ISSN 2354-0575

NHẬN DIỆN MẶT NGƯỜI BẰNG MẠNG NORON TÍCH CHẬP NỐI TẦNG ĐA NHIỆM

Vũ Thị Thềm[1], Nguyễn Quang Hoan[2], Nguyễn Thị Hồng[3]

[1] *Trung tâm GDNN-GDTX Gia Lộc, Hải Dương;*

[2] *Trường Đại học Sư phạm Kỹ thuật Hưng Yên;*

[3] *Học viện Công nghệ Bưu chính Viễn thông.*

Ngày tòa soạn nhận được bài báo: 16/10/2019

Ngày phản biện đánh giá và sửa chữa: 26/11/2019

Ngày bài báo được duyệt đăng: 06/12/2019

Tóm tắt:

Mục đích của bài báo là nhận diện mặt người theo các bước i) bước một: tách khuôn mặt trong ảnh, trích xuất các đặt trưng khuôn ii) bước hai phân tích, xác minh danh tính người cần nhận dạng. Bước thứ nhất thực hiện dựa trên mô hình mạng Noron tích chập nối tầng đa nhiệm (Multi-Task Cascaded Convolutional Networks: MTCNN) và bước thứ hai dựa trên mạng Noron tích chập (Convolutional Neural Network: CNN). Bài báo áp dụng phương pháp phát hiện khuôn mặt bằng MTCNN được thử nghiệm với độ chính xác cao.

Từ khóa: *Mạng Noron tích chập nối tầng đa nhiệm, mạng Noron tích chập, nhận diện mặt người*

1. Giới thiệu

Hiện nay, có nhiều phương pháp phát hiện mặt người như: Haar Cascade, Histogram of Oriented Gradians (HOG), MTCNN... Phương pháp của Haar Cascade: cho kết quả nhanh nhất nhưng chỉ hoạt động tốt với mặt nhìn chính diện và dễ bị ảnh hưởng môi trường ánh sáng [1, 2, 3] Phương pháp HOG cho kết quả nhanh thứ hai, ít bị ảnh hưởng bởi ánh sáng môi trường ngoài nhưng hoạt động kém khi mặt bị nhiều che lấp. Phương pháp MTCNN xử lý chậm so với hai phương pháp kia nhưng hoạt động tốt ngay cả trong trường hợp mặt bị che nhiều và ít ảnh hưởng bởi ánh sáng môi trường bên ngoài [6]. Trong bài báo này, chúng tôi sử dụng phương pháp MTCNN- gồm nhiều mạng CNN xếp chồng. để phát hiện khuôn mặt

2. Mạng Noron CNN

Kiến trúc cơ bản của CNN thường có 4 lớp: lớp tích chập (Convolutional Layer), lớp kích hoạt phi tuyến , lớp co (Pooling Layer), lớp kết nối đầy đủ (Fully Connected Layer). Tuỳ theo mục đích mà bài toán yêu cầu, mỗi mô hình được người thiết kế thêm hoặc bớt các lớp trên để hệ thống đạt được độ chính xác mong muốn và chi phí tính toán thấp. Dưới đây là chi tiết về 4 lớp cơ bản của một mạng CNN. [7,10]

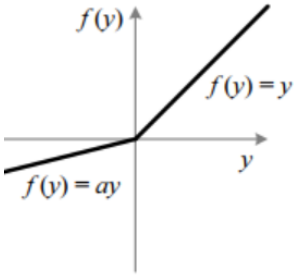
Lớp tích chập: thể hiện sự liên kết cục bộ và được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc có kích thước nhỏ. Đầu vào của các lớp là ảnh biểu diễn bởi ma trận có kích thước: [$H*W*D$] trong đó, W: chiều rộng, H: chiều cao, D: độ sâu, hay số lớp màu của ảnh với bộ lọc được sử dụng là ma trận vkích thước FxF. Bộ lọc này dịch chuyển lần lượt qua từng vùng cho đến khi toàn bộ ảnh được quét, kết quả là ma trận điểm ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào. Kích thước này được quyết định tùy theo kích thước các khoảng trắng được thêm ở viền bức ảnh gốc. Khi đó đầu ra của ảnh với mỗi lớp được tính:

$$\left(\frac{H-F+2P}{S}+1\right)*\left(\frac{H-F+2P}{S}+1\right)*K \qquad (1)$$

trong đó: F: Kích thước bộ lọc; S: bước trượt của bộ lọc; P: kích thước khoảng trắng phía ngoài viền của ảnh gốc; K: Số lượng bộ lọc.

Khi đưa ảnh vào lớp tích chập, đầu ra là một loạt ảnh ứng với các bộ lọc được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên và cập nhật trong quá trình huấn luyện.

Lớp kích hoạt phi tuyến đảm bảo tính phi tuyến của mô hình huấn luyện sau khi thực hiện một loạt các phép tính toán tuyến tính qua các lớp tích chập. Lớp kích hoạt phi tuyến sử dụng các hàm kích hoạt phi tuyến như *ReLU, LeakyReLU, Maxout. ELU, Sigmoid*, *tanh*… [8] để giới hạn biên độ đầu ra. Hai hàm kích hoạt được dùng trong bài này là *PReLU* và Sigmoid. Hàm *PReLU* được cải thiện từ hàm *ReLU* được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả [9].



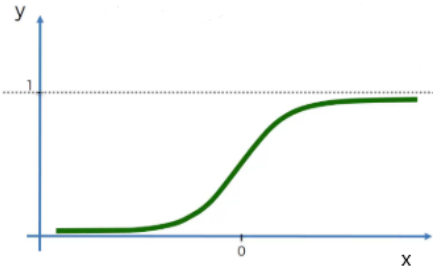
Hình 1: Đồ thị mô tả hàm PreLU

Lớp *PReLU* được áp dụng ngay sau lớp tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự. Hàm Sigmoid đưa ra kết quả trong khoảng (0,1) được sử dụng ở lớp cuối cùng cho sác xuất khuôn mặt nằm trong lớp nào. Hàm kích hoạt *PReLU* có dạng

$$f(y_i) = \begin{cases} y_i & \text{nếu } y_i > 0 \\ a_i\, y_i & \text{nếu } y_i \le 0 \end{cases} \quad (2)$$

Hàm Sigmoid hay hàm Softmax có dạng như sau:

$$y = \frac{1}{1+e^{-x}} \quad (3)$$



Hình 2: Đồ thị hàm Sigmoid hay Softmax

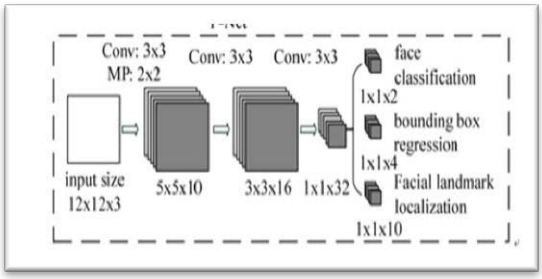
Lớp co: nằm sau lớp kích hoạt phi tuyến nhằm giảm kích thước ảnh đầu ra trong khi vẫn giữ các thông tin quan trọng của ảnh vào. Việc giảm kích thước dữ liệu làmg bớt các tham số, tăng hiệu quả tính toán. Lớp co sử dụng cửa sổ trượt quét các vùng ảnh rồi chọn một giá trị đại diện cho các điểm ảng vùng đó. Có hai phương thức co được sử dụng nhất là lấy giá trị điểm ảnh cực đại (Max Pooling: MP) hoặc lấy giá trị trung bình các điểm ảnh trong vùng cục bộ (Avarage Pooling: AP).

Lớp kết nối đầy đủ: được thiết kế tương tự như mạng noron truyền thống. Tất cả các điểm ảnh được kết nối đầy đủ với các nơ ron trong lớp. So với mạng nơ ron truyền thống [4], các ảnh vào của lớp này có kích thước giảm, đồng thời vẫn đảm bảo các thông tin quan trọng của ảnh. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ ron truyền thống.

3. Cấu trúc mạng MTCNN cho nhận diện

MTCNN hoạt động theo ba bước [5], mỗi bước dùng một mạng noron riêng lần lượt là: mạng đề xuất P-Net (*Proposal Network*) nhằm dự đóan các vùng trong ảnh ví dụ là vùng chứa khuôn mặt (Hình 3); mạng tinh chế R-Net (Refine Network) sử dụng đầu ra của P-Net để loại bỏ các vùng không phải khuôn mặt (Hình 4); và mạng đầu ra (*Output Network*): sử dụng đầu ra R-Net để đưa ra kết quả cuối cùng với 5 điểm đánh dấu khuôn mặt: 2 điểm mắt, 1 điểm mũi và 2 điểm khóe miệng (Hình.5) Bằng cách thử các cấu trúc khác nhau, chúng tôi chọn được một cấu trúc phù hợp có độ chính xác cao cho bài toán như sau

3.1. Mạng *P-Net*



Hình 3: Mạng P-Net

Mạng P-Net sử dụng kiến trúc CNN gồm 3 lớp tích chập và 1 lớp co.

Đầu vào cửa sổ trượt với kích thước 12x12x3 (3 tương ứng với 3 màu: đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường).

Lớp tích chập 1: Lớp tích chập với số bộ lọc: 10,

stride = 1, padding = 0, hàm kích hoạt = PReLU, kích thước bộ lọc: 3 x 3 x 10, số lượng tham số: (3 x 3 x 3 +1) x 10 = 280, đầu ra: 10x10x10

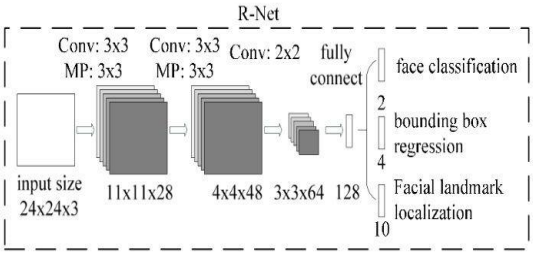
Lớp co: Maxpooling kích thước 2x2, stride = 2; padding = 0; Khi đó kích thước đầu ra của dữ liệu giảm đi 1/2, với chiều sâu được giữ nguyên còn 5x5x10.

Lớp tích chập 2: Kích thước đầu vào 5x5x10, số bộ lọc: 16, stride = 1, padding = 0, hàm kích hoạt=PReLU, kích thước bộ lọc: 3 x 3 x 16, số tham số: (3x3x10+1)x16=1456, đầu ra: 3x3x16.

Lớp tích chập 3: Kích thước đầu vào 3x3x16, số bộ lọc: 32, stride = 1, padding = 0, hàm kích hoạt=Softmax, kích thước bộ lọc: 3 x 3 x 32, số lượng tham số: (3 x 3 x 16 +1) x 32 = 4640, đầu ra: 1x1x32.

Kết quả của *P-Net:* mạng phân được 3 cụm gồm cụm thứ nhất có 2 bộ lọc kích thước 1x1 nhận dạng khuôn mặt, cụm thứ hai có 4 bộ lọc kích thước 1x1 đóng khung 4 vị trí hộp giới hạn và cụm còn lại có 10 bộ lọc kích thước 1x1 đóng khung 10 vị trí khuôn mặt.

3.2. Mạng R-Net



Hình 4: Mạng R-Net

Trong bước *R-Net* sử dụng kiến trúc CNN gồm 3 lớp tích chập, 2 lớp co và 1 lớp kết nối đầy đủ. Đầu vào cửa sổ trượt với kích thước 24x24x3 (3 tương ứng với 3 màu: đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường).

Lớp tích chập 1: Kích thước đầu vào 24x24x3, số bộ lọc: 28, stride=1, padding=0, hàm kích hoạt=*PReLU*, kích thước bộ lọc: 3x3x28, số tham số: (3x3x3+1)x28=812, kích thước đầu ra: 22x22x28

Lớp co: sử dụng kỹ thuật *MP* kích thước 3x3, stride=2; padding=1; kích thước đầu ra của dữ liệu giảm 1/2, chiều sâu dữ liệu giữ nguyên là 11x11x28.

Lớp tích chập 2: Kích thước đầu vào 11x11x28, số bộ lọc: 48, stride=1, padding=0, hàm kích hoạt=*PReLU*, kích thước bộ lọc: 3x3x28, số tham số: (3x3x28 +1)x48=12144, kích thước đầu ra: 9x9x48

Lớp co: sử dụng kỹ thuật *MP* kích thước 3x3, stride = 2; padding =0; đầu ra: 4x4x8

Lớp tích chập 3: Kích thước đầu vào 4x4x48, số bộ lọc: 64, stride=1, padding=0, hàm kích hoạt=*PReLU*, kích thước bộ lọc: 2x2x64, số lượng tham số: (2x2x48+1)x64= =12352, kích thước đầu ra: 3x3x64.

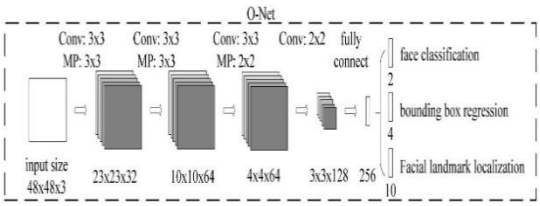
Lớp kết nối đầy đủ: Kích thước đầu vào 3x3x64, hàm kích hoạt=softmax, số tham số: (3x3x64+1)x128=73856, đầu ra: 128.

Kết quả của *R-Net* phân được 3 cụm gồm cụm thứ nhất có 2 lớp nhận dạng khuôn mặt, cụm thứ hai có 4 lớp đánh dấu vị trí hộp giới hạn và cụm còn lại có 10 lớp vị trí khuôn mặt

3.3. Mạng O-Net

Mạng *O-Net* sử dụng CNN gồm 4 lớp tích chập, 2 lớp co, 1 lớp kết nối đầy đủ. Đầu vào cửa sổ trượt có kích thước 48x48x3 (trong đó số 3 tương ứng với 3 màu: đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường).

Lớp tích chập 1: Kích thước đầu vào 48x48x3, số bộ lọc: 32, stride=1, padding=2, hàm kích hoạt=*PReLU*, kích thước bộ lọc: 3x3x32, số lượng tham số: (3x3x3+1)x32=896, đầu ra: 46x46x32.



Hình 5: Mạng O-Net

Lớp co: sử dụng kỹ thuật *MP* kích 3x3, stride=2; padding=1 ; Khi đó kích thước đầu ra của dữ liệu giảm đi với chiều sâu được giữ nguyên còn 23x23x32.

Lớp tích chập 2: Kích thước đầu vào: 23x23x32, số bộ lọc: 64, stride=2, padding=1, hàm kích hoạt=PReLU, kích thước bộ lọc: 3x3x28, số lượng tham số: (3x3x32+1)x64 = 18496, đầu ra: 12x12x64

Lớp co: sử dụng kỹ thuật *MP* kích thước 3x3, *stride=1; padding=0*; khi đó kích thước đầu ra của dữ liệu giảm với chiều sâu được giữ nguyên còn 10x10x64.

Lớp tích chập 3: Kích thước đầu vào: 10x10x64, số bộ lọc: 64, stride=1, padding=0, hàm kích hoạt=PReLU, kích thước bộ lọc: 3x3x32, số lượng tham số: (3x3x64+1)x64=36928, đầu ra: 8x8x64.

Lớp co: sử dụng kỹ thuật *MP* kích thước 2x2, stride=2; padding=0 ; khi đó kích thước đầu ra của dữ liệu giảm 1/2 với chiều sâu được giữ nguyên tức là: 4x4x64.

Lớp tích chập 4: Kích thước đầu vào 4x4x64, số bộ lọc: 128, stride1, padding=0, hàm kích hoạt=*PReLU*, kích thước bộ lọc: 2x2x128, số lượng tham số: (2x2x64+1)x124=32896, kích thước đầu ra: 3x3x128.

Lớp kết nối đầy đủ với kích thước đầu vào 3x3x128, hàm kích hoạt=softmax, số lượng tham số: (3x3x128+1)x256=295168, kích thước đầu ra lớp kết nối đầy đủ: 256.

Kết quả của *O-Net* phân được 3 cụm gồm cụm thứ nhất có 2 lớp nhận dạng khuôn mặt, cụm thứ hai có 4 lớp đánh dấu vị trí hộp giới hạn và cụm còn lại có 10 lớp vị trí khuôn mặt

4. Cài đặt chương trình

4.1 Dữ liệu cho bài toán

Nguồn dữ liệu dùng cho huấn luyện mạng (theo luật học lan truyền ngược [7]) thử nghiệm được thu thập trên Internet (MS1M-ArcFace) với: 1.020 mẫu và có tổng số khuôn mặt: 85.000.

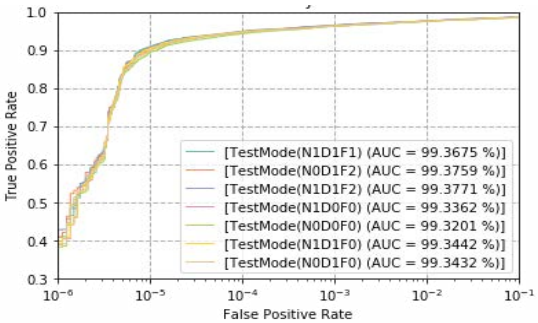
4.2 Cài đặt chương trình

Khởi tạo các thông số để huấn luyện mạng gồm tốc độ học (Learning Rate): 0.1; hệ số quán tính: 0.9; sai số cực tiểu:5.10^{-4} ; số lần học tối đa: 100.000; độ dài bó (steps): 128; số bó (batch_size): 10.000. Các mẫu huấn luyện mạng: 76.500 mẫu huấn luyện (chiếm 90% tổng số mẫu); 8.500 mẫu kiểm thử (chiếm 10% tổng số mẫu dữ liệu). Môi trường được sử dụng để huấn luyện mô hình nhận dạng là Debian, ngôn ngữ Python phiên bản 3.7.5 với Framework dùng cho huấn luyện mô hình là Caffe, MXNET; phần cứng: card đồ họa NVIDIA Geforce 840M, CPU Intel Core i7– 9400, 8GB RAM; Tốc độ 400 mẫu/giây; thời gian huấn luyện 22 ngày; số lớp ra 512 (x_i thuộc R_d, *d=512*); hàm mất mát : *ArcFace*. Điều kiện dừng của huấn luyện là đạt số vòng huấn luyện tối đa, hoặc sai số tuyệt đối (giá trị hàm mất mát) nhỏ hơn sai số cực tiểu.

4.3 Đánh giá và bàn luận kết quả

4.3.1Tiêu chí đánh giá

Có nhiều cách đánh giá một mô hình nhận mẫu (hay phân lớp) Tuỳ vào những bài toán khác nhau mà chúng ta sử dụng các phương pháp khác nhau. Các phương pháp thường được sử dụng là độ chính xác của hệ: (Accuracy Score) của ma trận nhầm lẫn (Confusion Matrix), đường đặc tính hoạt động của bộ nhận mẫu (Receiver Operating Characteristic curve: ROC) [9] v.v…ROC được dùng phổ biến để đánh giá các kết quả của bài toán nhận diện (Hình 6) [8]



Hình 6: Đồ thị đánh giá

Kết quả nhận dạng mặt người đạt độ chính xác theo dữ liệu kiểm thử đã nêu là 99.34 %.

4.3.2 Hàm mất mát

Hàm mất mát Arcface [2, 3, 4] là hàm cải tiến từ hàm Softmax. Do hàm Softmax bình thường không đủ để thực thi độ tương tự cao Công thức hàm mất mát:

$$L1 = -\frac{1}{m}\sum_{i=1}^{m} log\frac{e^{W_{y_i}^T x_i + b_i}}{\sum_{j=1}^{n} e^{W_{y_i}^T x}} \qquad (3)$$

trong đó, $x_i \in R^d$: đầu vào thứ *i*, y_i: đầu ra thứ *i*, $W \in R^{dxn}$: ma trận trọng số, $b \in R^n$ độ sai lệch.

5. Kết luận

Bài viết này trích rút đặc trưng xác định khuôn mặt người bằng MTCNN được thử nghiệm với độ chính xác cao. Hướng phát triển tiếp theo của bài báo là thử nghiệm với các đặc trưng đa dạng của ảnh, những tác động của nhiễu và độ tương phản khác nhau của ảnh. Một số trường

hợp như: thay đổi số lớp xử lý, lọc nhiễu để có đầu vào lớp đầy đủ khác nhau; tỷ lệ mẫu học/mẫu thử khác nhau… cho kết quả thế nào sẽ được nghiên cứu, công bố tiếp theo ở các bài báo và công trình khoa học công nghệ khác.

Tài liệu tham khảo

[1] Jiankang Deng, Jia Guo, Niannan Xue, Stefanos Zafeiriou, Imperial College London, InsightFace, FaceSoft (2016) ”ArcFace: Additive Angular Margin Loss for Deep Face Recognition”, Stefanos Zafeiriou Imperial College London.

[2] Y. Wen, K. Zhang, Z. Li, and Y. Qiao (2016). “A Discriminative Feature Learning Approach for Deep Face Recognition”. *European Conference on Computer Vision*, pages 499–515. Springer.

[3] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao (2017) “Range Loss for Deep Face Recognition with Long-Tail”. IEEE Conf. on Computer Vision and Pattern Recognition.

[4] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE, (2016) ”Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks”,

[5] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua (2015) “A Convolutional Neural Network Cascade for Face Detection,” IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5325-5334.

[6]. Nguyễn Thanh Tuấn (2019), “Deep Learning cơ bản”, trang 106 đến 110.

[7], Nguyễn Quang Hoan, Nguyễn Thị Trang, Nguyễn Thị Huyền, Trương Quốc Khánh, Nguyễn Thị Hoa. (2016). Kết hợp mạng nơ ron với giải thuật di truyền ứng dụng cho lớp bài toán nhận mẫu. Tạp chí KHCN DHSPKTHY, ISSN 2354-0575, số11/9, Tr. 57-62

[8] Pham Dinh Khanh (2017) “Giới thiệu về đường cong ROC [online]”, từ < https://rstudio-pubs-static.s3.amazonaws.com/267441_5459af9d83ae44f18a13aea4a479f31f.html>

[9] Jatayu (2018), “Hàm kích hoạt PreLU [online]”, từ <https://medium.com/@shoray.goel/

prelu-activation-e294bb21fefa>

[10] Sik-Ho-Tsang (2018) từ <https://medium.com/coinmonks/review-prelu-net-the-first-to-surpass-human-level-performance-in-ilsvrc-2015-image-f619dddd5617>.

FACE PATTERN RECOGNITION USING MULTI-TASK CASCADED CONVOLUTIONAL NEURAL NETWORKS

Abstract:

The article provides 2 steps of identifying a human face including taking out the faces in the image; extracting the features of that face and from the information obtained after analyzing, and verifying the identity of the person. The first step is based on the model of Multi-Task Cascaded Convolutional Networks (MTCNN) and the second step is based on the Convolutional Neural Network (CNN). The method of detecting faces by MTCNN due to the results after identification with high accuracy even if faces are obscured.

***Keywords**: Integrated neural network connected concurrently multitasking, neural network integrated with convolution, face pattern recognition*