ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

Môn học: Tính đoán đa phương tiện

Học kỳ II (2019-2020)

**PHÂN LOẠI TUỔI DỰA TRÊN KHUÔN MẶT**

Sinh viên:   
1. Nguyễn Thịnh Quyền - 18521322

2. Võ Huy Khôi - 18520949

3. Lê Đoàn Nhật Minh - 18521101

Lớp: **CS232.K21**

Giảng viên: Mai Tiến Dũng

**Thành phố Hồ Chí Minh, 06 tháng 08 năm 2020**

**MỤC LỤC**

[**A. LÝ DO CHỌN ĐỀ TÀI** 2](#_Toc47813101)

[**B. MÔ TẢ BÀI TOÁN** 2](#_Toc47813102)

[**I. Input – Output** 2](#_Toc47813103)

[**II. Dataset** 2](#_Toc47813104)

[**C. CÁC PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN** 4](#_Toc47813105)

[**I. Sử dụng machine learning truyền thống** 4](#_Toc47813106)

[**II. Sử dụng deep learing** 12](#_Toc47813107)

# 

# **A. LÝ DO CHỌN ĐỀ TÀI**

Một trong những thách thức đối với facial vision hiện nay không chỉ là phát hiện và khoanh vùng khuôn mặt mà còn phải cho biết thông tin đặc điểm của khuôn mặt. Đây là project tiền đề cho các thiết bị bổ trợ thị giác cho những người bị suy giảm thị lực. Mạng xã hội có thể sử dụng thông tin về tuổi để suy luận tốt hơn về bối cảnh của ảnh.

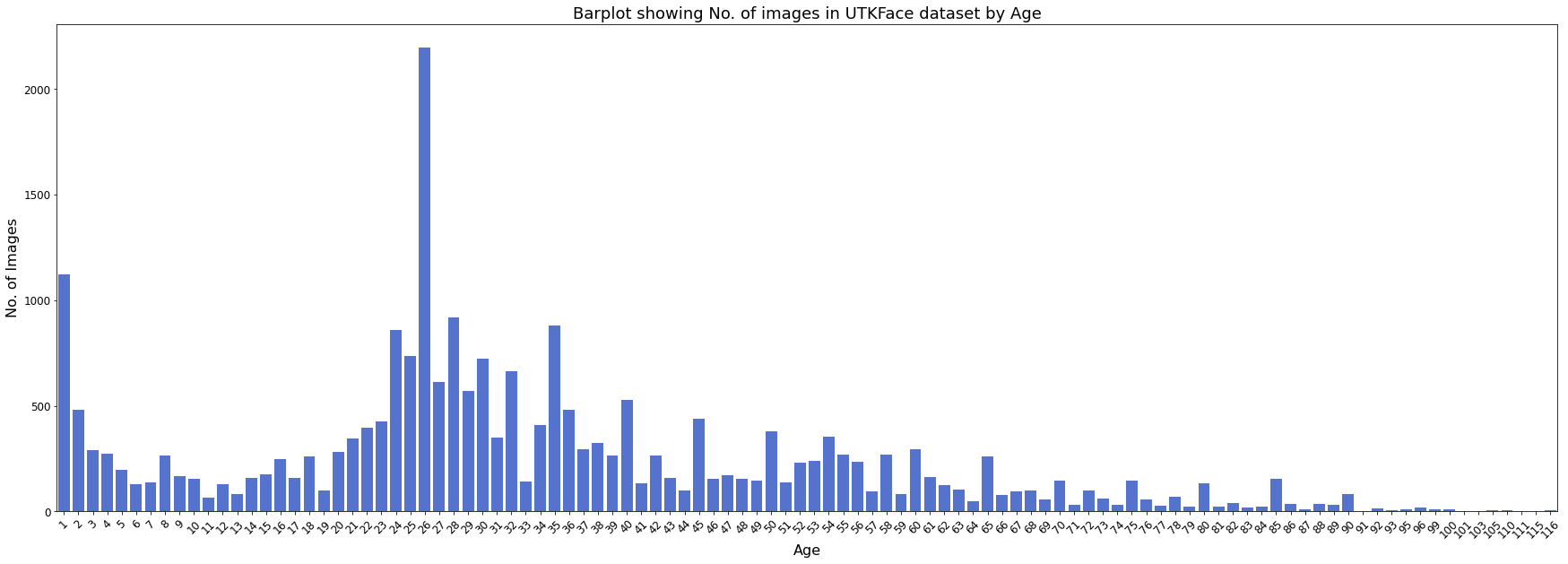
# **B. MÔ TẢ BÀI TOÁN**

## **I. Input – Output**

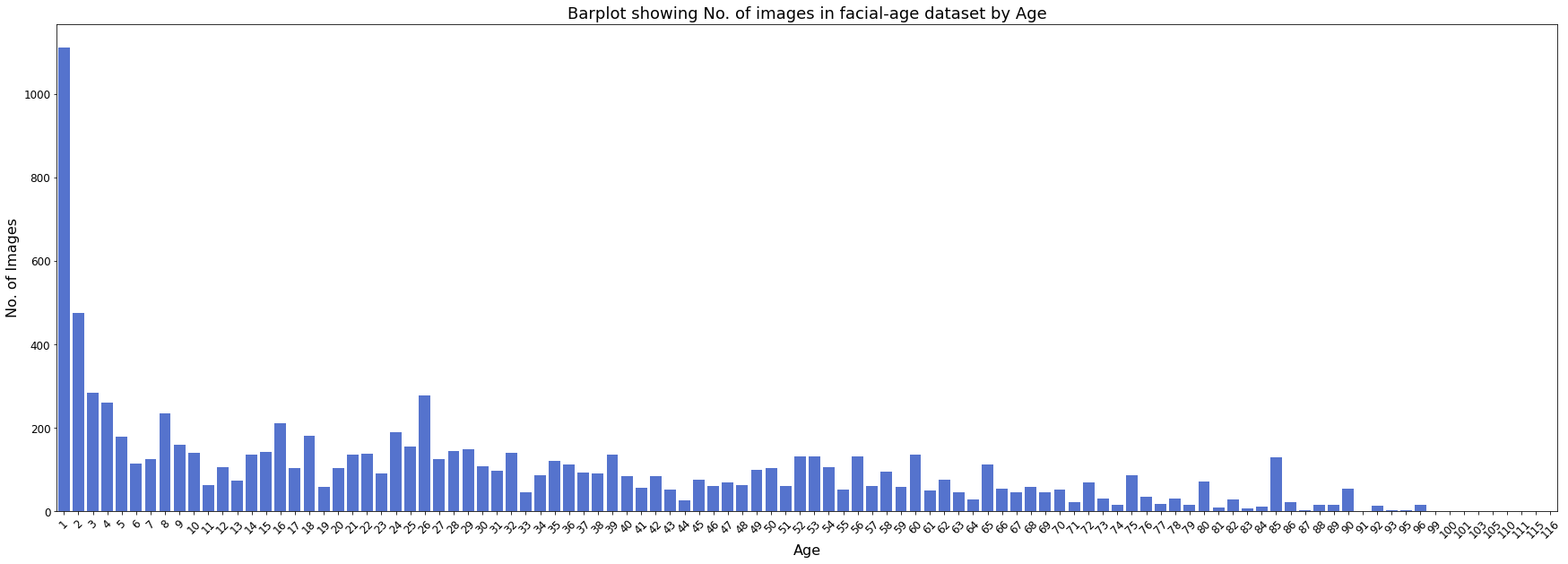
* Input: hình ảnh chứa khuôn mặt của một hoặc nhiều người
* Ouput: bức ảnh với khuôn mặt được khoanh vùng và dự đoán phân khúc tuổi của từng khuôn mặt.

## **II. Dataset**

* UTKFace dataset: 23708 bức ảnh RGB ở định dạng JPG và size ảnh: 200 x 200 pixels với label đã được đính sẵn trong phần tên ảnh.

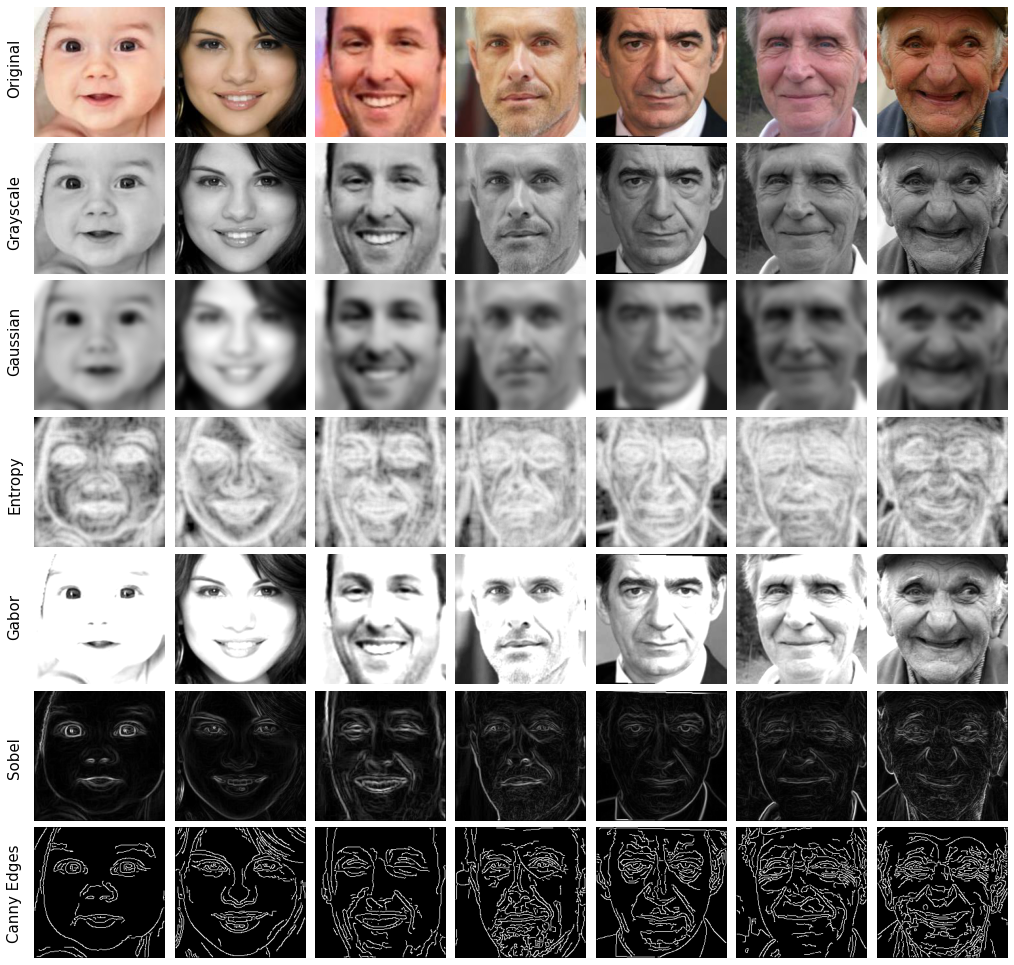


* Facial age dataset của Kaggle: 9788 ảnh RGB của khuôn mặt người với độ tuổi từ 1 – 99. Định dạng: PNG và size ảnh: 200 x 200 pixels.



# **C. CÁC PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN**

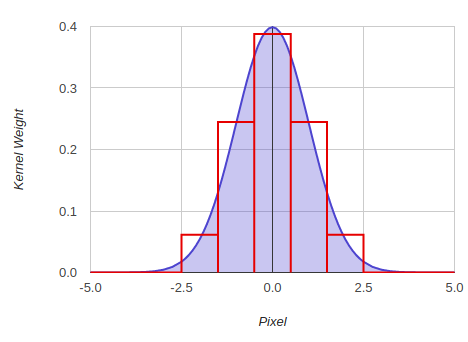
## **I. Sử dụng machine learning truyền thống**

1. Feature extraction

Sau khi thử qua các bộ lọc như Canny Edges, Sobel, Gabor, Gaussian…có thể nhận thấy bộ lọc Canny Edges là bộ lọc thể hiện rõ nhất sự phân hóa giữa những nhóm người ở các độ tuổi từ trẻ đến già (Trẻ thì có ít nếp nhăn hơn già)

**1. Giải thuật tìm cạnh Canny**

Thuật toán Canny (Canny Edge Detection) là một giải thuật phát hiện cạnh nổi tiếng được phát triển năm 1986 bởi John. F Canny. Nó là một giải thuật nhiều bước. Các bước chính:

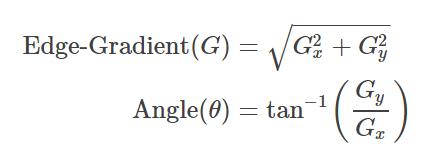
**a. Loại bỏ nhiễu**

Làm mờ ảnh, giảm nhiễu dùng bộ lọc Gaussian kích thước 5x5. Kích thước 5x5 thường hoạt động tốt cho giải thuật Canny. Dĩ nhiên bạn cũng có thể thay đổi kích thước của bộ lọc làm mờ cho phù hợp

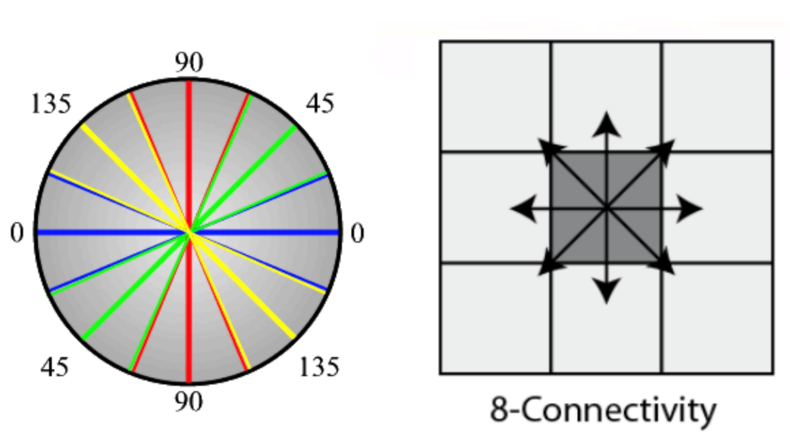
Minh họa cách xác định các trọng số của Gaussian Kernel.

**b. Tìm Gradient của ảnh**

Ảnh sau khi được làm mịn sẽ được được nhân chập với Sobel Kernel để tìm đạo hàm bậc nhất theo cả chiều dọc (GxGx) và chiều ngang (GyGy). Từ hai thành phân đó, chúng ta có thể tìm giá trị edge gradient và hướng gradient cho mỗi điểm ảnh như sau:



Hướng gradient là luôn luôn vuông góc với cạnh. Nó được làm tròn vào một trong bốn hướng 00,450,900,1350.



**c. Loại bỏ các điểm không phải cực đại**

Sau khi tìm được gradient và hướng, quét toàn bộ bức ảnh để loại bỏ các điểm ảnh không thuộc cạnh. Tại mỗi điểm ảnh, kiểu tra xem liệu nó có phải là một điểm cực đại địa phương (a local maximum) trong khu vực xung quanh nó theo hướng của gradient. Xem ví dụ dưới đây:

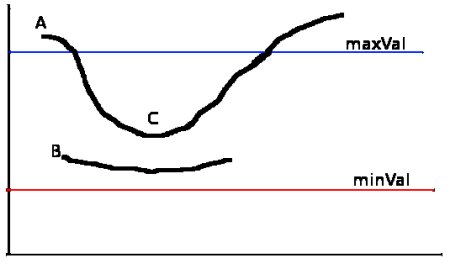


Điểm A là thuộc cạnh theo chiều dọc. Hướng của gradient là vuông góc với cạnh. Điểm B và điểm C thuộc hướng của gradient. Vì vậy, giá trị edge gradient của điểm A được so sánh với điểm B và điểm C để xem liệu nó có phải là cực đại địa phương phương hay không (tức nó có lớn hơn cả B và C không). Nếu nó là cực đại địa phượng, nó sẽ được xét ở bước tiếp theo. Ngược lại, nó sẽ được gán bằng 0 (suppressed). Vì vậy, bước này được gọi là Non-maximun Suprression, tức loại bỏ các điểm không phải cực đại.

Tóm lại, kết quả của bước này là một ảnh nhị phân với “các cạnh mỏng”.

**d. Lọc ngưỡng**

Bước này xét xem trong các cạnh tìm được ở bước trước cái nào thực sự là cạnh, cái nào không phải. Chúng ta làm điều này bằng cách sử dụng giá trị nguỡng dưới (minVal) và ngưỡng trên (maxVal). Bất kỳ điểm ảnh nào có độ lớn của gradient lớn hơn maxVal thì được xem là chắc chán thuộc cạnh, nhỏ hơn minVal thì được xem là chắc chắn không thuộc cạnh. Những điểm ảnh còn lại sẽ được xét dựa trên kết nối của nó với các điểm ảnh đã được phân loại. Nếu nó có kết nối với một điểm ảnh chắc chắn thuộc cạnh nó sẽ được xem là một phần của cạnh. Ngược lại, nó không thuộc cạnh. Xem hình minh họa dưới đây:



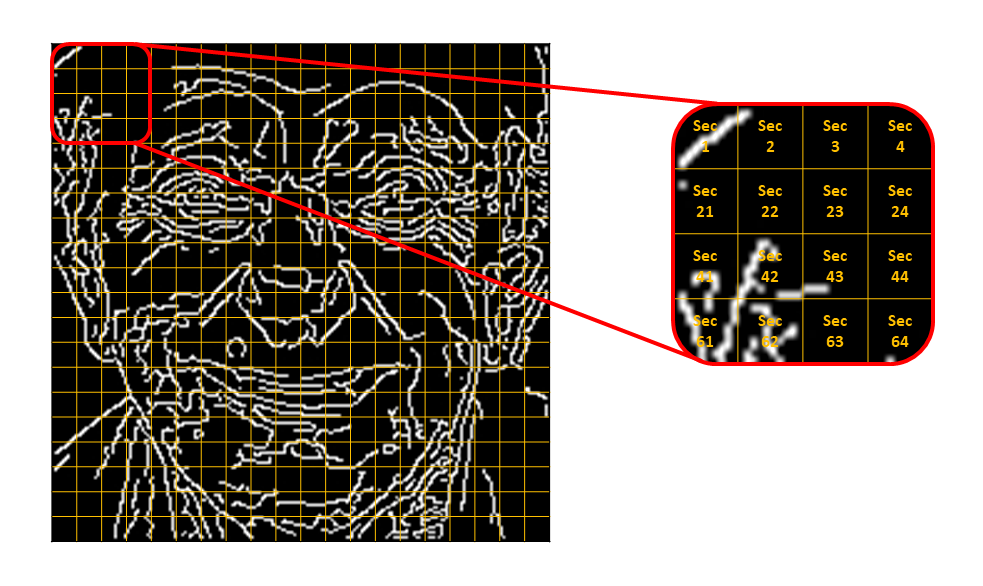
Điểm A là lớn hơn maxVal nên nó được xem chắc chắn là cạnh. Mặc dù điểm C là ở giữa maxVal và minVal nhưng nó có kết nối với điểm A nên nó cũng được xem là thuộc cạnh. Điểm B cũng nằm giữa maxVal và minVal như C nhưng nó không có kết nối với bất kỳ điểm nào chắc chắn thuộc cạnh nên nó bị loại bỏ. Vì vậy, việc chọn giá trị minVal và maxVal là rất quan trọng.

Ngoài ra, bước này cũng loại bỏ các điểm ảnh nhiễu với giả thiết rằng cạnh phải dài.

Kết thúc bước này ta nhận được kết quả cuối cùng.

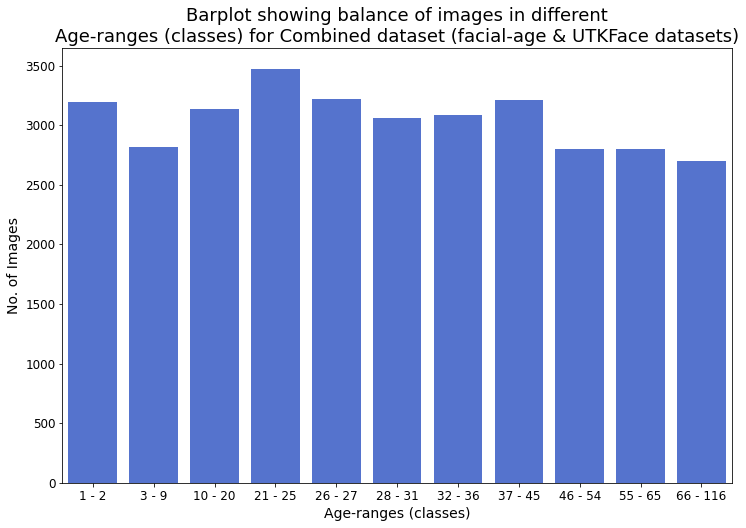
**2. Trích xuất đặc trưng**

* Sau khi lọc ảnh qua filter, cắt các bức ảnh 200 x 200 pixels thành các section 10 x 10 pixels.
* Tính độ lệch trung bình và độ lệch chuẩn cho từng section.
* Chúng ta sẽ có được 800 giá trị vô hướng cho mỗi bức ảnh và được chuyển thành các dataframe để sử dụng như feature cho model.

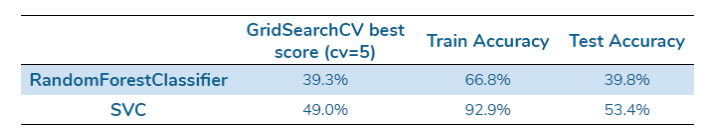


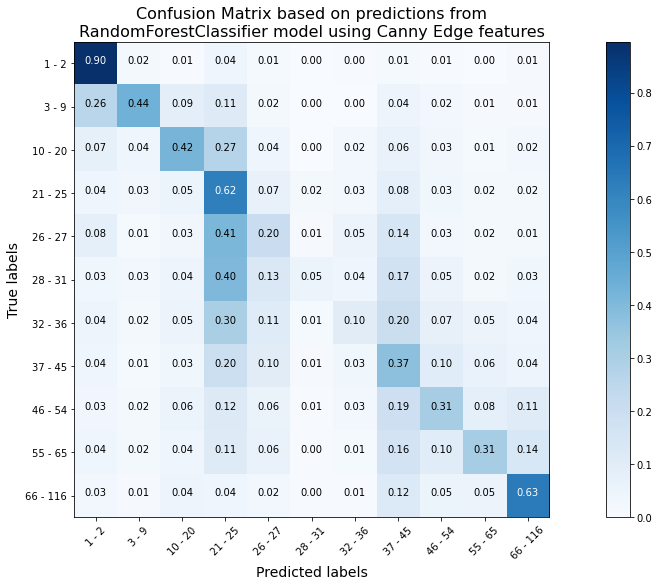
**3. Xử lí data**

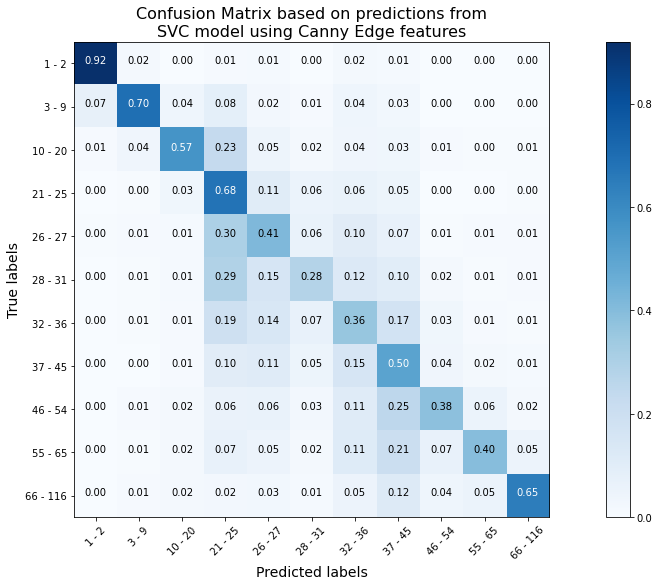
* Tổng: 33,468 ảnh
* 11 class: [“1-2”, “3-9”, “10-20”, “21-25”, “26-27”, “28-31”, “32-36”, “37-45”, “46-54”, “55-65”, “66-116” ]
* 70% train (23.440 ảnh) và 30% test (10,046 ảnh)

**4. Các model để train**

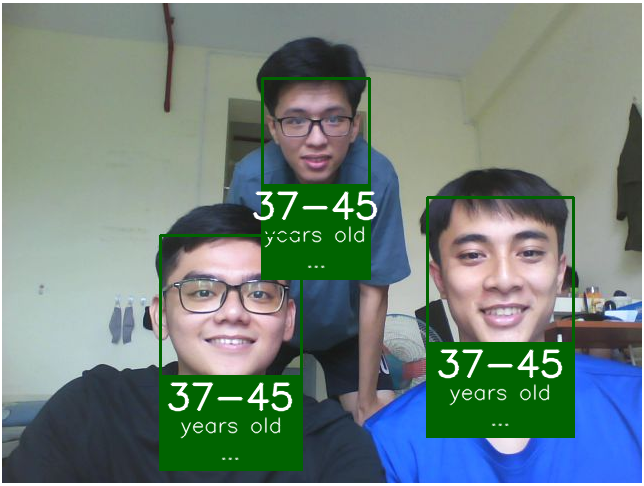
* sklearn.ensemble.RandomForestClassifier
* sklearn.svm.SVC
* sklearn.model\_selection.GridSearchCV ( tìm các parameter tốt nhất cho model)

**5. Kết quả  
**

****

****

Nhận xét: Độ chính xác khá thấp và đúng chủ yếu ở các nhóm tuổi 1-2, 3-9, 66-116



**Ảnh test model**

Ví dụ ở trên cho thấy sử dụng machine learning truyền thống để dự đoán đưa ra độ chính xác thấp, thậm chí sai hoàn toàn (Cả 3 đều 20 tuổi nhưng kết quả trả về là từ 37-45 tuổi)

Do đó chúng ta có hướng tiếp cận mới, sử dụng deep learning.

## **II. Sử dụng deep learing**

**1. Xử lí data**

* 33, 486 ảnh với kích thước 200x200 (ảnh xám) ta sẽ có 33,468 x 40,000

Khi train với kích thước lớn như vậy tràn RAM là điều chắc chắn xảy ra.

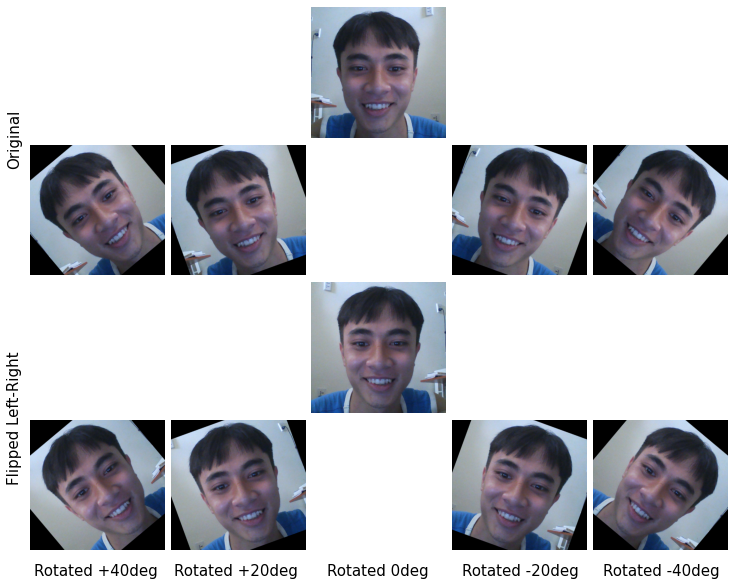
* Giải pháp:

Sử dụng tf.data

Sử dụng ảnh xám

Tăng kích thước datasets (234,400 ảnh)

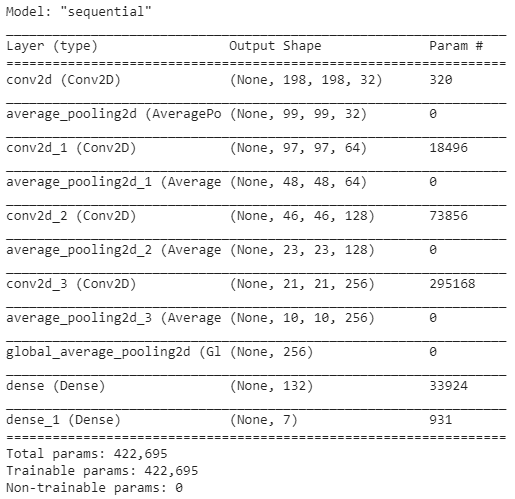
Giảm bớt class age-range [“1-2”, “3-9”, “10-20”, “21-27”, “28-45”, ”46-65”, “65-116”]



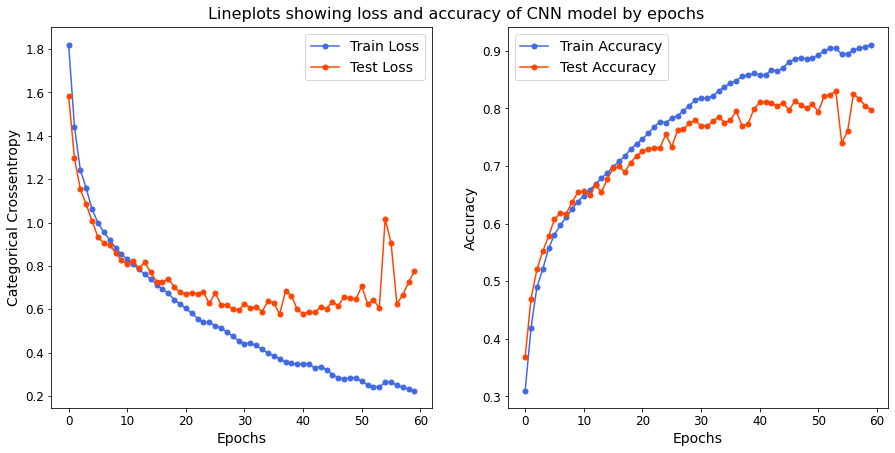
**Tăng kích thước data bằng cách xoay ảnh**

**2. Train model**

* CNN model

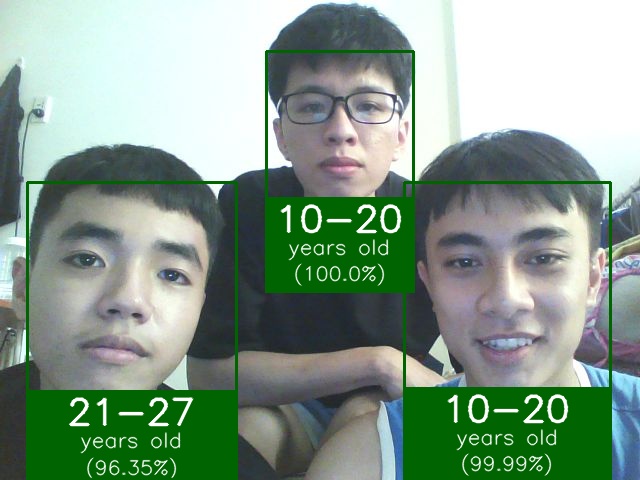


**Kết quả train**



Kết luận: Độ chính xác sau khi sử dụng deep learning cao hơn.

Một số ví dụ minh họa:



Mặc dù phần trăm đúng của model dựa trên bộ test khá cao, nhưng để dự đoán tuổi của 1 người ngoài đời thật thì ta cần dựa vào rất nhiều dữ liệu như màu da, điều kiện sống, quốc tịch. ví dụ như diện viên park bo young đây thì thật khó để dự đoán tuổi nếu không biết cô ấy là ai và chỉ dựa vào khuôn mặt.

