|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  Đồ án thiết kế 3:  Đề tài:  Xây dựng hệ thống phát hiện và phán đoán vị trí đối tượng trong tương lai  Giảng viên hướng dẫn: PGS.TS Nguyễn Hữu Trung  Sinh viên thực hiện: Bùi Trung Kiên - 20172638  Hà Nội, 6-2021 |

Lời nói đầu

**Thị giác máy tính** là một lĩnh vực khoa học liên ngành đề cập đến cách [máy tính](https://en.wikipedia.org/wiki/Computer" \o "Máy vi tính) có thể đạt được sự hiểu biết cấp cao từ hình ảnh hoặc video kỹ thuật số. Từ góc độ kỹ thuật, nó tìm cách hiểu và tự động hóa các nhiệm vụ mà hệ thống thị giác của con người có thể thực hiện.

Các nhiệm vụ về thị giác máy tính bao gồm các phương pháp thu thập, xử lý, phân tích và hiểu hình ảnh kỹ thuật số cũng như trích xuất dữ liệu chiều cao từ thế giới thực để tạo ra thông tin số hoặc biểu tượng, ví dụ như dưới dạng quyết định. Hiểu theo ngữ cảnh này có nghĩa là chuyển đổi hình ảnh trực quan (đầu vào của võng mạc) thành mô tả thế giới có ý nghĩa đối với các quá trình suy nghĩ và có thể gợi ra hành động thích hợp. Sự hiểu biết hình ảnh này có thể được coi là sự tách rời thông tin biểu tượng từ dữ liệu hình ảnh bằng cách sử dụng các mô hình được xây dựng với sự hỗ trợ của hình học, vật lý, thống kê và lý thuyết học tập.

Ngành khoa học về thị giác máy tính liên quan đến lý thuyết đằng sau các hệ thống nhân tạo trích xuất thông tin từ hình ảnh. Dữ liệu hình ảnh có thể ở nhiều dạng, chẳng hạn như chuỗi video, chế độ xem từ nhiều máy ảnh, dữ liệu đa chiều từ máy quét 3D hoặc thiết bị quét y tế. Ngành công nghệ của thị giác máy tính tìm cách áp dụng các lý thuyết và mô hình của nó vào việc xây dựng các hệ thống thị giác máy tính.

Các lĩnh vực phụ của thị giác máy tính bao gồm tái tạo cảnh, phát hiện sự kiện, theo dõi đối tượng, nhận dạng đối tượng, ước tính tư thế 3D, học tập, lập chỉ mục, ước tính chuyển động, hỗ trợ trực quan, tạo mô hình cảnh 3D và phục hồi hình ảnh.

Nắm bắt được xu hướng này, trong khi tiếp cận môn đồ án thiết kế I, nhóm em đã lựa chọn đề tài “Xây dựng máy chủ nhận dạng đối tượng từ nhiều luồng camera”. Hệ thống có thể ứng dụng trong việc giám sát và đảm bảo an ninh.

Trong qua trình làm đồ án III, chúng em xin được gửi lời cảm ơn chân thành tới:

* Thầy giáo hướng dẫn PGS.TS Nguyễn Hữu Trung giảng viên trường Đại học Bách Khoa Hà Nội đã giúp đỡ, tạo điều kiện cho chúng em hoàn thành đồ án.

Mục lục

[CHƯƠNG I. XÂY DỰNG MẠNG KẾT NỐI MÁY CHỦ VỚI NHIỀU MÁY KHÁCH 8](#_Toc74422852)

[1. Đặt Vấn Đề 8](#_Toc74422853)

[1.1. Socket là gì ? 8](#_Toc74422854)

[1.2. Tại sao sử dụng socket ? 8](#_Toc74422855)

[1.3. Mô tả hoạt động socket 8](#_Toc74422856)

[1.4. Phân loại socket 10](#_Toc74422857)

[2. Triển Khai Mô Hình 10](#_Toc74422858)

[CHƯƠNG II. XỬ LÝ DỮ LIỆU TẠI MÁY KHÁCH 12](#_Toc74422859)

[1. Đọc dữ liệu từ camera 12](#_Toc74422860)

[2. Phát hiện đối tượng 13](#_Toc74422861)

[3. Đưa ra dự đoán về hướng di chuyển của đối tượng 17](#_Toc74422862)

[4. Mã hóa dữ liệu 19](#_Toc74422863)

[CHƯƠNG III. XỬ LÝ DỮ LIỆU TẠI MÁY CHỦ 20](#_Toc74422864)

[1. Giải mã dữ liệu 20](#_Toc74422865)

[2. Phát dữ liệu hình ảnh 20](#_Toc74422866)

[3. Cảnh báo tới người giám sát thông qua giao diện 21](#_Toc74422867)

[CHƯƠNG IV: TỔNG KẾT 23](#_Toc74422868)

[**\*Tài Liệu Tham Khảo** 24](#_Toc74422869)

Danh mục ký hiệu và chữ viết tắt

Danh sách hình vẽ

Danh mục bảng biểu

Tóm tắt đồ án

Với yêu cầu cần phát triển hệ thống phát hiện và theo dõi đối tượng, có thể thiết kế

hệ thống gồm 2 thành phần chính gồm: máy chủ và các máy khách. Khi đó, các máy khách sẽ có nhiệm vụ phát hiện đối tượng và tính toán các thông số phục vụ cho viêc dự đoán rồi gửi dữ liệu về máy chủ. Khi máy chủ nhận được dữ liệu sẽ tiến hành xử lý và gửi tín hiệu dự báo tới máy khách được chỉ định.

Vì vậy, những vấn đề chính cần làm sáng rõ trong báo cáo bao gồm:

* Xây dựng mạng kết nối giữa máy chủ và nhiều máy khách.
* Xử lý dữ liệu tại máy khách.
* Xử lý dữ liệu tại máy chủ.
* Các tính năng bổ xung hỗ trợ hệ thống.

# CHƯƠNG I. XÂY DỰNG MẠNG KẾT NỐI MÁY CHỦ VỚI NHIỀU MÁY KHÁCH

1. Đặt Vấn Đề

Do hệ thống được thiết kế nhằm trao đổi dữ liệu giữa máy chủ và các máy khách nên việc đảm bảo độ trễ không ở mức quá cao là việc cực kỳ quan trọng vì việc theo dõi sẽ trở nên vô nghĩa khi độ trễ lớn.

Khi muốn ứng dụng các hoạt hoạt động realtime thì có rất nhiều các giao diện lập trình ứng dụng mạng như: socket, ZMQ, WebRTC, …Tuy nhiên, trong phạm vi bài báo cáo này em sẽ sử dụng socket cho việc xây dựng mạng.

* 1. Socket là gì ?

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữu liệu qua internet. Giữa 2 chương trình chạy trên mạng muốn liên kết cần có liên kết giao tiếp hai chiều. Điêm cuối của liên kết được gọi là socket.

* 1. Tại sao sử dụng socket ?

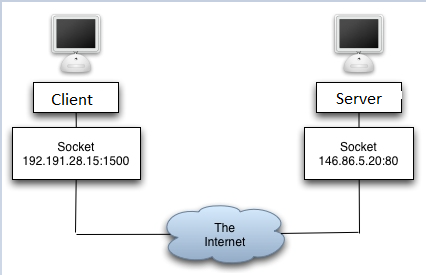
Nếu không sử dụng socket, theo các phương pháp cũ thì các thành viên trong mạng sẽ kiểm tra api sau 1 quãng thời gian nghỉ nhất định và nếu có sự thay đổi về mặt dữ liệu thì sẽ cập nhật. Nếu sử lý như vậy chắc chắn cả client và server đều mệt mỏi.

Socket giải quyết vấn đề này 1 cách hoàn hảo, tiết kiệm tài nguyên cho cả client và server.

* 1. Mô tả hoạt động socket

Socket hoạt động trên cả giao thức TCP và UDP.

Socket = IP Address + Số Port



Hình 1:Socket

* Khởi tạo kết nối từ client tới server

1. Client có địa chỉ IP1(‘192.191.28.15’) và port 1500 sẽ sử dụng cặp (IP1,port) = (IP1,1500) để kết nối tới server có địa chỉ IP2 (‘146.86.5.20’) và port 80.
2. Sau khi client được server xác thực thành công và có các thông tin cần thiết, cổng 25 sẽ đc mở cho IP2.
3. Khi có email mới, server sẽ kiểm tra kết nối tới (IP1,25) còn tồn tại hay không, nếu có sẽ tiến hành gử thông báo về client.

* Duy trì kết nối

Sở dĩ 2 máy có thể duy trì được kết nối là do **port đã được mở và sẽ không đóng** cho đến khi chiều bên kia gửi tín hiệu muốn chấm dứt bằng cách gửi gói tin RST. Trong trường hợp chiều bên kia ngắt kết nối mà không gửi RST thì kết nỗi vẫn sẽ được đóng sau một khoảng timeout nào đó được quy định ở quá trình Keep-Alive.

Quá trình Keep-Alive có 3 thuộc tính để quyết định có đóng kết nối hay không:

1. tcp\_keepalive\_time: Khoảng thời gian không có tín hiệu. Mặc định là 7200s.
2. tcp\_keepalive\_intvl: Khoảng thời gian chờ chiều bên kia hồi đáp. Mặc định là 75s.
3. tcp\_keppalive\_probles: Số lần sẽ thử lại nếu việc giao tiếp gặp lỗi. Mặc định là 9.

Quá trình Keep-Alive sẽ diễn ra như sau:

1. Client mở kết nối TCP.
2. Sau một khoảng thời gian tcp\_keepalive\_time, nếu như server kia im lặng không có tín hiệu gì. Client sẽ gửi đi cờ ACK (kích thước rất nhỏ, có thể không được xem là một package) đến server và chờ hồi đáp.
3. Server có hồi đáp ACK hay không ?

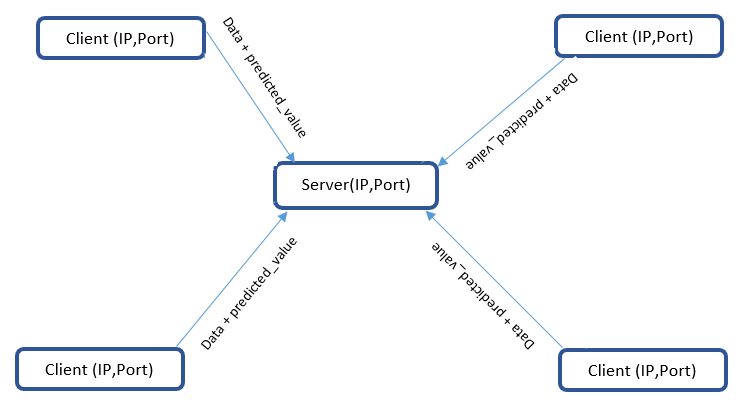
* Nếu không hồi đáp(3.1): Kiểm tra xem số lần thử lại đã vượt quá tcp\_keppalive\_probles hay chưa ? Nếu chưa thì tiến hành đợi sau khoảng tcp\_keepalive\_intvl rồi gửi lại ACK và quay lại bước 3. Nếu đã vượt quá tcp\_keepalive\_probes thì gửi RST đến server (không quan tâm đến kết quả trả về) rồi đóng kết nối.
* Nếu server hồi đáp:
  1. Nếu hồi đáp đúng => reset lại các thuộc tính rồi quay về bước 2.
  2. Nếu hồi đáp sai => chuyển sang 3.1

Như vậy thì mặc định, client socket sẽ mở: tcp\_keepalive\_time + tcp\_keepalive\_intvl \* tcp\_keepalive\_probes = 7200 + 75 \* 9 (giây) = 2h11p nếu như không nhận được RST của server.

* 1. Phân loại socket
* Stream Socket: Dựa trên giao thức TCP( Tranmission Control Protocol), việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Do đó, hình thức này được gọi là socket hướng kết nối.
  + Ưu điểm: Có thể dùng để liên lạc theo mô hình client và sever. Nếu là mô hình client /sever thì sever lắng nghe và chấp nhận từ client. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn. Đồng thời, mỗi thông điệp gửi phải có xác nhận trả về và các gói tin chuyển đi tuần tự.
  + Hạn chế: Có một đường kết nối (địa chỉ IP) giữa 2 tiến trình nên 1 trong 2 tiến trình kia phải đợi tiến trình kia yêu cầu kết nối.
* Datagram Socket: Dựa trên giao thức UDP( User Datagram Protocol) việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 quá trình. Do đó, hình thức này được gọi là socket không hướng kết nối.
  + Ưu điểm: Do không yêu cầu thiết lập kết nối, không phải có những cơ chế phức tạp nên tốc độ giao thức khá nhanh, thuận tiện cho các ứng dụng truyền dữ liệu nhanh như chat, game…
  + Hạn chế: Ngược lại với giao thức TCP thì dữ liệu được truyền theo giao thức UDP không được tin cậy, có thế không đúng trình tự và lặp lại.

1. Triển Khai Mô Hình

Từ những lý thuyết cơ bản được nêu ra ở trên, có thể đưa ra mô hình thiết kế máy chủ đa máy khách như sau:



Hình 2: Mô Hình Mạng Hệ Thống

**Mô tả hoạt động:**

1. Máy khách sẽ xử lý dữ liệu hình ảnh thu thập được từ camera và đưa ra giá trị dự đoán (predicted\_value) phù hợp và gửi giá trị đó cùng dữ liệu hình ảnh ( đã được tổng hợp và mã hóa ) về server.
2. Máy chủ sẽ tiến hành giải mã dữ liệu được gửi về. Dữ liệu hình ảnh sẽ được giải mã và phát trên giao diện của máy chủ. Giá trị dự đoán sẽ được giải mã và đưa ra cảnh báo ngay trên giao diện của máy chủ.

**Lựa chọn thiết bị cho hệ thống:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thiết bị | Hình ảnh | Vai trò |
| 1 | HPE ML30 Gen10 4LFF E-2134 P06761-B21 2TB | HPE ML30 Gen10 4LFF E-2134 P06761-B21 2TB | Máy chủ server |
| 2 | NVIDIA Jetson Nano Developer Kit (V3) |  | Máy khách |
| 3 | Intel Wireless AC8265 |  | Module Wifi cho Jetson Nano |
| 4 | Camera | Amazon.com: 8MP IMX219-160 Camera Compatible with NVIDIA Jetson Nano and  Raspberry Pi Compute Module 3/3+,8 Megapixels Camera Module IMX219 Sensor  3280×2464 Resolution 160 Degree FOV: Computers &amp; Accessories | Thu thập hình ảnh |

# CHƯƠNG II. XỬ LÝ DỮ LIỆU TẠI MÁY KHÁCH

Như đã trình bày ở phần trên, nhiệm vụ của máy khách bao gồm:

* + Đọc dữ liệu hình ảnh từ camera.
  + Phát hiện đối tượng trong hình ảnh.
  + Đưa ra dự đoán về hướng di chuyển của đối tượng.
  + Mã hóa các dữ liệu.

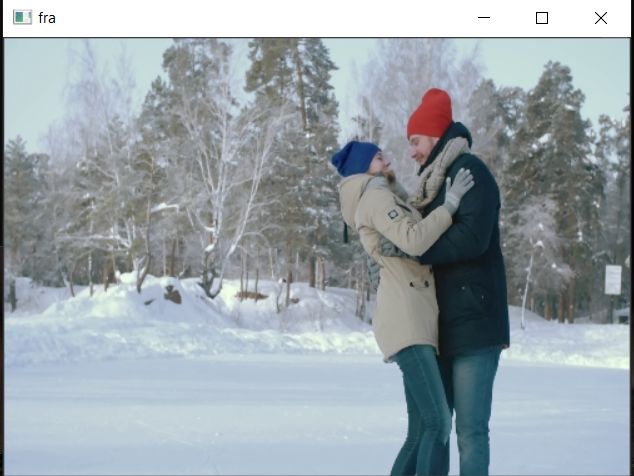
1. Đọc dữ liệu từ camera

Để máy tính đọc được dữ liệu hình ảnh từ camera là khá đơn giản. Các thư viện trong python hỗ trợ rất tốt về vấn đề này.

Trong báo cáo này, OpenCV được sử dụng xuyên suốt do dễ tiếp cận và vô cùng phù hợp với việc xử lý thị giác máy tính thời gian thực.

Thật may mắn là OpenCV có cung cấp 1 class là VideoCapture có chức năng capturing video từ camera hay đọc video từ 1 file video.

Việc triển khai cũng vô cùng đơn giản chỉ cần 1 vài dòng code python + OpenCV là đã có thể hoàn toàn truy cập và làm việc với luồng dữ liệu hình ảnh thu được từ camera hay file video.



Hình 3: Đọc video với lớp VideoCapture

1. Phát hiện đối tượng

Để có thể xác định chính xác đối tượng, cần phải thực hiện 2 nhiệm vụ:

* Nhận dạng đối tượng
* Nhận diện khuôn mặt
  1. Nhận dạng đối tượng

Nhận dạng đối tượng là quá trình nhận dạng các đối tượng trong video hoặc hình ảnh.

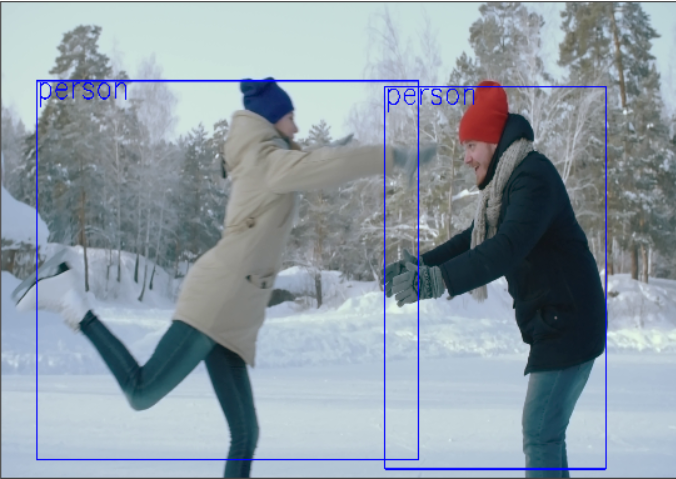
Giải pháp cơ bản của OD là cắt riêng từng vùng ảnh rồi đưa vùng ảnh đó vào mạng CNN để so sánh xem giống cái gì. Nếu phát hiện vật thể trong vùng ảnh nào đó thì lưu lại để làm bounding box.

Cách làm đơn giản này gặp khá nhiều khó khăn bởi các vật thể có kích thước khác nhau giữa các ảnh, góc quay, hướng quay khác nhau giữa các ảnh nên số lượng vùng ảnh phải rất lớn và “cực phong phú đa dạng” về kích thước, tỷ lệ thì may ra mới phát hiện được hết các vật thể trong ảnh. Điều này làm cho thuật toán chạy chậm và tốn nhiều tài nguyên tính toán.

Một trong những mô hình về OD nổi trội là R-CNN. Nhóm kỹ thuật R-CNN chủ yếu sử dụng các vùng để khoanh vùng các đối tượng trong ảnh. Mạng không xem xét toàn bộ hình ảnh, chỉ xem xét các phần của hình ảnh có khả năng chứa một đối tượng cao hơn.

Tuy nhiên, mô hình YOLO còn làm được tốt hơn R-CNN. YOLO là You Only Look Once và đúng như vậy, nó không xem xét từng vùng mà nhìn toàn bộ bức ảnh một lần sau đó predict ra class, bounding box cho từng object trong hình. Chính vì vậy, **ưu điểm lớn nhất của việc sử dụng YOLO là tốc độ tuyệt vời** - nó cực kỳ nhanh và có thể xử lý 45 khung hình mỗi giây. YOLO cũng hiểu biểu diễn đối tượng tổng quát.

Về cách hoạt động của YOLO, thay vì xét các vùng trong ảnh như R\_CNN thì YOLO chia ảnh thành một lưới vuông với kích thước giả sử là NxN mỗi ô. Sau đó nó duyệt từng ô trong ảnh, với mỗi ô sẽ thử m cái anchor box. Với mỗi anchor box đó thì sẽ dự đoán class và offset của bounding box của vật thể. Cuối cùng thì các bouding box có xác suất lớn nhất trên ngưỡng sẽ được giữ lại áp dụng một thuật toán có tên Non Max Supression (NMS) để chỉ giữ lại các bounding box tốt nhất và output ra kết quả cho người dùng.



Hình 4: Nhận dạng đối tượng với YOLOv3

* 1. Nhận diện khuôn mặt

Sau khi nhận dạng được đối tượng, để xác định chính xác hơn nữa đối tượng theo dõi, công cụ nhận diện khuôn mặt là 1 lựa chọn phù hợp để phát hiện và loại bỏ các nhận dạng trước đó không cần thiết khác.

Về cơ bản, Nhận diện khuôn mặt cần thực hiện các công việc: Phát hiện khuôn mặt, đào tạo và nhận diện.

* + 1. Phát hiện khuôn mặt

Năm 2018, tính năng nhận diện khuôn mặt trở nên cực kỳ dễ dàng nhờ sự bùng nổ về khả năng thị giác của máy tính. OpenCV chắc chắn là thư viện mã nguồn mở yêu thích của nhiều sinh viên và các nhà nghiên cứu.

Có rất nhiều mô hình nhận diện khuôn mặt đã được phát triển như: Haar, DNN, HoG, MMOD,… Tuy nhiên, mô hình DNN có vẻ tỏ ra vượt trội hơn so với các mô hình còn lại khi cho kết quả trả về có độ chính xác tương đối cao cũng như có fps trong khoảng 45-47. Tuy nhiên, MMOD lại tỏ ra vượt trội hơn hẳn về tốc độ khi fps đạt tới 310 fps khi hoạt động trên GPU.

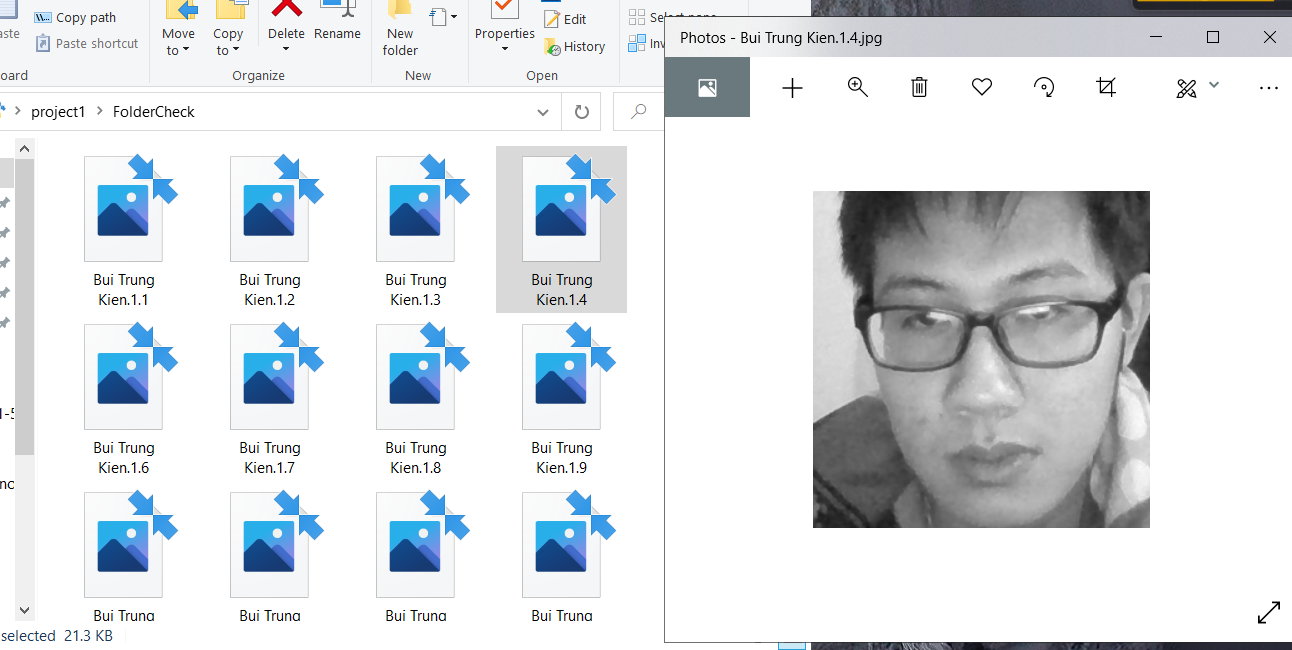
Vì vậy, tốt hơn là sử dụng phương pháp OpenCV - DNN vì nó khá nhanh và rất chính xác, ngay cả đối với các khuôn mặt có kích thước nhỏ. Nó cũng phát hiện khuôn mặt ở nhiều góc độ khác nhau.



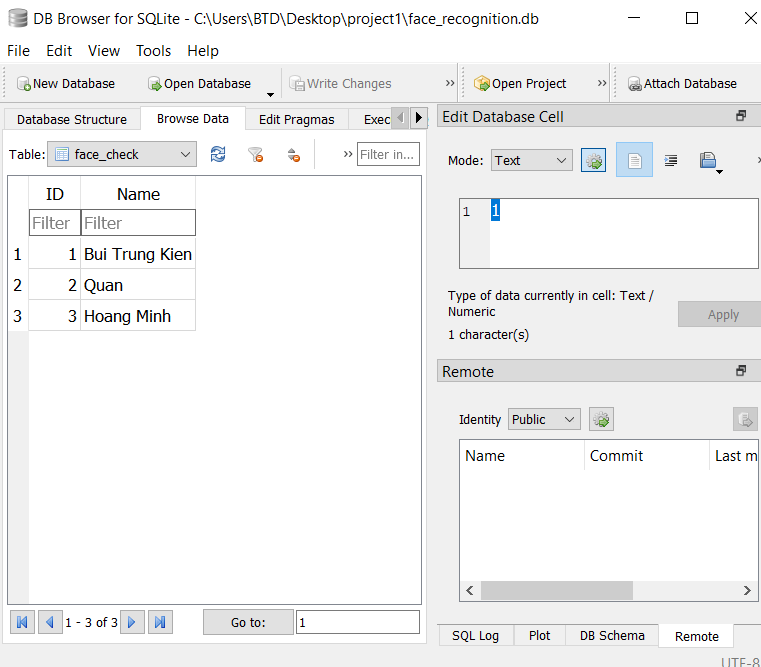
Hình 5: Phát hiện khuôn mặt với OpenCV-DNN

* + 1. Đào tạo

Trước khi tiền hành đào tạo cho máy tính thì cần phải có dữ liệu mẫu để đào tạo. Dữ liệu mẫu là các hình ảnh mức xám chứa khuôn mặt của đối tượng cần theo dõi. Việc thu thập có thể thực hiện bằng nhiều cách khác nhau miễn cho ra được kết quả mong muốn.

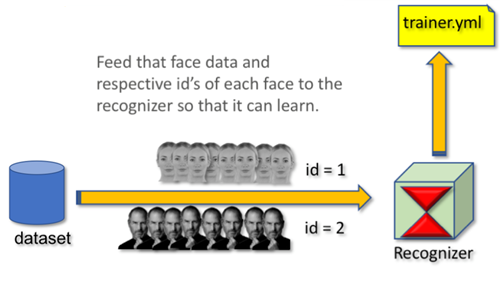


Hình 6: Dữ liệu mẫu phục vụ cho việc đào tạo



Hình 7: Thông tin của đối tượng được đào tạo

**Mô hình đào tạo:**



Hình 8: Mô hình đào tạo với OpenCV Recognizer

Trong giai đoạn này, chúng ta sẽ lấy tất cả dữ liệu người dùng từ tập dữ liệu đã thu thập được và Đào tạo với OpenCV Recognizer. Điều này được thực hiện với hàm train\_data.py và sau quá trình huấn luyện sẽ nhận được tệp .yml.

Thuật toán được sử dụng để trích xuất thông tin hình ảnh và thực hiện so sánh với hình ảnh đầu vào là LBPH.

Toàn bộ thông tin về LBPH đã được chỉ ra rõ ràng và chi tiết trong [1].

* + 1. Nhận diện

Tiếp theo, giai đoạn cuối cùng trong nhận dạng khuôn mặt. Tại đây, chúng ta sẽ chụp một khuôn mặt trên máy ảnh của mình và nếu người này đã được chụp và đào tạo khuôn mặt trước đó, trình nhận dạng sẽ đưa ra "dự đoán" trả về id và chỉ số của người đó, cho thấy mức độ tin cậy của người nhận với trận đấu này.



Hình 9: Kết quả nhận diện

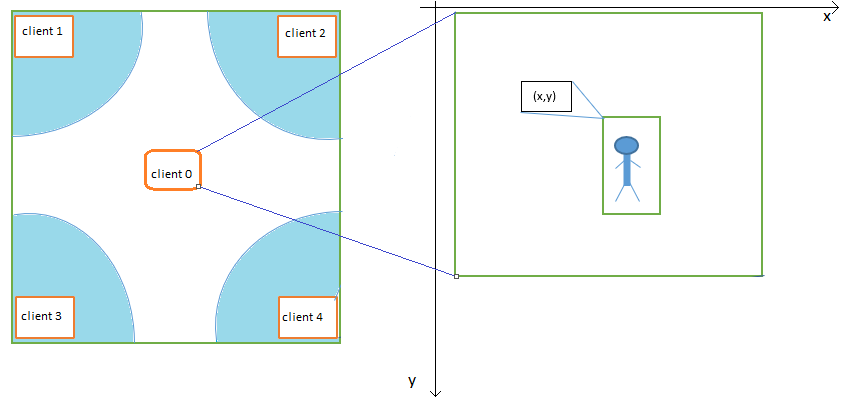
1. Đưa ra dự đoán về hướng di chuyển của đối tượng

Về cơ bản, để có thể phán đoán vị trí của đối tượng trong tương lai cần dựa vào các kết quả trả về trạng thái hiện tại và quá khứ của đối tượng.

Trong mô hình, các camera có vai trò như những con mắt giám sát cũng như vị trí của các camera là các vị trí mà đối tượng có thể di chuyển tới trong tương lai. Vì vậy, các camera cần được bố trí 1 cách chính xác và đồng bộ với nhau để có thể tránh được trường hợp phán đoán sai.

Quá trình phát hiện đã cho chúng ta biết được vị trí cụ thể của đối tượng trong ảnh với 1 bounding box. Nhận thấy, các bounding box này thể hiện cho vị trí của đối tượng nên hoàn toàn có thể dùng tọa độ của bounding box để có thể xét sự thay đổi để đưa ra phán đoán. Và để thuận tiện, có thể lấy luôn tọa độ (x,y) của bounding box để tiến hành tính toán vì việc lấy bất kỳ 1 tọa độ nào trong bounding box thì tất cả đều cho ra 1 kết quả giống nhau.

Để làm rõ hơn, có thể xét 1 mạng lưới phát hiện và phán đoán đơn giản.



Hình 10: Mô hình với 5 máy khách

Có thể thấy trong hình 10 là 1 mạng lưới 5 máy khách giúp phát hiện và phán đoán vị chí. Vùng xanh da trời được hiểu là vùng quan sát được của camera. Khi 1 đối tượng trong diện theo dõi đi vào vùng theo dõi của camera, máy khách sẽ tiến hành xử lý OD và FD sau đó tính toán, đưa ra dự đoán dựa vào giá trị của (x,y) theo thời gian.

Với mô hình trên, các trường hợp có thể xảy ra là:

* x↑ và y↑: Đối tượng đang di chuyển và sẽ tới vị trí đặt client 4
* x↑ và y↓: Đối tượng đang di chuyển và sẽ tới vị trí đặt client 2
* x↓ và y↑: Đối tượng đang di chuyển và sẽ tới vị trí đặt client 3
* x↓ và y↓: Đối tượng đang di chuyển và sẽ tới vị trí đặt client 1
* Chỉ x↑: Đối tượng đang di chuyển và sẽ tới vị trí giữa 2 client 2 và 4
* Chỉ x↓: Đối tượng đang di chuyển và sẽ tới vị trí giữa 2 client 1 và 3
* Chỉ y↑: Đối tượng đang di chuyển và sẽ tới vị trí giữa 2 client 3 và 4
* Chỉ y↓: Đối tượng đang di chuyển và sẽ tới vị trí giữa 2 client 1 và 2

Tất nhiên trong thực tế, các client sẽ đại diện cho tên con đường mà nó được lắp đặt.

Với mô hình trên, giá trị của tọa độ (x,y) đóng vai trò chủ chốt trong việc dự đoán. Vì vậy, để có độ chính xác cao cần tìm khoảng thay đổi phù hợp cho việc đưa ra giá trị dự đoán. Giá trị dự đoán sẽ thể hiện cho 3 th chính: Đối tượng di chuyển, đối tượng không di chuyển, và không phát hiện được đối tượng.

1. Mã hóa dữ liệu

Như đã biết, việc truyền trực tiếp hình ảnh về server sẽ khiến tốn băng thông, gây ra độ trễ lớn khi truyền trực tiếp. Vì vậy, việc mã hóa dữ liệu trước khi truyền sẽ giúp cải thiện các vấn đề trên.

Trong báo cáo này, pickle sẽ được sử dụng để chuyển đổi khung hình và giá trị dự đoán sang dạng byte. Sau đó, các byte sẽ được đóng gói dưới dạng hàng đợi với thư viện struct để tạo thông báo và gửi tới máy chủ qua socket.

Tuy nhiên, với mô hình mã hóa sơ sài trên sẽ gây nguy hiểm nếu như tín hiệu truyền đi là tín hiệu quan trọng cần đc bảo mât. Vì vậy, tùy vào đặc điểm của dữ liệu cần truyền có thể chọn cách mã hóa phù hợp.

# CHƯƠNG III. XỬ LÝ DỮ LIỆU TẠI MÁY CHỦ

Tiếp đến, nhiệm vụ của máy chủ bao gồm:

* Giải mã các dữ liệu.
* Phát dữ liệu hình ảnh lên giao diện.
* Dùng dữ liệu dự đoán để cảnh báo thông qua giao diện.

1. Giải mã dữ liệu

Về cơ bản, quá trình giải mã là ngược lại của quá trình mã hóa. Trong phần mã hóa, dữ liệu sẽ được chuyển sang byte rồi sau đó được đóng gói dưới dạng hàng đợi để tạo thống báo.

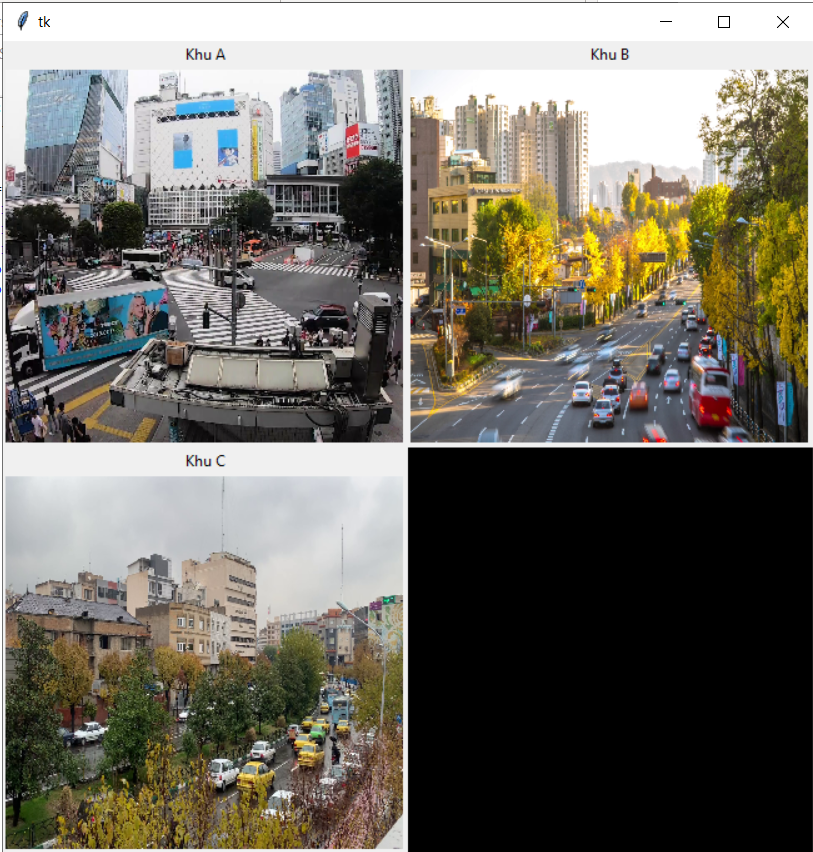
Ở máy chủ, tiến hành đọc thông báo và tính toán độ dài hàng đợi và giải nén các giá trị được đóng gói về giá trị ban đầu trước khi đóng gói của nó. Sau đó, sử dụng module pickle để đọc và giải nén dữ liệu về dạng ban đầu trước khi được mã hóa sang byte.

1. Phát dữ liệu hình ảnh

Python cung cấp các tùy chọn khác nhau để phát triển giao diện người dùng đồ họa (GUI). Các tính năng quan trọng nhất được liệt kê dưới đây.

* **Tkinter** - Tkinter là giao diện Python cho bộ công cụ Tk GUI đi kèm với Python.
* **wxPython** - Đây là một giao diện Python mã nguồn mở cho bộ công cụ GUI của wxWidgets.
* **PyQt** -Đây cũng là giao diện Python cho thư viện GUI Qt đa nền tảng phổ biến.
* **JPython** - JPython là một cổng Python dành cho Java, cho phép các tập lệnh Python truy cập liền mạch vào các thư viện lớp Java trên máy cục bộ .

Báo cáo này sẽ tập trung vào Tkinter để xây dựng chương trình cho máy chủ. Tkinter là thư viện GUI tiêu chuẩn cho Python. Python khi kết hợp với Tkinter cung cấp một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng mạnh mẽ cho bộ công cụ Tk GUI. Việc tạo 1 ứng dụng GUI bằng Tkinter cũng hết sức dễ dàng.



Hình 11: Giao diện hiển thị của server

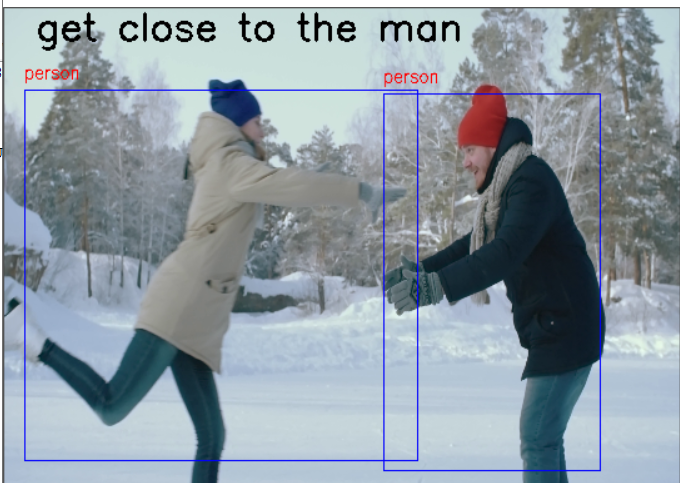
Server sẽ sử dụng Tkinter làm khuôn khổ, dùng thư viện OpenCV và Tkinter Canvas để hiển thị các luồng video. Để có thể hiển thị nhiều luồng video trong 1 giao diện , sử dụng module Threading.

1. Cảnh báo tới người giám sát thông qua giao diện

Cách đơn giản nhất để thông báo cho người giám sát là hiện lời nhắc cảnh báo ngay trên màn hình.

Như đã trình bày ở trên, máy khách phát hiện ra đối tượng sẽ tính toán và đưa ra giá trị dự đoán và gửi giá trị này cùng khung hình về server. Và với mô hình mạng như trong Chương I, máy chủ hoàn toàn có thể biết được địa chỉ cụ thể của máy khách nào đang phát hiện và xác định địa chỉ cụ thể của máy khách sẽ phát hiện được đối tượng trong tương lai theo giá trị dự đoán.

Để đơn giản, hoàn toàn có thể dùng cấu trúc switch-case để so sánh và tìm ra cảnh báo phù hợp cho từng trường hợp cụ thể và in nó ra màn hình được chỉ định.



Hình 12: Thử nghiệm kết quả dự đoán

Trong hình ảnh trên thử nghiệm với đối tượng là người phụ nữ. Khi cặp giá trị (x,y) của bounding box quanh người phụ nữ có giá trị x tăng, y thay đổi không đáng kể sẽ xuất hiện thông báo “Get close to the man”.

# CHƯƠNG IV: TỔNG KẾT

# **\*Tài Liệu Tham Khảo**