Facial Keypoints Detection

Ngô Thành Phát[1] - 19521994
Âu Thiên Phước[2] - 19522050

***Abstract*— Trong dự án của chúng tôi. chúng tôi muốn xác định vị trí các chính trên khuôn mặt trong một hình ảnh nhất định bằng cách sử dụng các kiến trúc học sâu để không chỉ có được tổn thất thấp hơn cho nhiệm vụ phát hiện mà còn đẩy nhanh quá trình đào tạo và thử nghiệm cho các ứng dụng trong thế giới thực. Chúng tôi đã xây dựng một mạng lưới thần kinh tích chập cơ bản làm đường cơ sở của chúng tôi. Và chúng tôi sẽ đề xuất một cách tiếp cận để định vị tốt hơn các tọa độ của các điểm bàn phím khuôn mặt với các tính năng được giới thiệu khác với đầu vào thô. Và cuối cùng chúng tôi sử dụng các thao tác khác nhau trên bộ dữ liệu để tính toán vectơ đầu ra cuối cùng. Các kết quả thí nghiệm đã cho thấy hiệu quả của các cấu trúc sâu đối với các tác vụ phát hiện bàn phím khuôn mặt đã cải thiện một chút hiệu suất của phát hiện so với các phương pháp cơ bản.**

I. Giới Thiệu

A. Động lực

Động lực chính của chúng tôi cho dự án này là sự quan tâm của chúng tôi trong việc áp dụng học sâu vào các vấn đề quan trọng với mục đích sử dụng phù hợp. Các ứng dụng của nghiên cứu này rất nhiều và quan trọng, bao gồm phân tích nét mặt, sinh trắc học hoặc nhận dạng khuôn mặt, chẩn đoán y tế về biến dạng khuôn mặt. Ý tưởng rằng chúng tôi đang làm việc để giải quyết một vấn đề mở và quan trọng với tất cả những ứng dụng khả thi này đã truyền cảm hứng rất nhiều chúng ta. Chúng tôi đặc biệt chọn cuộc thi Kaggle phát hiện điểm chính trên khuôn mặt vì nó mang lại cho chúng tôi nhiều cơ hội thử nghiệm nhiều cách tiếp cận và mô hình mạng lưới thần kinh khác nhau để giải quyết một vấn đề khác. Yếu tố cạnh tranh cũng cho phép chúng tôi đánh giá kết quả so với cộng đồng lớn hơn, và so sánh, tính hiệu quả của các phương pháp của chúng tôi so với các phương pháp thay thế. Cuối cùng, chúng tôi được thúc đẩy bởi những thách thức liên quan đến vấn đề. Phát hiện các điểm chính trên khuôn mặt là một vấn đề đầy thách thức với các biến thể trong cả đặc điểm khuôn mặt cũng như điều kiện hình ảnh. Đặc điểm khuôn mặt khác nhau tùy theo kích thước, vị trí, tư thế và biểu cảm, trong khi điều kiện hình ảnh thay đổi theo độ sáng và góc xem. Những biến thể phong phú này, kết hợp với sự cần thiết cho cao dự đoán tọa độ chính xác (ví dụ góc chính xác của một con mắt) khiến chúng tôi tin rằng đây sẽ là một chủ đề sâu sắc và thú vị

B. Báo cáo vấn đề

Nhiệm vụ này là một cuộc thi kaggle hiện đang hoạt động, được chỉ định[1], đã bắt đầu vào tháng 5 năm 2013 và đã hết hạn vào tháng 12 năm 2016. Mục tiêu của chúng tôi là xác định vị trí 15 bộ điểm chính trên khuôn mặt khi được cung cấp một hình ảnh khuôn mặt thô. Đầu vào là một tập hợp các hình ảnh khuôn mặt thô 96×96 chỉ có các giá trị pixel thang độ xám và đầu ra là một vectơ 30 chiều, biểu thị (x, y) tọa độ của 15 bộ điểm chính trên khuôn mặt Dưới đây là ví dụ đầu vào(hàng đầu tiên) và đầu ra (hàng thứ hai)

Fig. 1. Input and Output Example.

Trong dự án của chúng tôi, chúng tôi sẽ sử dụng các cấu trúc sâu để phát hiện các điểm chính trên khuôn mặt, cấu trúc này có thể học tốt từ các khuôn mặt khác nhau và khắc phục sự khác biệt giữa các khuôn mặt của những người khác nhau hoặc của

các điều kiện khác nhau ở mức độ lớn. Mô hình được sử dụng rộng rãi là mạng lưới thần kinh tích chập sẽ được thiết kế làm đường cơ sở trong dự án của chúng tôi.Quan trọng nhất, chúng tôi đã sử dụng mô hình Resnet-50 [2] để khám phá một số kỹ thuật để giảm độ phức tạp tính toán để phát hiện các điểm bàn phím khuôn mặt. Khi so sánh với các mô hình cơ sở của chúng tôi, chúng tôi có thể thấy một sự cải thiện tuyệt vời khi sử dụng các mô hình khởi động trước để dự đoán vị trí của các điểm bàn phím khuôn mặt. Những đóng góp của bài viết này bao gồm:

1. Khám phá hiệu suất của các cấu trúc sâu khác nhau trong nhiệm vụ phát hiện các điểm bàn phím khuôn mặt và đánh giá hiệu quả bằng cách sử dụng tổn thất MSE.

2. Sử dụng mô hình cơ sở và mô hình Resnet-50 phát hiện các điểm bàn phím khuôn mặt. Mặc dù mô hình Resnet-50 được đào tạo về nhiệm vụ phân loại hình ảnh trên ImageNet, kết quả thí nghiệm đã cho thấy các tính năng trung gian có thể thích nghi với nhiệm vụ phát hiện điểm chính khuôn mặt tốt.

3. Tiến hành các thí nghiệm trên các bộ dữ liệu trong thế giới thực từ Kaggle Challenge và khái quát mô hình của chúng tôi có thể dễ dàng mở rộng sang nhiệm vụ phát hiện khuôn mặt khác.

Trong Phần 2,chúng tôi mô tả kiến trúc học sâu cơ bản và mô hình Resnet-50 của chúng tôi một cách chi tiết. Bộ dữ liệu từ Thử thách Kaggle mà chúng tôi sử dụng cho bài viết này được phân tích trong Phần 3. Kết quả thử nghiệm được tóm tắt trong Phần 4. Chúng tôi thực hiện khái quát hóa bài toán trong phần 5. Và Chúng tôi kết thúc công việc của chúng tôi trong Phần 6.

II. Tiếp cận kỹ thuật

Trong phần này, đầu tiên chúng ta sẽ xây dựng mô hình xử lý vấn đề phát hiện điểm chính và giới thiệu các số liệu hiệu suất. Để giải quyết vấn đề này, chúng tôi đã xây dựng một mô hình nơ-ron tích chập đơn giản dựa trên ý tưởng của VGG . Sau đó là phần chính của phần này, mô hình Resnet-50 [2] sẽ được giới thiệu và phân tích chi tiết. Đặc biệt, chúng tôi sẽ thể hiện những lợi thế ấn tượng của nó so với các mô hình nơ-ron tích chập truyền thống,

A. Simple CNN

Chúng ta có một kiến trúc CNN chứa 5 lớp chính, và 1 lớp chính được kết nối đầy đủ và một lớp đầu ra mục tiêu. Mỗi lớp chính chứa 1 lớp tích chập(convotional layer) và một lớp tổng hợp(pooling layer) có kích thước 2x2. Lớp tích chấp đầu tiên có 16 bộ lọc, lớp 2 32, lớp 3 64, lớp 4 128,lớp 5 256. Tất cả bộ lọc này có kích thước 3x3 và chúng ta sẽ sử dụng chức năng kích hoạt(activation) 'relu' trong những lớp này. Các giá trị được làm phẳng ở lớp thứ 5(Flatten()) sẽ được đưa đến lớp kết nối đầy đủ với 512 nút(nodes) sau đó các giá trị cuối cùng sẽ được gửi đến lớp đầu ra có 30 nút với chức năng lỗi bình phương trung bình. Sơ đồ kiến trúc của CNN của chúng tôi được hiển thị trong Hình 2

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 96, 96, 16)        160

 max_pooling2d (MaxPooling2D  (None, 48, 48, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 48, 48, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 24, 24, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 24, 24, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 12, 12, 64)       0
 2D)

 conv2d_3 (Conv2D)           (None, 12, 12, 128)       73856

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 128)        0
 2D)

 conv2d_4 (Conv2D)           (None, 6, 6, 256)         295168

 max_pooling2d_4 (MaxPooling  (None, 3, 3, 256)        0
 2D)

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 512)               1180160

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 30)                15390

=================================================================
Total params: 1,587,870
Trainable params: 1,587,870
Non-trainable params: 0
```

Fig. 2. Thông số của mô hình

B. Resnet-50

ResNet gần như tương tự với các mạng nơ-ron tích chập cơ bản gồm có convolution, pooling, activation và fully-connected layer. Những kiến trúc trước đây thường cải tiến độ chính xác nhờ gia tăng chiều sâu của mạng CNN[2]. Nhưng thực

nghiệm cho thấy đến một ngưỡng độ sâu nào đó thì độ chính xác của mô hình sẽ bão hòa và thậm chí phản tác dụng và làm cho mô hình kém chính xác hơn. Ý tưởng chính của Resnet là sử dụng kết nối tắt. Các kết nối tắt (skip connection) giúp giữ thông tin không bị mất bằng cách kết nối từ layer sớm trước đó tới layer phía sau và bỏ qua một vài layers trung gian. Và chúng ta sẽ triển khai từng bước giống tác giả đã thực nhưng bỏ qua phần tăng cường dữ liệu. Sơ đồ kiến trúc Resnet-50 mặc định của chúng tôi được hiển thị trong Hình 3

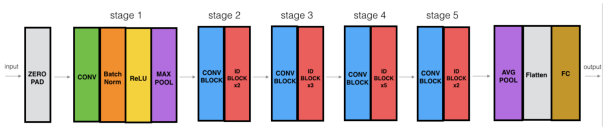

Fig. 3. Kiến trúc Resnet-50

"ID BLOCK" trong hình trên là viết tắt của từ Identity block và ID BLOCK x3 nghĩa là có 3 khối Identity block chồng lên nhau. Nội dung hình trên như sau :

Zero-padding : Input với (3,3)

Stage 1 : Tích chập (Conv1) với 64 filters với shape(7,7), sử dụng stride (2,2). BatchNorm, MaxPooling (3,3).

Stage 2 : Convolutiontal block sử dụng 3 filter với size 64x64x256, f=3, s=1. Có 2 Identity blocks với filter size 64x64x256, f=3.

Stage 3 : Convolutional sử dụng 3 filter size 128x128x512, f=3,s=2. Có 3 Identity blocks với filter size 128x128x512, f=3.

Stage 4 : Convolutional sử dụng 3 filter size 256x256x1024, f=3,s=2. Có 5 Identity blocks với filter size 256x256x1024, f=3.

Stage 5 :Convolutional sử dụng 3 filter size 512x512x2048, f=3,s=2. Có 2 Identity blocks với filter size 512x512x2048, f=3.

The 2D Average Pooling : sử dụng với kích thước (2,2).

The Flatten.

Fully Connected (Dense) : sử dụng softmax activation.

Chúng ta thực hiện thay đổi đầu vào trở thành 96x96x1 và thay đổi đầu ra trở thành:

GlovalAveragePoooling2D(): Tổng hợp toàn cầu ngưng tụ tất cả các bản đồ tính năng thành một bản đồ duy nhất, gộp tất cả các thông tin liên quan vào một bản đồ có thể dễ dàng hiểu được bằng một lớp phân loại dày đặc thay vì nhiều lớp

Dropout(): Trong mạng neural network, kỹ thuật dropout là việc chúng ta sẽ bỏ qua một vài unit trong suốt quá trình train trong mô hình, những unit bị bỏ qua được lựa chọn ngẫu nhiên. Ở đây, chúng ta hiểu "bỏ qua - ignoring" là unit đó sẽ không tham gia và đóng góp vào quá trình huấn luyện (lan truyền tiến và lan truyền ngược).Về chức năng là để chống over-fitting

Fully Connected(Dense): Sử dụng LeakyReLU activation

Sơ đồ kiến trúc Resnet-50 sẽ sử dụng của chúng tôi được hiển thị trong Hình 4

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 96, 96, 3)         6

 leaky_re_lu (LeakyReLU)     (None, 96, 96, 3)         0

 resnet50 (Functional)       (None, 3, 3, 2048)        23587712

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 30)                61470

=================================================================
Total params: 23,649,188
Trainable params: 23,596,068
Non-trainable params: 53,120
_________________________________________________________________
```

Fig. 4. Kiến trúc Resnet-50

Giải pháp ResNet đưa ra đơn giản hơn và tập trung vào cải thiện thông tin thông qua độ dốc của mạng. Sau ResNet hàng loạt biến thể của kiến trúc này được giới thiệu. Thực nghiệm cho thấy những kiến trúc này có thể được huấn luyện mạng nơ ron với độ sâu hàng nghìn lớp và nó nhanh chóng trở thành kiến trúc phổ biến nhất trong thị giác máy tính

III. Bộ Dữ liệu

Bộ dữ liệu của chúng tôi sử dụng được cung cấp bởi University of Montreal Medical , bao gồm dữ liệu đào tạo gồm 7049 hình ảnh 96x96x1 thang độ xám, với tọa độ (x, y) được gắn nhãn cho 15 điểm chính trên khuôn mặt. Các điểm chính trên khuôn mặt bao gồm bên phải, bên trái và trung tâm của mắt, lông mày, mũi, môi trên và môi dưới. Nhiệm vụ của chúng tôi là phát triển một mô hình có thể dự đoán các vị trí cụ thể của các điểm chính

trên khuôn mặt này trên các hình ảnh của bộ thử nghiệm 1783 với hàm mất mát MSE (Lỗi bình phương trung bình).

Trong số 7049 hình ảnh đào tạo, tồn tại một tập hợp con riêng biệt của 2140 được dán nhãn đầy đủ và chính xác, trong khi những ảnh còn lại đều không được gắn nhãn đầy đủ. Chúng ta sẽ sử dụng hai hàm xử lý dữ liệu trong thư viện Pandas đó là Dropna() và Fillna(). Về cơ bản Dropna loại bỏ những phần tử bị missing data, ngược lại với Dropna thì Fillna sẽ điền vào những cột còn thiếu. Chúng ta thực hiên cả hai phương thức cho từng mô hình

Chúng ta sẽ có hai bộ dữ liệu bao gồm đào tạo: 2140 ảnh và 7049 ảnh được chia ngẫu nhiên thành tập huấn luyện, tập xác thực với tỷ lệ 85 và 15.

15 điểm bàn phím khuôn mặt trên một khuôn mặt nhất định được liệt kê dưới đây:

left_eye_center	right_eye_center
left_eye_inner_corner	left_eye_outer_corner
right_eye_inner_corner	right_eye_outer_corner
left_eyebrow_inner_end	left_eyebrow_outer_end
right_eyebrow_inner_end	right_eyebrow_outer_end
nose_tip	mouth_left_corner
mouth_right_corner	mouth_center_top_lip
mouth_center_bottom_lip	

Fig. 5. 15 facial keypoints

IV. Thực Nghiệm

Trong phần này, chúng tôi sẽ chứng minh cách chúng tôi tiến hành các thí nghiệm trên mô hình cơ sở và mô hình Resnet-50 một cách chi tiết. Đầu tiên, chúng tôi giới thiệu các số liệu đánh giá để đánh giá hiệu suất của tất cả các mô hình trong cả độ chính xác phát hiện và tổn thất.

Để đánh giá các mạng của chúng tôi, chúng tôi so sánh tổn thất sau một số lượng kỷ nguyên nhất định (200). Các đường cong của tổn thất giảm được thể hiện như sau:

A. Số liệu đánh giá (Evaluation Metrics)

Đối với các thử nghiệm của chúng tôi, với tư cách là chỉ số đánh giá tổn thất hồi quy, chúng tôi sử dụng lỗi bình phương trung bình (MSE) giữa vectơ tọa độ điểm chính thực và vectơ dự đoán.

Mean Squared Error (MSE) có lẽ là số liệu phổ biến nhất được sử dụng cho các bài toán hồi quy.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

Fig. 6.

Về cơ bản, nó tìm thấy sai số bình phương trung bình giữa các giá trị được dự đoán và thực tế. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.Trong đó n là số điểm dữ liệu, y là giá trị quan sát và ^y là giá trị dự đoán. Nó thực hiện điều này bằng cách lấy khoảng cách từ các điểm đến đường hồi quy (những khoảng cách này là "sai số") và bình phương chúng. Bình phương là rất quan trọng để giảm độ phức tạp với các dấu hiệu tiêu cực. Nó cũng tạo ra nhiều trọng lượng hơn cho sự khác biệt lớn hơn

B. Thiết lập thử nghiệm

Chúng tôi tiến hành các thí nghiệm trên tổng cộng 4 mô hình, 2 trong số đó là các mô hình cơ sở và 2 là mô hình Resnet-50. Nền tảng thử nghiệm là máy tính xách tay Lenovo 4 core 1.8 GHz GPU + bộ nhớ 8GB 1867MHz và sử dụng ngôn ngữ lập trình Python. Mã thực hiện bằng GPU được cung cấp ngẫu nhiên Google Colab.

C. Kết quả thực nghiệm

Các kết quả thí nghiệm bao gồm hai phần. Đầu tiên, chúng tôi tập trung vào độ chính xác của phát hiện bằng cách so sánh tổn thất đào tạo, xác nhận và kiểm tra giữa tất cả 4 mô hình được minh họa ở trên. Tiếp theo chúng tôi sẽ thực hiện so sánh của hiệu suất của mô hình trên bộ kiểm định.

Cả tổn thất đào tạo và xác thực trong cả 4 mô hình đều giảm khi số kỷ nguyên tăng lên. Con số cụ thể có thể được tìm thấy trong bảng bên dưới

Iter/10^{-3}	CNN_drop		CNN_fill		Resnet-50_Drop		Resnet-50_fill	
	Train	Valid	Train	Valid	Train	Valid	Train	Valid
50	0.001	0.003	6.1045e-04	0.0012	1.58	5.34	6.94	7.2
100	5.6210e-04	0.0029	3.6358e-04	0.0012	1.09	4.5	5.84	5.41
150	3.7468e-04	0.0031	2.9410e-04	0.0011	1.01	4.45	5.38	4.91
200	26664e-04	0.0029	2.9001e-04	0.0011	1.01	4.45	5.1	4.63

Fig. 7.

- Đối với một mô hình CNN cho bộ dữ liệu dropout , đây là kiến trúc sâu đơn giản dựa

ý tưởng VGG. Và hiệu suất của mô hình đạt mức tốt nhất ở epochs thứ 114 là 0.5981 và tổn thất giảm gần như hội tụ

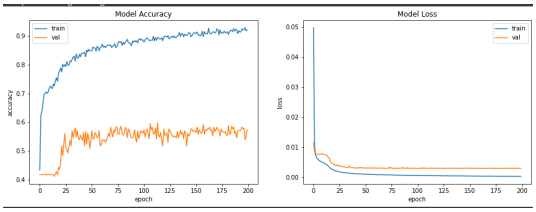

Fig. 8.

- Đối với mô hình CNN cho bộ dữ liệu fill , kết quả tổn thất được thể hiện ở cột thứ ba trong bảng, so với mô hình CNN dropout , tổn thất tốt hơn một chút và hiệu suất được cải thiện lên đến 0.8053 ở epochs 38 và 0.7665 ở epoch 200.

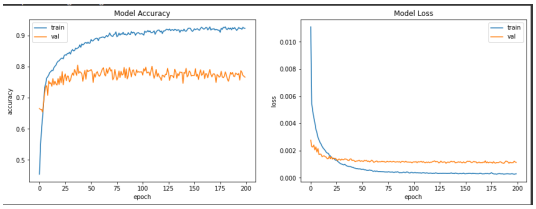

Fig. 9.

- Đối với mô hình Resnet-50 cho bộ dữ liệu drop , kết quả tổn thất được thể hiện ở cột thứ tư trong bảng, hiệu suất mô hình giai đoạn đầu có sự tăng trưởng rõ rệt trong khoảng epochs từ 25 đến 50 sau đó biến thiên và duy trì ở mốc 0.5763. Để giải thích cho việc này, chúng tôi nghĩ có lẽ do bộ dữ liệu quá nhỏ(2140 ảnh).

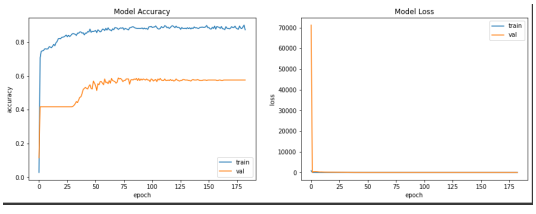

Fig. 10.

- Mô hình Resnet-50 cho bộ dữ liệu fill , kết quả tổn thất được thể hiện ở cột thứ năm trong bảng. Tương tự như mô hình Resnet-50 với bộ dữ liệu drop thì hiệu suất của mô hình này tăng trưởng ổn định trong 50 epochs đầu tiền và duy trì ở các epochs tiếp sau.

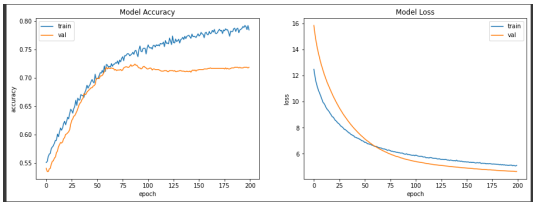

Fig. 11.

Sau đó chúng tôi thực hiện so sánh hiệu suất của các mô hình được thực nghiệm, chi tiết được hiển thị chi tiết ở bảng bên dưới:

	CNN_drop	CNN_fill	Resnet-50_drop	Resnet-50_fill
Accuracy	0.92	0.92	0.91	0.7842
Val_Accuracy	0.59	**0.76**	0.59	0.71

Fig. 12.

Xem xét tổng quan, hiệu suất có mô hình Resnet-50 không đạt được kỳ vọng của chúng tôi. Với lượng tham số lên đến con số 26 triệu thì chúng tôi nghĩ mô hình này phức tạp so với bài toán phát hiện điểm chính khuôn mặt. Tuy nhiên chúng tôi sẽ tiếp tục làm việc để xem xét những tham số ảnh hưởng đến vấn đề hiệu suất này.

V. Khái quát bài toán

Chúng ta cần phải đảm bảo những hình ảnh đầu vào là những hình ảnh khuôn mặt màu xám có kích thước 96x96 chứ không phải một hình ảnh có nhiều khuôn mặt ở những vị trí khác nhau, điều này làm giảm khả năng khái quát của bài toán. Để giải quyết vấn đề này, chúng ta cần một hệ thống có thể giúp chúng ta phát hiện ra những khuôn mặt trong ảnh. Để thực hiện chúng ta có thể tạo ra một model khác hoặc sử dụng thuật toán nhận diện khuôn mặt Haar Cascade

Phát hiện đối tượng Sử dụng các phân loại tầng dựa trên tính năng Haar là một phương pháp phát hiện đối tượng hiệu quả được đề xuất bởi Paul Viola và Michael Jones trong bài báo của họ, "Rapid Object Detection using a Boosted Cascade of Simple Features" vào năm 2001. Tính năng Haar: Độ sáng của các

vùng trên khuôn mặt là khác nhau. Ví dụ: Vùng mắt tối hơn vùng má, vùng mũi sáng hơn vùng hai bên. Sử dụng cửa số 24x24 để đánh các đặc trưng của ảnh. Tại mỗi pixel, bạn sẽ trích xuất những các đặc điểm,và sẽ phân loại các điểm mà bạn đã trích xuất là khuôn mặt hay không phải khuôn mặt.

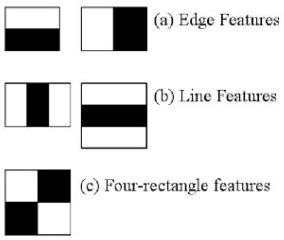

Fig. 13.

Đây là một mô hình đã được đào tạo từ trước và cho phép mọi người sử dụng mà không phải đào tạo lại từ đầu. Tuy nhiên chúng phải xem xét một số yêu cầu khi sử dụng. Bao gồm: ảnh chính diện(khuôn mặt đủ lớn), ánh sáng tốt, không vật cản trước gương mặt, nước da sáng.

A. *Thử ngiệm ảnh trên Simple CNN*

Fig. 14. Input

VI. Hướng cải thiện và phát triển

A. *Dữ liệu*

Áp dụng thêm các kỹ thuật Data Augmentation (flip, rotane, alter brightness, ...). Chọn

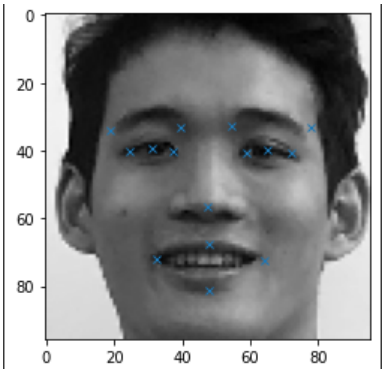

Fig. 15. Output

lựa phù hợp các kỹ thuật tăng cường khác nhau để phù hợp với bộ dữ liệu. Tăng cường sự đa dạng của dữ liệu bằng cách thu thập thêm nhiều ảnh về các khuôn mặt ở độ tuổi khác nhau.

B. *Mô hình*

Áp dụng thêm nhiều pre-trained model khác nhau để có thể tìm được mô hình phù hợp nhất với bài toán và bộ dữ liệu.Áp dụng một số kỹ thuật như thay đổi cấu trúc mô hình, tùy chỉnh tham số để có thể cải thiện mô hình hơn.

References

[1] 1. Facial Keypoint Detection Competition. Kaggle, 7 May 2013. Web. 31 Dec. 2016. ¡https://www.kaggle.com/c/facial-keypointsdetection¿. 1.2, 1.3
[2] https://phamdinhkhanh.github.io/2020/05/31/CNNHistory.html4-c
[3] https://arxiv.org/abs/1409.1556.
[4] https://www.kaggle.com/code/jiesun2007/facial-keypoint-detection-resnet50-imageaug
[5] https://www.youtube.com/watch?v=vC3bTziLRTA
[6] http://cs231n.stanford.edu/reports/2016/pdfs/007.Report.pdf