



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**ĐỀ TÀI**

# AGE & GENDER DETECTION

**Lớp: CS231.M12.KHCL**

**Giảng viên: TS. Mai Tiến Dũng**

**Thành viên:**

Nguyễn Hoài Nam – 18521126

Nguyễn Dương Hải – 19521464

Trịnh Tuấn Nam – 19521874

Thành phố Hồ Chí Minh, ngày 25 tháng 12 năm 2021

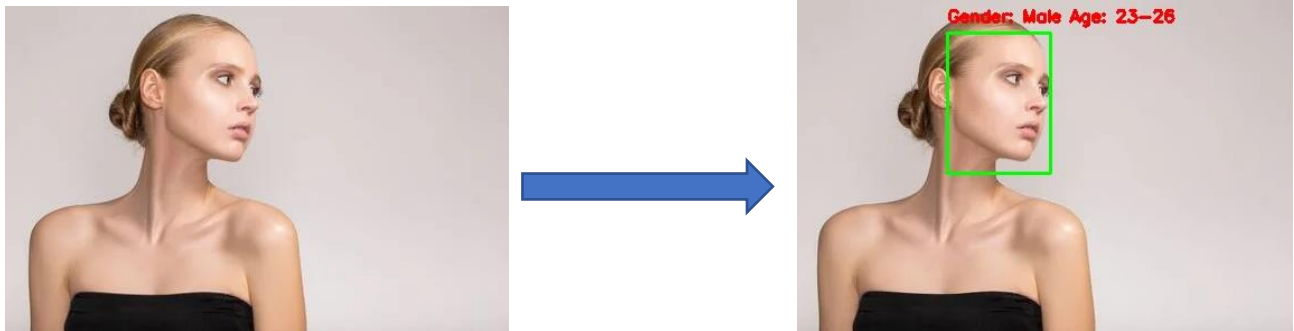


# MỤC LỤC

<b>I.</b>	<b>Giới thiệu .....</b>	<b>2</b>
<b>II.</b>	<b>Face Detection .....</b>	<b>2</b>
	1. Haar Cascade .....	3
	2. MTCNN .....	4
	3. YOLOv4 .....	5
	4. So sánh và lựa chọn .....	6
<b>III.</b>	<b>Dataset .....</b>	<b>7</b>
<b>IV.</b>	<b>Age Detection .....</b>	<b>8</b>
	1. Xử lý dữ liệu .....	8
	2. Lựa chọn thuật toán ML .....	10
	3. Kết quả .....	11
<b>V.</b>	<b>Gender Detection .....</b>	<b>12</b>
	1. SVM .....	12
	2. CNN .....	13
	3. So sánh và lựa chọn .....	17
<b>VI.</b>	<b>Thực nghiệm .....</b>	<b>18</b>
<b>VII.</b>	<b>Phân chia công việc .....</b>	<b>18</b>
<b>#</b>	<b>Tài liệu tham khảo .....</b>	<b>19</b>

# **I. Giới thiệu**

- Bài toán này là phát hiện độ tuổi giới tính thông qua gương mặt của một người. Bằng cách phát hiện gương mặt của một người trong một bức ảnh sau đó sẽ thực hiện các thao tác để có thể đưa ra được tuổi và giới tính dự đoán được của gương mặt được phát hiện trong ảnh.
- Cụ thể:
  - Input: Một bức ảnh có xuất hiện gương mặt của người.
  - Output: Một bức ảnh tương tự input nhưng sẽ có thêm bounding box gương mặt và một đoạn văn bản cho biết tuổi và giới tính dự đoán được.



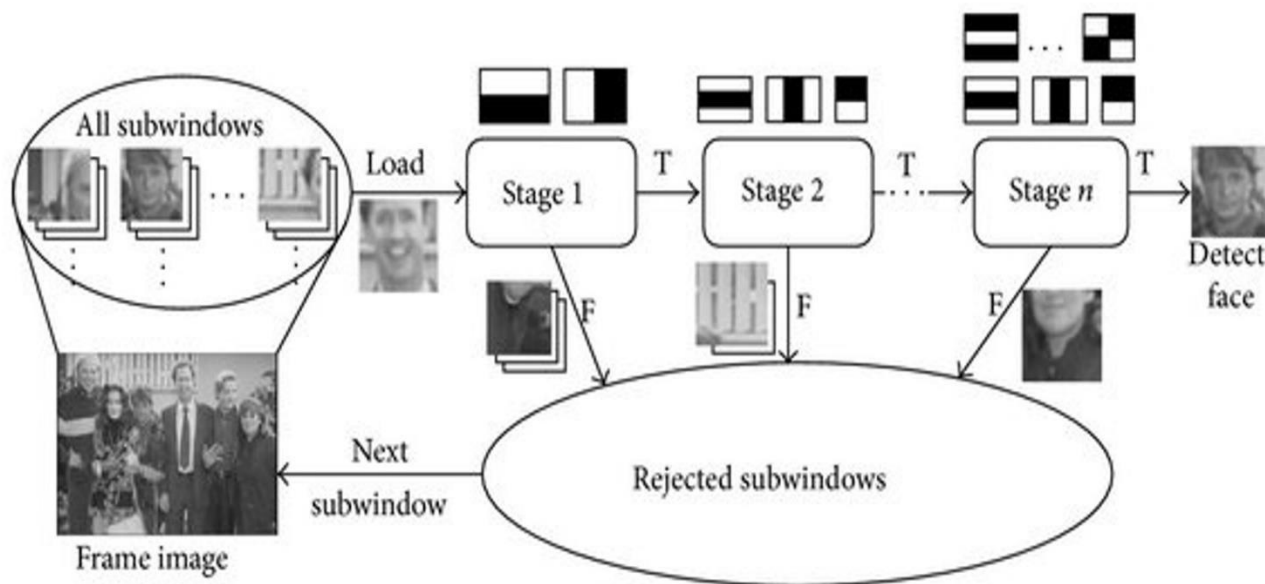
# **II. Face Detection**

- Phát hiện gương mặt là bài toán phổ biến trong lĩnh vực Computer vision. Đây là bài toán tiên quyết cho khá nhiều bài toán khác và ứng dụng nhiều trong đời sống ví dụ như trong các hệ thống an ninh, các loại điện thoại, tivi thông minh, .... Trong bài toán này phát hiện gương mặt cũng đóng một phần quan trọng. Đây là bước đầu tiên để thực hiện bài toán.
  - Các hướng tiếp cận:
    - Haar cascade
    - Multi-task Cascaded Convolutional Networks (MTCNN)
    - YOLOv4

# 1. Haar cascade

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”.

Các bước phát hiện gương mặt của Haar Cascade



Hình 1.1 Minh họa các bước phát hiện gương mặt

**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).

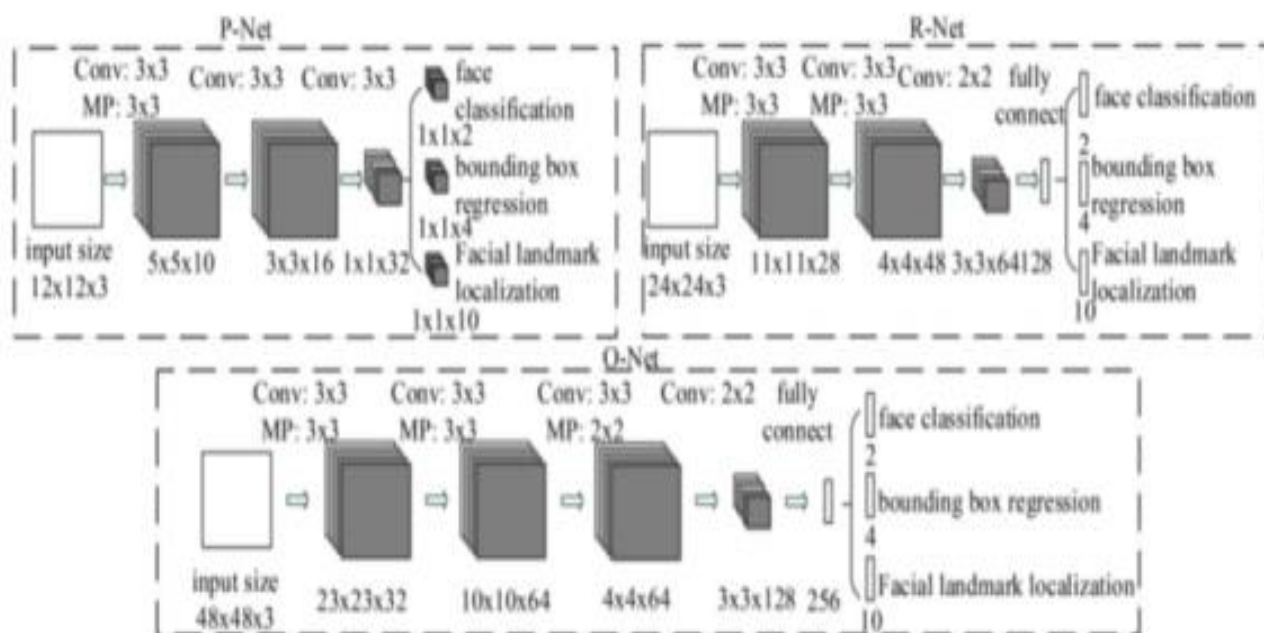
**Bước 2:** Hình ảnh sẽ được đi qua **n** bộ dò theo cách xếp tầng trong đó mỗi bộ là sự kết hợp giữa các đặc trưng haar (cạnh, đường thẳng, hình vuông,...). Hình ảnh sẽ được gán một **giá trị tin cậy**.

**Bước 3:** Hình ảnh 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.

## 2. MTCNN

Mạng MTCNN sẽ là một trong những phương pháp giúp chúng ta nhận diện gương mặt trong ảnh. Mạng hoạt động với 3 lớp mạng khác biệt, tương trưng cho 3 giai đoạn đó là **Proposal Network**(P-Net), **Refine Network**(R-Net) và **Output Network**(O-net)

- **P-Net:** Đầu vào của mạng P-Net sẽ là một loạt ảnh được copy từ một ảnh theo nhiều kích thước khác nhau và được xếp từ nhỏ đến lớn để tạo thành một **Image Pyramid**. Đầu ra sẽ là những vùng có thể là khuôn mặt trong tấm ảnh (trong đó có nhiều vùng không phải gương mặt). Ở lớp này thực hiện nhanh nhưng độ chính xác sẽ không cao.
- **R-Net:** Đầu vào sẽ là đầu ra của P-Net. Đầu ra sẽ là những vùng là gương mặt (loại bỏ bớt những vùng không phải gương mặt từ lớp P-Net).
- **O-Net:** Đầu vào là đầu ra của lớp R-Net. Đầu ra sẽ là kết quả cuối cùng (vùng xuất hiện gương mặt với độ chính xác cao nhất) cùng với 5 facial landmark (2 mắt, mũi, 2 bên khóe miệng).



Hình 2.1: Cấu trúc 3 mạng của MTCNN

### 3. YOLOv4

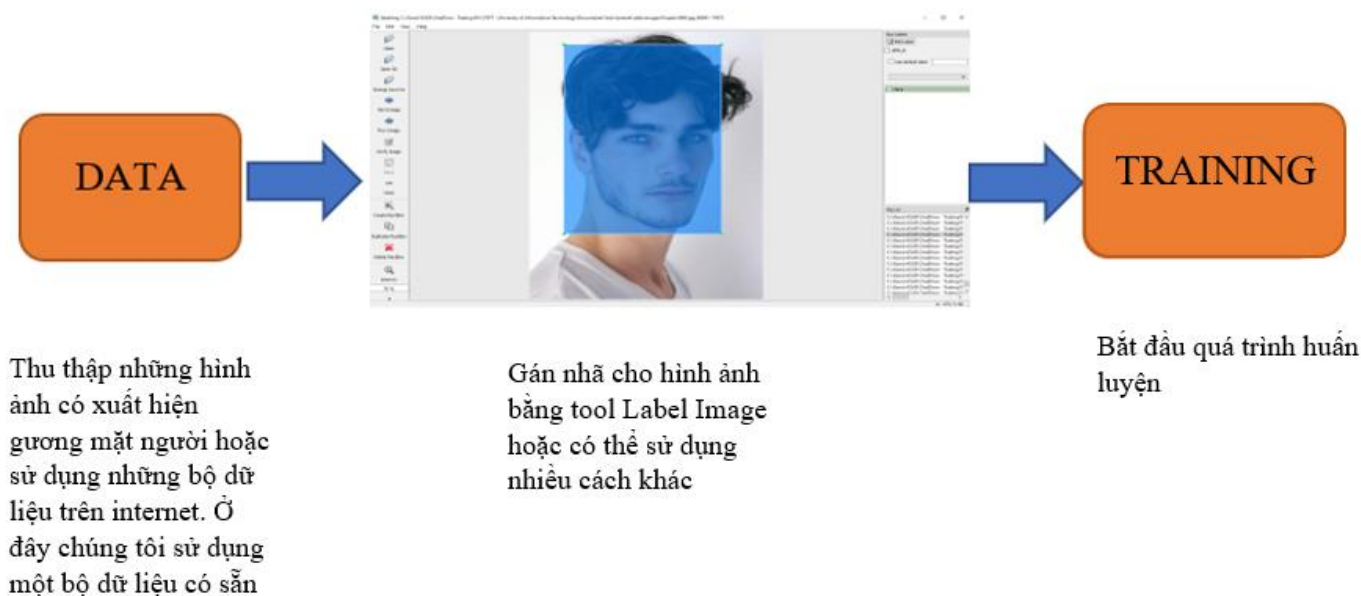
YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

YOLOv4 có nhiều sự cải tiến đặc biệt giúp tăng độ chính xác và tốc độ hơn đối với những phiên bản cũ.

Cấu trúc của v4 được tác giả chia làm bốn phần:

- **Backbone** (xương sống).
- **Neck** (cổ).
- **Dense prediction** (dự đoán dày đặc)- sử dụng các one-stage-detection như YOLO hoặc SSD.
- **Sparse Prediction** (dự đoán thưa thớt) – sử dụng các two-stage-detection như RCNN.

Trong bài toán này có thể sử dụng **yolov4** để có thể phát hiện gương mặt



Dataset: <https://www.kaggle.com/ashwingupta3012/human-faces>

Tool LabelImg: <https://github.com/tzutalin/labelImg>

#### 4. So sánh và lựa chọn

Thông qua các thử nghiệm có thể rút ra được một vài so sánh của 3 hướng tiếp cận trên

Haar Cascade	MTCNN	Yolov4
Tốc độ nhanh nhất	Tốc độ nhanh thứ 3	Tốc độ nhanh thứ 2
Hoạt động tốt với gương mặt chính diện	Hoạt động tốt ngay cả khi mặt bị che lấp và nhiều góc mặt khác nhau	Hoạt động tốt ngay cả khi mặt bị che lấp và nhiều góc mặt khác nhau
Dễ bị ảnh hưởng bởi ánh sáng môi trường	Ít bị ảnh hưởng bởi ánh sáng môi trường	Ít bị ảnh hưởng bởi ánh sáng môi trường
Dễ dàng cài đặt	Dễ dàng cài đặt	Tốn khá nhiều thời gian để cài đặt và training

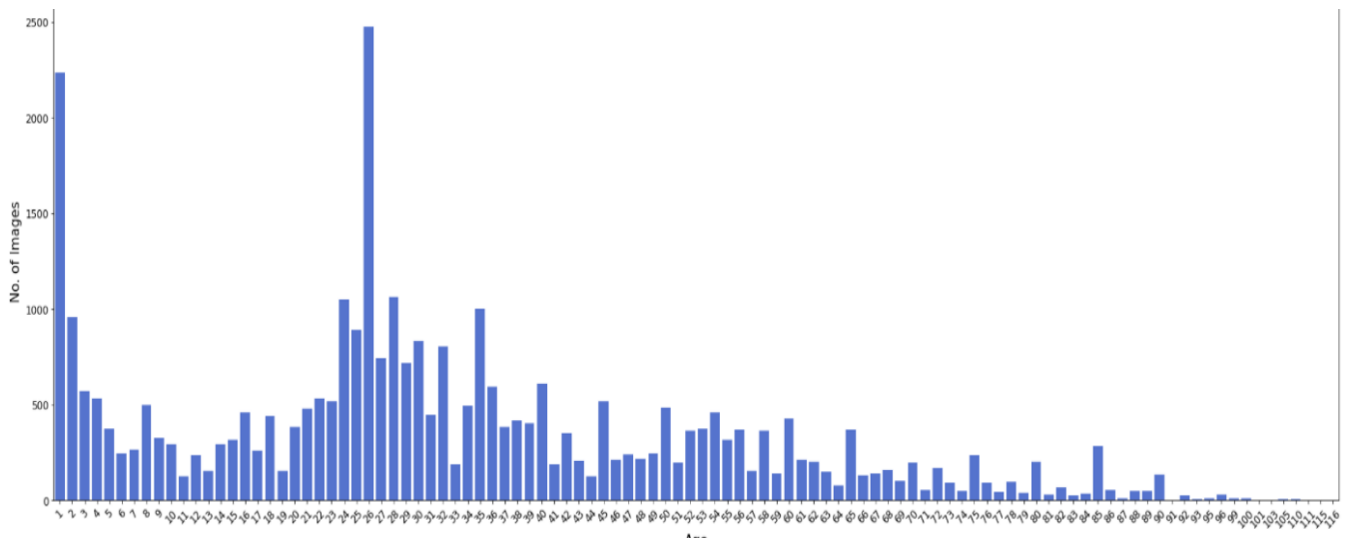
Bảng 1.1: So sánh ưu nhược điểm của 3 phương pháp

Từ bảng so sánh trên có thể thấy mỗi phương pháp đều có những ưu nhược điểm riêng. Mỗi phương pháp sẽ phù hợp với những loại bài toán khác nhau. Ở đây chúng tôi quyết định **sử dụng MTCNN** để có thể áp dụng vào bài toán lớn ban đầu.



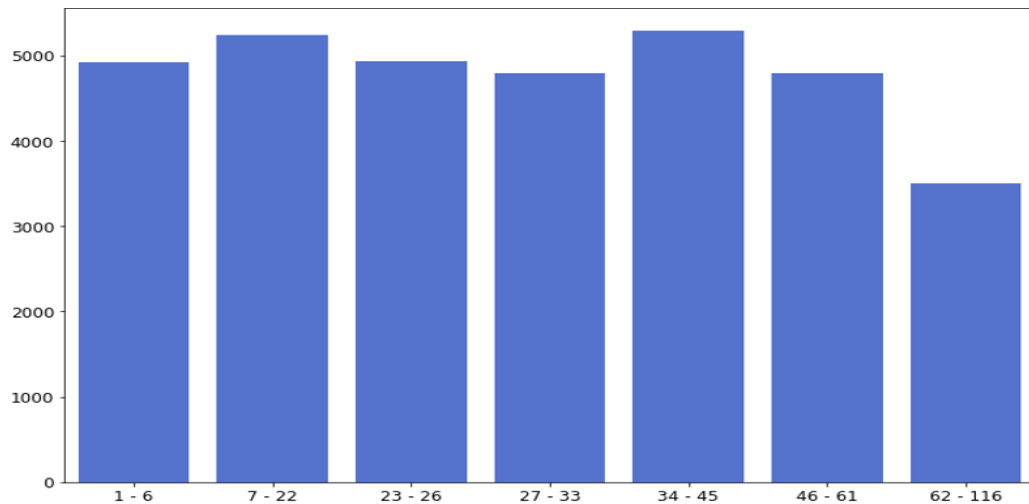
### III. Dataset

Trong kỷ nguyên của Covid-19, chúng ta trở nên tin cậy hơn vào các tương tác ảo như cuộc họp Zoom / Trò chuyện nhóm. Các video webcam livestream này đã trở thành một nguồn dữ liệu phong phú để khám phá. Ví dụ, bài viết này sẽ khám phá trường hợp sử dụng của dự đoán tuổi, giới tính và cảm xúc có thể giúp nhân viên bán hàng hiểu khách hàng của họ tốt hơn.



Hình 2.1: Biểu đồ lượng dữ liệu trên từng độ tuổi

- Trên đây là biểu đồ biểu thị lượng dữ liệu trên từng độ tuổi trong dataset mà team đã thu thập. Có thể thấy dataset đang có sự mất cân bằng khi có độ tuổi sở hữu lượng lớn dữ liệu như 1, 26, trong khi các độ tuổi còn lại có lượng dữ liệu thấp hơn đáng kể.
- Vì vậy team đã quyết định chia lớp cho dataset sao cho số lượng dữ liệu tại mỗi lớp là không quá chênh lệch như dataset ban đầu, bên cạnh đó số lượng lớp cũng phải hợp lý, nếu quá nhiều thì sẽ bị tình trạng giống với dataset trước khi chia lớp và nếu nhiều quá thì bài toán sẽ trở nên kém quan trọng.
- Và team đã quyết định chia dataset thành 7 lớp tương ứng với 7 khoảng tuổi để phù hợp với các yêu cầu đề ra ban đầu.

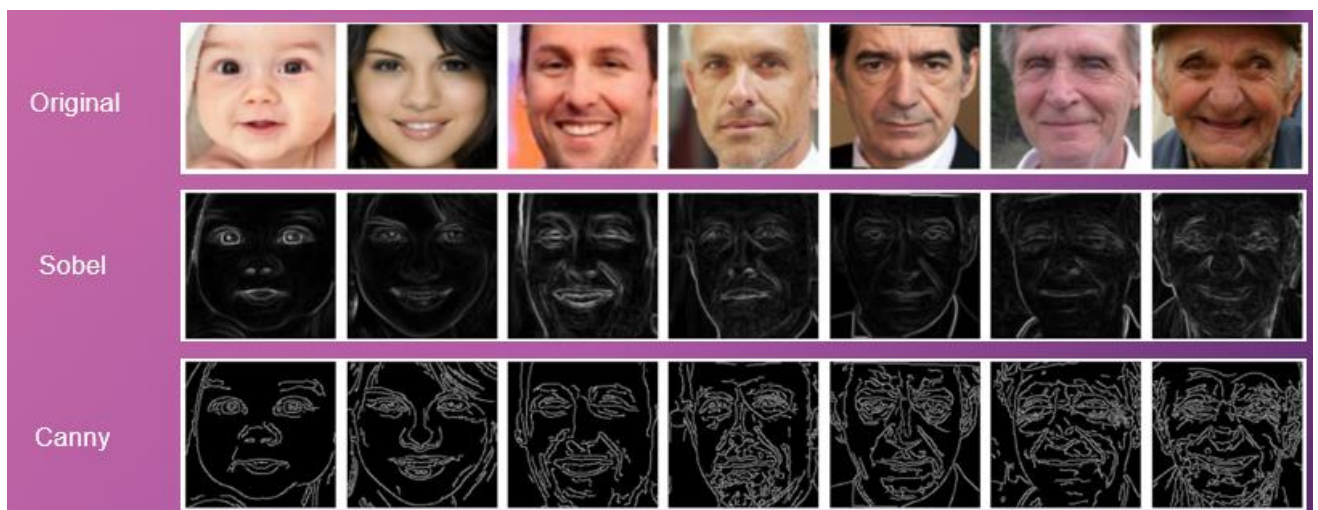


Hình 2.2: Dataset được chia thành 7 khoảng tuổi

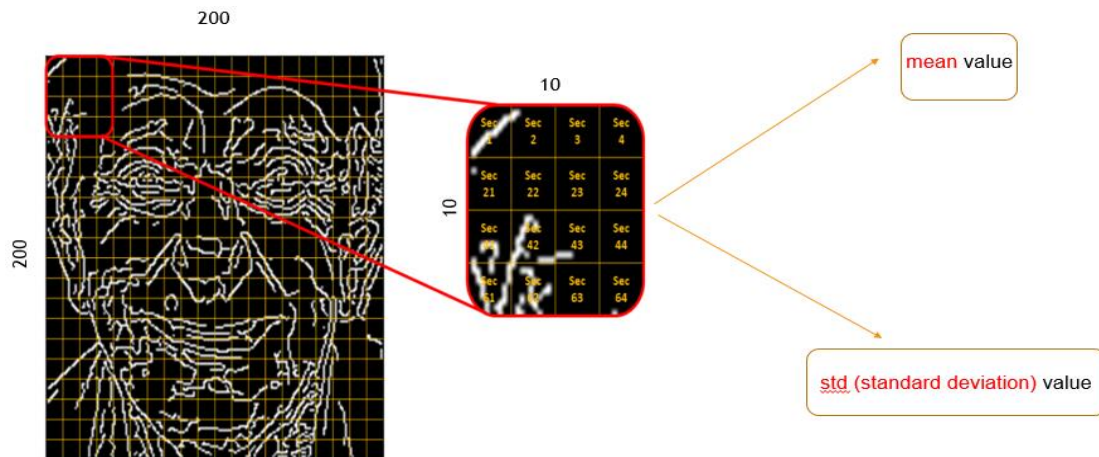
## IV. Age Detection

### 1. Xử lý dữ liệu

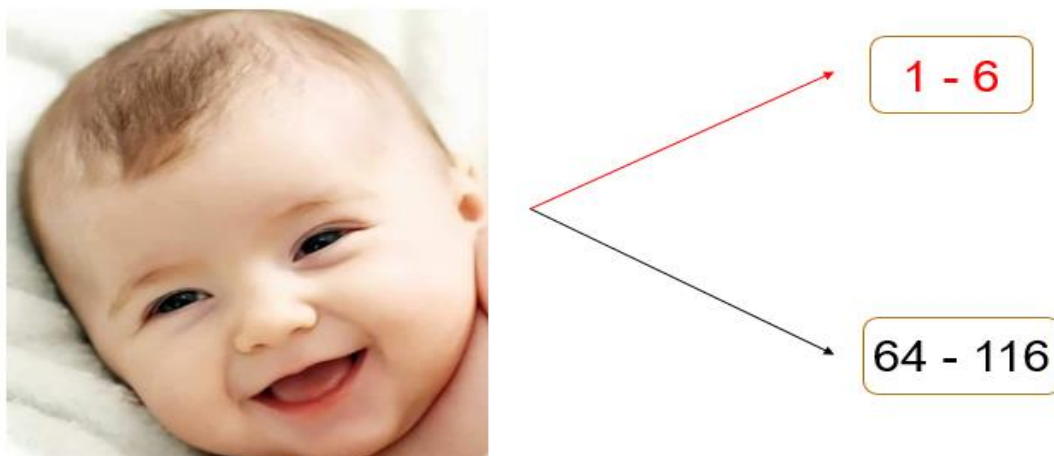
- Sau khi được chia lớp, dataset sẽ đi qua bước trích xuất đặc trưng cho từng ảnh.
- Phương pháp trích xuất đặc trưng mà team chọn là trích xuất đặc trưng CẠNH. Lý do là vì team nhận ra con người khi càng lớn tuổi thì gương mặt sẽ càng xuất hiện nhiều nếp nhăn. Phương pháp phát hiện cạnh sẽ giúp nhận diện được các nếp nhăn trên gương mặt, từ đó giúp cho dự đoán tuổi được chính xác hơn.
- Công việc của team là lựa chọn ra phương pháp trích xuất cạnh phù hợp với yêu cầu đề ra.
- Hình bên dưới là sự so sánh giữa 2 phương pháp trích xuất đặc trưng cạnh là Sobel và Canny.



- Ta có thể thấy là Canny có thể trích xuất tốt ở các cạnh ngang, dọc, cạnh tròn và các cạnh ở góc, Sobel thì trích xuất tốt các cạnh ngang, dọc. Điểm mạnh của Sobel là nó có thể phân biệt được mức độ đậm nhạt trên từng cạnh, từ đó có thể phù hợp trong các bài toán khác. Nhưng trong bài toán dự đoán độ tuổi dựa trên mật độ nếp nhăn trên gương mặt thì Canny đang thể hiện tốt hơn khi cho ra kết quả đường nét rất rõ ràng.
- Cuối cùng, team đã quyết định chọn phương pháp trích xuất cạnh là Canny.
- Sau đó, team tiến hành chia nhỏ ảnh size 200x200 pixels thành các sections size 10x10 pixels và tính giá trị trung bình và giá trị độ lệch chuẩn cho mỗi section.



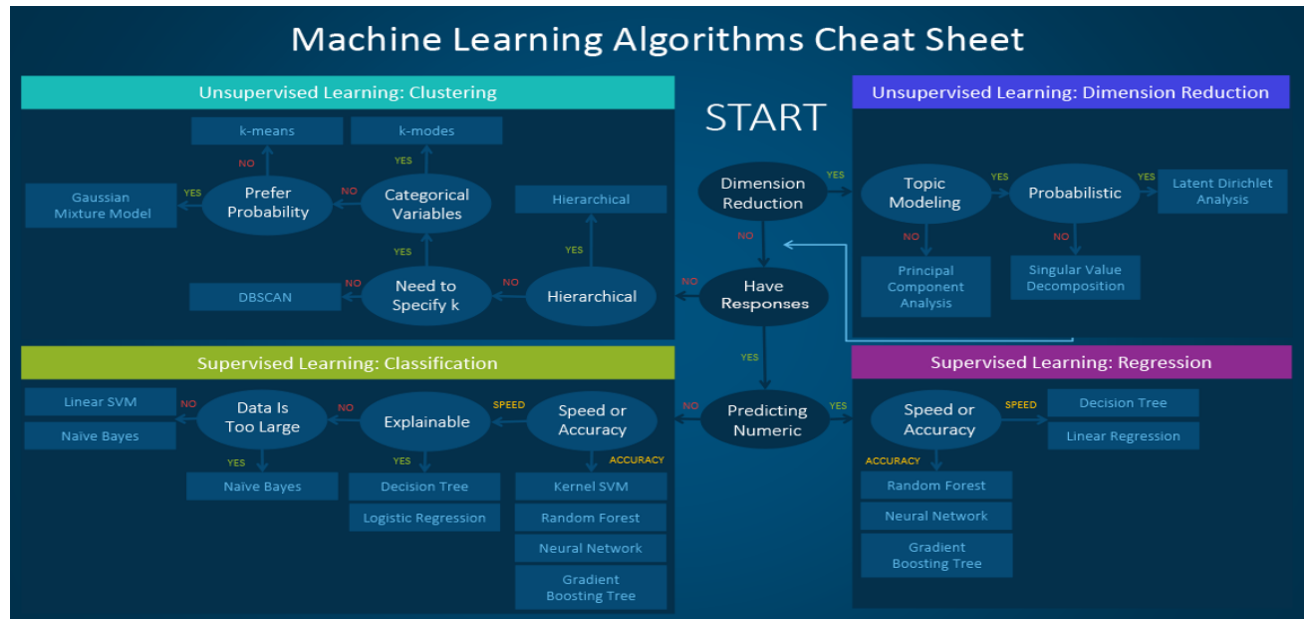
- Phương pháp này có 2 mục đích. Một là làm giảm khối lượng tính toán từ việc phải tính 40000 giá trị pixels trên mỗi tấm ảnh thì giờ đây chỉ cần phải tính toán 800 giá trị trên mỗi ảnh. Hai là việc chuẩn hóa giá trị pixel sẽ giúp áp dụng dự đoán trên tấm ảnh mới được tốt hơn và được sử dụng khi chuyển sang các mô hình Deep Learning.



Hình 2.4: Hình ảnh thực nghiệm kết quả

- Trên là kết quả dự đoán mà team đã thực nghiệm với một kết quả màu đỏ là ảnh đã được chuẩn hóa thì model sẽ dự đoán đúng độ tuổi của em bé, còn ảnh khi không được chuẩn hóa thì model sẽ dự đoán sai lệch đi khá nhiều, mặc dù accuracy sau khi training model là gần như tương đương nhau.

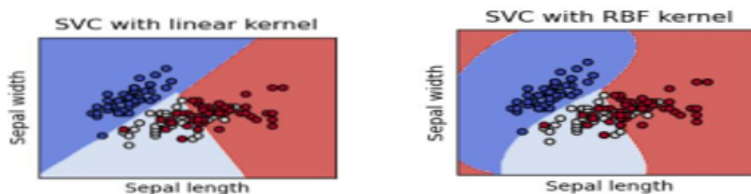
## 2. Chọn thuật toán Machine Learning



Hình 2.5: Sơ đồ lựa chọn thuật toán Machine Learning

- Team dựa vào sơ đồ trên hình 2.5 để lựa chọn thuật toán Machine Learning cho bài toán Dự đoán độ tuổi vốn có bản chất là một bài toán Classification. Team muốn đánh giá dựa trên Accuracy nên đã chọn model SVM làm model dự đoán tuổi.
- Đây là model được sử dụng nhiều nhất trong bài toán Classification. Nó có độ chính xác khá tốt và tốc độ nhanh hơn Random Forest.
- Team sử dụng model SVM có sẵn trong thư viện sklearn và sẽ sử dụng 2 kernel khác nhau là **rbf** và **linear** để xem kết quả thực nghiệm như thế nào.

### 3. Kết quả



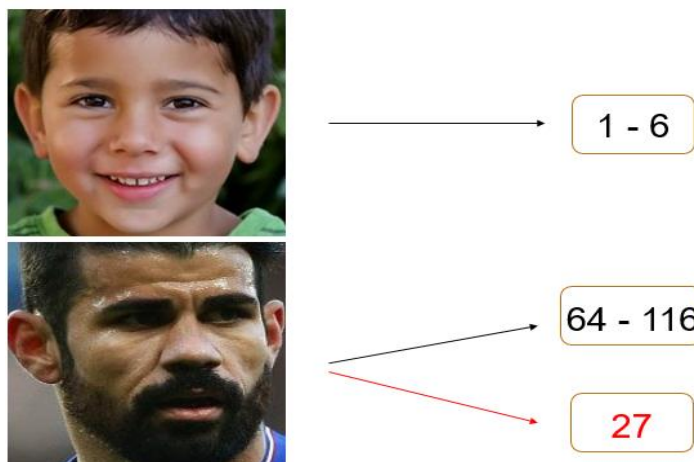
104 classes	KERNEL	
	LINEAR	RBF
TRAINING ACC	96.8%	65.8%
TESTING ACC	44.8%	30.7%

7 classes	KERNEL	
	LINEAR	RBF
TRAINING ACC	60.8%	84.8%
TESTING ACC	46.8%	56.1%

Hình 2.6: Kết quả thực nghiệm

- Trên đây là kết quả thực nghiệm khi sử dụng model SVM. Có thể thấy hiện tượng overfitting đã xuất hiện khi không chia lớp cho dataset, sau khi dataset được chia lớp thì hiện tượng overfitting đã được khắc phục phần nào và bên cạnh đó cũng giúp cho Accuracy của model trở nên tốt hơn.
- Dựa vào kết quả ở trên, team quyết định sử dụng model SVM với kernel RBF làm model chính thức cho bài toán dự đoán tuổi của mình. Lý do giải thích cho việc Accuracy chỉ có được ~ **56%** thì một là do Canny trích xuất cạnh cả ở ngoài gương mặt (background) nên làm giảm sự chính xác của model khi dự đoán. Hai là có những người già nhưng họ lại có gương mặt trẻ trung, ngược lại người trẻ lại sở hữu gương mặt già nua. Cuối cùng là do chất lượng ảnh làm hiển thị sai lệch nếp nhăn trên gương mặt so với sự thật.





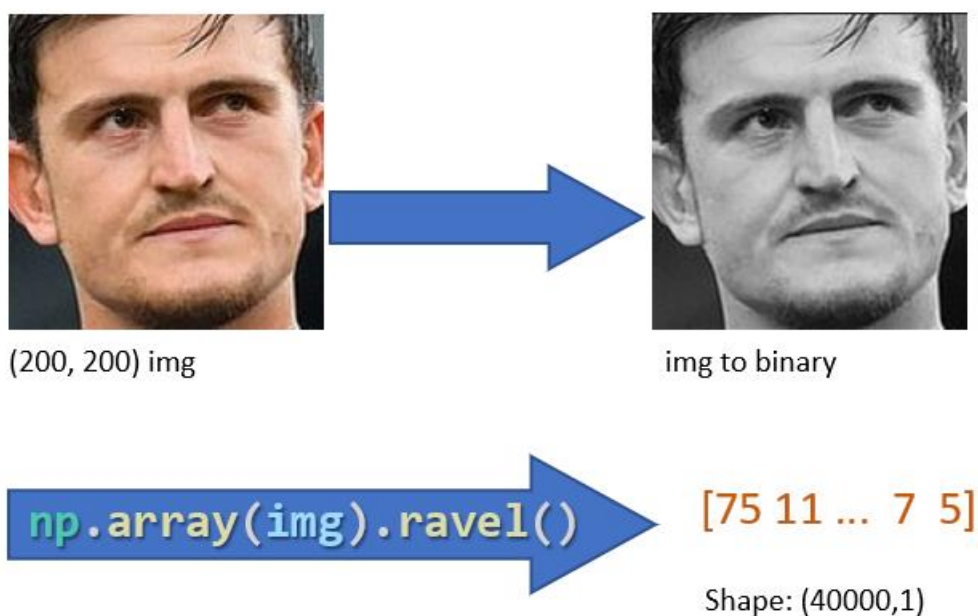
- Đây là hai trường hợp mà team đã thực nghiệm để thấy rằng một trường hợp model của team đã dự đoán đúng độ tuổi của cậu bé trong hình là từ 1 -> 6 tuổi. Còn trong trường hợp còn lại thì model lại dự đoán tuổi của anh chàng trong hình bị sai lệch khá nhiều so với sự thật mà lý do team đã nêu ở trên đó là anh chàng này mới 27 tuổi nhưng gương mặt lại nhìn già hơn độ tuổi khá nhiều nên việc model dự đoán sai cũng không có gì đáng trách.

## V. Gender Detection

- Trong phần này, vẫn bộ dataset ở phần III, team đã sử dụng hai phương pháp để thực hiện nhận diện độ tuổi cho đối tượng trong ảnh, đó là: SVM và CNN.

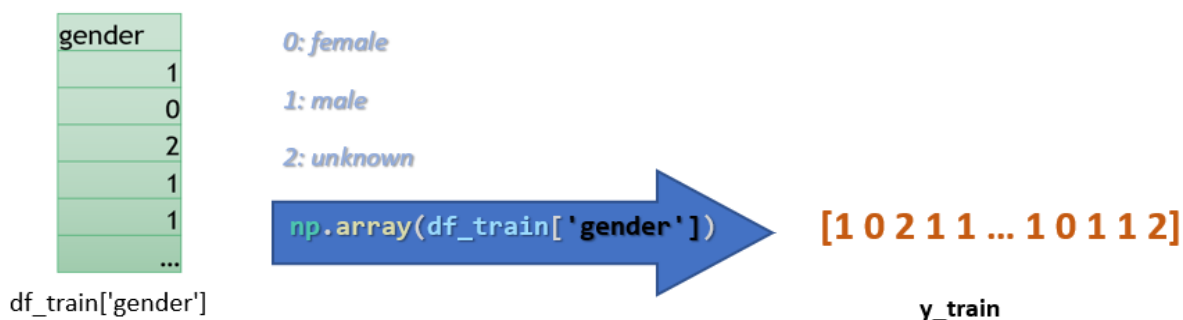
### 1. Supported Vector Machine (SVM)

Với phương pháp SVM, hình ảnh đầu vào được đọc vào dưới dạng binary, sau đó được reshape về 1D.



Sau đó, tập hợp tất cả các ảnh trong tập training sẽ là một mảng 2D, sau đó tiến hành chia các giá trị cho **255**. Lí do là để độ hội tụ nhỏ hơn nên quá trình training nhanh hơn và chính xác hơn.

Về phần label cho dữ liệu training, chỉ cần đọc cột label đã được team gán nhãn từ trước trong bộ training, kết quả trả về 1 mảng 1D như sau:



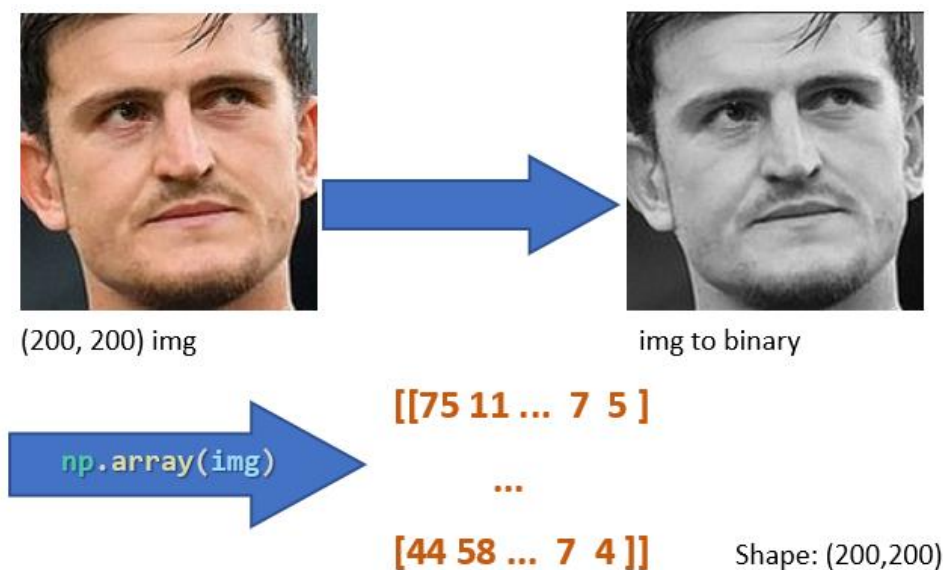
Sau khi đã có được rút trích đặc trưng các ảnh trong tập train và label của chúng. Quá trình training model SVM diễn ra, sau cùng model sẽ được lưu trữ lại để thực hiện testing và predicting:



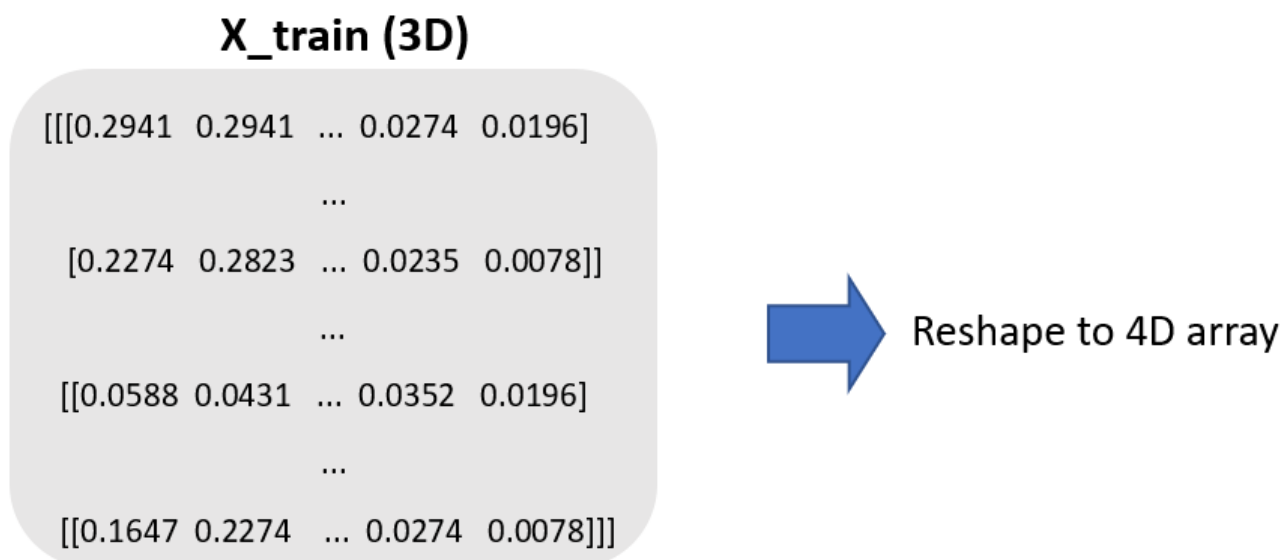
Tập testing cũng sẽ được xử lý tương tự.

## 2. Convolutional Neural Network (CNN)

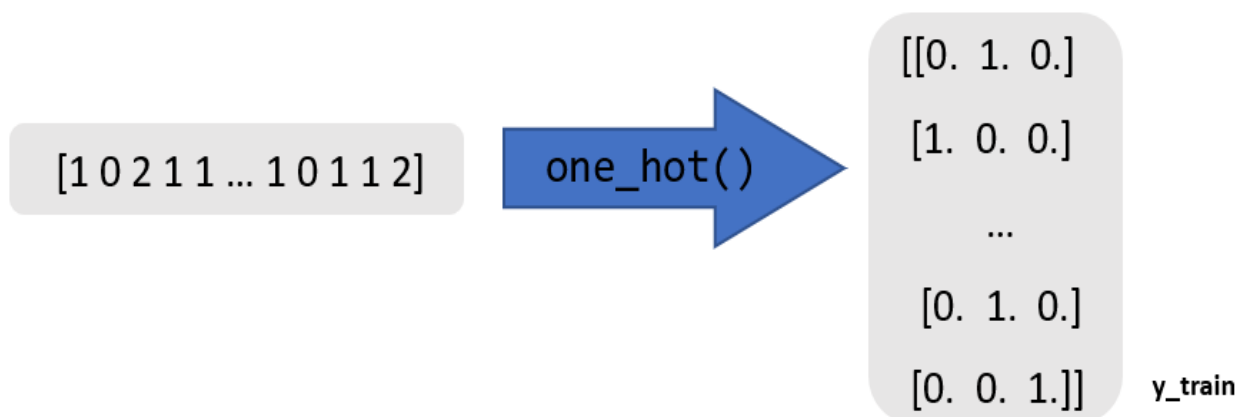
Không như SVM, khi trích xuất đặc trưng ảnh trong CNN ta không đọc từng ảnh dưới dạng 1D, mà là dạng 2D:



Rút trích đặc trưng tất cả các ảnh sẽ trả về 1 mảng 3D, nhưng mô hình CNN cần một mảng 4D, vì thế, cần reshape về đúng dạng.



Về phần xử lý label trong CNN, tương tự như SVM, nhưng ta cần thêm một bước nữa, đó là `one_hot()` để đầu ra là 1 mảng đơn trội như sau:



Trong CNN, ngoài 2 bộ training và testing được xử lý tương đồng, nhóm có sử dụng thêm 1 bộ nữa, đó là bộ validation, việc xử lý cũng tương tự như trên 2 bộ trên.

Sau cùng là training: ở đây team đã tự build một model mạng CNN để phù hợp với kích thước bộ dữ liệu, dữ liệu training sẽ đi qua các lớp `Conv2D()` và `Maxpooling()` xen kẽ nhau, cho đến khi tới 3 lớp cuối cùng bao gồm: 1 bước lớp `Flatten()` và 2 lớp `Dense()` trong đó lớp `Dense()` cuối cùng có số `units=3` – đó cũng là số class trong dự đoán giới tính. Cụ thể như sau:

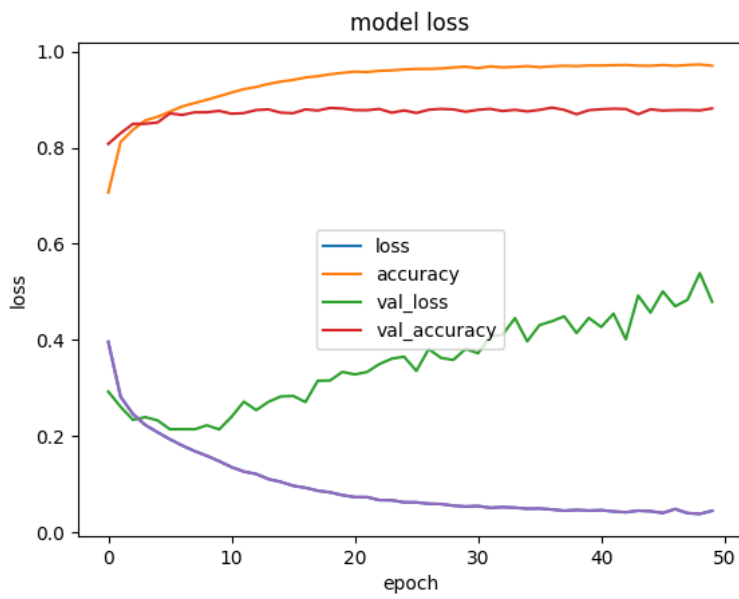


Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 198, 198, 16)	160
max_pooling2d (MaxPooling2D)	(None, 99, 99, 16)	0
conv2d_1 (Conv2D)	(None, 97, 97, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 46, 46, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_3 (Conv2D)	(None, 21, 21, 64)	18496
max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 128)	131200
dense_1 (Dense)	(None, 3)	387
=====		
Total params: 201,059		
Trainable params: 201,059		
Non-trainable params: 0		

Về vấn đề chọn số lượng epochs để training mô hình CNN, qua quá trình thực nghiệm, team đã có được 2 biểu đồ như sau:

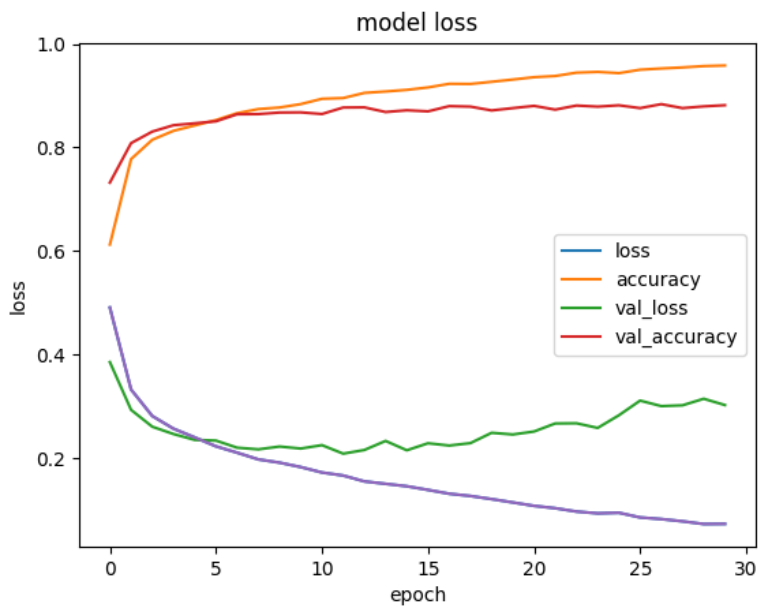
----- 50 epochs -----



-- Test Accuracy: 88.32 %

-- Test Loss: 0.42316874523015

----- 30 epochs -----

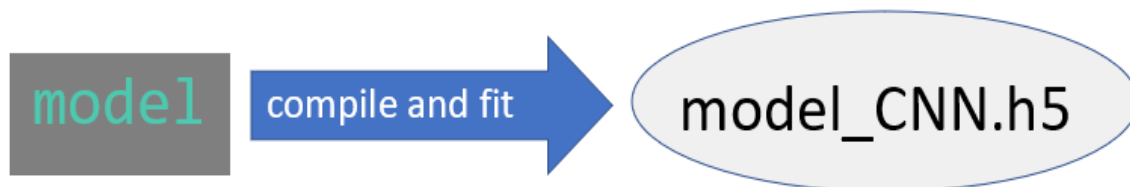


-- Test Accuracy: 88.06 %



-- Test Loss: 0.3133225440979004

Nhận thấy việc training với 50 epochs có vẻ cho kết quả của biểu đồ không “đẹp” bằng 30 epochs, vì giá trị loss trong tập validation ở 50 epochs có xu hướng tăng lên quá cao từ giao đoạn số epochs bằng 30. Vì thế team thống nhất sử dụng **30 epochs**.

Sau cùng kết quả training sẽ được lưu lại:



### 3. So sánh và lựa chọn

	SVM (kernel='rbf')						CNN					
	• Accuracy: 83.85 %						• Accuracy: 88.06 %					
	• Slowly predict and less accurate						• Predict faster and more accurate					
												
predict	0	0	2	1	1	2	0	0	0	1	1	2
true	0	0	0	1	1	0	0	0	0	1	1	0

Kết quả thực nghiệm cho thấy, CNN có ưu thế hơn SVM, nhất là về tốc độ. Cho nên team đã thống nhất **sử dụng CNN** cho bài toán lớn ban đầu.

## VI. Thực nghiệm

Kết quả thực nghiệm đã được nhóm quay lại và đính vào slide cuối của file Power Point.

Nhận xét:

- Về phần detect khuôn mặt: hầu như model luôn luôn nhận ra khuôn mặt ở trong điều kiện ánh sáng khá phức tạp như trong môi trường nhóm đã thực hiện.
- Về phần dự đoán tuổi: người trong video demo có label tuổi đúng là 7-22, mô hình đã dự đoán được nhãn trên, nhưng vẫn còn một số frame dự đoán sai.
- Về phần dự đoán giới tính: người trong video demo có giới tính là “Male”. Nhìn chung hầu hết các frame đều dự đoán đúng, vẫn còn tồn tại số ít frame dự đoán sai khi khuôn mặt người có sự di chuyển nhanh.

## VII. Phân chia công việc

<i>Thành viên</i>	<i>Công việc</i>
<b>Trịnh Tuấn Nam</b>	<b>Face Detection</b>
<b>Nguyễn Hoài Nam</b>	<b>Age Detection</b>
<b>Nguyễn Dương Hải</b>	<b>Gender Detection</b>

## # Tài liệu tham khảo:

- <https://www.miai.vn/2019/08/09/yolo-series-2-cach-train-yolo-de-detect-cac-object-dac-thu/>
- <https://github.com/ipazc/mtcnn>
- <https://viblo.asia/p/nhan-dien-khuon-mat-voi-mang-mtcnn-va-facenet-phan-1-Qbq5QDN4lD8>
- <https://towardsdatascience.com/face-detection-using-mtcnn-a-guide-for-face-extraction-with-a-focus-on-speed-c6d59f82d49>
- <https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P>
- <https://www.miai.vn/2020/05/25/yolo-series-train-yolo-v4-train-tren-colab-chi-tiet-va-day-du-a-z/>
- <https://towardsdatascience.com/yolov4-in-google-colab-train-your-custom-dataset-traffic-signs-with-ease-3243ca91c81d>
- <https://viblo.asia/p/haar-cascade-la-gi-luan-ve-mot-ky-thuat-chuyen-dung-de-nhan-biet-cac-khuon-mat-trong-anh-E375zamdlGW>
- [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
- <https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/>
- <https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/>
- [https://colab.research.google.com/github/kozodoi/website/blob/master/\\_notebooks/2021-03-08-image-mean-std.ipynb#scrollTo=CIYKt5CQDILS](https://colab.research.google.com/github/kozodoi/website/blob/master/_notebooks/2021-03-08-image-mean-std.ipynb#scrollTo=CIYKt5CQDILS)
- <https://kozodoi.me/python/deep%20learning/pytorch/tutorial/2021/03/08/image-mean-std.html>
- <https://learnopencv.com/edge-detection-using-opencv/>
- <https://www.ijert.org/research/comparison-of-canny-edge-detector-with-sobel-and-prewitt-edge-detector-using-different-image-formats-IJERTCONV2IS03009.pdf>

---

*Cảm ơn Thầy đã xem !*