# Tổng quan thiết kế dự án

### Đặt vấn đề

Hiện nay công nghệ thông mình đang phát triển rất mạnh mẽ. Chẳng hạn có thể kể đến như tivi điều khiển bằng giọng nói, hay các thiết bị được điều khiển thông qua ứng dụng di động,… Với sự phát triển mạnh mẽ của công nghệ IoT và trí tuệ nhân tạo, việc điều khiển thiết bị ngày càng trở nên đa dạng hơn chứ không gói gọn ở việc điều khiển bằng tay thông thường. Có thể nói công nghệ trí tuệ nhân tạo đã góp phần quan trọng làm cho các thiết bị trở nên thông minh hơn.

Chúng ta có thể nghe nhiều tới thiết bị điều khiển bằng giọng nói. Tuy nhiên, việc điều khiển thiết bị bằng cử chỉ có vẻ khá xa lạ. Bằng cử chỉ của bàn tay ta có thể ứng dụng vào giải quyết rất nhiều vấn đề. Có thể kể tới những lĩnh vực như ứng dụng cho người khuyết tật (ngôn ngữ cử chỉ), điều khiển thiết bị trong nhà thông minh,… Chính vì những ứng dụng này mà nhóm em đã quyết định thực hiện đề tài liên quan đến điều khiển thiết bị thông qua cử chỉ bàn tay để điều khiển các thiết bị thông mình. Mô tả sơ bộ về dự án sẽ trình bày chi tiết ở phần sau.

### Mô tả bài toán từ góc độ người dùng

Ở dự án này, thông qua việc nhận dạng cử chỉ của bàn tay, người dùng có thể điều khiển các thiết bị trong nhà như đèn điện, tivi, điều hòa…

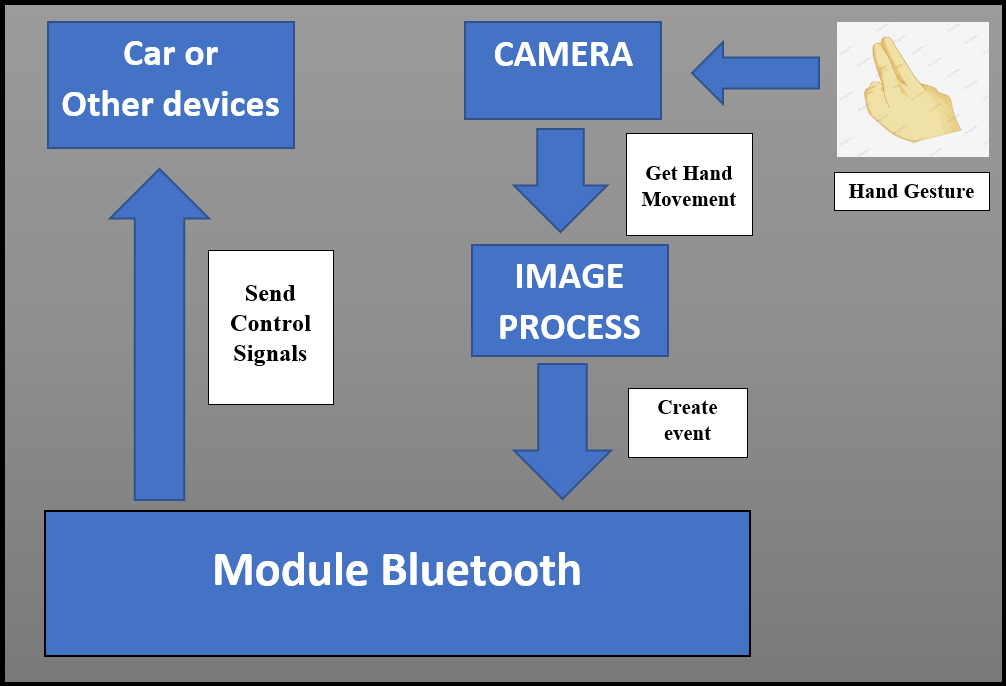
Một ví dụ thực tế: Người dùng đứng cách xa tivi 3m. Trên màn hình tivi có hình một con trỏ chuột. Mỗi khi ngón tay người dùng di chuyển, con trỏ chuột trên màn hình sẽ di chuyển theo. Mỗi khi người dùng nháy đúp hai lần trên ngón tay của mình, con trỏ chuột trên màn hình cũng sẽ click hai lần liên tiếp. Việc này khiến cho người dùng thao tác với màn hình tivi giống như 1 chiếc màn hình cảm ứng, tuy nhiên người dùng không cần chạm ngón tay trực tiếp vào màn hình, thay vào đó người dùng có thể đứng từ xa và làm việc đó.

### Ứng dụng thực tế của bài toán

Hiện nay  các thiết bị thông minh đang hướng đến việc loại bỏ các thiết bị điều khiển. Thay vào đó chúng sử dụng trực tiếp các khả năng có sẵn của con người như giọng nói, ngón tay,... Việc điều khiển thiết bị qua cử động bàn tay hoặc giọng nói sẽ là xu hướng của tương lai.

Để làm một Demo như một minh chứng cho việc ứng dụng nhận diện cử chi tay trong cuộc sống hằng ngày. Nhóm đã quyết định thực hiện điều khiển xe thông qua cử chỉ tay với dữ liệu được truyền thông qua bluetooth hoặc wifi.

### Sơ đồ khối của hệ thống



Hệ thống sẽ gồm camera nhận dạng chuyển động của bàn tay, từ đó tạo ra sự kiện, sau đó gửi tín hiệu tương ứng với từng cử chỉ qua module bluetooth để điều khiển xe di chuyển.

## Thiết kế hệ thống

### Phân tích sơ bộ hệ thống

Hệ thống có thể được chia làm hai module chính là : Tầng ứng dụng xử lý chuyển động của bàn tay và module điều khiển xe với dữ liệu được gửi qua bluetooh.

Ở module xử lý chuyển động của bàn tay, camera sẽ nhận dữ liệu hình ảnh là video stream trực tiếp của camera. Sau đó sẽ trích xuất video thành các frame ảnh và thực hiện dectect cử chỉ của tay. Với mỗi cử chỉ tay như vậy sẽ tương ứng với một tín hiệu là số 1, 2 ,3 … Sau khi thu được tín hiệu một cách chính xác. Module này sẽ tiếp tục thực hiện pair với bluetooth và gửi dữ liệu qua nó.

Module thứ hai là module điều khiển xe.. Ở module này, nó sẽ lắng nghe liên tục thông điệp mà module xử lý chuyển động bàn tay qua socket. Với mỗi tín hiệu thu được ta sẽ thực hiện các chức năng mong muốn như điều khiển xe tiến lùi, quay trái, quay phải, tăng giảm tốc độ xe.

### Thiết kế xử lý chuyển động bàn tay

#### Phương hướng giải quyết

Hiện nay, lĩnh vực thị giác máy tính đang phát triển rất mạnh. Hàng loạt các thuật toán xử lý ảnh tiên tiến đã ra đời. Cùng với đó là việc phát triển mạnh của deep learning đã góp phần rất lớn vào giải quyết bài toán theo dõi và nhận diện chuyển động. Ở bài toán này, để đơn giản thì ta sẽ chia thành hai vấn đề chính là theo dõi (tracking) theo chuyển động của bàn tay và nhận diện chuyển động để tạo ra các sự kiện gửi cho module điều khiển xử lí.

Vấn đề theo dõi chuyển động của bàn tay có nhiều phương hướng giải quyết. Có thể đơn giản sử dụng một số thuật toán xử lý ảnh phổ biến như Color object detect, Background subtraction,… Ta phải đánh giá tính khả thi của các thuật toán để đưa ra lựa chọn thích hợp phù hợp với ứng dụng.

Ở vấn đề còn lại, nhận diện chuyển động của bàn tay ta cũng có nhiều phương án giải quyết. Xử lý ảnh đơn giản ta có thể tìm được bao quanh bàn tay hay còn gọi là contour. Từ đó xử lý và đưa ra những dự đoán về chuyển động như một ngón tay, hai ngón tay hoặc nắm đấm,… Ngoài ra để nâng cao độ chính xác ta có thể sử dụng những giải thuật deep learning phổ biến trong công nghệ nhận dạng ảnh như Convolutional neural networks (CNNs).

#### Phân tích ưu nhược điểm của các phương án

Thứ nhất là việc lựa chọn giữa color object detect và background subtraction. Ở giải thuật Color object detect, ta sẽ theo dõi vật dựa trên màu của vật thể, ở đây là bàn tay người, Bằng việc sử dụng phổ màu HSV ta có thể nhận biết được vật thể đang ở đâu trong ảnh và từ đó xác định được ô bao quanh vật. Tuy nhiên thuật toán sẽ mất đi độ chính xác khi trong ảnh có vật gì đó màu giống da của người, lúc đó việc dựa theo màu sẽ không còn ý nghĩa. Còn ở giải thuật Background subtraction ta sử dụng một ảnh nền và với mỗi frame mới ta tiến hành trừ đi ảnh nền đó để phát hiện ra vật chuyển động. Về cơ bản thuật toán hoạt động khá tốt, chỉ có tồn tại là thuật toán nhạy với ảnh nắng, khi bàn tay che mất ánh nắng thì ánh sáng thay đổi, thuật toán sẽ hoạt động không tốt.

Về giải thuật nhận diện chuyển động bàn tay bằng cách tìm contour và nhận dạng sự thay đổi của contour để đưa ra chuyển động, ta thấy ưu điểm nổi trội đó là đơn giản, chi phí bộ nhớ thấp. Nhưng bù lại độ chính xác không cao bằng sử dụng giải thuật Deep Learning. Ở giải thuật Deep Learning, ta có thể thấy được ưu thế vượt trội ở độ tin cậy với kĩ thuật nhận dạng với độ chính xác cao. Tuy nhiên sẽ tiêu tốn một chút tài nguyên trong việc tính toán. Mặc dù vậy, với công nghệ phần cứng mạnh như hiện nay thì đây có thể không là vấn đề nghiêm trọng.

#### Lựa chọn giải thuật

Sau khi phân tích ưu nhược điểm của từng giải thuật, vì mục đích đơn giản và tối ưu hóa cho ứng dụng nhóm đã lựa chọn giải thuật cho từng vấn đề. Ở phần theo dõi chuyển động bàn tay nhóm sẽ sử dụng giải thuật background subtraction và vấn đề nhận dạng chuyển động sẽ được giải quyết bằng một mạng CNNs.

#### Platform và công cụ giải quyết

Để ứng dụng những giải thuật đã lựa chọn ở trên, cần lựa chọn platform, công cụ được hỗ trợ nhiều hiện nay. Có thể kể đến là bộ thư viện opencv, các framework hỗ trợ Deep Learning như pytorch, keras, tensorflow,.. Để thuận tiện trong việc phát triển thì nhóm sẽ sử dụng Opencv trên nền tảng ngôn ngữ python để giải quyết cả vấn đề đặt ra. Việc các framework Deeplearning được hỗ trợ mạnh trên ngôn ngữ python nên python sẽ là một lựa chọn không tồi. Ngoài ra hiện nay mã nguồn mở cho xử lý ảnh trên nền tảng ngôn ngữ python rất nhiều nên ta có thể được hỗ trợ nhiều trong quá trình phát triển, bảo trì.

Như vậy nhóm đã thống nhất lựa chọn :

* Nền tảng phát triển : Linux.
* Thư viện hỗ trợ : OpenCv.
* Framework hỗ trợ : keras, tensorflow, scikit-learn
* Ngôn ngữ lập trình: Python

## Các bước triển khai

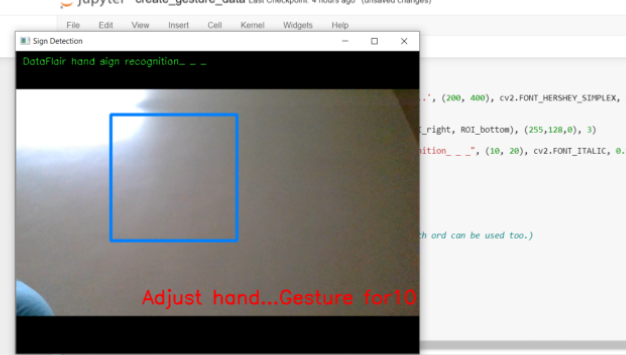
#### Điều kiện tiên quyết

Các thư viện và phần mềm cần thiết dùng cho dự án:

* Python (3.7.4)
* IDE (Jupyter)
* Numpy (version 1.16.5)
* cv2 (openCV) (version 3.4.2)
* Keras (version 2.3.1)
* Tensorflow (version 2.0.0)

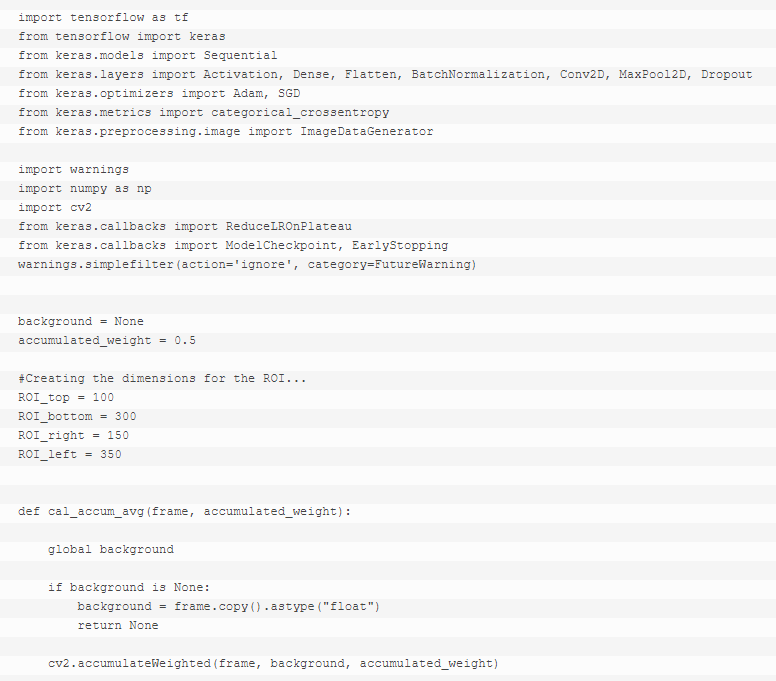
#### Tạo dataset cho Project nhận diện cử chỉ tay.

#### Chúng ta sẽ lấy dữ liệu từ video cam và mọi khung hình phát hiện bàn tay trong ROI (vùng quan tâm) được tạo sẽ được lưu trong thư mục cử chỉ.

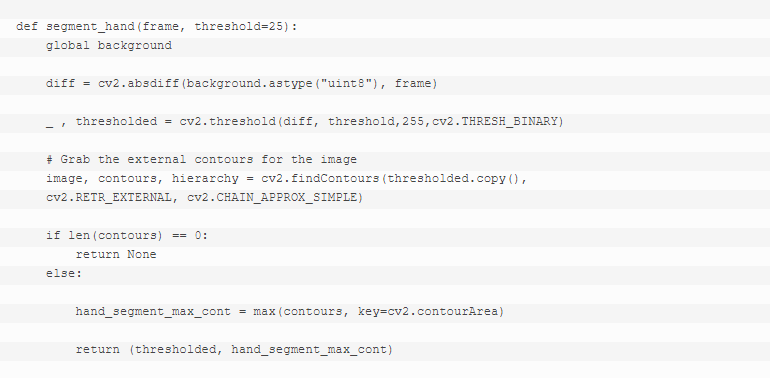
Bây giờ để tạo tập dữ liệu, chúng ta lấy nguồn cấp dữ liệu cam trực tiếp bằng OpenCV.

Để phân biệt giữa nền, chúng ta tính toán trung bình có trọng số tích lũy cho background và sau đó trừ giá trị này khỏi các frame chứa một số đối tượng ở trước background . Có thể được coi là foreground.

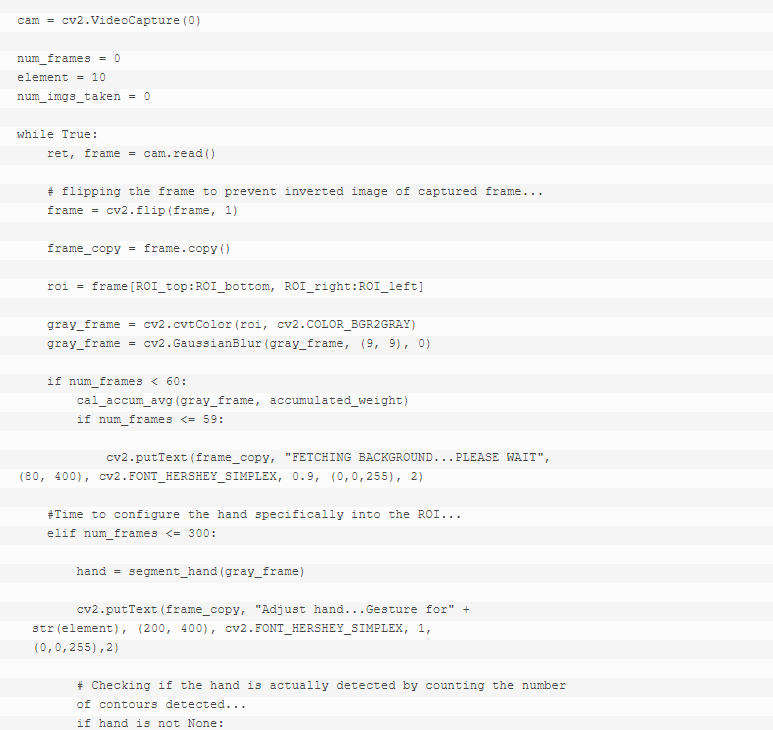
Sau khi có trung bình tích lũy cho background , chúng ta trừ nó khỏi mỗi frame mà chúng ta đọc sau 60 khung hình để tìm bất kỳ đối tượng nào bao phủ background .

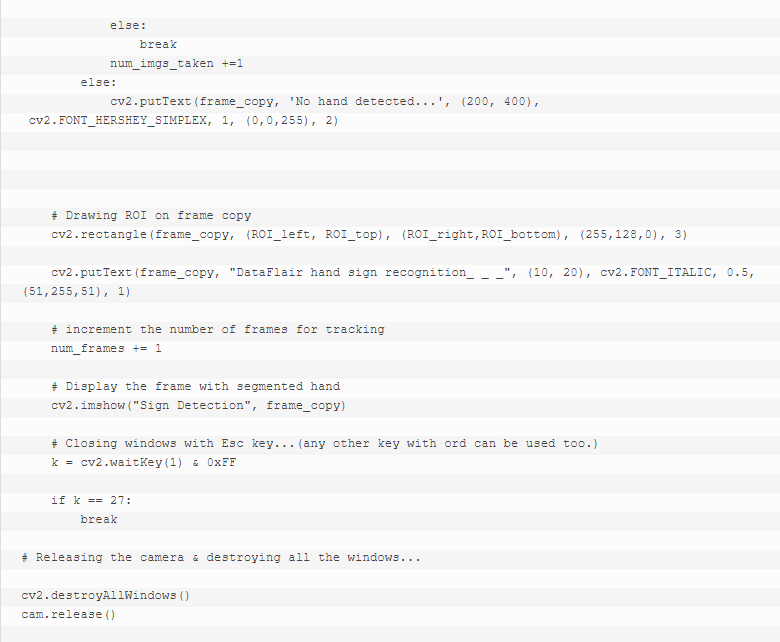


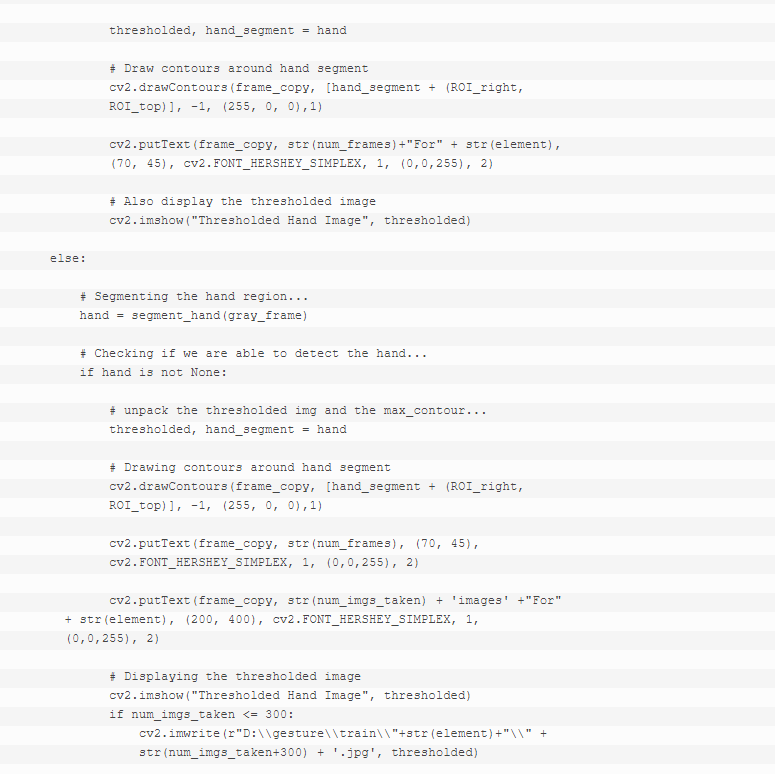
Bây giờ chúng ta tính toán giá trị threshold cho mọi frame và xác định các đường bao bằng cv2.findContours và trả về các đường viền tối đa (các đường bao ngoài cùng nhất cho đối tượng) thông qua hàm segment\_hand().



Khi các đường bao được phát hiện (hoặc bàn tay hiện diện trong ROI), Chúng ta bắt đầu lưu hình ảnh của ROI trong bộ data\_train và data\_test tương ứng cho chữ cái hoặc số mà chúng ta đang phát hiện.



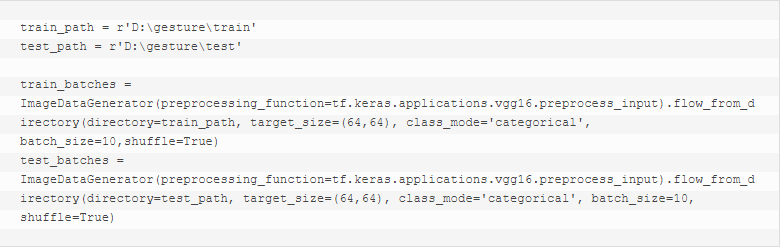




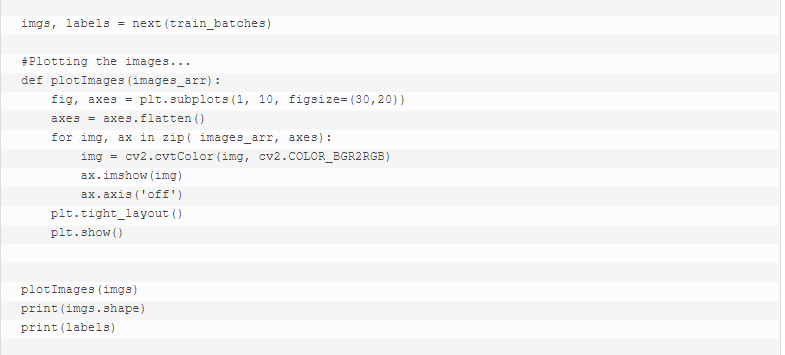
#### Training CNN.

Bây giờ trên tập dữ liệu đã tạo, chúng ta sẽ train một mạng CNN.

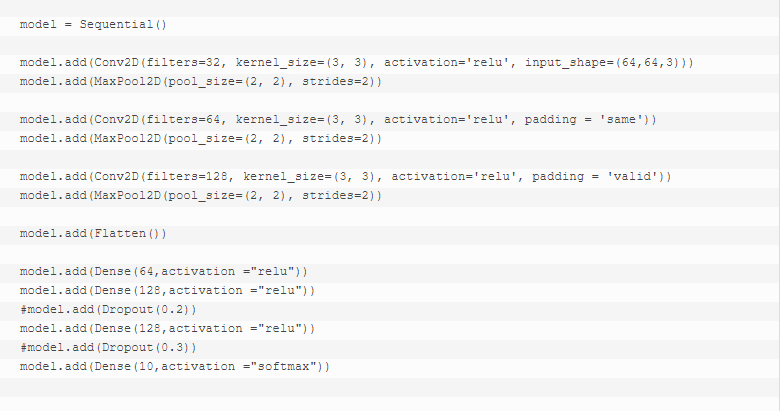
Đầu tiên, chúng ta tải dữ liệu bằng ImageDataGenerator của keras, qua đó chúng ta có thể sử dụng hàm flow\_from\_directory để tải dữ liệu train và test, và mỗi tên của các thư mục sẽ là tên class cho các tệp tin được tải.

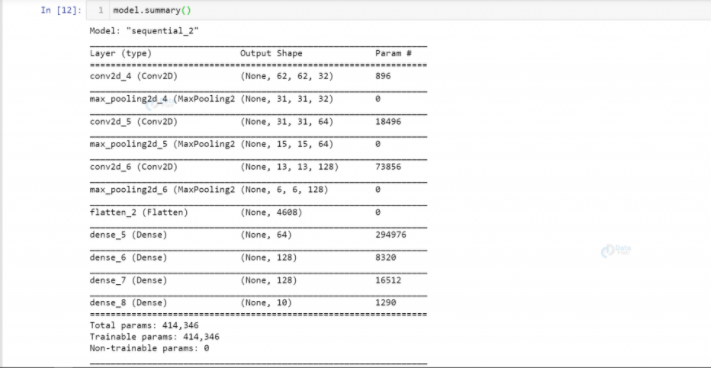


Hàm plotImages dùng để vẽ các hình ảnh của tập dữ liệu được tải.



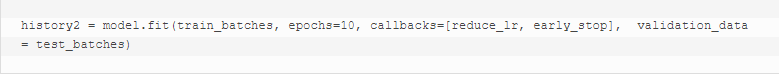
Bây giờ chúng ta thiết kế CNN như sau (hoặc tùy thuộc vào một số thử nghiệm và lỗi có thể sử dụng các siêu tham số khác)



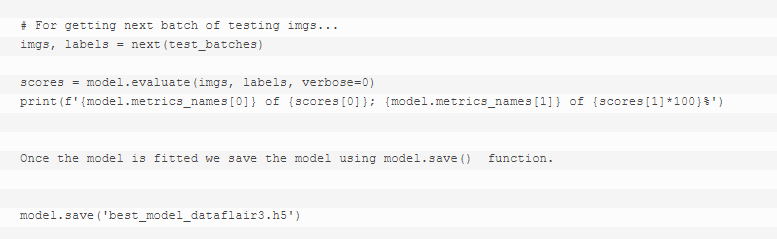


Ngoài ra, chúng ta sử dụng hai thuật toán tối ưu hóa khác nhau - SGD (có nghĩa là trọng số được cập nhật ở mọi trường hợp train) và Adam.

Sau đó chúng ta sẽ thực hiện trainning model trên bộ dữ liệu data\_train với 10 chu kì (có thể thay đổi tùy theo sự lựa chọn thông số của người dùng).

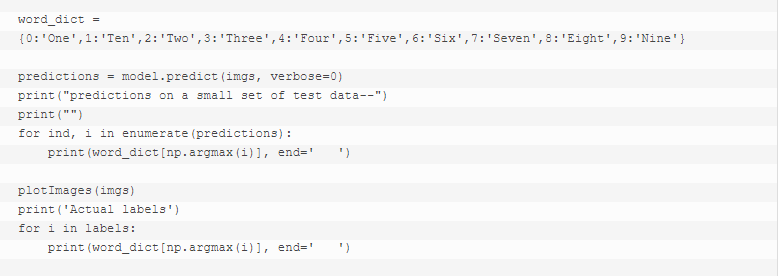


Bước tiếp là thực hiện đánh giá model và lưu lại.



Bước cuối cùng là tiến hành thử nghiệm mô hình để kiểm tra xem mọi thứ có hoạt động như chúng ta mong đợi hay không.

Word\_dict là từ điển chứa tên các nhãn khác nhau dùng để dự đoán.

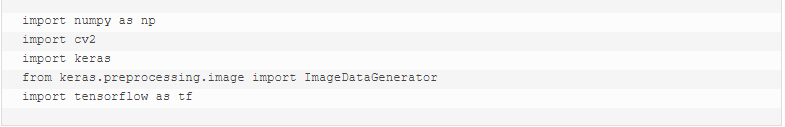


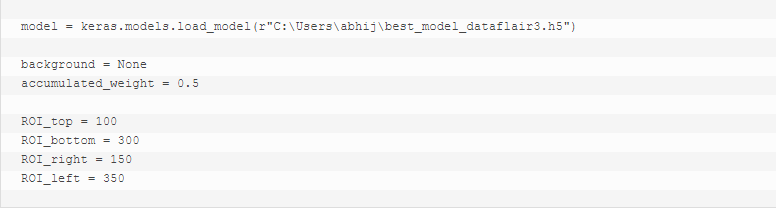
#### Dự đoán cử chỉ tay.

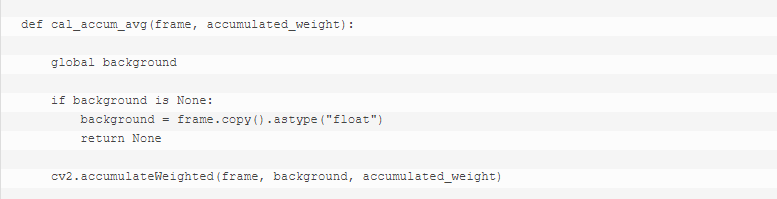
Trong trường hợp này, chúng ta tạo một vùng giới hạn để phát hiện ROI và tính toán trọng số trung bình như khi tạo tập dữ liệu. Điều này được thực hiện để xác định bất kỳ đối nào ở foreground.

Bây giờ chúng ta tìm đường viền tối đa và nếu đường viền được phát hiện có nghĩa là một bàn tay được phát hiện.

Chúng ta tải mô hình đã lưu trước đó bằng keras.models.load\_model và cung cấp hình ảnh bàn tay làm đầu vào cho mô hình để dự đoán.







Phát hiện bàn tay trực tiếp từ cam.

