**Thực hiện tiền xử lý ảnh**

1. Chuẩn bị dữ liệu.

Để bắt đầu với việc tiền xử lý ảnh, ta sẽ thực hiện việc thu thập ảnh. Đề tài là nhận diện 10 đối tượng bất kì để nhận diện khuôn mặt vậy nên ta cần thu thập ảnh chân dung của 10 người khác nhau. Và 10 đối tượng được chọn là:

* Albert Eintein
* Dmitri Mendeleev
* Galileo Galilei
* Issac Newton
* Marie Curie
* Michael Faraday
* Nicola Tesla
* Otto Hahn
* Thomas Edison
* Alexander Fleming

Ta sẽ có 1 tập datasets trong đó chia là 2 tập train\_data và test\_data, mỗi tập sẽ bao gồm 10 đối tượng trên và trong mỗi đối tượng có từ 15 đến 20 tấp ảnh chân dung của họ.

1. Lọc ảnh.

Ở bước này, ta thực hiện lọc (crop) các khuôn mặt từ các bức ảnh và cho các khuôn mặt đó vào trong một tập dữ liệu khác.

Để có thể cắt được khuôn mặt từ trong một tấp ảnh, trước hết ta phải thực hiện detect được khuôn mặt trong tập ảnh. Để làm được điều này ta sẽ chọn phương pháp phát hiện khuôn mặt phổ biến gọi là phương pháp Haar\_cascade.

Vậy thế nào là phương pháp Haar\_cascade?

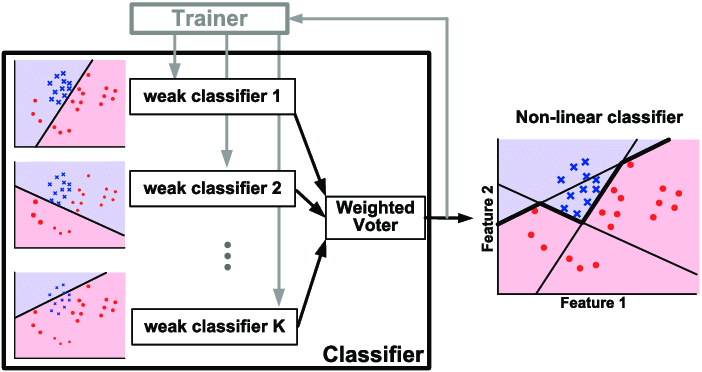
Haar\_cascade là một mô module phát hiện đối tượng do OpenCV trong một dự án viết ra, trong đó có “**haarcascade\_frontalface\_default.xml**” là một model được xây dựng và đóng gói thành một file .xml, model này được sử để hổ trợ để nhận diện khuôn mặt con người.

Về cách hoạt động của model Haar\_cascade:

Nói một cách đơn giản là model này sẽ sử dụng các bộ lọc Haar, các bộ lọc này chỉ trích xuất các đặc trưng Haar (các đặc trưng của khuôn mặt), ta sẽ trượt (cascade) bộ lọc này trên một bức ảnh, việc này giống với các trượt của sổ bộ lọc trong mô hình nơ tron tích chập CNN. Qua đó ta có thể trích xuất ta được các đặc điểm trên khuôn mặt từ đó phát hiện được khuôn mặt.

Model sẽ sử dụng một cửa sổ trượt có kích thước 20x20 pixel, trong cửa sổ trước sẽ chứa các bộ lọc Haar khác nhau, bộ lọc Haar cũng không khác bộ lọc bình thường, chỉ là nó sẽ trích suất ra các đặt trưng quan trong của khuôn mặt mà thôi. Ví dụ như bộ lọc cạnh ngang sẽ được đặt ở vị trí giữa con mắt với lông mày vì ở đâu có sự khác này về màu sắc hay bộ lọc cạnh thẳng sẽ đặt ở sống mũi vì nó khác với hai bên má về góc độ ánh sáng hay bộ lọc lấy phần trung tâm đặt ở con mắt, … các bộ lọc như thế sẽ được đặt trong cửa sổ trượt, và cứ mỗi lần di chuyển của sổ trước các bộ lọc sẽ thực thi nhiệm vụ của nó. Nếu tất cả bộ lọc để bắt được hình ảnh có chứa khuôn mặt thì nó sẽ ngừng khung cửa sổ trượt lại và lưu các đầu ra của từng bộ lọc lại, sau đó mới tiếp tục trượt, nếu không thì sẽ bỏ qua và bỏ kết quả các bộ lọc. Sau khi trượt quả một lần và thu lại kết quả, cửa sổ trượt và bộ lọc có thể sẽ tiến hành nhỏ đi để trượt thêm một lần nữa trên kết quả vừa rồi, đến khi bộ lọc nhỏ vừa đủ với kích thước khuôn mặt trên hình thì ngừng lại và cho ra kết quả.

Trên thực tế có rất nhiều loại bộ lọc khoản 160k bộ lọc khác nhau nhưng để chọn một lại bộ lọc tốt thì không dể. Vì lẽ đó thay vì chọn bộ lọc ta sẽ tiến hành cải tiến nó để cho ra kết quả tốt hơn và phương pháp cải tiến là Adaboost. Phương pháp này sẽ thực hiện cải tiến các bộ lọc bằng cách gộp các bộ lọc lại với nhau (đa phần là các bộ lọc tuyến tính) để tạo thành bộ lọc phi tuyến tính có tác dụng lọc tốt hơn.

*Hình 1. Hình ảnh minh họa phương pháp Adaboost*

Để rõ ràng hơn về phương pháp lọc, phát hiện khuôn mặt bằng haar\_cascade, ta có thể xem quá trình thực hiện của nó thông qua clip sau:

https://www.youtube.com/watch?v=L0JkjIwz2II

1. Cải thiện ảnh.

Ở bước này, ta sẽ xử lý các ảnh khuôn mặt cắt được:

Bước 1: Resize các bức ảnh để cho tất cả các ảnh có cùng kích thước.

Bước 2: Lọc nhiễu cho ảnh.

Bước 3: Tăng cường ảnh.

Bước 4: Lưu ảnh vào một tập dữ liệu khác.

Giải thích từng bước:

Trong bước thứ nhất, chúng ta sẽ dừng lệnh resize của thư viện opencv để thực hiện giảm kích thước cũng như đồng bộ về kích thước cho tất cả các ảnh. Tất cả các ảnh sẽ được đưa về cùng kích thước là 200x200x3 (trong đó width = 200, height = 200)

Trong bước thứ hai, ta thực hiện lọc nhiễu cho ảnh, vì ảnh lấy từ trên mạng về cho nên rất dể xảy ra các nhiễu có thế kể đến đa phần là nhiễu muỗi tiêu vậy nên cách hiệu quả trong việc giảm nhiễu, ta dùng bộ lọc trung vị cho ảnh. Với bộ lọc này thư viện opencv cung cấp cho chúng ta hàm medianBlur.

\*Bộ lọc trung vị là một bộ lọc phi tuyến (non linear filters) dùng cho việc lọc nhiễu vết đốm và nhiễu muối tiêu. Cách thực hiện của bộ lọc trung vị, đầu tiên tạo một của sổ lọc 3x3, sau đó cho cửa sổ lọc trượt trên ảnh, nó sẽ lấy giá trị tại từng vị trí, sau đó sắp xếp lại các giá trị của từng pixels theo thứ tự từ nhỏ đến lớn, từ đó lấy 1 giá trị trung bình của dãy các giá trị của pixels sau khi sắp xếp và gán vào các pixel hàm xóm.

Trong bước thứ 3, ta thực hiện tăng cường ảnh bằng bộ lọc sharpen để làm nét các đường viền của ảnh lên. Với bộ lọc này thư viện opencv cung cấp cho chúng ta hàm filter2D.

Sau khi đã hoàn thành được các bước trên ta sẽ cho ra được 2 tập dữ liệu mới là train\_new\_data, và valid\_new\_data, trong mỗi tập dữ liệu này có 10 đối tượng ảnh trên và trong mỗi đối tượng ảnh trên thì có các ảnh khuôn mặt đã được thực hiện tiền xử lý.