

# XÁC ĐỊNH CÁC HUYỆT ĐẠO MASSAGE TRÊN ĐẦU NGƯỜI

Nguyễn Hoài Nam<sup>1</sup> & PGS.TS Nguyễn Trường Thịnh<sup>2</sup>

<sup>1</sup>Khoa Đào tạo chất lượng cao, Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh, Việt Nam

<sup>2</sup>Khoa Cơ Khí Chế Tạo Máy, Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh, Việt Nam

Thông tin liên hệ: [19146219@student.hcmute.edu.vn](mailto:19146219@student.hcmute.edu.vn)

Mã số sinh viên: 19146219

## TÓM TẮT

*Trong một xã hội đang ngày càng phát triển, con người dần biết chú trọng đến chất lượng cuộc sống và sức khỏe của bản thân, nhưng lại đòi hỏi ít tốn kém thời gian và chi phí. Có một liệu pháp đã có từ xa xưa bắt nguồn từ Y học cổ truyền Việt Nam, có thể đáp ứng được nhu cầu chăm sóc, cải thiện, chữa lành và tăng cường sức khỏe nhưng lại không cần dùng thuốc, hạn chế được thời gian và chi phí đáng kể. Massage các huyết đạo trên cơ thể chính là liệu pháp đó và đặc biệt trong bài báo này tác giả chủ yếu đề cập đến massage các huyết đạo trên đầu người. Việc massage các huyết đạo có thể thực hiện ngay tại nhà mà không cần đến các phòng khám, tuy nhiên việc xác định chính xác các huyết đạo lại là điều vô cùng quan trọng. Xây dựng mô hình dự đoán xác định các điểm huyết đạo trên đầu người là một giải pháp tốt cho nhu cầu này. Bài toán đưa ra áp dụng CNN (mạng neural tích chập) là một mô hình thuộc Deep Learning hỗ trợ cho việc xây dựng mô hình nhận diện điểm một cách thông minh và chính xác. Trong bài báo này sẽ giới thiệu quy trình tạo ra một mô hình mạng neuron tích chập dùng để xác định các huyết đạo massage trên đầu người bao gồm các bước như tạo tập dữ liệu, tiền xử lý dữ liệu, xây dựng các lớp tích chập, kiểm chứng và ứng dụng cho người dùng.*

**Từ khóa:** Artificial Intelligence, Deep Learning, CNN, Keypoint Detection

## 1. Giới thiệu

Trong những năm gần đây, việc sử dụng kết hợp các liệu pháp điều trị giữa y học cổ truyền và y học hiện đại đang dần phổ biến rộng rãi. Con người ngày một linh hoạt hơn trong vấn đề chăm sóc và cải thiện sức khỏe, để vẫn đảm bảo được nhu cầu sức khỏe nhưng tiết kiệm được thời gian và chi phí, họ đã tìm đến y học cổ truyền với các liệu pháp

massage huyết đạo có thể thực hiện ngay tại nhà mà không cần tìm đến các phòng khám. Máu huyết lưu thông, điều hòa hô hấp, cải thiện tim mạch, giảm đau, xua tan căng thẳng, mệt mỏi, thư giãn, điều trị mất ngủ và ngăn chặn các bệnh mãn tính là lợi ích từ liệu pháp massage các huyết đạo. Nhưng việc xác định chính xác các điểm huyết đạo không phải điều đơn giản và có thể cần tới các

chuyên gia, bác sĩ đông y có kinh nghiệm. Nhận thấy sự khó khăn trên, tác giả đã lên ý tưởng và thiết kế một mô hình mạng neural nhân tạo để nhận diện các điểm huyết đạo. Do số lượng huyết đạo trên cơ thể người là khá lớn (108 huyết đạo), nên tác giả đã giới hạn và chỉ thực hiện việc xác định các huyết đạo massage chính trên đầu người.

Trong nghiên cứu này, mô hình xây dựng áp dụng CNN (Convolution Neural Network) là một kiến trúc mạng được ứng dụng phổ biến và rộng rãi trong các bài toán phân tích và nhận diện hình ảnh mang lại độ chính xác và hiệu quả cao.

Ứng dụng thực tiễn của mô hình là sau khi thu thập dữ liệu hình ảnh đầu người từ camera, hình ảnh sẽ được đưa vào mô hình trải qua các bước xử lý, sau đó so sánh với các hình ảnh đã được gắn nhãn vị trí các điểm huyết đạo trước đó. Trải qua quá trình đánh giá thông qua các thuật toán phức tạp bên trong, mô hình sẽ tiến hành dự đoán tọa độ các điểm huyết đạo tương ứng với vị trí khuôn mặt của hình ảnh đưa vào và hiển thị cho người dùng biết các vị trí huyết đạo thông qua trực quan hóa điểm dữ liệu. Bên cạnh đó, mô hình dự đoán huyết đạo này, còn áp dụng cho các loại máy móc, thiết bị

massage trong việc tự tìm ra các điểm huyết đạo để thực hiện các chức năng của chúng.

Đối với mô hình, tác giả giới hạn nhận diện 6 điểm huyết đạo massage chính trên đầu người là: Thái Dương, Dương Bạch, Toàn Trúc, Ấn Đường, Đầu Duy và Bách Hội.

## 2. PHƯƠNG PHÁP

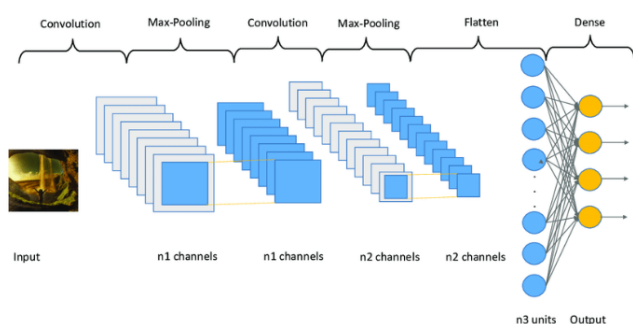
Mô hình sẽ được xây dựng dựa trên nền tảng Mạng nơron tích chập (Convolutional Neuron Network) đây là một trong những mô hình của Deep Learning. Tập hợp các thuật toán để có được mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp để xử lý cấu trúc phức tạp. CNN là một lớp của mạng nơron sâu, được áp dụng phổ biến nhất để phân tích hình ảnh trực quan. CNN được thiết kế với mục đích xử lý dữ liệu thông qua nhiều lớp mạng. Ngoài ra, CNN có thể giúp tạo ra được hệ thống thông minh, phản ứng với độ chính xác khá cao.

Mục đích chính của CNN là trích xuất ra các đặc trưng và giảm được kích thước của dữ liệu đầu vào. Convolutional: là một ma trận nhỏ được gọi với tên kernel, filter hay feature detection với kích thước nhỏ trượt trên ma trận đầu vào. Có thể nói Convolutional là lớp rất quan trọng vì đây thành phần thực hiện tính toán, tìm ra các

đặt trung. Các yếu tố chính trong một convolutional layer là: stride, padding, filter map và feature.

Cấu tạo cơ bản của một mô hình CNN bao gồm các thành phần:

- Convolutinal layers.
- Relu layers.
- Pooling layers.
- Fully connected layers.



Hình 1. Convolution Neural Network

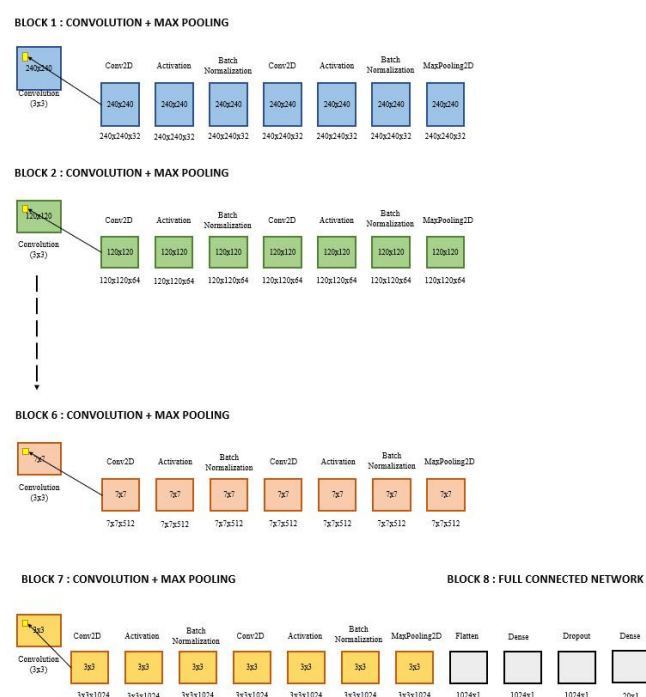
### 3. TRIỂN KHAI

Đầu tiên là bước tiền xử lý dữ liệu, dữ liệu thô ban đầu là các tấm ảnh chụp khuôn mặt ở nhiều góc xoay. Sau đó, dựa vào lý thuyết y học cổ truyền Việt Nam để tiến hành gắn nhãn cho các vị trí của các điểm huyết đạo đã xác định trước lên các tấm ảnh và tiến hành xuất các giá trị tọa độ của các điểm huyết đạo đó. Tiếp theo, tiến hành lọc và làm sạch dữ liệu bằng cách loại bỏ các thông tin không cần thiết. Dữ liệu sau quá trình tiền xử lý cần bao gồm 2 yếu tố:

- Tọa độ của điểm dữ liệu
- Tên lớp mà điểm đó thuộc vào

Sau quá trình tiền xử lý các dữ liệu đầu vào, tác giả sẽ đi đến công đoạn vô cùng quan trọng và cốt lõi đó là việc xây dựng mô hình. Dưới đây là mô hình mà tác giả đã xây dựng, mô hình có tổng cộng 8 Block và sẽ được trình bày cụ thể sau đây:

Hình 2. Mô hình CNN tự xây dựng



Ở Block 1, có thể thấy ảnh đầu vào với kích thước là  $240 \times 240 \times 3$ , tác giả sử dụng 32 kernel với kích thước là  $3 \times 3$  để trượt qua từng tấm ảnh gốc nhằm làm nổi bật đặc trưng của các bức ảnh. Sử dụng thêm “padding = same” để giữ lại kích thước ban đầu của ảnh và “use\_bias: False” tức không sử dụng vector thiên vị. Lúc này kích thước ảnh đã trở thành  $32 \times 32$  với 32 lớp màu

tương đương 32x32x32. Sau đó, tác giả thêm một lớp Activation LeakyReLU (alpha = 0.1) là một biến thể nổi tiếng ReLU phi tuyến để có thể xấp xỉ được các phân bố phức tạp và không làm thay đổi kích thước ban đầu của dữ liệu ảnh đầu vào. Tiếp theo là lớp Batch Normization để phân bố của dữ liệu đồng nhất với dữ liệu đầu vào ban đầu vì qua các lớp như Convolution hay Activation thì phân bố của dữ liệu so với đầu vào là rất lớn. Tiếp tục thêm các lớp tương tự như trên để tăng độ phức tạp và độ dày đặc cho mô hình. Cuối cùng tác giả thêm 1 lớp MaxPooling2D dùng để chọn ra những đặc trưng nổi bật nhất làm tăng độ chính xác đồng thời cũng làm giảm kích thước ảnh.

Block 2 sẽ tương tự như Block 1 tuy nhiên tác giả sẽ giảm kích thước ảnh xuống thành 120x120 nhưng số lượng lớp màu sẽ tăng lên thành 64.

Từ Block 3,4,5 và 6 sẽ tương tự như Block 2 qua mỗi Block kích thước ảnh sẽ giảm dần đi một nửa và số lượng lớp màu sẽ tăng lên. Đến Block 7 kích thước ảnh được giảm xuống chỉ còn 3x3 và số lượng lớp màu lúc này tăng lên thành 1024.

Cuối cùng là Block 8, sau khi sử dụng các lớp tích chập để trích xuất được các đặc trưng

của ảnh thì ảnh sẽ qua 1 lớp Flatten để trải phẳng ma trận 2 chiều thành vector 1 chiều trước khi đưa vào lớp Dense. Tiếp theo là lớp Dropout, thông thường khi tắt cả các đặc trưng được kết nối với lớp Dense, nó có thể gây ra sự quá tải trong bộ dữ liệu đào tạo. Quá tải xảy ra khi mô hình hoạt động rất tốt trên dữ liệu đào tạo gây ra tác động tiêu cực đến hiệu suất của mô hình khi được sử dụng trên dữ liệu mới. Để khắc phục vấn đề này, một lớp Dropout cần được sử dụng. Cuối cùng, tác giả sẽ thu nhỏ số lớp sao cho phù hợp với số lượng giá trị cần dự đoán. Ở đây có 10 điểm huyết đạo tức 20 giá trị đầu ra trong đó có 10 giá trị tọa độ x và 10 giá trị tọa độ y tương ứng.

Mô hình sẽ được cụ thể hóa thông qua bản tóm tắt như sau:

Model: "sequential"			
Layer (type)	Output Shape	Param #	
conv2d (Conv2D)	(None, 240, 240, 32)	864	max_pooling2d_2 (MaxPooling2D (None, 30, 30, 96)) 0
leaky_re_lu_1 (LeakyReLU)	(None, 240, 240, 32)	0	conv2d_6 (Conv2D) (None, 30, 30, 128) 118892
batch_normalization_1 (Batch Normalization (None, 240, 240, 32))	128		leaky_re_lu_6 (LeakyReLU) (None, 30, 30, 128) 0
conv2d_1 (Conv2D) (None, 240, 240, 32)	9216		batch_normalization_6 (Batch Normalization (None, 30, 30, 128)) 512
leaky_re_lu_1 (LeakyReLU) (None, 240, 240, 32)	0		conv2d_7 (Conv2D) (None, 30, 30, 128) 147456
batch_normalization_1 (Batch Normalization (None, 240, 240, 32))	128		leaky_re_lu_7 (LeakyReLU) (None, 30, 30, 128) 0
max_pooling2d (MaxPooling2D) (None, 120, 120, 32)	0		batch_normalization_7 (Batch Normalization (None, 30, 30, 128)) 512
conv2d_2 (Conv2D) (None, 120, 120, 64)	18432		max_pooling2d_3 (MaxPooling2D (None, 15, 15, 128)) 0
leaky_re_lu_2 (LeakyReLU) (None, 120, 120, 64)	0		conv2d_8 (Conv2D) (None, 15, 15, 256) 294912
batch_normalization_2 (Batch Normalization (None, 120, 120, 64))	256		leaky_re_lu_8 (LeakyReLU) (None, 15, 15, 256) 0
conv2d_3 (Conv2D) (None, 120, 120, 64)	36864		batch_normalization_8 (Batch Normalization (None, 15, 15, 256)) 1024
leaky_re_lu_3 (LeakyReLU) (None, 120, 120, 64)	0		conv2d_9 (Conv2D) (None, 15, 15, 256) 589824
batch_normalization_3 (Batch Normalization (None, 120, 120, 64))	256		leaky_re_lu_9 (LeakyReLU) (None, 15, 15, 256) 0
max_pooling2d_1 (MaxPooling2D (None, 60, 60, 64))	0		batch_normalization_9 (Batch Normalization (None, 15, 15, 256)) 1024
conv2d_4 (Conv2D) (None, 60, 60, 96)	85206		max_pooling2d_4 (MaxPooling2D (None, 7, 7, 256)) 0
leaky_re_lu_4 (LeakyReLU) (None, 60, 60, 96)	0		conv2d_10 (Conv2D) (None, 7, 7, 512) 1179648
batch_normalization_4 (Batch Normalization (None, 60, 60, 96))	384		leaky_re_lu_10 (LeakyReLU) (None, 7, 7, 512) 0
conv2d_5 (Conv2D) (None, 60, 60, 96)	82944		batch_normalization_10 (Batch Normalization (None, 7, 7, 512)) 2048
leaky_re_lu_5 (LeakyReLU) (None, 60, 60, 96)	0		conv2d_11 (Conv2D) (None, 7, 7, 512) 2359296
batch_normalization_5 (Batch Normalization (None, 60, 60, 96))	384		leaky_re_lu_11 (LeakyReLU) (None, 7, 7, 512) 0
			batch_normalization_11 (Batch Normalization (None, 7, 7, 512)) 2048

max_pooling2d_5 (MaxPooling2)	(None, 3, 3, 512)	0
conv2d_12 (Conv2D)	(None, 3, 3, 1024)	4718592
leaky_re_lu_12 (LeakyReLU)	(None, 3, 3, 1024)	0
batch_normalization_12 (Batch Normalization)	(None, 3, 3, 1024)	4896
conv2d_13 (Conv2D)	(None, 3, 3, 1024)	9437184
leaky_re_lu_13 (LeakyReLU)	(None, 3, 3, 1024)	0
batch_normalization_13 (Batch Normalization)	(None, 3, 3, 1024)	4896
max_pooling2d_6 (MaxPooling2)	(None, 1, 1, 1024)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 1024)	1049600
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 20)	20500
Total params: 20,120,116		
Trainable params: 20,119,668		
Non-trainable params: 448		

Hình 3. Bảng tóm tắt mô hình CNN tự xây dựng

Tiếp theo tác giả sẽ tiến hành xây dựng các hàm tối ưu cho quá trình huấn luyện.

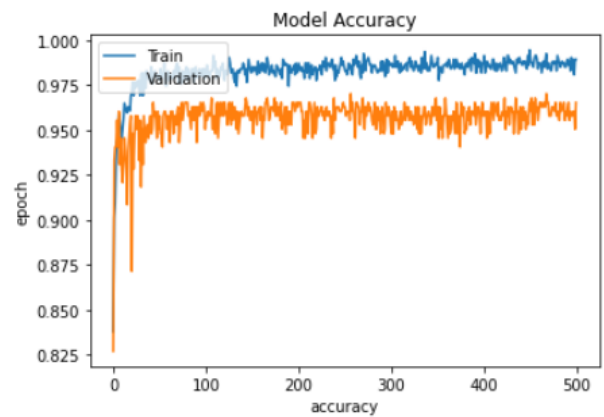
- **Optimizer = Adamax:** Là một biến thể của Adam dựa trên tiêu chuẩn vô cực. Adamax đôi khi vượt trội hơn adam, đặc biệt là trong các mô hình có nhúng.
- **Batch\_size = 4:** Số lượng dữ liệu trong 1 lần lấy ra để học là 4.
- **Epochs = 500:** Số lần huấn luyện là 500.
- **Loss = mean\_squared\_error:** hàm mất mát sai số toàn phương trung bình được sử dụng phổ biến trong các bài toán nhận diện.
- **Metrics = accuracy:** Thông số accuracy dùng để đánh giá mô hình.

Các hàm tối ưu và thông số trên đã được tác giả thử nghiệm nhiều lần để tìm ra kết quả tốt nhất cho mô hình Sau khi đã lựa chọn các thông số và hàm tối ưu tác giả tiến hành quá trình huấn luyện.

```
Epoch 491/500
237/237 [=====] - 7s 32ms/step - loss: 6.1210 - accuracy: 0.989
4 - val_loss: 31.7072 - val_accuracy: 0.9630
Epoch 492/500
237/237 [=====] - 7s 31ms/step - loss: 6.7350 - accuracy: 0.986
2 - val_loss: 28.8672 - val_accuracy: 0.9654
Epoch 493/500
237/237 [=====] - 7s 31ms/step - loss: 7.2743 - accuracy: 0.988
4 - val_loss: 22.9244 - val_accuracy: 0.9556
Epoch 494/500
237/237 [=====] - 7s 30ms/step - loss: 5.9459 - accuracy: 0.983
1 - val_loss: 27.2181 - val_accuracy: 0.9685
Epoch 495/500
237/237 [=====] - 7s 31ms/step - loss: 6.1287 - accuracy: 0.990
5 - val_loss: 18.8580 - val_accuracy: 0.9580
Epoch 496/500
237/237 [=====] - 8s 32ms/step - loss: 5.7247 - accuracy: 0.985
2 - val_loss: 33.7961 - val_accuracy: 0.9508
Epoch 497/500
237/237 [=====] - 7s 31ms/step - loss: 6.4331 - accuracy: 0.989
4 - val_loss: 21.7142 - val_accuracy: 0.9685
Epoch 498/500
237/237 [=====] - 7s 31ms/step - loss: 5.8672 - accuracy: 0.981
0 - val_loss: 25.1033 - val_accuracy: 0.9685
Epoch 499/500
237/237 [=====] - 7s 31ms/step - loss: 7.2161 - accuracy: 0.989
4 - val_loss: 23.5442 - val_accuracy: 0.9586
Epoch 500/500
237/237 [=====] - 7s 30ms/step - loss: 6.2760 - accuracy: 0.989
4 - val_loss: 28.3248 - val_accuracy: 0.9654
```

Hình 4. Quá trình huấn luyện

Trải qua quá trình huấn luyện là 500 lần, tác giả sẽ tiến hành đánh giá chất lượng của mô hình.



Hình 5. Biểu đồ đánh giá độ chính xác mô hình

- Sai số của tập dữ liệu huấn luyện là với 6.0695 độ chính xác là 0.9905.
- Sai số của tập dữ liệu kiểm chứng là 33.7961 với độ chính xác là 0.9508.

	<b>Huấn luyện</b>	<b>Kiểm chứng</b>
<b>Độ chính xác</b>	99.05	95.08



Có thể nhận thấy sai số của tập dữ liệu huấn luyện và dữ liệu kiểm chứng là khá cao. Nhưng độ chính xác của cả hai tập đều trên 90%. Với sai số cùng với độ chính xác như trên, mô hình sẽ có khả năng dự đoán đúng trên dưới 75%.

#### 4. KIỂM NGHIỆM MÔ HÌNH

Sau quá trình huấn luyện, tác giả tiến hành kiểm nghiệm và đánh giá trên tập dữ liệu kiểm tra và cả tập dữ liệu thực tế.

- Dữ liệu hình ảnh tập kiểm tra

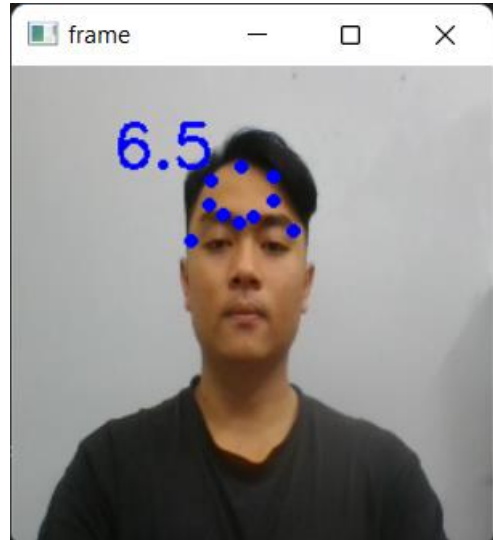


- Dữ liệu hình ảnh tập thực tế



#### 5. MÔ PHỎNG THỜI GIAN THỰC

Ngoài ra mô hình cũng có thể áp dụng cho việc mô phỏng theo thời gian thực:



Nhìn chung, mô hình nhận diện các điểm huyết đạo ở thời gian thực còn khá tệ một phần do nhiều yếu tố như nhiễu ánh sáng, độ phân giải của camera máy tính và phần cốt yếu là do lượng dữ liệu huấn luyện còn quá

ít, bên cạnh đó mô hình tự xây dựng vẫn chưa thực sự tốt. Có thể thấy tỷ lệ nhận diện chính xác nằm ở khoảng 30-40%.

## 6. ĐÁNH GIÁ VÀ KẾT LUẬN

Tỷ lệ xác định tọa độ các điểm huyết đạo của mô hình với tập dữ liệu kiểm tra là khá chính xác lên đến 80-90%. Còn tỷ lệ xác định đúng tọa độ các điểm huyết đạo của mô hình với dữ liệu hình ảnh thực tế bên ngoài là tương đối chính xác khoảng 70-80%. Trừ trường hợp các góc mặt khó, hình ảnh mờ và vị trí góc mặt với số lượng dữ liệu hình ảnh ít dẫn đến việc dự đoán chưa tối ưu và chính xác.

Nhìn chung, mô hình đưa ra kết quả dự đoán là tương đối ổn, chưa có sự chính xác tuyệt đối. Tuy nhiên, với số lượng dữ liệu đầu vào là 1350 tấm ảnh nên khó có thể đòi hỏi mô hình dự đoán với độ chính xác cao. Cần bổ sung thêm nhiều dữ liệu hơn với nhiều góc chụp khác nhau của khuôn mặt để mô hình có thể cải thiện khả năng nhận diện tốt hơn. Đồng thời, nên bổ sung dữ liệu hình ảnh của nhiều người hơn nữa để việc nhận diện huyết đạo trở nên đa dạng hơn.

Theo nhận định của tác giả mô hình xây dựng chưa thực sự tốt, ngoài việc bổ sung dữ liệu cũng cần cải thiện các thông số, bổ sung các hàm tối ưu, tối ưu hóa các lớp nơ ron và các

thành phần bên trong nhằm cải thiện khả năng dự đoán của mô hình.

## TÀI LIỆU THAM KHẢO

- [1] Vũ Hữu Tiệp: Machine Learning cơ bản
- [2] Nguyễn Thanh Tuấn, Deep Learning Cơ Bản.
- [3] Chris Albon. 2018, Python Machine Learning Cookbook
- [4] TopDev, Thuật toán CNN – Convolution Neural Network, <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>