc ít tài nguyên để giải mã. Các chuẩn nén từ thấp đến cao cần tài nguyên của máy tính: MJPEG -> H264 -> H265
- Độ trễ: Nếu độ trễ cao, thì máy tính cần nhiều tài nguyên để giải mã và hiển thị video một cách liền mạch.

Các thông số của camera IP có thể ảnh hưởng đến tài nguyên của máy tính khi giải mã video của camera, do đó cần đánh giá kỹ các thông số này để chọn một máy tính có đủ tài nguyên để xử lý video của camera một cách mượt mà và ổn định.

Trong trường hợp không cần tốc độ cao thì có thể sử dụng cấu hình camera tiêu chuẩn sau:

- Độ phân giải: 1280x720
- Tốc độ khung hình: 6 - 12 frame.
- Chuẩn nén: H264

## 2.6 Mã hóa Base64.

Base64 là một phương thức mã hóa 2 chiều từ tập tin nhị phân (binary) sang chuỗi ký tự (string) và ngược lại, nó thường được sử dụng để mã hóa các tập tin đa phương tiện (hình ảnh, âm thanh, video,...). Ký tự 64 trong Base64 là đại diện cho 64 ký tự trong bảng mã ASCII. Base64 thường được sử dụng trong việc truyền tải email. Tuy nhiên, chúng tôi dùng Base64 để mã hóa hình ảnh phục vụ mục đích sử dụng dữ liệu hình ảnh giữa các ngôn ngữ lập trình được đơn giản nhất.

### 3. Cài đặt

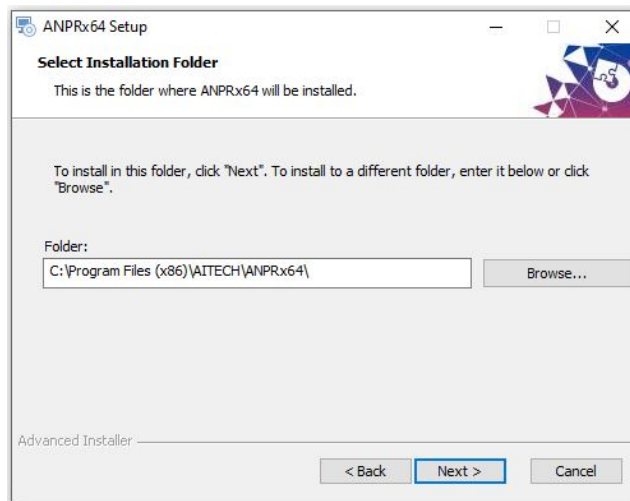
Chọn tệp AI.ANPRV5.exe nhấn ENTER hoặc đúp vào file để chạy cài đặt.

Màn hình xuất hiện :



Nhấn “Next” để tiếp tục cài đặt hoặc nhấn “Cancel” để hủy thao tác cài đặt.

Tiếp đến sẽ hiển thị 1 màn hình mới:

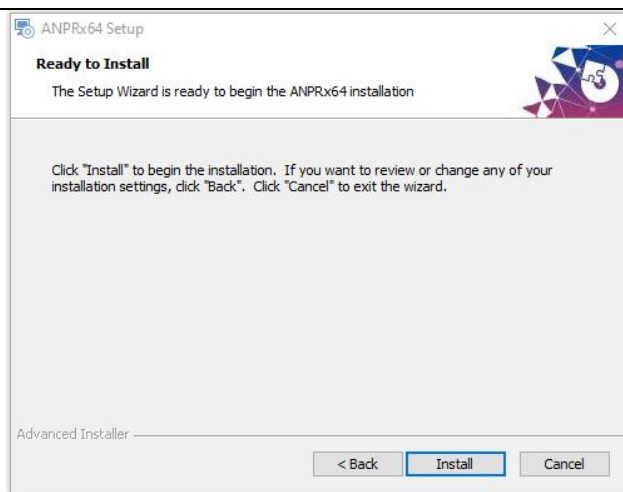


Folder: Chương trình mặc định cài trên thư mục “C:\Program Files (x86)\AITECH\ANPRx64V5” nếu bạn muốn thay đổi thư mục khác thì nhấn “Browse” chọn đến thư mục cần cài đặt.

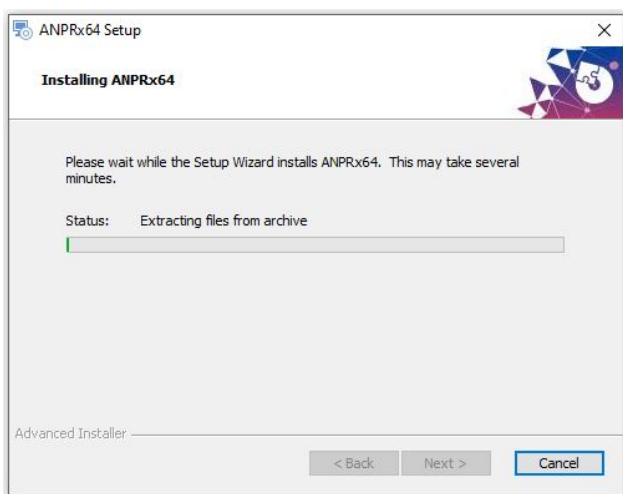
Nhấn nút “Next” để tiếp tục cài đặt.

Tiếp đến sẽ hiển thị màn hình chấp nhận cài đặt:



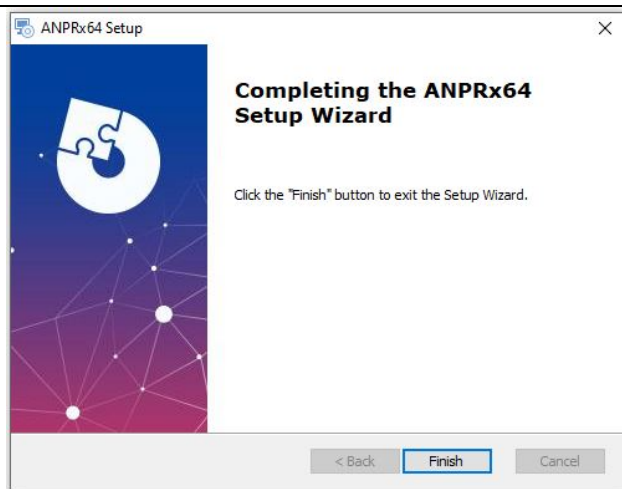


Nhấn “Install” để cài đặt.



Chương trình đang chạy tiến trình cài đặt.

Khi chạy xong nó sẽ hiện thông báo tiến trình cài đặt thành công như hình.



Nhấn “Finish” kết thúc tiến trình cài đặt.

## 4. License.

### 4.1 License

License chúng tôi cung cấp là: License phần mềm theo máy tính (còn gọi là "perpetual license") là một loại giấy phép mà khi người sử dụng mua một phần mềm thì sẽ được sử dụng phần mềm đó vĩnh viễn trên một máy tính cụ thể mà họ đã mua giấy phép.

Nói cách khác, người sử dụng không cần phải trả thêm phí để sử dụng phần mềm trên máy tính đó trong tương lai, trừ khi họ muốn nâng cấp phiên bản phần mềm hoặc mua một giấy phép khác để sử dụng trên một máy tính khác.

Hiện tại, chúng tôi cấp license trên website <https://aitech.com.vn/> hoặc có thể liên hệ trực tiếp để được tư vấn cụ thể hơn.

### 4.2 Activate.

#### - Bước 1:

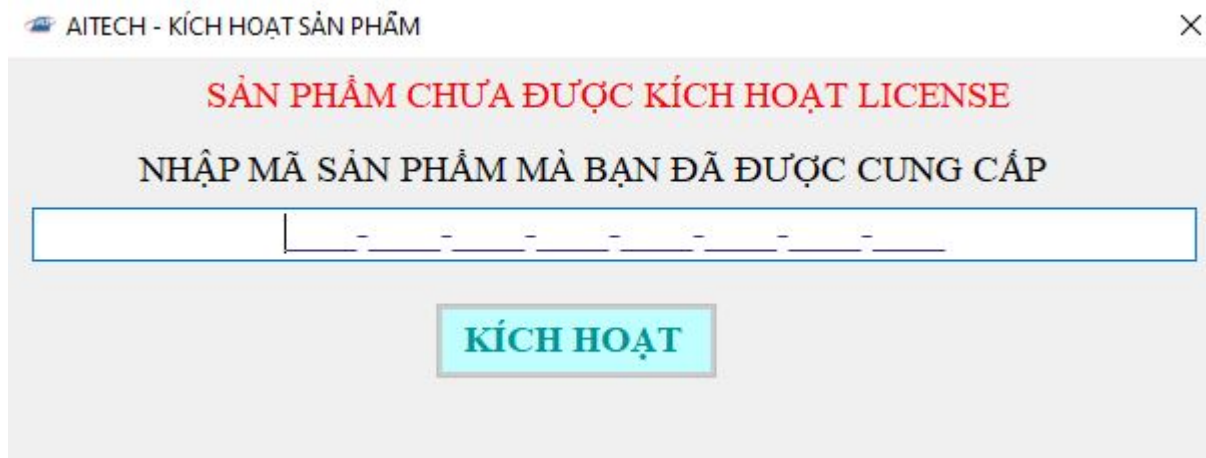
Chạy chương trình “**aiAnprActivate.exe**” trong thư mục cài đặt

**C:\Program Files (x86)\AITECH\ANPRx64V5\Bin\aiAnprActivate.exe**

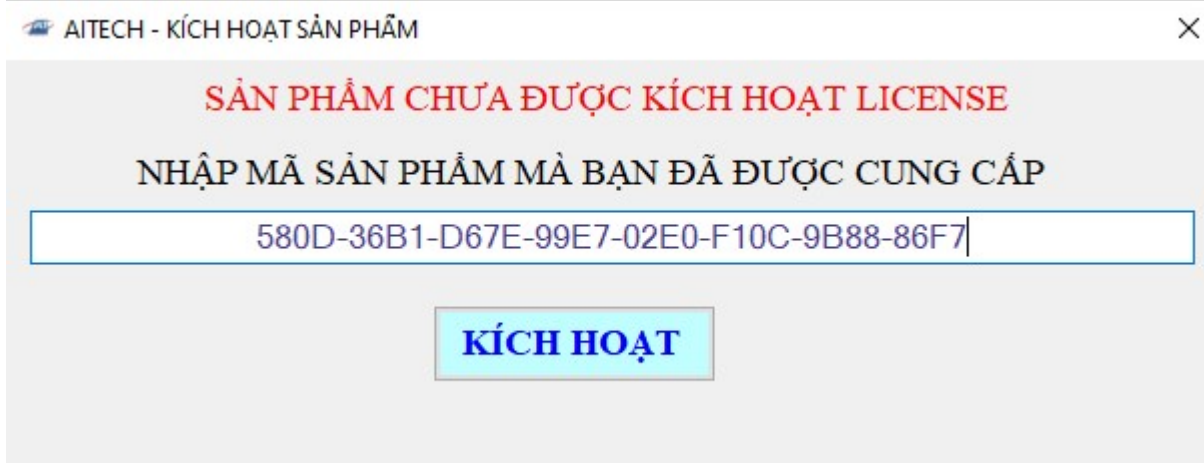
PC > Local Disk (C:) > Program Files (x86) > AITECH > ANPRx64V5 > Bin

Name	Date modified	Type	Size
decode	5/1/2023 10:45 AM	File folder	
x64	5/1/2023 10:45 AM	File folder	
ai.anpr.dll	5/1/2023 5:38 PM	Application exten...	350 KB
aiAccr.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiAnpr.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiAnprActivate.exe	4/30/2023 9:52 PM	Application	130 KB
aiAnprX64.dll	5/2/2023 11:15 PM	Application exten...	86 KB
aianprX64.tlb	4/30/2023 3:18 PM	TLB File	10 KB
aiCamera.dll	4/28/2023 3:17 PM	Application exten...	1,025 KB
aiCodec.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiCodedec.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiConnect.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiControl.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiDecode.dll	4/9/2023 5:53 PM	Application exten...	1,131 KB
aiDevice.dll	4/9/2023 5:53 PM	Application exten...	1,100 KB
aiForm.dll	4/9/2023 5:53 PM	Application exten...	1,116 KB
aiH264.dll	4/9/2023 5:53 PM	Application exten...	1,084 KB
aiH265.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiLicense.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiMjpeg.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiOnvif.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB

- **Bước 2:**



Nhập mã sản phẩm mà bạn được cung cấp.



- **Bước 3:**

“KÍCH HOẠT”



Chúng tôi có cung cấp license dùng thử trên website <https://aitech.com.vn/> , bạn có thể đăng ký dùng thử.

### 5. Tích hợp AI.ANPR SDK vào phần mềm

AI.ANPR SDK sau khi cài đặt là một tập hợp các giao diện lập trình ứng dụng API COM cho phép các ứng dụng Windows truy cập vào các tính năng của module để sử dụng.

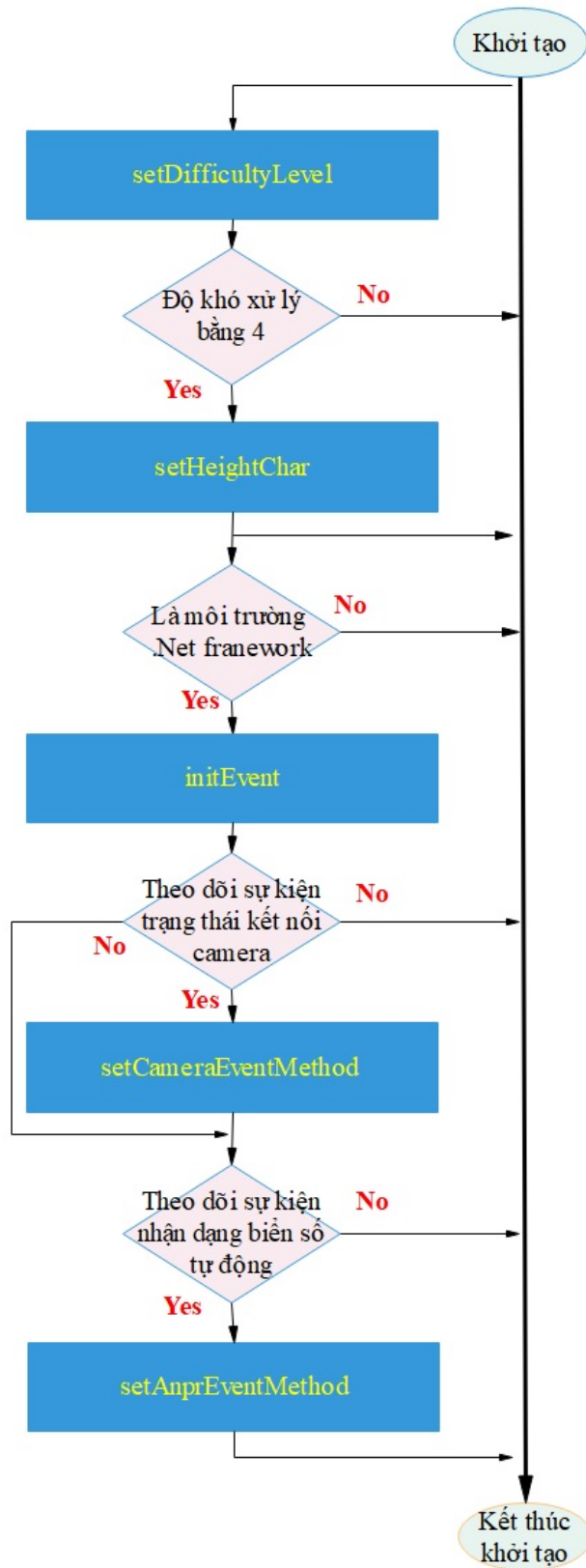
Sau khi cài đặt và kích hoạt license, chúng ta có thể dùng thử bằng phần mềm chạy thử với hình ảnh và đường link RTSP cung cấp sẵn để sử dụng trong quá trình tích hợp.

Module được cài đặt kèm theo mã nguồn tích hợp của một số ngôn ngữ lập trình thông dụng tại thư mục cài đặt: **C:\Program Files (x86)\AITECH\AIANPRV5\DemoSourceCode**

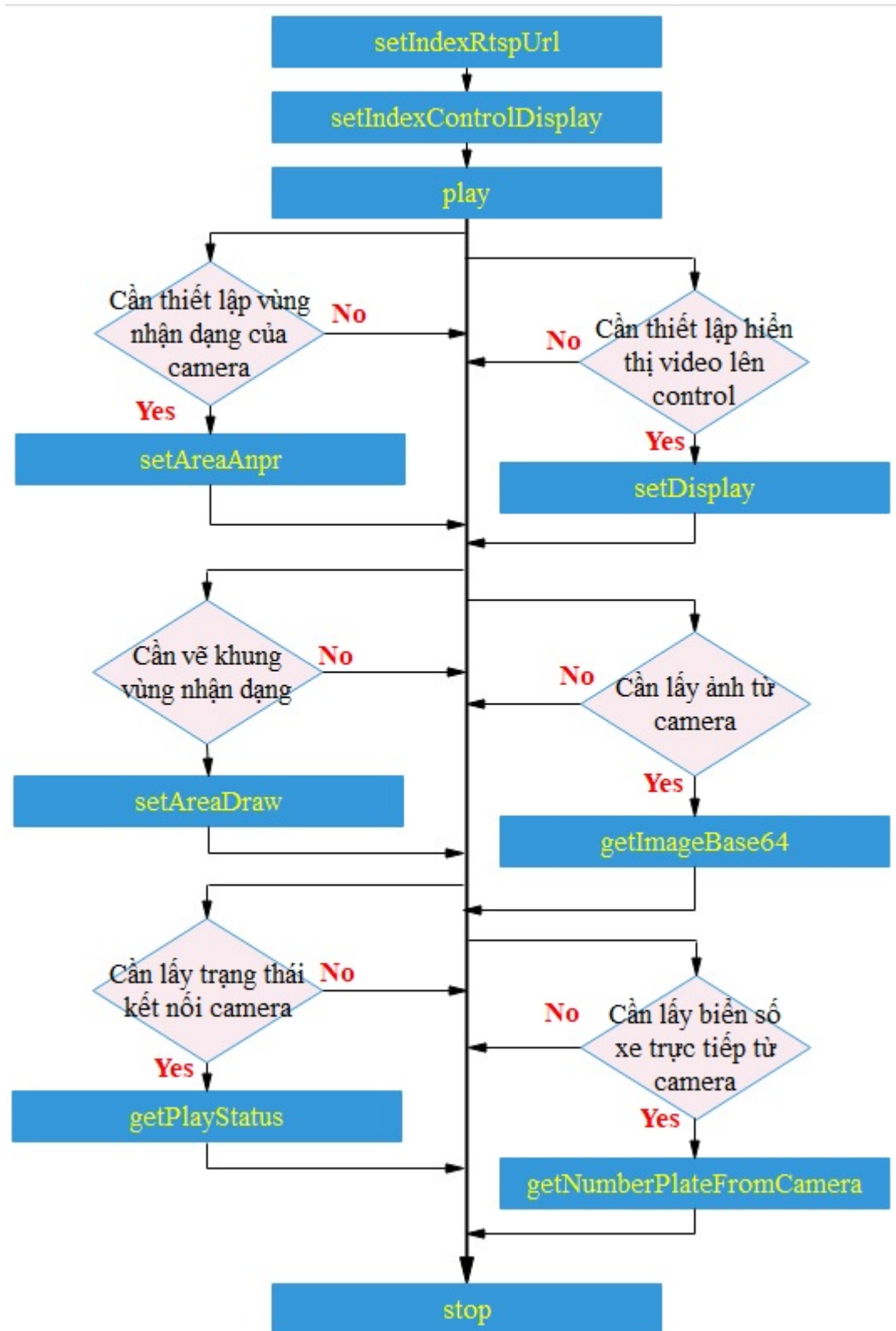
## 5.1 Quy trình khởi tạo

Đối với ngôn ngữ lập trình khác nhau thì cách khởi tạo kết nối AP COM cũng khác nhau, các khởi tạo ngôn ngữ lập trình thông dụng.

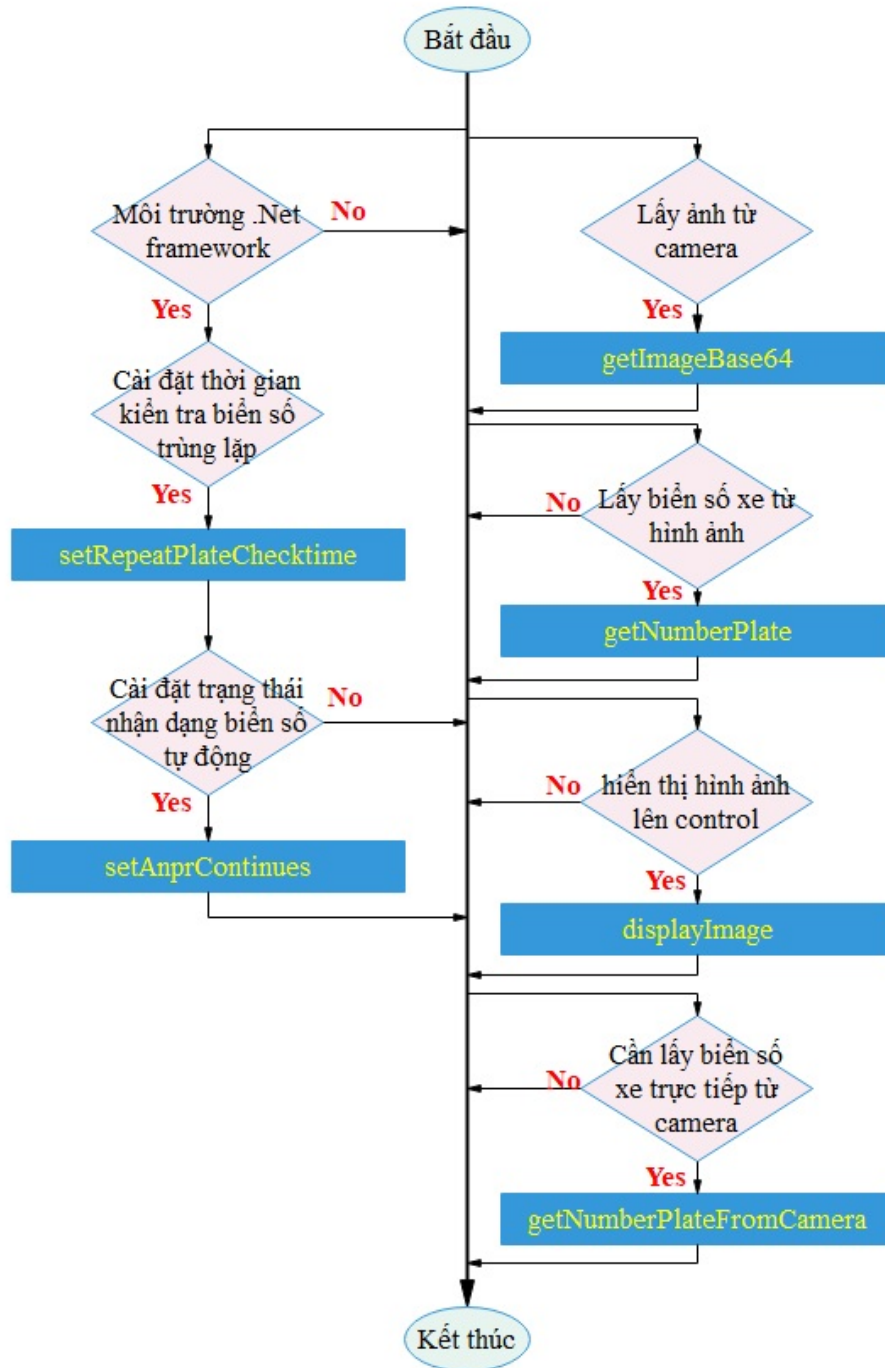
Ngôn ngữ lập trình	Khởi tạo API COM
C#	<pre>Type cameraCom dynamic aicamera cameraCom = Type.GetTypeFromProgID("Aitech.ClsAiAnprX64V5") aicamera = Activator.CreateInstance(cameraCom, false)</pre>
VB.Net	<pre>Dim cameraCom As Type Dim aicamera As Object cameraCom = Type.GetTypeFromProgID("Aitech.ClsAiAnprX64V5") aicamera = Activator.CreateInstance(cameraCom, False)</pre>
C++	<pre>#import "C:\Program Files (x86)\AITECH\AICAMERA\aiCameraX64.tlb" aiCameraX86::IClsAiRtspPlayerX86Ptr aicamera = nullptr SUCCEEDED(CoInitialize(0)) SUCCEEDED(aicamera.CreateInstance("Aitech.ClsAiAnprX64V5"))</pre>
Python	<pre>import win32com.client aicamera = win32com.client.Dispatch("Aitech.ClsAiAnprX64V5")</pre>



## 5.2 Quy trình kết nối camera IP

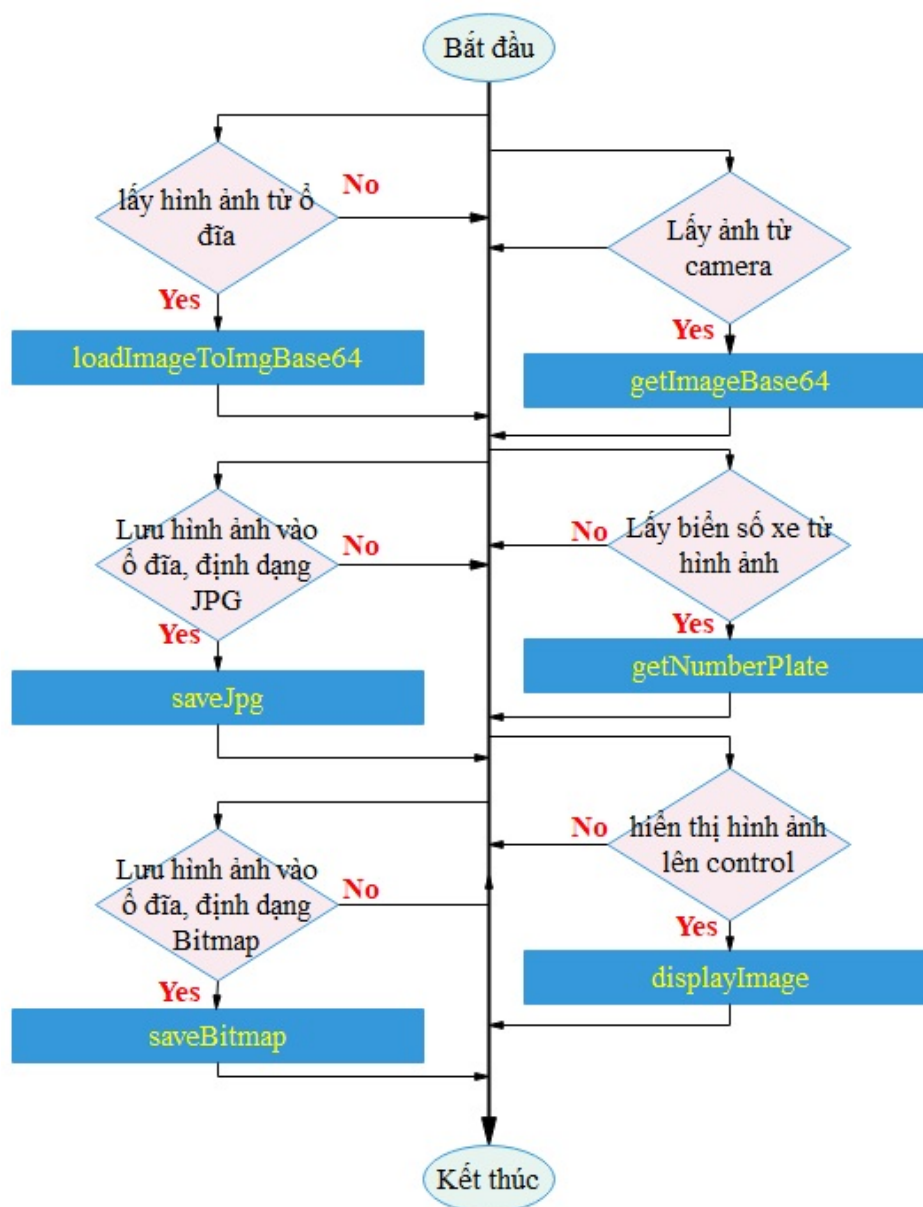


### 5.3 Quy trình xử lý nhận dạng biển số xe





## 5.4 Quy trình xử lý hình ảnh.



## 6. Các “Phương thức” sử dụng

### 6.1 setDifficultyLevel

- Cấu trúc:

`void setDifficultyLevel (int level)`

- Tham số:

- o level: Cài đặt độ khó xử lý ảnh của SDK, có 4 mức độ khó xử lý: 1, 2, 3, 4

- Trả về: Không

- Kết quả: : Cài đặt độ khó xử lý ảnh của SDK, độ khó xử phụ thuộc vào ứng dụng và mức độ hỗ trợ của camera nhận dạng biển số. Độ khó đề nghị
  - o Mốc số 2: Cho ứng dụng như: Trạm cân, bãi xe thông minh, cổng ra vào... Chỉ nhận dạng biển số đơn lẻ, không cần tốc độ quá nhanh, dùng camera không cần phải là loại chuyên dụng giá thành cao.
  - o Mốc số 4: Đối với các dự án cần tốc độ nhanh, xử liên tục xe đang di chuyển, sử dụng camera là loại chuyên dụng, lắp đặt đúng kỹ thuật như: Dự án ETC, dự án Trạm Cân Động ...

## 6.2 `initEvent`

- Cấu trúc:  
`void initEvent()`
- Tham số: Không
- Trả về: Không
- Kết quả: Khởi tạo môi trường nhận trạng thái kết nối camera (online, offline), kết quả khi nhận dạng biển số xe tự động

## 6.3 `setIndexRtspUrl`

- Cấu trúc:  
`void setIndexRtspUrl(int cameraIndex, string rtspUrl)`
- Tham số:
  - o `cameraIndex`: Số thứ tự camera
  - o `rtspUrl`: Đường link RTSP của camera
- Trả về: Không
- Kết quả: đường link RTSP `rtspUrl` của camera vào số thứ tự **CameraIndex**

## 6.4 `setIndexControlDisplay`

- Cấu trúc:  
`void setIndexControlDisplay(int cameraIndex, int displayControlHandle)`
- Tham số:
  - o `cameraIndex`: Số thứ tự camera
  - o `displayControlHandle`: Handle của control hiển thị video

- Trả về: Không
- Kết quả: Video camera sẽ hiển thị lên control

### 6.5 setDisplay

- Cấu trúc: **void setDisplay(int cameraIndex, Boolean displayValue)**
- Tham số:
  - o cameraIndex: Số thứ tự camera
  - o displayValue: Trạng thái muốn hiển thị hay không (true, false)
- Trả về: Không
- Kết quả: Hiển thị hoặc không hiển thị video lên control

### 6.6 play

- Cấu trúc:  
**void play(int cameraIndex)**
- Tham số:
  - o cameraIndex: Số thứ tự camera
- Trả về: Không
- Kết quả: Bắt đầu giải mã tín hiệu video tại vị trí **CameraIndex**

### 6.7 setCameraEventMethod

- Cấu trúc:  
**void setCameraEventMethod(object consumer, string eventName)**
- Tham số:
  - o Consumer: Đối tượng chứa “phương thức” eventName( This (c#), Me (vb.net)..)
  - o EventName: Tên “phương thức” (callback ) nhận kết quả. Cấu trúc của “Phương thức” nhận kết quả: **void cameraCallback (int cameraIndex, Boolean playStatus)**
- Trả về: Không
- Kết quả: Đăng ký “phương thức” nhận kết quả trả về trạng thái kết nối camera.

### 6.8 cameraCallback

- Cấu trúc:

**void cameraCallback** (int cameraIndex, Boolean playStatus)

- Tham số:
  - cameraIndex: Thứ tự camera
  - playStatus: Trạng thái camera
- Trả về: Không
- Kết quả: Sự kiện trả về trạng thái camera. **cameraCallback** ( Có thể đổi tên phương thức).

## 6.9 setAnprEventMethod

- Cấu trúc:
 

**void setAnprEventMethod**(object consumer, string eventName)
- Tham số:
  - Consumer: Đối tượng chứa “phương thức” **anprCallback**( This (c#), Me (vb.net)..)
  - EventName: Tên “phương thức” (callback ) nhận kết quả. Cấu trúc của “Phương thức” nhận kết quả: **void anprCallback**(int cameraIndex, string plateNum)
- Trả về: Không
- Kết quả: : Đăng ký “phương thức” nhận kết quả trả về kết quả nhận dạng biển số xe tự động

## 6.10 anprCallback

- Cấu trúc:
 

**void cameraCallback** (int cameraIndex, Boolean plateNum)
- Tham số:
  - cameraIndex: Số thứ tự camera
  - plateNum: Biển số xe trả về
- Trả về: Không
- Kết quả: : Phương nhận kết quả trả về kết quả nhận dạng biển số xe tự động.

## 6.11 setAnprContinues

- Cấu trúc:

---

**void setAnprContinues(int cameraIndex, Boolean anprContinuesStatus)**

- Tham số:
  - cameraIndex: Số thứ tự camera
  - anprContinuesStatus: Trạng thái tự động nhận dạng biển số xe.
- Trả về: Không
- Kết quả: Cài đặt trạng thái nhận dạng biển số xe, **chức năng chỉ mở ta khi độ khó xử lý được cài đặt ở mức số 4.**

### 6.12 setAreaAnpr

- Cấu trúc:

**void setAreaAnpr(int cameraIndex, int topOffset, int bottomOffset, int leftOffset, int rightOffset)**
- Tham số:
  - cameraIndex: Số thứ tự camera
  - topOffset: Số pixel loại bỏ lề trên khi nhận dạng
  - bottomOffset: Số pixel loại bỏ lề dưới khi nhận dạng
  - leftOffset: Số pixel loại bỏ lề trái khi nhận dạng
  - rightOffset: Số pixel loại bỏ lề phải khi nhận dạng
- Trả về: Không
- Kết quả: Loại bỏ một số vùng khi nhận dạng biển số tự động, sẽ làm tăng tốc độ, giảm nhiễu và tăng chính xác.

### 6.13 setAreaDraw

- Cấu trúc:

**void setAreaDraw(int cameraIndex, bool drawEnable)**
- Tham số:
  - cameraIndex: Số thứ tự camera
  - drawEnable: Trạng thái có vẽ khung ( vùng nhận dạng lên video không)
- Trả về: Không
- Kết quả: Vẽ hoặc không vẽ vùng nhận dạng biển số

## 6.14 setHeightChar

- Cấu trúc:

```
void setHeightChar(int minPixelHeight, int maxPixelHeight)
```

- Tham số:

- o minPixelHeight: Độ cao ( số pixel) nhỏ nhất ký tự trên biển số khi nhận
- o maxPixelHeight: Độ cao ( số pixel) lớn nhất ký tự trên biển số khi nhận

- Trả về: Không

- Kết quả: Cài đặt độ cao tối thiểu và tối đa khi nhận dạng biển số, chức năng này có thể dùng để định vị xe khi nhận dạng tự động. Chức năng chỉ cần đặt 1 lần và có động lên đến khi tắt chương trình. **Chức năng chỉ mở ta khi độ khó xử lý được cài đặt ở mức số 4.**

## 6.15 setRepeatPlateChecktime

- Cấu trúc:

```
void setRepeatPlateChecktime(int cameraIndex, int checktime)
```

- Tham số:

- o cameraIndex: Số thứ tự camera
- o checktime: Tính bằng **Milisecond**, thời gian kiểm tra trùng lặp biển số khi nhận dạng tự động.

- Trả về: Không

- Kết quả: Cài đặt thời gian kiểm tra trùng lặp biển số khi nhận dạng tự động, nếu không cài đặt thì biển số sẽ không trả về kết quả nếu biển số nhận dạng trùng với biển số lần nhận dạng trước. **Chức năng chỉ mở ta khi độ khó xử lý được cài đặt ở mức số 4.**

## 6.16 getNumberPlate

- Cấu trúc:

```
string getNumberPlate(string imageBase64, ref string plateImageBase64)
```

- Tham số:

- o imageBase64: Hình ảnh dạng base64 đưa vào để nhận dạng biển số xe.
- o plateImageBase64: Hình ảnh biển số xe được tách ra (nếu có)

- Trả về: Kiểu **string** là biển số xe trả về

Kết quả: Lấy được biển số xe và hình của vùng chứa biển số xe.

### 6.17 `getNumberPlateWithArea`

- Cấu trúc:

```
string getNumberPlateWithArea(string imageBase64, ref string originalImageBase64, ref string plateImageBase64, int topOffset, int bottomOffset, int leftOffset, int rightOffset)
```

- Tham số:

- `imageBase64`: Hình ảnh dạng base64 đưa vào để nhận dạng biển số xe.
  - `originalImageBase64`: Trả lại hình ảnh sau khi loại bỏ: lề trên, lề dưới, lề trái, lề phải
  - `plateImageBase64`: Hình ảnh biển số xe được tách ra (nếu có)
  - `topOffset`: Số pixel loại bỏ lề trên khi nhận dạng
  - `bottomOffset`: Số pixel loại bỏ lề dưới khi nhận dạng
  - `leftOffset`: Số pixel loại bỏ lề trái khi nhận dạng
  - `rightOffset`: Số pixel loại bỏ lề phải khi nhận dạng
- Trả về: Kiểu `string` là biển số xe.
- Kết quả: Lấy được biển số xe, hình của vùng chứa biển số xe, hình ảnh sau khi loại bỏ vùng không cần nhận dạng biển số.

### 6.18 `saveBitmap`

- Cấu trúc:

```
Boolean saveBitmap(string imageBase64, string fullFileName)
```

- Tham số:

- `imageBase64`: Hình ảnh dạng base 64
  - `fullFileName`: đường dẫn muốn lưu hình
- Trả về: kiểu `Boolean` lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng `Bitmap` vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

### 6.19 `saveJpg`

- Cấu trúc:

---

**Boolean saveJpg**(string imageBase64, string fullFileName)

- Tham số:
  - imageBase64: Hình ảnh dạng base64
  - fullFileName: đường dẫn lưu hình
- Trả về: kiểu **Boolean** lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng **JPG** vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

### 6.20 displayImage

- Cấu trúc:

**Boolean saveBitmap**(string imageBase64, string fullFileName)
- Tham số:
  - imageBase64: Hình ảnh dạng base 64
  - fullFileName: đường dẫn muốn lưu hình
- Trả về: Kiểu **Boolean** lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng **Bitmap** vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

### 6.21 loadImageToImgBase64

- Cấu trúc:

**string loadImageToImgBase64**(string fullFileName)
- Tham số:
  - fullFileName: Đường dẫn hình ảnh trong có trong ổ cứng
- Trả về: Kết quả kiểu **string** là hình ảnh dạng base64
- Kết quả: Tải hình ảnh lưu từ ổ đĩa thành hình ảnh dạng Base64

### 6.22 getPlayStatus

- Cấu trúc:

**Boolean getPlayStatus**(int cameraIndex)
- Tham số:
  - CameraIndex: Số thứ tự camera



- Trả về: Kiểu **Boolean** là trạng thái của camera
- Kết quả: Trả về trạng thái kết nối camera

### 6.23 getAnprStatus

- Cấu trúc:  
**Boolean** getAnprStatus(**int** cameraIndex)

- Tham số:
  - o CameraIndex: Số thứ tự camera

Trả về: kiểu **Boolean** là trạng thái nhận dạng biển số tự động của camera.

- Kết quả : Lấy trạng thái nhận dạng tự động. **Chức năng chỉ mở ta khi độ khó xử lý được cài đặt ở mức số 4.**

### 6.24 getImageBase64

- Cấu trúc:  
**string** getImageBase64(**int** cameraIndex)

- Tham số:
  - o cameraIndex: Số thứ tự camera
- Trả về: Kiểu **string** là hình ảnh dạng base64

- Kết quả: Lấy hình ảnh dạng Base64 từ camera có vị trí CameraIndex

### 6.25 getNumberPlateFromCamera

- Cấu trúc:  
**string** getNumberPlateFromCamera(**int** cameraIndex, **ref string** originalImageBase64, **ref string** plateImageBase64);

- Tham số:
  - o cameraIndex: Số thứ tự camera
  - o originalImageBase64: Trả lại hình ảnh đã nhận dạng biển số xe.
  - o plateImageBase64: Hình ảnh biển số xe được tách ra (nếu có)
- Trả về: Kiểu **string** là biển số xe.
- Kết quả: Chụp hình và trả về kết quả biển số xe từ tín hiệu camera.

## 6.26 Dispose

- Cấu trúc:  
`void dispose()`
- Tham số: Không
- Trả về: Không
- Kết quả: Dừng tất cả camera đang hoạt động

## 7. Thông tin công ty TNHH Công nghệ A.I

Công ty TNHH CÔNG NGHỆ A.I

Địa chỉ: Số 425 Đường TCH 21, Phường Tân Chánh Hiệp, Quận 12, Tp. HCM

Điện thoại: 0286 250 7888

Hotline: 0914452090

Email: [info@aitech.com.vn](mailto:info@aitech.com.vn)

Website: <https://aitech.com.vn>

Fanpage: <https://www.facebook.com/SoftwareComputervision>