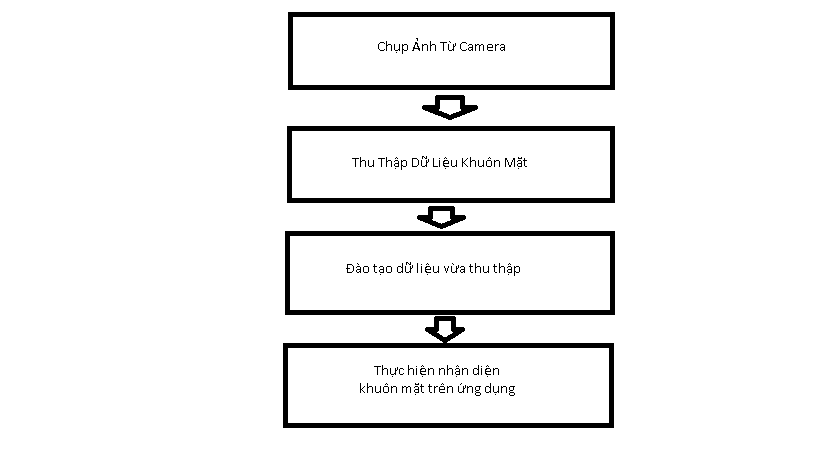
Đề tài

Xây Dựng Ứng Dụng Nhận Diện Khuôn Mặt

1. Ý Tưởng:



1. Gen code

Nếu bạn có L hình ảnh chế độ, điều đó có nghĩa đó là hình ảnh kênh đơn - thường được hiểu là thang độ xám. Điều Lđó có nghĩa là chỉ lưu trữ Độ chói. Nó rất nhỏ gọn, nhưng chỉ lưu trữ một thang độ xám, không phải màu.

<https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php>

<https://stackoverflow.com/questions/54727642/what-does-roi-gray-function-do-in-opencv>

3.Lý thuyết máy học MTCNN

-MTCNN là viết tắt của Multi-task Cascaded Convolutional Networks. Nó là bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi detect khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong task. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng…

## -Tổng quan về MTCNN (Multi-task Cascaded Convolutional Networks):

### 1.1 Stage 1: P-Net

Trước hết, một bức ảnh thường sẽ có nhiều hơn một người - một khuôn mặt. Ngoài ra, những khuôn mặt thường sẽ có kích thước khác nhau. Ta cần một phương thức để có thể nhận dạng toàn bộ số khuôn mặt đó, ở các kích thước khác nhau. MTCNN đưa cho chúng ta một giải pháp, bằng cách sử dụng phép Resize ảnh, để tạo một loạt các bản copy từ ảnh gốc với kích cỡ khác nhau, từ to đến nhỏ, tạo thành 1 **Image Pyramid.**

## Ảnh có chứa văn bản Mô tả được tạo tự động

Với mỗi một phiên bản copy-resize của ảnh gốc, ta sử dụng **kernel** 12x12 pixel và stride = 2 để đi qua toàn bộ bức ảnh, dò tìm khuôn mặt. Vì các bản copies của ảnh gốc có kích thước khác nhau, cho nên mạng có thể dễ dàng nhận biết được các khuôn mặt với kích thước khác nhau, mặc dù chỉ dùng 1 kernel với kích thước cố định (Ảnh to hơn, mặt to hơn; Ảnh nhỏ hơn, mặt nhỏ hơn). Sau đó, ta sẽ đưa những kernels được cắt ra từ trên và truyền qua mạng P-Net (Proposal Network). Kết quả của mạng cho ra một loạt các bounding boxes nằm trong mỗi kernel, mỗi bounding boxes sẽ chứa tọa độ 4 góc để xác định vị trí trong kernel chứa nó (đã được normalize về khoảng từ (0,1)) và điểm confident (Điểm tự tin) tương ứng.

## 

## Để loại trừ bớt các bounding boxes trên các bức ảnh và các kernels, ta sử dụng 2 phương pháp chính là lập mức Threshold confident - nhằm xóa đi các box có mức confident thấp và sử dụng NMS (Non-Maximum Suppression) để xóa các box có tỷ lệ trùng nhau (Intersection Over Union) vượt qua 1 mức threshold tự đặt nào đó. Hình ảnh dưới đây là minh họa cho phép NMS, những box bị trùng nhau sẽ bị loại bỏ và giữ lại 1 box có mức confident cao nhất.

## Ảnh có chứa văn bản, tạo dáng Mô tả được tạo tự động

## Sau khi đã xóa bớt các box không hợp lý, ta sẽ chuyển các tọa độ của các box về với tọa độ gốc của bức ảnh thật. Do tọa độ của box đã được normalize về khoảng (0,1) tương ứng như kernel, cho nên công việc lúc này chỉ là tính toán độ dài và rộng của kernel dựa theo ảnh gốc, sau đó nhân tọa độ đã được normalize của box với kích thước của của kernel và cộng với tọa độ của các góc kernel tương ứng. Kết quả của quá trình trên sẽ là những tọa độ của box tương ứng ở trên ảnh kích thước ban đầu. Cuối cùng, ta sẽ resize lại các box về dạng hình vuông, lấy tọa độ mới của các box và feed vào mạng tiếp theo, mạng R.

### 1.2 Stage 2: R-Net

## 

## Mạng R (Refine Network) thực hiện các bước như mạng P. Tuy nhiên, mạng còn sử dụng một phương pháp tên là **padding**, nhằm thực hiện việc chèn thêm các zero-pixels vào các phần thiếu của bounding box nếu bounding box bị vượt quá biên của ảnh. Tất cả các bounding box lúc này sẽ được resize về kích thước 24x24, được coi như 1 kernel và feed vào mạng R. Kết quả sau cũng là những tọa độ mới của các box còn lại và được đưa vào mạng tiếp theo, mạng O.

### 1.3 Stage 3: O-Net

## 

## Cuối cùng là mạng O (Output Network), mạng cũng thực hiện tương tự như việc trong mạng R, thay đổi kích thước thành 48x48. Tuy nhiên, kết quả đầu ra của mạng lúc này không còn chỉ là các tọa độ của các box nữa, mà trả về 3 giá trị bao gồm: 4 tọa độ của bounding box (out[0]), tọa độ 5 điểm landmark trên mặt, bao gồm 2 mắt, 1 mũi, 2 bên cánh môi (out[1]) và điểm confident của mỗi box (out[2]). Tất cả sẽ được lưu vào thành 1 dictionary với 3 keys kể trên.

## Ảnh có chứa văn bản, trong nhà, tìm kiếm Mô tả được tạo tự động

## -Thuật toán LBP trong FaceRecognizer

## -LBP (Các mẫu nhị phân cục bộ) là một cách để trích xuất các tính năng đặc trưng của một đối tượng (Có thể là khuôn mặt, cốc cà phê hoặc bất cứ thứ gì có hình đại diện). Thuật toán của LBP thực sự rất đơn giản và có thể được thực hiện thủ công. (ngưỡng pixel + phép toán số học cấp pixel.)

## 

## -OpenCV chuyển đổi hình ảnh LBP thành biểu đồ để lưu trữ thông tin không gian của các khuôn mặt.Trong đó chia hình ảnh LBP thành các vùng cục bộ có kích thước m và trích xuất biểu đồ cho từng vùng và nối chúng.

## -Sau khi có thông tin về khuôn mặt của một người (1 nhãn), phần còn lại rất đơn giản. Trong quá trình suy luận, nó tính toán LBP của mặt thử nghiệm, chia các vùng và tạo biểu đồ. Sau đó, so sánh khoảng cách euclidian của nó với biểu đồ của các khuôn mặt được huấn luyện. Nếu nó nhỏ hơn giá trị dung sai, nó được tính là khớp. (Cũng có thể sử dụng các phương pháp khoảng cách khác, giá trị tuyệt đối của khoảng cách chi bình phương, v.v.)

## -Haar Cascade:

### Haar Cascade là gì?

Haar Cascade là một thuật toán được tạo ra dựa trên những tính năng đó để phát hiện đối tượng (có thể là khuôn mặt, mắt, tay, đồ vật,…) được đề xuất vào năm 2001 bởi **Paul Viola** và **Michael Jones** trong bài báo của họ với khẳng định “Phát hiện đối tượng một cách nhanh chóng bằng cách sử dụng tầng (Cascade) tăng cường các tính năng đơn giản”.

Triển khai ban đầu được sử dụng để phát hiện khuôn mặt chính diện và các đặc điểm như Mắt, Mũi và Miệng. Tuy nhiên, có nhiều đặc trưng Haar được đào tạo trước đó trong **GitHub** của họ cho các đối tượng khác cũng như cho toàn bộ cơ thể, thân trên, thân dưới, nụ cười và nhiều đồ vật khác.

Nói một cách dễ hiểu hơn, Haar Cascade là gì? Là một lớp model có thể giúp chúng ta nhận diện khuôn mặt (Haar Cascade face detection) Haar Cascade sử dụng các tầng Haar và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (Cascade) và tạo thành **một cỗ máy nhận diện** khuôn mặt hoàn chỉnh.

### Haar Cascade hoạt động như thế nào

Trong bài báo của tác giả *Paul Viola và Michael Jones* đã trình bày một phương pháp mới và nhanh hơn để xử lý hình ảnh và phát hiện khuôn mặt bằng cách sử dụng các đặc điểm hình chữ nhật như hình dưới đây. Các **đặc điểm hình chữ nhật** tương tự như nhân được sử dụng để phát hiện các đặc điểm khác nhau của khuôn mặt như mắt và các nốt như trong hình minh họa.



Các tính năng hình chữ nhật được chạy lần lượt trên hình ảnh và tổng số pixel nằm trong phần màu trắng được trừ cho tổng số pixel nằm trong phần màu đen.

**Trong đó**

* a) Là các bộ lọc bắt các cạnh trong ảnh
* b) Bắt các đường thẳng trong ảnh
* c) Về đặc trưng 4 hình vuông

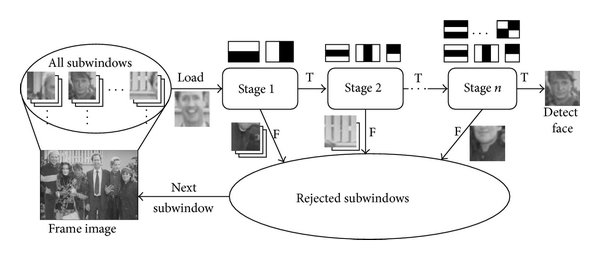
Trong hình minh họa bên dưới, đặc điểm hình chữ nhật đầu tiên đang tính toán cho sự khác biệt về cường độ giữa vùng mắt và vùng má trên khuôn mặt. Và đặc điểm hình chữ nhật thứ hai là **đo sự chênh lệch**về cường độ giữa hai vùng mắt và sống mũi. Bộ lọc Haar chỉ có thể nhìn cụ thể vào một vùng trong cửa sổ để nhận diện.



Vì vậy, sử dụng các đặc điểm hình chữ nhật này trên một hình ảnh, chúng ta có thể tạo ra hàng nghìn **điểm đặc trưng** cho hình ảnh. Tuy nhiên, việc tính toán tổng số pixel trong các vùng trắng và đen trong toàn bộ ảnh có thể là một hoạt động tốn kém, đặc biệt là đối với các ảnh lớn.

Các tác giả cũng đề xuất một phương pháp được gọi là **ảnh tích phân** có thể đạt được tính toán tương tự bằng cách thực hiện các phép toán chỉ trên bốn điểm ảnh. Có thể sử dụng hàng nghìn hình ảnh được gắn nhãn để chuyển đổi chúng thành bản đồ tính năng tầng HAAR và đào tạo nó bằng cách sử dụng **mô hình học máy** để tạo bộ phân loại nhận diện khuôn mặt.

Để hiểu hơn về cách Haar Cascade hoạt động thì mình sẽ đi sâu hơn về từng bước nhận diện khuôn mặt của Haar Cascade nhé!!!



* **Bước 1:** Hình ảnh (đã được gửi đến bộ phân loại) được **chia thành các phần nhỏ**(hoặc các cửa sổ con như trong hình minh họa).
* **Bước 2:** Chúng tôi đặt N không có bộ dò theo cách xếp tầng trong đó mỗi bộ phát hiện sự kết hợp của các loại **đặc trưng khác nhau** từ các hình ảnh (ví dụ: đường thẳng, cạnh, hình tròn, hình vuông) được truyền qua. Giả sử khi việc trích xuất đối tượng địa lý được thực hiện, mỗi phần phụ được gán một **giá trị tin cậy**.
* **Bước 3:** Hình ảnh (hoặc hình ảnh phụ) có độ tin cậy cao nhất được phát hiện dưới dạng khuôn mặt và được gửi đến **bộ tích lũy** trong khi phần còn lại bị từ chối. Do đó, Cascade tìm nạp khung hình / hình ảnh tiếp theo nếu còn lại và bắt đầu lại quá trình.

### 1.1. Dự đoán giới tính

Họ đã đóng khung Dự đoán Giới tính như một bài toán phân loại. Lớp đầu ra trong mạng dự đoán giới tính thuộc loại softmax với 2 nút biểu thị hai lớp “Nam” và “Nữ”.

### 1.2. Dự đoán độ tuổi

Tốt nhất, nên tiếp cận Dự đoán tuổi như một bài toán Hồi quy vì chúng ta đang mong đợi một số thực làm đầu ra. Tuy nhiên, ước tính tuổi chính xác bằng cách sử dụng hồi quy là một thách thức. Ngay cả con người cũng không thể dự đoán chính xác tuổi dựa trên cách nhìn của một người. Tuy nhiên, chúng tôi có một ý tưởng về việc họ đang ở độ tuổi 20 hay 30. Vì lý do này, khôn ngoan hơn nếu đặt vấn đề này là một bài toán phân loại trong đó chúng ta cố gắng ước tính nhóm tuổi của người đó. Ví dụ: độ tuổi trong phạm vi 0-2 là một nhóm duy nhất, 4-6 là nhóm tuổi khác lớp học và như vậy.

Tập dữ liệu Đối tượng có 8 lớp được chia thành các nhóm tuổi sau [(0 - 2), (4 - 6), (8 - 12), (15 - 20), (25 - 32), (38 - 43), ( 48 - 53), (60 - 100)]. Do đó, mạng dự đoán tuổi có 8 nút trong lớp softmax cuối cùng cho biết các độ tuổi được đề cập.