

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CƠ KHÍ CHẾ TẠO MÁY

—oOo—



HCMUTE

PROJECT

Trí Tuệ Nhân Tạo

Xác Định Các Huyệt Đạo Massage Trên Đầu Người
(Dựa Vào Y Học Cổ Truyền VN)

MÔN: Trí tuệ nhân tạo
GVHD: Nguyễn Trường Thịnh
SVTH: Đinh Văn Bình
MSSV: 19146308

TP. Hồ Chí Minh, ngày 23 tháng 06 năm 2022

MỤC LỤC

PHẦN I: TỔNG QUAN	1
1.1 Giới thiệu đề tài.....	1
1.2 Lý do chọn đề tài.....	1
1.3 Mục tiêu nghiên cứu	2
1.4 Phương pháp nghiên cứu	2
PHẦN II: CƠ SỞ LÝ THUYẾT	3
2.1 Cơ sở lý thuyết về huyệt đạo	3
2.1.1 Huyệt đạo là gì?	3
2.1.2 Phương pháp xác định huyệt đạo	3
2.1.3 Huyệt đạo ở đầu	8
2.2 Thuật toán CNN – Convolutional Neural Network.....	11
2.2.1 CNN – Convolutional Neural Network là gì?.....	11
2.2.2 Cấu trúc của mạng CNN – Convolutional Neural Network	11
2.2.2.1 Trường tiếp nhận cục bộ	11
2.2.2.2 Trọng số chia sẻ	15
2.2.2.3 Lớp tổng hợp	16
2.2.3 Convolutional là gì?	18
PHẦN III: ỨNG DỤNG - KẾT QUẢ THỰC NGHIỆM	19
3.1 Dataset.....	19
3.2 Training.....	20

PHẦN IV: KẾT LUẬN.....	25
4.1 Kết luận	25
4.2 Ưu điểm – Nhược điểm.....	25
4.3 Hướng phát triển.....	25
TÀI LIỆU THAM KHẢO	26

PHẦN I: TỔNG QUAN

1.1 Giới thiệu đề tài

Hiện nay, tình hình hiện tại của dịch Covid-19 đã giảm sút nhưng hậu Covid-19 đã để lại cho con người nhiều hậu quả xấu ảnh hưởng đến sức khỏe sau này. Do đó con người cần có những phương pháp cải thiện sức khỏe cho bản thân sau dịch Covid-19 đầy gian nan và khó khăn. Ngoài những phương pháp điều trị thông thường bằng thuốc hay rèn luyện thể dục thì việc điều trị bằng phương pháp Y học cổ truyền cũng được con người áp dụng khá nhiều vào lúc này. Massage đầu có lợi cho con người, làm giảm căng thẳng, kích thích tuần hoàn máu não, thúc đẩy thư giãn và giảm các triệu chứng đau đầu căng thẳng, hay trong nghiên cứu của Keshavarz đã chỉ ra rằng xoa bóp đầu làm giảm kích động ở bệnh nhân Alzheimer cao tuổi. Với những lợi ích của việc gội đầu và massage đầu, vì vậy nó rất cần thiết cho con người. Hiện tại, việc chăm sóc bệnh nhân sau các đợt điều trị đặc biệt như xạ trị, phẫu thuật, vv được thực hiện bởi nhân viên y tế tại bệnh viện, điều này gây ra tình trạng thiếu nhân sự trầm trọng. Với sự phát triển của khoa học công nghệ, việc nghiên cứu ứng dụng vào thực tế là rất cần thiết để giải quyết khó khăn về con người, giảm quá tải cho nhân viên y tế, giảm nguy cơ lây nhiễm giữa các bệnh nhân.

1.2 Lý do chọn đề tài

Việc áp dụng AI – Trí tuệ nhân tạo vào y tế là một hướng phát triển mới cho thời đại này, chúng ta có thể dùng AI phát hiện những huyết đạo trên đầu của con người bằng cách cho nó học và có thể hiểu nhận diện được các huyết đạo ấy mà không cần phải dùng những phương pháp xác định cổ truyền hay xưa nay của cha ông ta. Vì vậy em quyết định chọn dự án ” Xác Định Các Huyết Đạo Massage Trên Đầu Người (Dựa Vào Y Học Cổ Truyền VN)” để có thể giúp cho bộ y tế giải quyết vấn đề xác định huyết đạo một cách nhanh và chính xác nhất.

1.3 Mục tiêu nghiên cứu

Nghiên cứu được cơ sở lý thuyết về huyết đạo.

Nghiên cứu được thuật toán CNN.

Áp dụng thuật toán CNN với cơ sở lý thuyết về huyết đạo để cho AI học.

1.4 Phương pháp nghiên cứu

Nghiên cứu lý thuyết qua kaggle về object detection.

Tìm hiểu qua các video object detection bằng CNN.

Xác định kích thước đầu của bệnh nhân, điểm được quy ước là điểm start point là huyết đầu tiên trên khuôn mặt bệnh nhân.

Tính được giá trị thốn của mỗi bệnh nhân, dự đoán các huyết đạo còn lại của bệnh nhân

PHẦN II: CƠ SỞ LÝ THUYẾT

2.1 Cơ sở lý thuyết về huyết đạo

2.1.1 Huyết đạo là gì?

Huyết đạo có nhiều tên gọi khác nhau như: du huyết, khí huyết, cốt huyết,... Nhưng huyết đạo vẫn là tên gọi được dùng nhiều nhất. Có nhiều định nghĩa về huyết đạo nhưng theo ghi chép từ cuốn Linh khu thiên thập nhị nguyên thì đó là nơi để lưu thông thần khí ra và vào cơ thể.

Huyết đạo được phân bố, rải đều khắp cơ thể. Trên cơ thể của chúng ta có 365 huyết đạo, trong đó có 257 tiểu huyết, 108 đại huyết. Đặc biệt, trong số 108 đại huyết trên cơ thể có đến 36 điểm là tử huyết – những huyết này có thể dẫn có thể tới tử vong hoặc nguy hiểm nếu bạn tác động không đúng cách.

Huyết đạo có mối liên hệ chặt chẽ với hoạt động của các chi, các cơ quan trong cơ thể, việc tìm hiểu về huyết đạo giúp bạn hiểu hơn về cơ thể mình để sống khỏe mạnh nhất.

2.1.2 Phương pháp xác định huyết đạo

Trên lâm sàng, việc châm đạt hiệu quả hay không, phụ thuộc khá nhiều vào việc xác định đúng vị trí huyết, vì có nhiều khi, chẩn đoán đúng bệnh nhưng châm không đúng huyết thì hiệu quả cũng không thể đạt được. Vì vậy, cần phải nắm vững phương pháp lấy huyết cho chính xác.

Bằng những kinh nghiệm tỉ mỉ và lâu dài, các nhà châm cứu xưa và nay đã tìm ra 1 số phương pháp giúp lấy huyết như sau:

h.1- Phương Pháp Đo Lấy Huyết

Phương pháp này có 2 cách:

Chia Đoạn Từng Phần Cơ Thể: phương pháp này gọi là 'Cốt Độ Pháp' được ghi tỉ mỉ trong thiên 'Cốt Độ' (Linh Khu 14). theo đó:

- + Cơ thể con người được chia 38 phần ngang và dọc.
- + Chiều cao mọi người từ đầu đến chân là 75 thốn.

+ Thốn được phân bằng 1/75 chiều cao của mỗi người.

Cụ thể được phân chia như sau:

Mốc Vị Trí Của Cơ Thể	Đơn Vị Đo Theo Linh Khu
+ Từ chân tóc trán đến chân tóc gáy	12 thốn
+ Giữa 2 góc tóc trán (2 huyết Đầu Duy)	09 thốn
+ Giữa chân tóc trán đến chân tóc gáy	12 thốn
+ Giữa 2 lông mày (Ấn Đường) đến chân tóc trán	03 thốn
+ Chân tóc gáy đến huyết Đại Chùy	03 thốn
+ Giữa 2 huyết Hoàn Cốt (giữa 2 mỗm trâm chũm)	09 thốn
+ Từ bờ trên xương ức (huyết Thiên Đột) đến góc 2 cung sườn (huyết Trung Đỉnh)	09 thốn
+ Từ huyết Trung Đỉnh đến giữa rốn (huyết Thần Khuyết)	08 thốn
+ Giữa rốn đến bờ trên xương mu (huyết Khúc Cốt)	6,5 thốn
+ Khoảng cách giữa 2 đầu vú	08 thốn
+ Khoảng cách của 2 góc trên -trong xương bả vai	06 thốn
+ Đỉnh của nách tới bờ xương cụt (huyết Chương Môn)	12 thốn
+ Từ huyết Chương Môn đến huyết Hoàn Khiêu (ngang mấu chuyển lớn)	09 thốn
+ Từ huyết Hoàn Khiêu đến đỉnh ngang bờ trên xương bánh chè (huyết Hạc Đỉnh)	19 thốn
+ Từ huyết Đại Chùy (dưới mỗm gai đốt sống cổ 7) đến bờ dưới xương cùng	30 thốn
+ Từ ngang đầu nếp nách trước đến ngang khớp khuỷu tay	09 thốn
+ Từ ngang đầu nếp nách sau đến ngang khớp khuỷu	09 thốn

+ Lăn chỉ cổ tay đến lằn chi khớp khuỷu trước	12,5 thốn
+ Ngang khớp khuỷu sau đến ngang khớp cổ tay	12 thốn
+ Lăn chỉ cổ tay đến khớp bàn tay	04 thốn
+ Từ huyết Khúc Cốt đến ngang bờ trên lồi cầu trong xương đùi	18 thốn
Từ huyết Âm Lăng Tuyền (Ngang bờ dưới lồi củ trong xương chày) đến đỉnh mắt cá chân trong	13 thốn
+ Từ nếp nhượng chân (huyết Ủy Trung) đến đỉnh mắt cá chân ngoài	13 thốn
+ Từ bờ sau gót chân đến đầu ngón chân thứ 2	12 thốn
+ Từ ngang lồi cầu cao nhất của mắt cá chân trong đến mặt đất	03 thốn

Bảng 1: Xác định điểm huyết bằng thốn

Cách phân chia theo tiết đoạn này tương đối dễ lấy và định huyết 1 cách nhanh chóng, ngoài ra, còn tránh được sai lệch do sự cấu tạo của thân thể người bệnh và thầy thuốc. Thí dụ người bệnh có tay chân quá dài, lấy theo thốn tay dễ bị sai lạc.

Cách Dùng Các Phần Ngón Tay Người Bệnh Để Đo

Cách đo này, người xưa gọi là 'Đồng Thân Thốn'.

+ Đồng Thân Thốn là gì? Bảo người bệnh co đầu ngón tay giữa vào cho chạm đầu ngón tay cái thành hình vòng tròn, chỗ tận cùng bề ngang của 2 lằn chỉ lỏng giữa ngón tay trở được gọi là 1 đồng thân thốn, và thường được gọi tắt là 1 thốn.

+ Chiều ngang 4 ngón tay: bảo người bệnh duỗi bàn tay, ép sát 4 ngón tay (trừ ngón cái ra), bề ngang tính từ ngón út đến ngón trở được tính là 3 thốn. Cách đo này thường dùng để lấy những huyết có bề dài khoảng cách 3 thốn, thí dụ: huyết Tam Âm Giao (cách đỉnh mắt cá chân trong 3 thốn - Đẻ 4 ngang ngón tay lên đỉnh mắt cá chân trong, cuối của 4 ngang ngón tay này là huyết), Huyền Chung

+ Chiều ngang của 3 ngón tay (trừ ngón cái và ngón út) được coi là 2 thốn. Cách này dùng để lấy các huyết có khoảng cách 2 thốn như huyết Thủ Tam Lý, Phục Lưu, Nội Quan

+ Chiều ngang của 2 ngón tay giữa và trỏ tương đương 1, 5 thốn.

+ Chiều ngang qua gốc ngón tay cái (chỗ cao nhất khi gập ngón tay lại), tương đương 1 thốn, cũng gọi là 1 khoát.

Theo tạp chí ‘Thông Tin YHCTDT’ số 45/1984 về các loại thốn để đo đối với người Việt Nam cao trung bình 1m58 thì:

. Chiều dài trung bình thốn của đốt ngón tay giữa (thốn): 2, 11cm.

. Chiều dài trung bình thốn ngang 4 ngón tay: 2, 2cm.

. Chiều dài trung bình thốn ngang ngón cái (khoát): 2, 0cm.

Tỉ số chênh lệch giữa các loại thốn trên là vào khoảng 0, 1cm (0, 5%), và đối với thống kê học, thì tỉ số chênh lệch này không đáng kể và có thể chấp nhận được. Tuy nhiên, trong khoảng cách ngắn thì còn ít sai số và chênh lệch nhưng càng nhiều thì tỉ số càng lớn và sai sót càng nhiều. Vì vậy, nên dùng cách đo này khi cần đo khoảng cách ngắn mà thôi.

Phương Pháp Dùng Các Mốc Giải Phẫu Hoặc Hình Thể Tự Nhiên

Có rất nhiều vị trí gắn liền với 1 mốc điểm của giải phẫu cơ thể, vì vậy, có thể dùng ngay những vị trí xác định đó làm chuẩn để định huyết cho chính xác.

Dựa Vào Các Cấu Tạo Cố Định: Tai, mắt, mũi, miệng...

Thí dụ: Huyết Tình ở sát khoé mắt trong.

Huyết Thừa Tương ở đáy chỗ lõm giữa môi dưới.

Dựa Vào Các Nếp Nhăn Của da Làm Mốc

Thí dụ: Huyết Đại Lãng ở giữa nếp gấp cổ tay trong.

Huyết Ủy Trung ở giữa nếp gấp nhượng chân.

Dựa Vào Đặc Điểm Xương Làm Mốc

Thí dụ: Huyết Dương Khê ở đầu mỏm trâm quay.

Huyệt Đại Chùy ở dưới đầu móm gai đốt sống cổ thứ 7 ...

Dựa Vào Gân, Cơ Làm Chuẩn

Thí dụ: Huyệt Thừa Sơn ở đỉnh góc tạo nên bởi 2 thân cơ tiếp giáp nhau và cùng bám vào gân gót chân.

Huyệt Tý Nhu ở ngang chỗ bám của cơ Delta vào xương cánh tay.

Lấy Huyệt Dựa Vào Tư Thế Hoạt Động Của 1 Bộ Phận

Phương pháp này đòi hỏi người bệnh phải thực hiện 1 số động tác nhất định như co tay lại, cúi đầu xuống...

Thí dụ: Co tay vào ngực để lấy huyệt Khúc Trì.

Đứng thẳng người, tay áp vào đùi để lấy huyệt Phong Thị.

Cúi đầu xuống để lấy huyệt á Môn.

Lấy Huyệt Dựa Vào Cảm Giác Của Người Bệnh

Theo Cảm Giác Của Người Bệnh: vì huyệt là nơi dễ nhạy cảm và có phản ứng khi có bệnh, do đó, khi sờ ấn lên vùng huyệt, chỗ nào có biểu hiện đau nhiều nhất, đó thường là vị trí huyệt rõ nhất.

Theo Cảm Giác Của Thầy Thuốc: Khi cơ thể có bệnh, huyệt là nơi thông tin mạnh nhất, vì vậy, nó có thể thay đổi 1 số hình thái mà dùng mắt thường hoặc cảm giác ở tay có thể nhận biết được: chỗ huyệt đó mềm hơn, cứng hơn, nóng đỏ...

Phương pháp dựa trên cảm giác tương đối khá dễ nhưng còn nhiều hạn chế:

Không thể áp dụng cho trẻ nhỏ vì chúng chưa đủ trình độ mô tả chính xác các cảm giác khi được hỏi.

Thầy thuốc không có kinh nghiệm khó có thể nhận thấy những thay đổi đặc biệt nơi các huyệt trong cơ thể bệnh.

Dựa Vào Các Máy Móc Kỹ Thuật Hiện Đại

Dựa vào đặc tính thay đổi của từng huyết, nhất là sự thay đổi điện trở của huyết, các nhà nghiên cứu đã chế ra các máy đo điện trở để tìm ra vị trí của huyết 1 cách tương đối nhanh chóng và chính xác.

Phương pháp này bảo đảm nhiều mặt thiếu sót của thầy thuốc nhưng không phải là mọi thầy thuốc đều có điều kiện sắm máy cũng như không phải máy nào cũng có độ chính xác cao.

Dựa vào đặc điểm thay đổi cảm giác của huyết, nhất là các dấu hiệu đau khi ấn vào huyết, người ta đã chế ra các loại que dò giúp dễ ấn tìm ra vị trí huyết, nhất là khi tìm kiếm huyết có vị trí đờng kính nhỏ như huyết ở vùng mặt (diện châm) hoặc ở loa tai (nhĩ châm), ...

Trong thực tế lâm sàng, muốn chọn huyết nhanh và chính xác, phải tùy theo vị trí huyết mà chọn dùng 1 trong số những phương pháp nêu trên hoặc phối hợp cùng lúc 2 - 3 cách để hỗ trợ cho nhau.

Thí dụ: tìm huyết Nội Quan:

Có thể dùng 3 ngang ngón tay (2 thốn) đo từ giữa lằn chỉ cổ tay trong lên.

Gấp bàn tay vào cẳng tay cho gân cơ gan tay lớn và cơ gan tay bé nổi rõ dưới mặt da để dễ lấy huyết.

Như vậy, vừa phối hợp được cách lấy huyết theo YHCT vừa theo cách lấy huyết theo giải phẫu học của YHHĐ.

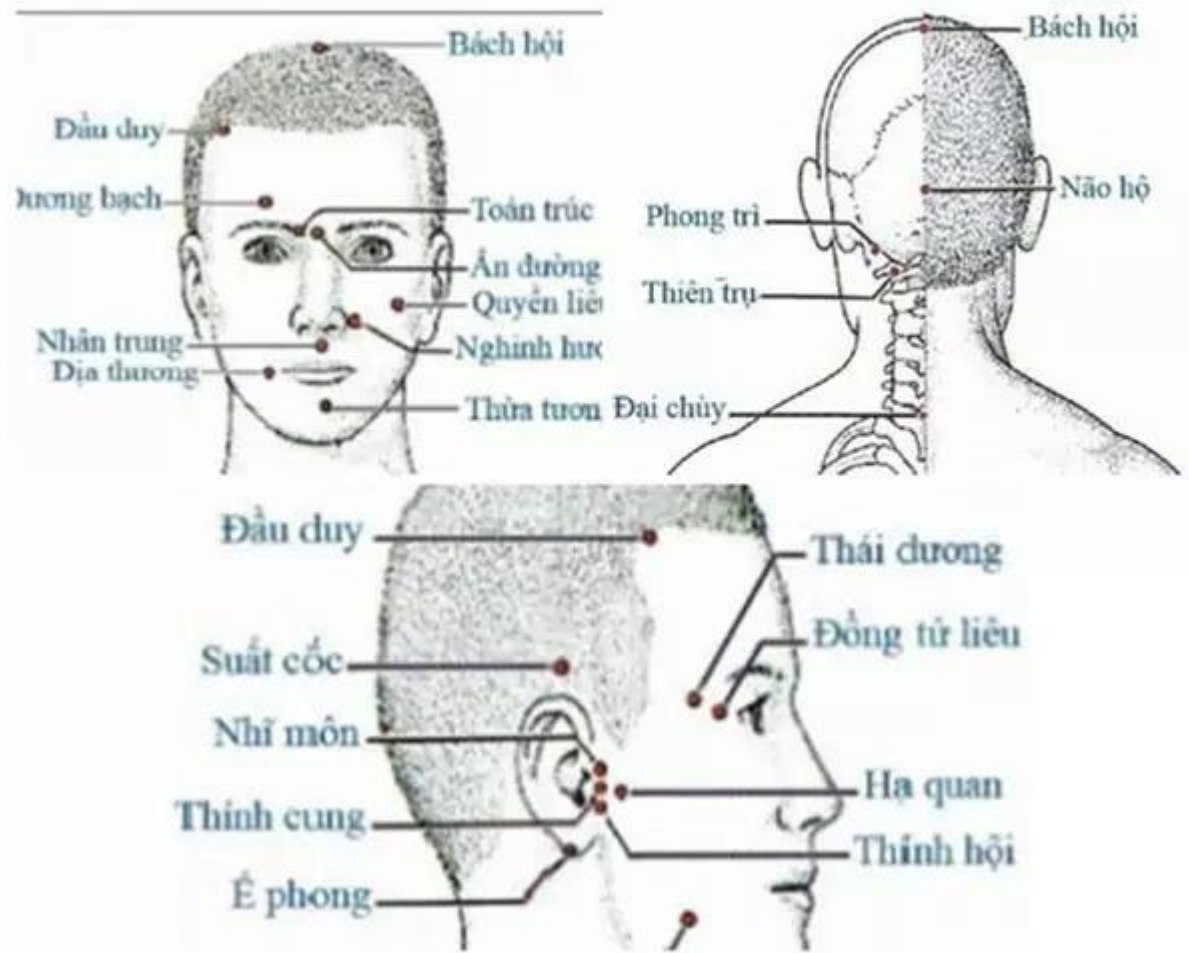
2.1.3 Huyết đạo ở đầu

Vị trí các huyết quan trọng vùng đầu, cổ

Huyết bách hội:

- Vị trí: tại giao điểm của tuyến chính giữa đỉnh đầu và đường nối liền phần đầu dọc theo trên của 2 bên tai

- Khi bị tác động: Khi bị đánh trúng bách hội huyết sẽ gây ra choáng váng, ngã xuống đất và bất tỉnh nhân sự ngay lập tức



Huyệt thần đình.

- Vị trí: Từ mép tóc trước trán lên 5cm
- Khi bị tác động: sẽ choáng váng, ảnh hưởng đến vùng não bộ

Huyệt thái dương.

- Vị trí huyệt thái dương: Tại vùng lõm phía đuôi chân mày
- Khi bị tác động: gây ra choáng váng, mắt tối sầm lại, ù tai.

Huyệt nhĩ môn

- Vị trí: Tại điểm khuyết ở trước vành tay, khi há miệng sẽ hiện ra chỗ lõm đó
- Khi bị tác động: Û tai, chóng váng đầu và ngã xuống đất

Huyệt tinh minh

- Vị trí: Huyệt tinh minh nằm tại góc khoe mắt phía trong, đầu chân mày
- Khi bị tác động: Có thể hôn mê sâu, nhẹ hơn thì hoa mắt và ngã xuống đất.

Huyệt nhân trung

- Vị trí: Nằm dưới chóp mũi
- Khi bị tác động: sẽ gây ra choáng váng, hoa mắt.

Huyệt á môn

- Vị trí: Huyệt á môn nằm ở phía sau ót, chỗ lõm giữa gai đốt sống cổ thứ 2 và đốt sống cổ thứ 1
- Khi bị tác động: Đập vào diên tủy là một phần não sau nối với tủy sống sẽ không thể nói được, gây choáng váng, đau đầu, ngã xuống đất và bất tỉnh.

Huyệt phong trì

- Vị trí: Huyệt phong trì nằm phía sau dái tai, phần lõm dưới xương chẩm
- Khi bị tác động: Đập vào trung diên tủy, hôn mê bất tỉnh ngay.

Huyệt nhân nghênh

- Vị trí: Vùng yết hầu, ngang với 2 bên khoảng 5cm

- Khi bị tác động: Khí huyết ứ đọng, choáng đầu.

2.2 Thuật toán CNN – Convolutional Neural Network

2.2.1 CNN – Convolutional Neural Network là gì?

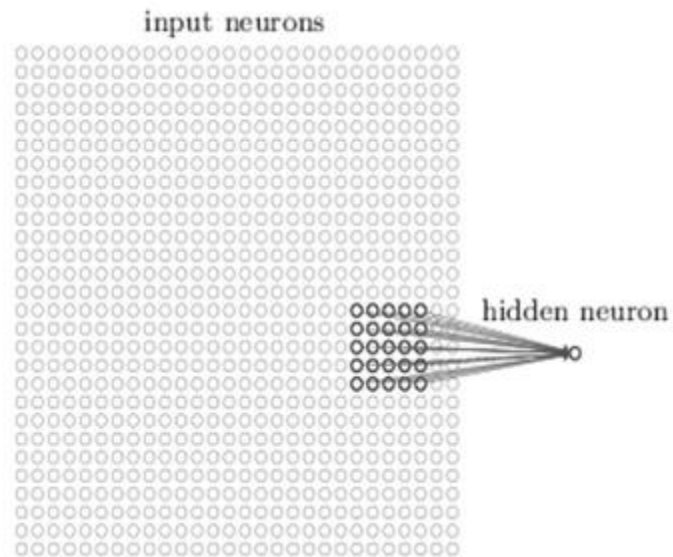
Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng (detection), chúng ta hãy cùng tìm hiểu về thuật toán này.

2.2.2 Cấu trúc của mạng CNN – Convolutional Neural Network

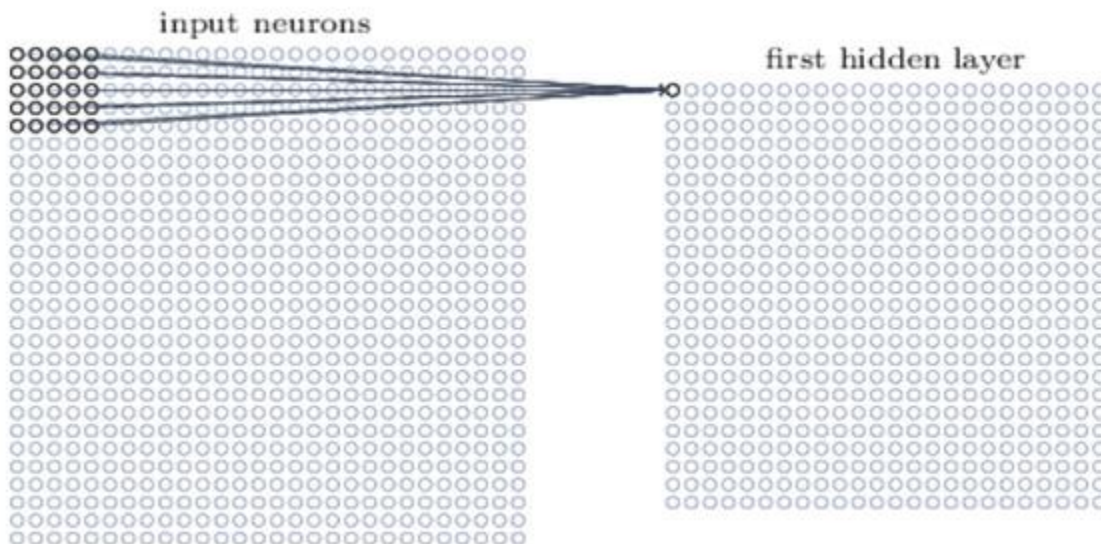
2.2.2.1 Trường tiếp nhận cục bộ

Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28×28 thì tương ứng đầu vào là một ma trận có 28×28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng ANN truyền thống thì chúng ta sẽ kết nối các neuron đầu vào vào tầng ảnh. Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các neuron đầu vào như một filter có kích thước 5×5 tương ứng $(28 - 5 + 1) = 24$ điểm ảnh đầu vào. Mỗi một kết nối sẽ học một trọng số và mỗi neuron ẩn sẽ học một bias. Mỗi một vùng 5×5 đây gọi là một trường tiếp nhận cục bộ.

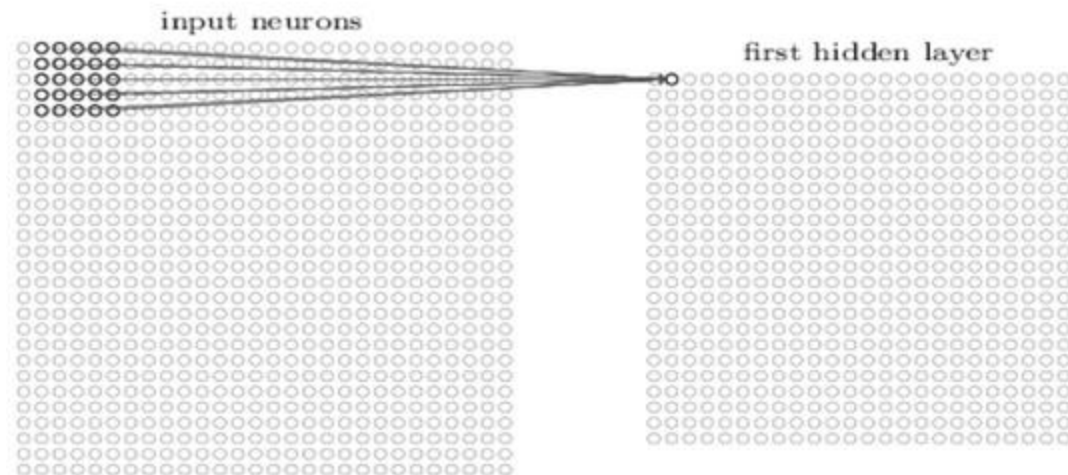


Một cách tổng quan, ta có thể tóm tắt các bước tạo ra 1 hidden layer bằng các cách sau:

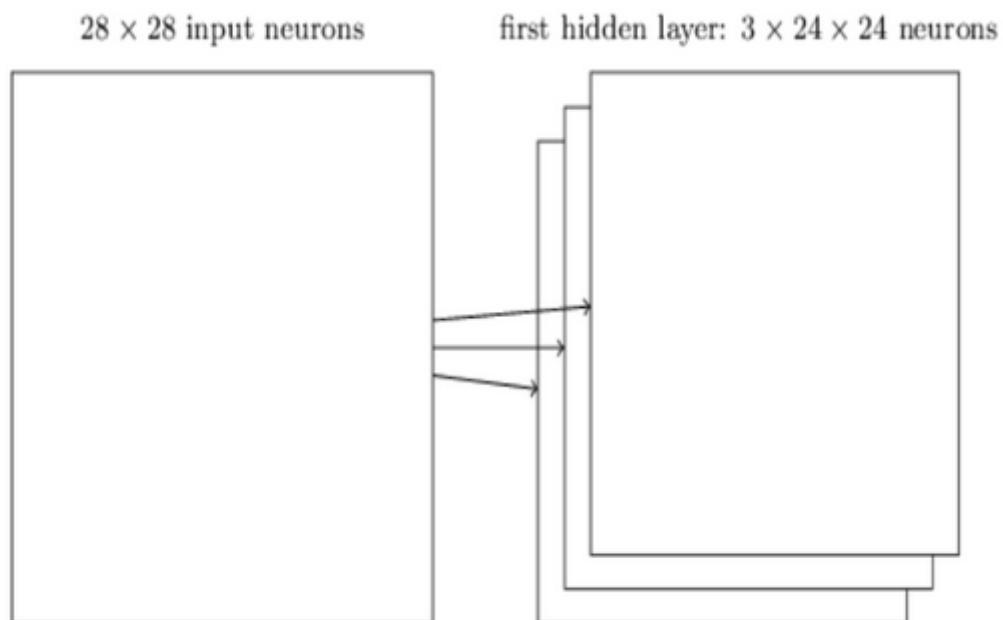
1. Tạo ra neuron ẩn đầu tiên trong lớp ẩn 1



2. Dịch filter qua bên phải một cột sẽ tạo được neuron ẩn thứ 2.



Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp đầu vào là feature map, trọng số xác định các đặc trưng là shared weight và độ lệch xác định một feature map là shared bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 feature map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một feature map.



Như vậy, local receptive field thích hợp cho việc phân tách dữ liệu ảnh, giúp chọn ra những vùng ảnh có giá trị nhất cho việc đánh giá phân lớp.

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN sử dụng 3 ý tưởng cơ bản:

- **các trường tiếp nhận cục bộ** (local receptive field)
- **trọng số chia sẻ** (shared weights)
- **tổng hợp** (pooling).

2.2.2.2 Trọng số chia sẻ

Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các nơ-ron trong lớp ẩn đầu sẽ phát hiện chính xác feature tương tự chỉ ở các vị trí khác nhau trong hình ảnh đầu vào. Chúng ta gọi việc map từ input layer sang hidden layer là một feature map. Vậy mối quan hệ giữa số lượng Feature map với số lượng tham số

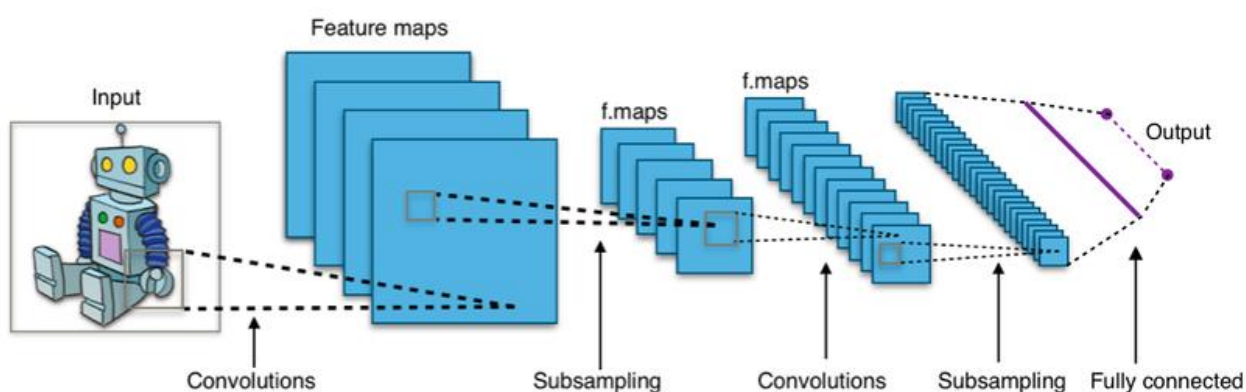
là gì?

Chúng ta thấy mỗi feature map cần $25 = 5 \times 5$ shared weight và 1 shared bias. Như vậy mỗi feature map cần $5 \times 5 + 1 = 26$ tham số. Như vậy nếu có 10 feature map thì có $10 \times 26 = 260$ tham số. Chúng ta xét lại nếu layer đầu tiên có kết nối đầy đủ nghĩa là chúng ta có $28 \times 28 = 784$ neuron đầu vào như vậy ta chỉ có 30 neuron ẩn. Như vậy ta cần $28 \times 28 \times 30$ shared weight và 30 shared bias. Tổng số tham số là $28 \times 28 \times 30 + 30$ tham số lớn hơn nhiều so với CNN. Ví dụ vừa rồi chỉ mô tả để thấy được sự ước lượng số lượng tham số chứ chúng ta không so sánh được trực tiếp vì 2 mô hình khác nhau. Nhưng điều chắc chắn là nếu mô hình có số lượng tham số ít hơn thì nó sẽ chạy nhanh hơn.

Tóm lại, một convolutional layer bao gồm các feature map khác nhau. Mỗi một feature map giúp detect một vài feature trong bức ảnh. Lợi ích lớn nhất của trọng số chia sẻ là giảm tối đa số lượng tham số trong mạng CNN.

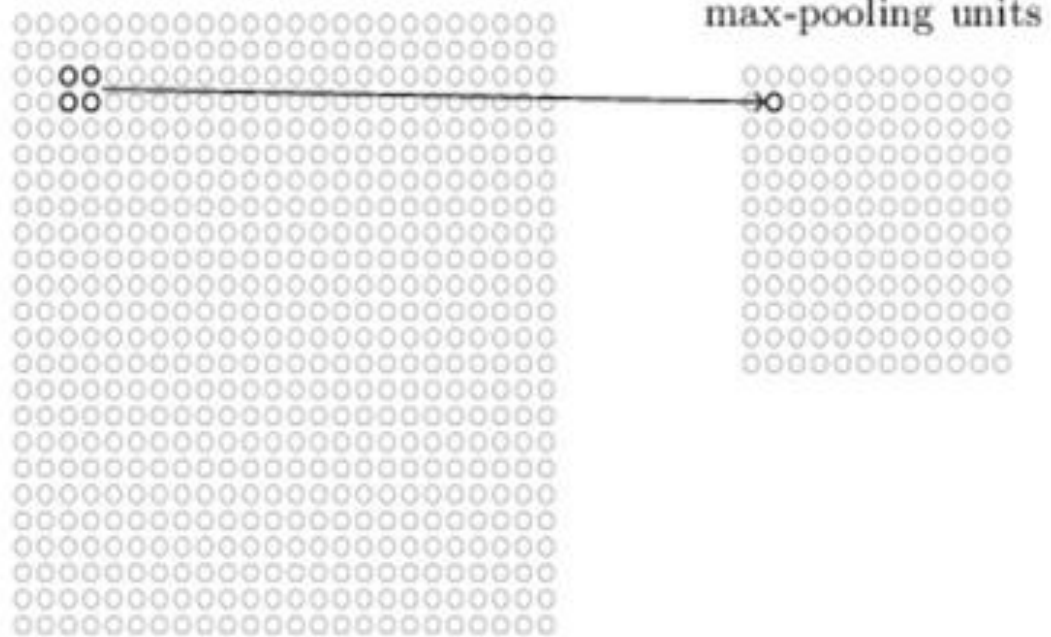
2.2.2.3 Lớp tổng hợp

Lớp pooling thường được sử dụng ngay sau lớp convolutional để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron.



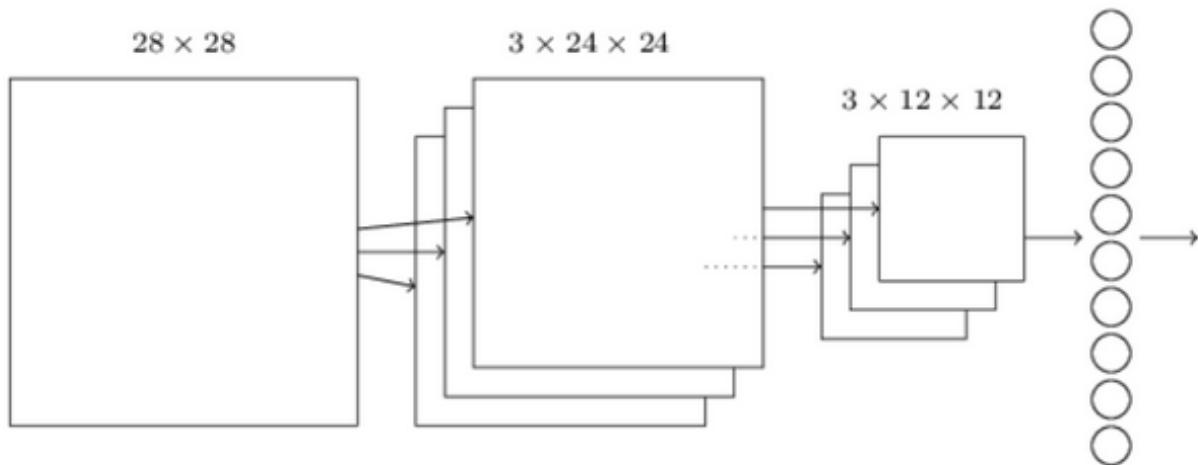
Thủ tục pooling phổ biến là max - pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2×2 .

hidden neurons (output from feature map)



Như vậy qua lớp Max Pooling thì số lượng neuron giảm đi phân nửa. Trong một mạng CNN có nhiều Feature Map nên mỗi Feature Map chúng ta sẽ cho mỗi Max Pooling khác nhau. Chúng ta có thể thấy rằng Max Pooling là cách hỏi xem trong các đặc trưng này thì đặc trưng nào là đặc trưng nhất. Ngoài Max Pooling còn có L2 Pooling.

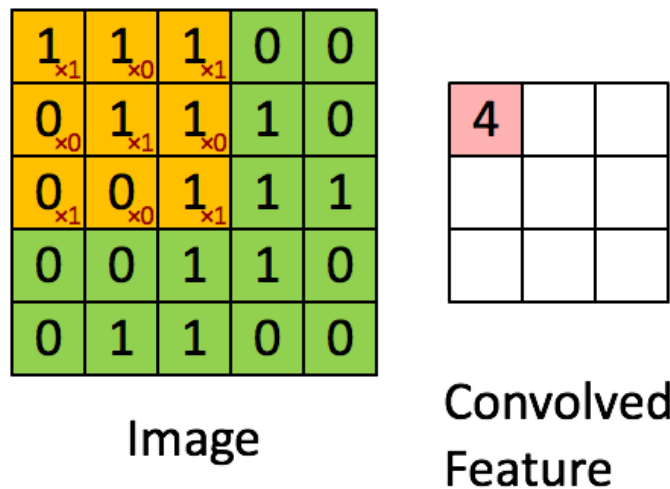
Cuối cùng ta đặt tất cả các lớp lại với nhau thành một CNN với đầu ra gồm các neuron với số lượng tùy bài toán.



2 lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (fully connected layer) Lớp này nối mọi neuron từ lớp max pooled tới mọi neuron của tầng ra.

2.2.3 Convolutional là gì?

Là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:



Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3×3 .

Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3×3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhân ma trận Filter với ma trận ảnh 5×5 bên trái.



PHẦN III: ỨNG DỤNG - KẾT QUẢ THỰC NGHIỆM

3.1 Dataset

Ảnh về bộ phận đầu của người có 454 ảnh là ảnh với những góc mặt khác nhau của mỗi người.

3.2 Training

Các thư viện cần thiết của Python

```
1 from keras.models import Sequential
2 from keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten
3 from tensorflow.keras.optimizers import Adam
4 from sklearn.preprocessing import StandardScaler
5 from keras.utils import np_utils
6 from sklearn.utils import shuffle
7 from tensorflow.keras.utils import load_img, img_to_array
8 import cv2
9 import matplotlib.pyplot as plt
10 import numpy as np
11 import pickle
12 import tensorflow as tf
13 import pandas as pd
```

Kết nối với Google Drive để lấy datasets

```
[ ] 1 X_DATA_PATH = '/content/drive/MyDrive/Cuoiky/data.pickle'
    2 Y_DATA_PATH = '/content/drive/MyDrive/Cuoiky/data.csv'
```

Xuất dữ liệu tọa độ các điểm huyết

```
[ ] 1 import pandas as pd
    2 y_data = pd.read_csv(Y_DATA_PATH)
    3 y_data.head(None)
```

	x1	y1	x2	y2	x3	y3	x4	y4	x5	y5	...	x7	y7	x8	y8	x9	y9	x10	y10	x11	y11
0	180	105	146	85	113	119	186	166	186	201	...	100	141	83	114	110	70	118	43	76	83
1	180	104	145	85	114	118	187	165	185	201	...	99	142	82	113	109	70	119	42	76	82
2	179	106	146	84	112	120	186	165	186	202	...	99	141	82	115	110	70	119	42	75	82
3	179	106	147	86	113	120	185	165	187	201	...	100	142	83	113	110	71	117	42	75	82
4	179	106	145	85	112	120	186	166	185	200	...	101	140	83	115	111	69	119	42	76	83
...
449	179	88	150	61	109	96	175	145	163	172	...	69	108	42	88	106	36	73	22	37	42
450	178	86	150	61	111	97	173	144	165	173	...	68	107	42	88	106	36	73	22	35	43
451	180	86	150	63	110	96	174	145	165	171	...	69	107	41	88	108	34	73	22	36	42
452	180	86	149	62	109	95	173	144	165	173	...	69	107	43	89	108	34	73	24	37	44
453	178	88	150	62	109	97	173	146	164	173	...	69	108	41	88	107	36	72	24	35	42

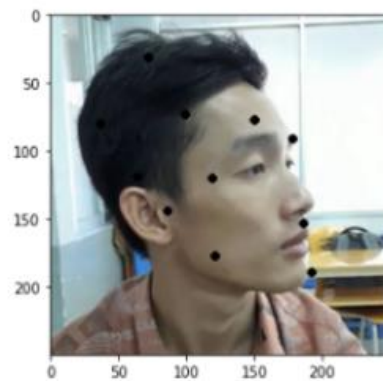
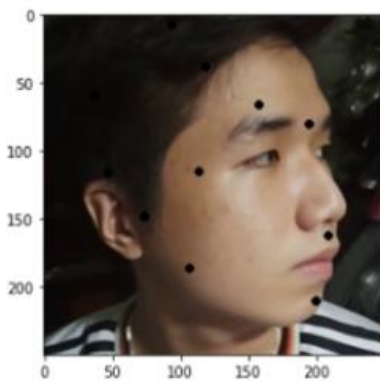
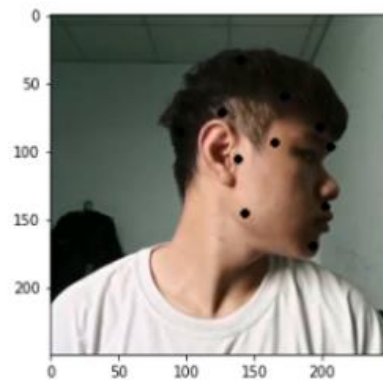
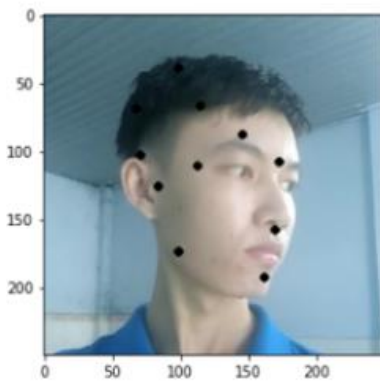
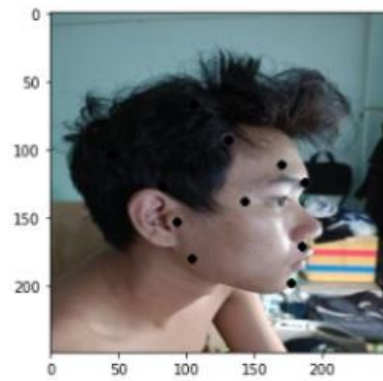
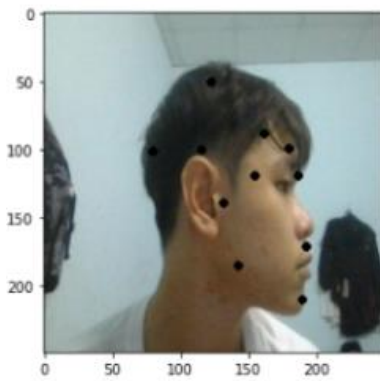
454 rows x 22 columns

Ảnh data input để training

```

1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import random
4 import cv2
5 #n = x_data.shape[0]
6 plt.figure(figsize = (15, 15))
7 plt.subplots_adjust(hspace = .2)
8 for i in range(6):
9     plt.subplot(3, 2, i + 1)
10    k = random.randint(0, 454)
11    img = x_data[k]
12    points = list(y_data.iloc[k])
13    points = [tuple(points[i : i + 2]) for i in range(0, len(points), 2)]
14    for x, y in points:
15        cv2.circle(img, (int(x), int(y)), 2, (0, 0, 0), 2)
16    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
17    _ = plt.suptitle('Sample images', size = 20)

```




```
[ ] 1 from sklearn.model_selection import train_test_split
2 import numpy as np
3 #x_data = x_data.reshape(-1, (250,250,3), 1)
4 input_shape = x_data.shape[1:4]
5 y_data = np.array(y_data, dtype = 'float32')
6 num_class = y_data.shape[1]
7 x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size = 0.1)
8 print('Input shape: ', input_shape)
9 print('Number of output: ', num_class)
10 print('x train shape: ', x_train.shape)
11 print('y train shape: ', y_train.shape)
12 print('x test shape: ', x_test.shape)
13 print('y test shape: ', y_test.shape)
```

```
Input shape: (250, 250, 3)
Number of output: 22
x train shape: (408, 250, 250, 3)
y train shape: (408, 22)
x test shape: (46, 250, 250, 3)
y test shape: (46, 22)
```

Tạo model

```
1 from keras.models import Sequential, load_model
2 from keras.layers import Dense, Conv2D, Dropout, BatchNormalization, MaxPooling2D, Flatten, LeakyReLU, Convolution2D, MaxPool2D
3 model = Sequential()
4
5 model.add(Convolution2D(32, (3,3), padding='same', use_bias=False, input_shape = input_shape))
6 model.add(LeakyReLU(alpha = 0.1))
7 model.add(BatchNormalization())
8
9 model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
10 model.add(LeakyReLU(alpha = 0.1))
11 model.add(BatchNormalization())
12 model.add(MaxPool2D(pool_size=(2, 2)))
13
14 model.add(Convolution2D(128, (3,3), padding='same', use_bias=False))
15 model.add(LeakyReLU(alpha = 0.1))
16 model.add(BatchNormalization())
17
18 model.add(Convolution2D(128, (3,3), padding='same', use_bias=False))
19 model.add(LeakyReLU(alpha = 0.1))
20 model.add(BatchNormalization())
21 model.add(MaxPool2D(pool_size=(2, 2)))
22
23 # Dense
24 model.add(Flatten())
25 model.add(Dense(512, activation = 'relu'))
26 # model.add(Dense(num_class, activation='softmax'))
27 model.add(Dense(num_class))
28 model.summary()
```

Training

```
[ ] 1 from tensorflow.keras.optimizers import Adam, SGD, RMSprop
2 model.compile(
3     optimizer = 'Adam',
4     loss = "mean_squared_error",
5     metrics = ['mae']
6 )
```

```
[ ] 1 from tensorflow.keras.callbacks import EarlyStopping
2 callback = EarlyStopping(monitor = 'val_mae', patience = 25)
3 hist = model.fit(x_train, y_train, epochs = 1000, batch_size = 4, validation_split = 0.1).history
```

```

Epoch 991/1000
92/92 [=====] - 9s 97ms/step - loss: 3.7946 - mae: 1.4841 - val_loss: 4.9231 - val_mae: 1.7734
Epoch 992/1000
92/92 [=====] - 9s 98ms/step - loss: 3.8644 - mae: 1.5036 - val_loss: 4.0998 - val_mae: 1.5893
Epoch 993/1000
92/92 [=====] - 9s 98ms/step - loss: 3.9449 - mae: 1.5173 - val_loss: 3.9784 - val_mae: 1.5354
Epoch 994/1000
92/92 [=====] - 9s 97ms/step - loss: 4.1657 - mae: 1.5524 - val_loss: 3.6099 - val_mae: 1.4703
Epoch 995/1000
92/92 [=====] - 9s 97ms/step - loss: 3.9403 - mae: 1.5159 - val_loss: 3.8299 - val_mae: 1.5094
Epoch 996/1000
92/92 [=====] - 9s 97ms/step - loss: 3.9431 - mae: 1.5093 - val_loss: 4.0475 - val_mae: 1.5748
Epoch 997/1000
92/92 [=====] - 9s 97ms/step - loss: 4.4020 - mae: 1.6174 - val_loss: 4.2415 - val_mae: 1.5854
Epoch 998/1000
92/92 [=====] - 9s 97ms/step - loss: 4.0316 - mae: 1.5282 - val_loss: 4.4514 - val_mae: 1.6325
Epoch 999/1000
92/92 [=====] - 9s 97ms/step - loss: 3.9668 - mae: 1.5217 - val_loss: 5.1379 - val_mae: 1.7800
Epoch 1000/1000
92/92 [=====] - 9s 97ms/step - loss: 3.9115 - mae: 1.5151 - val_loss: 4.1778 - val_mae: 1.6142

```

Độ chính xác của model – Bình phương sai số

```

1 final_loss, final_accuracy = model.evaluate(x_test, y_test)
2 print('Final loss: {:.4f}'.format(final_loss))
3 print('Final accuracy: {:.4f}'.format(final_accuracy))

```

```

2/2 [=====] - 3s 935ms/step - loss: 4.2542 - mae: 1.5542
Final loss: 4.2542
Final accuracy: 1.5542

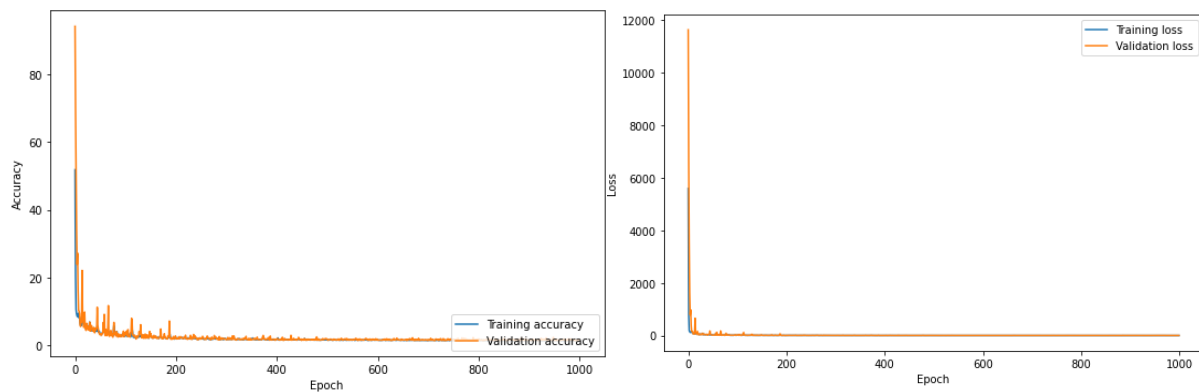
```

Đồ thị accuracy

```

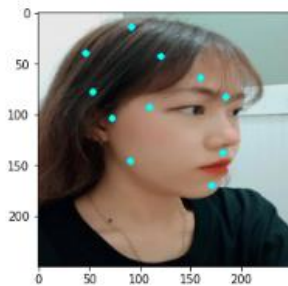
[ ] 1 import pandas as pd
2 import matplotlib.pyplot as plt
3 model_history = pd.DataFrame(hist)
4 # add epoch column
5 model_history['epoch'] = np.arange(1, len(model_history) + 1)
6 fig, (ax1, ax2) = plt.subplots(2, 1, figsize = (8, 10))
7 epochs = model_history.shape[0]
8 ax1.plot(np.arange(0, epochs), model_history['mae'], label = 'Training accuracy')
9 ax1.plot(np.arange(0, epochs), model_history['val_mae'], label = 'Validation accuracy')
10 ax1.legend(loc = 'lower right')
11 ax1.set_ylabel('Accuracy')
12 ax1.set_xlabel('Epoch')
13 ax2.plot(np.arange(0, epochs), model_history['loss'], label = 'Training loss')
14 ax2.plot(np.arange(0, epochs), model_history['val_loss'], label = 'Validation loss')
15 ax2.legend(loc = 'upper right')
16 ax2.set_ylabel('Loss')
17 ax2.set_xlabel('Epoch')
18 plt.tight_layout()
19 plt.show()

```



Test độ chính xác model

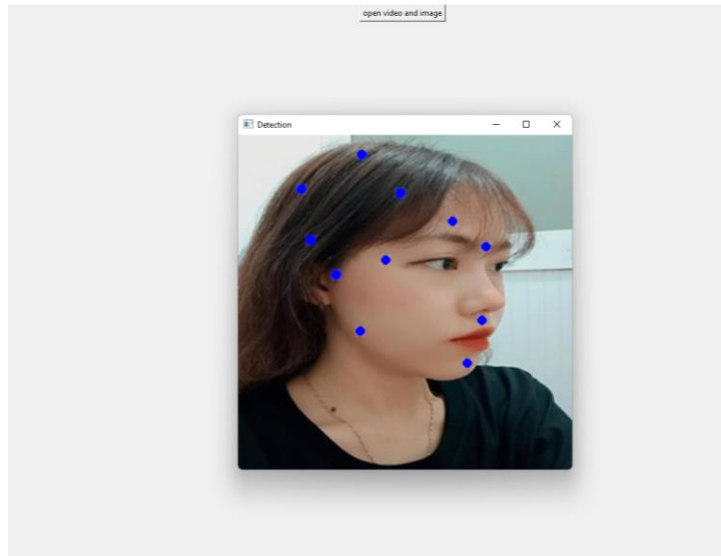
```
1 import matplotlib.pyplot as plt
2 import cv2
3 path = '/content/test.jpg'
4
5 img_input_test = cv2.imread(path)
6
7 # img_input_test = tf.keras.utils.load_img("/content/test.jpg", target_size = (250,250))
8 plt.imshow(img_input_test,cmap='gray')
9 plt.show()
10 img = tf.keras.utils.img_to_array(img_input_test)
11 img = img.astype('float32')
12 img /= 255
13
14 pred = model.predict(img.reshape(1,250,250,3))
15 #print(pred.shape)
16 points_tt = pred[:]
17 points_tt = points_tt.reshape(-1)
18 for j in range(11):
19     x = points_tt[2*j]
20     x = x.astype('int')
21     if x<0:
22         x=x*(-1)
23     y = points_tt[2*j+1]
24     y = y.astype('int')
25     if y<0:
26         y=y*(-1)
27     cv2.circle(img, (x,y) ,2, (255, 255, 0), 2)
28 plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
```



PHẦN IV: KẾT LUẬN

4.1 Kết luận

Với kết quả thu được thì mô hình CNN nhận dạng điểm huyết trên đầu người khá chính xác, nhưng vẫn còn sai số đáng kể do số lượng đưa vào cho máy học chưa đủ để có một mô hình nhận dạng tốt nhất.



4.2 Ưu điểm – Nhược điểm

Ưu điểm:

- Nhận diện theo thời gian thực, thuận tiện, nhanh chóng.
- Tỷ lệ chính xác khá cao và ổn định.

Nhược điểm:

- Vẫn còn nhận diện sai lệch một số điểm huyết.
- Dữ liệu ảnh cho máy học còn ít, thiếu sự đa dạng.

4.3 Hướng phát triển

- Nhận diện và xác định được huyết đạo trên toàn bộ cơ thể con người.
- Dự đoán được quỹ đạo của các điểm huyết khó xác định.
- Nâng cấp model, input và thuật toán để có thể cho ra kết quả chính xác hơn.

TÀI LIỆU THAM KHẢO

[1] TopDev, Thuật toán CNN – Convolutional Neural Network:

<https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>

[2] Wikipedia, Face Detection:

https://en.wikipedia.org/wiki/Face_detection

[3] Minh Trieu, N., & Thinh, N. T. (2021). Determining Trajectories for Hair Wash and Head Massage Robot based on Artificial Neural Network, 11(22), 10558

[4] Jouzi, M. (2009). Assessment of the effect of massage therapy on stroke patients. MedicalScience Journal of Islamic Azad Univesity-Tehran Medical Branch, 19(4), 256-261.

[5] VINMEC, Tìm hiểu về toàn bộ các huyết trên cơ thể:

<https://www.vinmec.com/vi/y-hoc-co-truyen/suc-khoe-thuong-thuc/tim-hieu-ve-toan-bo-cac-huyet-tren-co/>

[6] VINMEC, Cách lấy huyết để xoa bóp bấm huyết:

https://www.vinmec.com/y-hoc-co-truyen/suc-khoe-thuong-thuc/cach-lay-huyet-de-xoa-bop-bam-huyet/?link_type=related_posts