TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
Tập 21, Số 6 (2024): 1118-1130
Website: https://journal.hcmue.edu.vn
ISSN: 2734-9918

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
Vol. 21, No. 6 (2024): 1118-1130
https://doi.org/10.54607/hcmue.js.21.6.3645(2024)

Bài báo nghiên cứu

PHÁT TRIỂN MÔ HÌNH HỌC SÂU CHO TRA CỨU THÔNG TIN VẬT PHẨM TRONG GAME BẰNG HÌNH ẢNH

***Trịnh Huy Hoàng*[1*], *Trần Sơn Hải*[1], *Lê Hồng Thúy Vũ*[2]**

[1]*Trường Đại học Sư phạm Thành phố Hồ Chí Minh, Việt Nam*

[2]*Trường Đại học Ngoại ngữ – Tin học Thành phố Hồ Chí Minh, Việt Nam*

[*]*Tác giả liên hệ: Trịnh Huy Hoàng – Email: hoangth@hcmue.edu.vn*

Ngày nhận bài: 18-10-2022; ngày nhận bài sửa: 25-10-2022; ngày duyệt đăng: 04-3-2024

TÓM TẮT

Sự phát triển mạnh mẽ của công nghệ thông tin trên thế giới đã thúc đẩy ngành công nghiệp game trở nên phổ biến và đa dạng, mang lại sức hút mạnh mẽ dành cho nhiều người ở các lứa tuổi khác nhau. Nhiều loại game được phát triển một cách mới mẻ; mang lại nhiều cảm giác thú vị, cũng như mang lại tính giải trí cao, hơn nữa có những game tích hợp hoạt động quảng cáo kèm dịch vụ mang lại lợi nhuận lớn như trao đổi, buôn bán các vật phẩm trong game giả lập hay nhập vai. Ngày nay, nhờ vào học sâu, việc nhận dạng vật phẩm đã có những kết quả khả quan và giữ một vị trí quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nghiên cứu đề xuất mô hình học sâu MILU_MODEL_1 hỗ trợ tra cứu thông tin vật phẩm trong game bằng hình ảnh có độ chính xác cao nếu không bị nhiễu và cải tiến mô hình thành MILU_MODEL_2 đáp ứng với việc nhận diện vật phẩm có nhiễu. Ứng dụng chạy dựa trên Keras của Tensorflow, một trong những nền tảng mạnh mẽ nhất hiện nay. Việc huấn luyện dựa trên bộ dữ liệu thu thập riêng với các biểu tượng và thông tin cụ thể trích từ dự án game MILU của công ty Grateful Days.

Từ khóa: học sâu; lập trình game; truy vấn ảnh; vật phẩm trong game

1 Giới thiệu

Trong môi trường giáo dục, nhiều phụ huynh dần quan tâm hơn đến việc tập cho trẻ làm quen với lập trình game. Thế giới của trẻ em là thế giới đầy màu sắc và sáng tạo, mà hoạt động vui chơi vẫn là hoạt động chính. Do đó, học lập trình kết hợp tạo ra trò chơi đối với trẻ nhỏ chỉ bằng cách ghép các khối lệnh nhiều màu sắc hay viết các lệnh đơn giản để điều khiển được các nhân vật theo ý mình là rất phù hợp, trẻ sẽ được sáng tác câu chuyện của riêng mình để tạo ra các dự án đầu tay với các nhân vật ngộ nghĩnh không chỉ mang lại giá trị tinh thần mà còn giúp trẻ phát triển đam mê, ham học hỏi và hình thành thói quen cân nhắc xem mình nên chọn giải pháp nào là phù hợp nhất, biết cách lí giải để người khác hiểu và rồi lắng nghe - phản hồi - cải tiến.

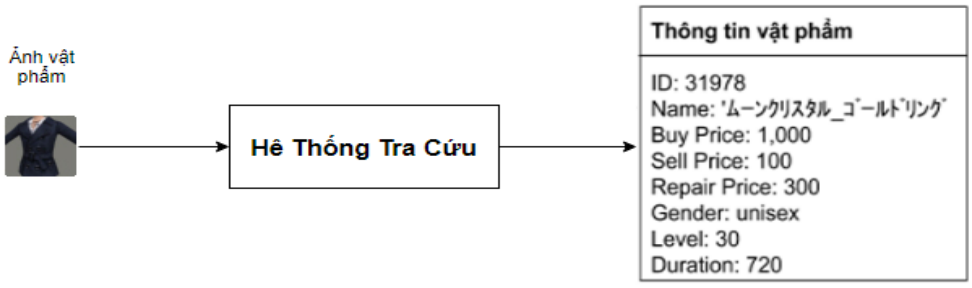
Cite this article as: Trinh Huy Hoang, Tran Son Hai, & Le Hong Thuy Vu (2024). Developing a deep learning model for searching item information in the game using images. *Ho Chi Minh City University of Education Journal of Science, 21*(6), 1118-1130.

Nghiên cứu xuất phát từ trò chơi MILU – một game mang tính mạng xã hội từ năm 2008 của Công ty Grateful Days. Sau hơn 10 năm ra đời và phát triển game vẫn đang được vận hành và yêu thích tại Nhật Bản. Đây là trò chơi mang tính chất mạng xã hội tập trung chủ yếu vào nội dung xây dựng nhân vật ảo như các vật phẩm thời trang, xây dựng nhà, kết bạn, trò chuyện, các sự kiện thời trang, kết hôn, câu cá… với số lượng người chơi đăng kí tới 3.456.733, số lượng người chơi hàng ngày (trung bình trong 7/2018): 5179 người chơi và tổng số lượng vật phẩm là 31, 120 vật phẩm (thông tin truy cập trang: https://www.milu.jp/). Với số lượng vật phẩm lớn và thời gian phát triển lâu dài nên có số lượng vật phẩm ít phổ biến và khá hiếm.

Bài toán đặt ra là khi người chơi muốn tìm kiếm thông tin cũng như cách để có được những vật phẩm xuất hiện trong các hình ảnh được chia sẻ trên diễn đàn nhưng không có thông tin nào khác của vật phẩm ngoài các hình ảnh của vật phẩm đó, công ti có mong muốn xây dựng một hệ thống hỗ trợ người chơi tìm kiếm thông tin vật phẩm bằng hình ảnh do người chơi cung cấp. Yêu cầu hệ thống có độ chính xác cao với mong muốn thông tin ảnh cần tìm nằm trong top 10 kết quả và có tốc độ xử lí nhanh.

Phát biểu bài toán: xây dựng hệ thống có khả năng phân tích hình ảnh và xác định vật phẩm game từ ảnh đầu vào. Sau cùng là xác định thông tin như: tên vật phẩm, giá tiền, miêu tả, cách tìm vật phẩm… từ cơ sở dữ liệu đã có.

- Input: ảnh chứa icon vật phẩm cần tìm.
- Output: thông tin vật phẩm như tên vật phẩm, giá tiền, độ hiếm...
- Điều kiện, ràng buộc: đầu vào là Tập tin ảnh, có chọn vùng có chứa icon cần tìm.
- Tiêu chí: thông tin ảnh cần tìm nằm trong top 10 kết quả.



Hình 1. *Sơ đồ tổng quát bài toán*

Nghiên cứu xây dựng mô hình giải quyết bài toán đặt ra đồng thời làm nền tảng cho việc phát triển game theo hướng mạng xã hội bao gồm các yêu cầu về nhận diện khuôn mặt và biểu cảm nhân vật theo người chơi.

2 Cơ sở lí thuyết và một số nghiên cứu liên quan

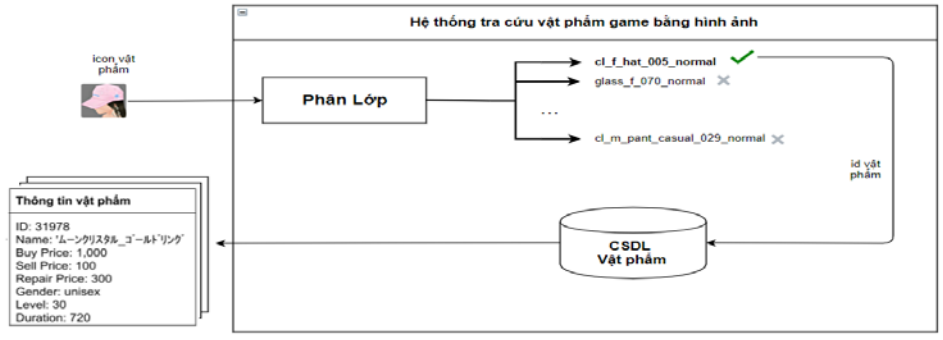
Bài toán tra cứu vật phẩm trong game bằng hình ảnh tìm ra các vật phẩm có hình ảnh giống với ảnh được truy vấn nhất trong cơ sở dữ liệu ảnh có sẵn. Các ảnh này có tên tương ứng với id duy nhất của vật phẩm, sau đó từ id ảnh được tìm thấy truy vấn trong cơ sở dữ liệu game để trả về các thông tin liên quan đến hình ảnh của vật phẩm được tra cứu, các

thông tin này bao gồm tên vật phẩm, giá tiền, cách sở hữu… Đây cơ bản cũng là một bài toán Content-Based Image Retrieval (Mohamed et al., 2019; Li et al., 2021), một bài toán tìm kiếm dựa trên nội dung của ảnh trong cơ sở dữ liệu là các hình ảnh. Tương tự như ứng dụng Google Images, người dùng nhập vào một ảnh, ứng dụng sẽ cho ra tất cả các kết quả liên quan đến ảnh bao gồm hình ảnh và bài viết.

Phương pháp phổ biến nhất cho bài toán CBIR là tìm kiếm theo độ tương đồng, tức là tìm kiếm sự giống nhau giữa ảnh được truy vấn với các ảnh trong cơ sở dữ liệu, sau đó trả về kết quả dựa trên mức độ giống nhau từ cao đến thấp. Phương pháp này có hai khó khăn cơ bản nhất đó là:

- Tìm được các đặc trưng ảnh tốt nhất để có thể so sánh độ tương đồng giữa các ảnh, hay nói cách khác là phải chọn các phương pháp rút trích đặc trưng ảnh phù hợp cho bộ dữ liệu ảnh.
- Tìm được độ đo sự tương đồng thích hợp cho bộ dữ liệu ảnh, ngoài ra các feature thường được biểu diễn dạng vector và có số chiều lớn từ vài trăm đến vài nghìn chiều. Việc so sánh các vector feature của ảnh truy vấn với toàn bộ ảnh trong cơ sở dữ liệu là rất mất thời gian và chi phí tính toán.

Để giải được bài toán tra cứu vật phẩm game bằng hình ảnh, ta cần giải quyết được hai khó khăn nêu trên. Về khó khăn thứ nhất, hướng tiếp cận phổ biến nhất hiện nay là dùng phương pháp học sâu, cụ thể là mô hình CNN cho bài toán phân loại ảnh nổi tiếng thường được sử dụng với các bài toán liên quan xử lí ảnh và cho độ chính xác tốt hơn nhiều các phương pháp truyền thống (H. S. Tran et al., 2016; Vo et al., 2017). Điểm nổi bật của mô hình CNN là khả năng tự học các đặc trưng trong suốt quá trình huấn luyện mạng, các đặc trưng nằm ở các tầng cuối của mạng trước khi đi qua tầng Softmax hoặc SVM để phân loại. Tuy nhiên, giải pháp này lại ảnh hướng lớn đến khó khăn thứ hai, vì các đặc trưng của mạng CNN thường có số chiều rất lớn. Hướng tiếp cận cho khó khăn thứ hai thường được sử dụng là Binary Hashing, phương pháp này chia vector feature thành 1 vector nhị phân có độ dài nhỏ hơn (hash code) (H. Tran et al., 2016; Tran et al., 2017). Sau khi có một mô hình giúp tìm hash code cho các ảnh việc tính toán độ tương đồng dễ dàng hơn vì chỉ làm việc với các vector có số chiều nhỏ và chỉ tính toán với các toán tử nhị phân đơn giản.



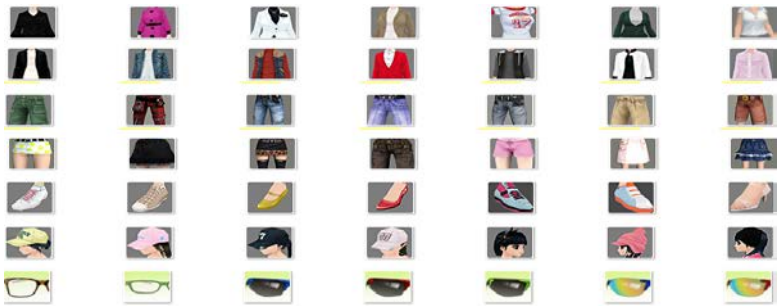
Hình 2. *Hệ thống tra cứu vật phẩm game bằng hình ảnh dùng phương pháp phân lớp ảnh*

Hiện tại, bài toán CBIR trong lĩnh vực game vẫn chưa có các công trình nghiên cứu nổi tiếng hay ứng dụng cụ thể. Tuy nhiên, trong các lĩnh vực khác đã có nhiều ứng dụng phổ biến như Google Images, tính năng tìm ảnh tương tự của các ứng dụng Flickr, Pinterest. Ngoài ra, một số công trình nghiên cứu về việc sử dụng CNN như một Feature Extractor cho bài toán CBIR như các nghiên cứu (Lin et al., 2015; Wan et al., 2014; Alzu'bi et al., 2017; Seo & Shin, 2019; Xu et al., 2019; Kolisnik et al., 2021). Tuy nhiên, các công bố này đều là sự kết hợp giữa các giải pháp kĩ thuật cho bài toán CBIR và kĩ thuật học sâu, không sử dụng hoàn toàn một phương pháp nào để giải bài toán như đề xuất của nghiên cứu.

3. Kết quả và thảo luận

3.1. Phân tích bộ dữ liệu ảnh trong game

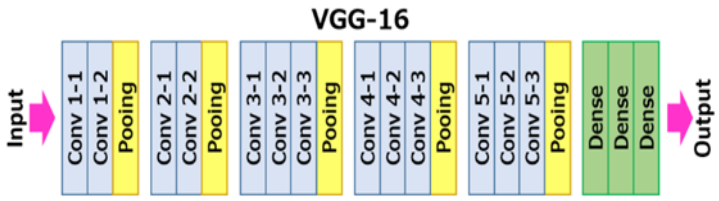
Bộ dữ liệu ảnh đầu vào là tập icon gốc trong game, với kích thước 46 x 46 pixel, định dạng JPG. Tổng icon vật phẩm trang bị hiện có là 5,756 icon. Các vật phẩm được chia thành 6 nhóm chính: body (áo), leg (quần), foot (giày dép), hand (găng tay), accessories (hoa tay, nhẫn, nón…) và outfit (bao gồm tất cả các phần trên). Do thời gian tiếp cận kiến thức tương đối ngắn và số lượng ảnh vật phẩm lớn nên nhóm chỉ sử dụng một phần dữ liệu để tiến hành nghiên cứu thử nghiệm. Tổng số icon sử dụng là 555/5,756 tương đương 9642% tổng số vật phẩm.



Hình 3. *Mẫu icon từ game*

Dựa vào bộ dữ liệu ảnh hiện có, với kích thước ảnh nhỏ giống nhau, sự khác biệt giữa các ảnh cùng loại là không nhiều do đó mô hình CNN cần có khả năng trích xuất đặc trưng cao. Như các kiến thức đã đề cập ở phần trên, để mạng CNN có khả năng trích xuất đặc trưng cao cần có nhiều tầng CONV chồng lên nhau. Điều này tương tự như kiến trúc CNN VGGNET của Simonyan và Zisserman (Simonyan & Zisserman, 2015).

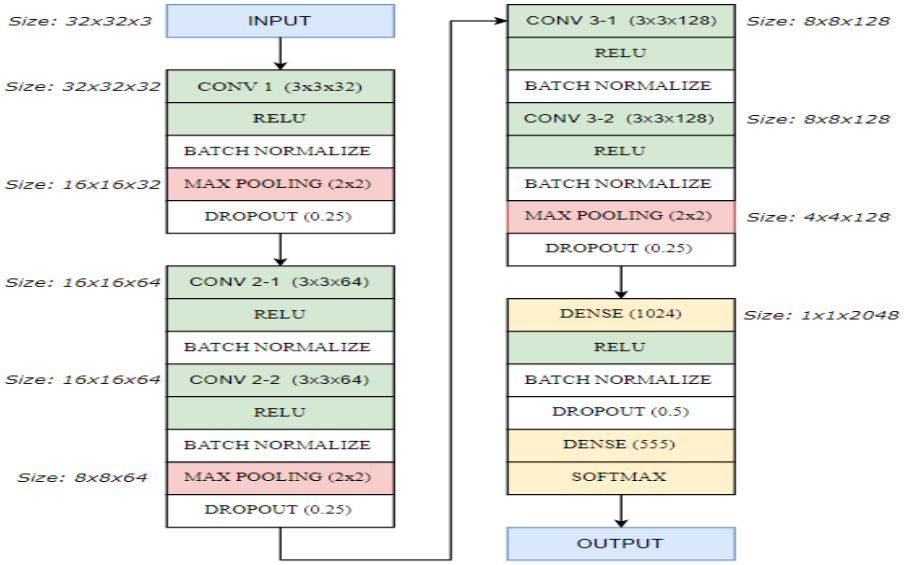
Kiến trúc này đạt được độ chính xác 92.7% và nằm trop top-5 kiến trúc có độ chính xác cao nhất của ImageNet với dataset hơn 14 triệu ảnh màu có độ phân giải cao thuộc về 1,000 lớp khác nhau. Kiến trúc này cải tiến thêm từ kiến trúc của AlexNet bằng cách thay thế filter kích thước lớn (11x11 và 5x5 trong các tầng CONV 1 và 2) với nhiều filter có kích thước 3x3. Giá trị đầu vào của kiến trúc là ảnh màu RGB có kích thước 224 x 224. Ảnh đầu vào được đưa qua nhiều tầng CONV chồng lên nhau, các filter này có kích thước rất nhỏ 3 x 3 (đó là kích thước nhỏ nhất để trích xuất các đặc trưng trái/phải, trên/dưới, chính giữa). Sử dụng filter 3x3 thay vì 11x11 ở Alexnet (7x7 ZFNet). Kết hợp nhiều tầng CONV 3x3 có hiệu quả hơn 1 tầng CONV kích thước lớn giúp mạng sâu hơn và giảm tham số tính toán cho mô hình. Nhược điểm của phương pháp này là tốn kém về vùng nhớ RAM khi huấn luyện mạng.



Hình 4. *Kiến trúc VGG-16*

3.2. Kiến trúc MILU_MODEL_1

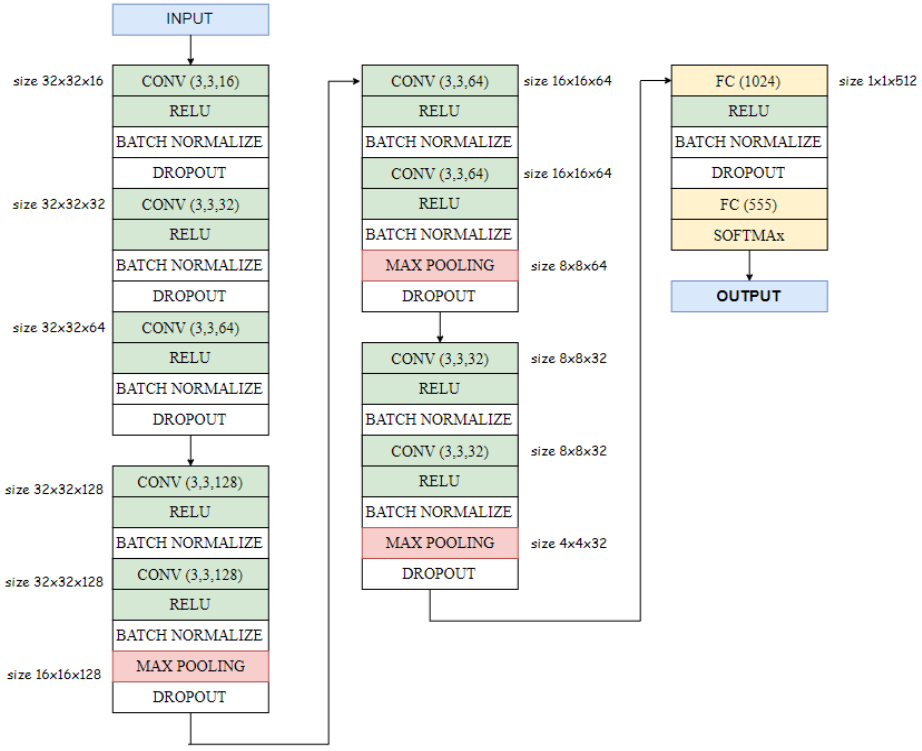
Kiến trúc VGG được thiết kế cho ảnh có kích thước lớn 224x224, do đó để phù hợp với bộ dữ liệu hiện có là bộ ảnh có kích thước 46 x 46, ta xây dựng lại kiến trúc tương tự với kiến trúc VGG nhưng với độ sâu thấp hơn để giảm thiểu vùng nhớ RAM và số lượng tham số, từ đó giảm thiểu thời gian huấn luyện mạng có tên là mô hình MILU_MODEL_1. Trong kiến trúc này ta có 7 tầng chứa trọng số (không kể các tầng NB) so với 16 tầng của kiến trúc mạng VGG16.



Hình 5. *Kiến trúc MILU_MODEL_1*

3.3. Kiến trúc MILU_MODEL_2

Chúng tôi chọn phương án mở rộng mạng theo chiều sâu, bằng cách thêm vào nhiều tầng CONV với chiến lược tăng dần số filter của mỗi tầng theo các mốc [16, 32, 64, 128] và giảm dần về [128, 64, 64, 32, 32]. Trong quá trình tăng số filter vẫn sẽ giữ nguyên không gian ảnh và trong quá trình giảm sẽ thực hiện giảm số chiều không giản ảnh (max pooling). Với chiến lược này mong muốn đạt được việc tăng số tầng CONV và số filter sau mỗi tầng sẽ tạo ra được bộ dữ liệu trừu tượng đa dạng và giảm dần số filter của tầng CONV cùng với không gian ảnh với mong muốn lưu giữ các đặc trưng trừu tượng cao nhất và giảm số tham số cho mạng. Phần lớn số lượng tham số nằm ở tầng FC như ở mạng VGG-16 có 7*7*512*4096 = 102,760,448 tham số (feed forwad), ở mạng MILU_MODEL_1 có 4*4*128 * 1024 = 2,097,152 tham số. Với mạng đề xuất ta có 4*4*32*1024 = 524,288 tham số.



Hình 6. *Mô hình kiến trúc MILU_MODEL_2*

Mô hình kiến trúc đề xuất sâu hơn so với mô hình VGGSmaller với 11 tầng so với 7 tầng chứa tham số và có số tham số thấp hơn 524,288 so với 2,097,152 tham số.

3.4. Thực nghiệm

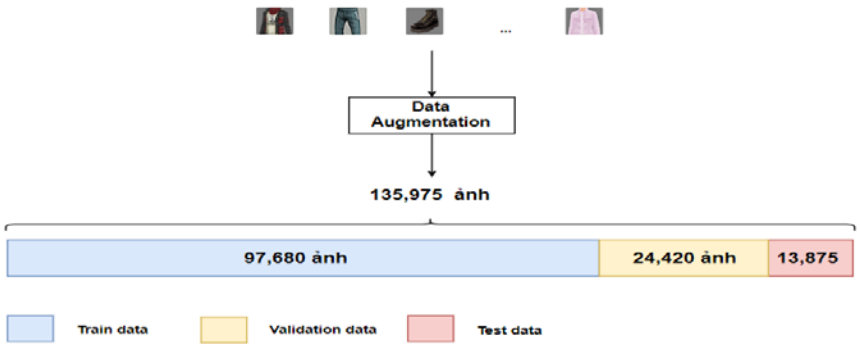
3.4.1. Dữ liệu đầu vào

Bộ dữ liệu có 555 icon ảnh thuộc về 555 vật phẩm khác nhau, so với bộ dữ liệu ảnh của ImageNet 1.2 triệu ảnh/ 1,000 lớp với trung bình 1 lớp sẽ 1200 ảnh để huấn luyện. Vì số dữ liệu ảnh hiện tại không đủ để huấn luyện mạng nên mục tiêu đặt ra là tìm thêm dữ liệu cho mỗi lớp trong tổng số 555 lớp hiện có. Một số cách để tăng cường dữ liệu mới như sau:

- Thu thập thủ công: công việc này đòi hỏi nhiều thời gian và công sức. Cụ thể nếu trong trường hợp 555 lớp hiện tại, người huấn luyện phải chụp lại ảnh icon của từng lớp ở các vị trí, góc độ, màu sắc và kích thước khác nhau để đa dạng hóa các biến thể dữ liệu của lớp đó.
- Kết hợp từ các dữ liệu sẵn có: kết hợp một số đặc điểm của các ảnh dữ liệu có sẵn, ví dụ chồng 2 ảnh lên nhau, hoặc thay thế khuôn mặt một người với tóc của người khác… Phương pháp này khá phức tạp, đôi khi không thể hiện đúng dữ liệu thực tế và không khả thi.
- Biến đổi tăng cường (augment): sử dụng các công cụ xử lí ảnh để giả định và tạo ra các biến thể khác nhau từ một ảnh gốc. Ví dụ như lật, xoay, dịch trái, phải, thay đổi tông màu…

Vì phương pháp Augment tương đối dễ thực hiện và tiết kiệm chi phí thời gian, nên người thực hiện sẽ sử dụng phương pháp này cho bộ dữ liệu hiện có. Một số quy tắc tạo biến thể mới như sau: lật, xoay, phóng to hay thu nhỏ, dịch chuyển, thay đổi độ sang và màu sắc.

Đối với bộ dữ liệu hiện có, ta có thể áp dụng toàn bộ các phương pháp trên để tạo ra bộ dữ liệu mới phong phú hơn từ một ảnh gốc. Cách áp dụng như sau:



Hình 7. *Bộ dữ liệu đạt được sau quá trình tăng cường*

3.4.2. Huấn luyện mạng

- **Giá trị đầu vào**

Vì kiến trúc mạng được thiết kế cho ảnh 32x32x3, và bộ dữ liệu hiện có là icon có kích thước 46x46x3, ta thực hiện scale ảnh về kích thước 32x32x3 trước khi huấn luyện. Ngoài ra, để quá trình hội tụ ổn định ta chuẩn hóa giá trị điểm ảnh từ [0-255] về [0-1].

- **Tiền xử lí dữ liệu**

Mỗi icon ảnh có kích thước lưu trữ trong khoảng 15kb-33kb, và tổng dung lượng ảnh có kích thước 400MB. Quá trình này chúng tôi sử dụng bộ thư viện numpy của python để tiền xử lí. Các thư mục ảnh sẽ lần lượt được load vào bộ nhớ và lưu vào một mảng numpy, đồng thời lưu nhãn của ảnh vào một mảng numpy khác với cùng chỉ số index. Các ảnh trước khi lưu vào mảng numpy sẽ được chuẩn hóa giá trị điểm ảnh về khoảng [0-1] có định dạng.**npz** (định dạng numpy) và có dung lượng 2.35G.

- **Thuật toán tối ưu (Optimizer):**

Sau khi tính toán được giá trị từ hàm lỗi, ta sử dụng giải thuật ***BGD,*** với toàn bộ tập dữ liệu để tính đạo hàm cập nhật lại giá trị của trọng số sao cho giá trị hàm lỗi đạt cực tiểu. Trong thực tế việc tính toán GD rất khó khăn nên ta sử dụng thuật toán biến thể **SGD** (Da, 2014) – một giải thuật đặc biệt hữu ích khi huấn luyện mạng CNN.

- **Bộ dữ liệu thử nghiệm thực tế từ game:**

Trong phần thử nghiệm này người thực hiện sử dụng các ảnh chụp trực tiếp màn hình chơi game, và cắt các icon vật phẩm để tạo bộ dữ liệu thử nghiệm. Các ảnh được cắt thủ công và có kích thước khác nhau. Ảnh chứa toàn bộ thông tin vật phẩm, minh họa trong điều kiện lí tưởng từ người dùng. Số ảnh thực nghiệm trong bộ dữ liệu này là 120 ảnh.

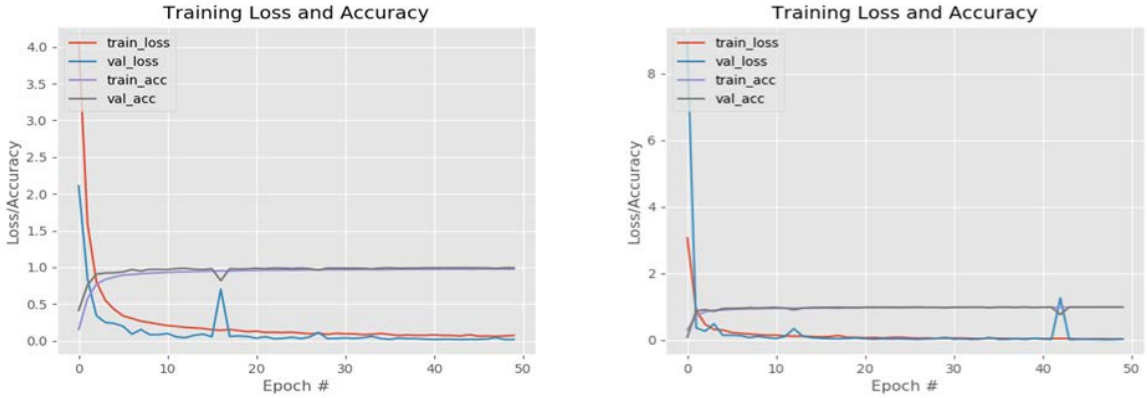
Hình 8. *Ảnh thử nghiệm thực tế từ game*

- **Huấn luyện mạng (training):** kiến trúc MILU_MODEL_1 và kiến trúc MILU_MODEL_2 với một số siêu tham số như sau: số ảnh huấn luyện 81,400 ảnh (2/3 tổng ảnh) và 122,100 ảnh (toàn bộ ảnh), số nơron ở tầng FC (128, 512, 1024).

3.4.3. Kết quả đạt được

- **Mô hình MILU_MODEL_1**

Với kiến trúc mô hình MILU_MODEL_1 các kết quả huấn luyện được thể hiện như các biểu đồ sau:



Hình 9. *Biểu đồ lỗi và chính xác sau khi huấn luyện kiến trúc* MILU_MODEL_1 *với tầng FC (128) trái, FC (1024) phải*

Kết quả sau khi thực nghiệm mạng với các bộ dữ liệu thử nghiệm như sau:

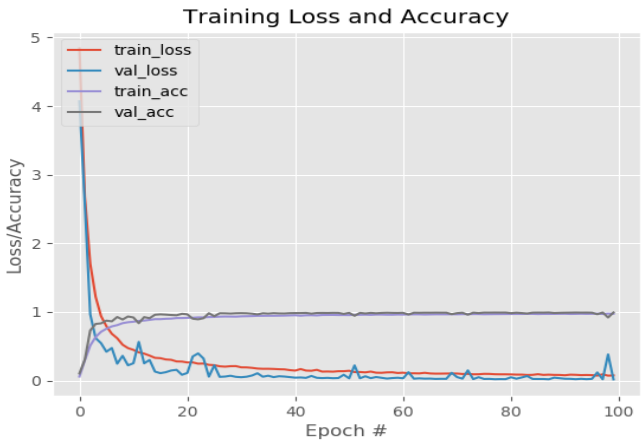
Bảng 1. *Kết quả thử nghiệm trên các mô hình mạng MILU_MODEL_1*

Tên mô hình	Kiến trúc	FC Nơron	Epochs	Test Data Augment	Real Data	Real with Noise
MILU_MODEL_1_128_50	MILU_MODEL_1	128	50	99.31%	100%	65.88%
MILU_MODEL_1_1024_50	MILU_MODEL_1	1024	50	99.42%	100%	62.35%
MILU_MODEL_1_128_100	MILU_MODEL_1	128	100	98.91%	90.83%	55.29%
MILU_MODEL_1_1024_100	MILU_MODEL_1	1024	100	99.06%	92.5%	48.23%
MILU_ 1_x3CONV128_1024	MILU_MODEL_1	1024	50	98.92%	93.33%	51.17%
MILU_ 1_MID_1024	MILU_MODEL_1	1024	50	99.5%	100%	61.7%

Kết luận: mô hình MILU_MODEL_1 với tầng FC (128) và số epochs 50 (***MILU_MODEL_1_128_50***) ***cho kết quả tốt nhất*** trên bộ dữ liệu thử nghiệm thực tế và trong điều kiện không có nhiễu với độ chính xác 100% trên 120 mẫu ảnh. Tuy nhiên, với bộ dữ liệu thử nghiệm với nhiễu (mất chi tiết, che phủ bởi màu khác) chỉ đạt 65.88% trên 170 mẫu ảnh. Theo kết quả trên ta thấy phần lớn ảnh dự đoán sai bị nhiễu về màu và các chi tiết màu nhỏ làm thay đổi đặc trưng ảnh.

- **Kiến trúc MILU_MODEL_2**

Với kiến trúc đề xuất như trên sau khi huấn luyện với 50 epoch và 100 epoch ta có các kết quả như sau:



Hình 10. *Kết quả sau khi huấn luyện kiến trúc MILU_MODEL_2 với 100 epochs*

Bảng 2. *Kết quả huấn luyện đạt giá trị tốt nhất trong các epochs trên mô hình kiến trúc mạng MILU_MODEL_2*

Tên mô hình	FC Nơron	Epochs	Epoch no	Train accuracy	Train loss	Validation accuracy	Validation Loss
MILU_MODEL 2_128_ 50	128	50	37	96.65%	9.96%	99.24%	1.9%
MILU_MODEL 2_1024_50	1024	50	42	98.38%	4.61%	99.29%	1.65%
MILU_MODEL 2_128_ 100	128	100	100	97.39%	7.56%	99.3%	1.73%

Bảng 3. *Kết quả thử nghiệm trên các mô hình kiến trúc mạng đề xuất MILU_MODEL_2*

Tên mô hình	Kiến trúc	FC Nơron	Epochs	Test Data Augment	Real Data	Real with Noise
MILU_MODEL_2_128_50	MILU_MODEL_2	128	50	99.21%	100%	64.70%
MILU_MODEL_2_1024_50	MILU_MODEL_2	1024	50	99.18%	100%	84.7%
MILU_MODEL_2_128_100	MILU_MODEL_2	128	100	99.2%	95%	60.58%

Kết luận: Kiến trúc mạng đề xuất MILU_MODEL_2 với tầng FC (1024) và 50 epochs cho kết quả tốt nhất (**MILU_MODEL_2_1024_50**). Tương tự như mô hình kiến trúc MILU_MODEL_1, tăng số epochs sẽ làm giảm độ chính xác của mạng. Mạng cho kết quả tốt với độ chính xác 100% trên 120 mẫu ảnh dữ liệu thực tế. Đồng thời mạng cũng cho kết quả tốt với các ảnh bị nhiễu hơn so với kiến trúc MILU_MODEL_1, đạt độ chính xác 84.7% so với 65.88% trên 170 mẫu ảnh thử.

Bảng so sánh kết quả đạt được giữa 2 kiến trúc MILU_MODEL_1 và kiến trúc đề xuất MILU_MODEL_2 như sau:

Bảng 4. *So sánh kết quả giữa 2 kiến trúc MILU_MODEL_1 và MILU_MODEL_2*

Tên mô hình	Kiến trúc	FC Nơron	Epochs	Test Data Augment	Real Data	Real with Noise
MILU_MODEL_1_128_50	MILU_MODEL_1	128	50	99.31%	100%	65.88%
MILU_MODEL_2_1024_50	MILU_MODEL_2	1024	50	99.18%	100%	84.7%

Hai mô hình kiến trúc cho kết quả tốt đối với bộ dữ liệu thử nghiệm lí tưởng (ảnh không bị mất chi tiết, nhiễu…). Tuy nhiên không đạt độ chính xác cao trên bộ dữ liệu có nhiễu (tự tạo, ít xuất hiện trong điều kiện thực tế).

Kết luận

- Với mô hình kiến trúc *MILU_MODEL_1* với kiến trúc dựa trên kiến trúc VGGNet với số filter ít hơn và nông hơn, mạng cho kết quả chính xác 100% trên 120 mẫu thử ảnh thực tế vật phẩm game. Mô hình cho kết quả tốt với ảnh bị mất hoặc nhiễu góc cạnh nhưng không tốt trong điều kiện ảnh bị mất mát chi tiết bên trong ảnh hoặc nhiễu màu với độ chính xác 65.88% trên 170 mẫu.
- Mô hình kiến trúc đề xuất *MILU_MODEL_2* cho kết quả chính xác 100% trên 120 mẫu thử ảnh thực tế vật phẩm game. Cho kết quả tốt hơn kiến trúc *MILU_MODEL_1* trên bộ dữ liệu thực tế bị nhiễu 84.7% so với 65.88% trên 170 mẫu thử. Cho kết quả không tốt với ảnh bị nhiễu với phần cạnh và các vùng nhiễu đa màu.
- Kiến trúc đề xuất MILU_MODEL_2 có độ sâu hơn kiến trúc *MILU_MODEL_1* nên thời gian huấn luyện mạng lâu hơn. Với kiến trúc *MILU_MODEL_1* thời gian huấn luyện trung bình dùng GPU là 30-35 phút, với kiến trúc đề xuất thời gian trung bình từ 70-75 phút.

4 Kết luận và hướng phát triển

Bài báo thực hiện nghiên cứu về học sâu cho bài toán phân loại hình ảnh. Bao gồm nghiên cứu về kiến trúc mạng nơron cơ bản MLP và kiến trúc state-of-the-art Convolutional Neural Network. Thực hiện nghiên cứu về các kiến trúc CNN nổi tiếng để tìm hướng đi và giải pháp cho bài toán tra cứu vật phẩm game hiện tại. Các kết quả trong nghiên cứu này dựa vào cải tiến mô hình từ VGGNET cho bài toán cần giải quyết.

Bài báo đề xuất mô hình kiến trúc cho bài toán nhận dạng vật phẩm trong game và thực nghiệm cho kết quả chính xác 100% cùng thời gian truy xuất nhanh chóng (30-50ms ảnh) đối với các ảnh bình thường không bị nhiễu hoặc mất chi tiết. Đối với các ảnh mất chi tiết trong khoảng 10-25% hoặc bị nhiễu ứng dụng cho kết quả với độ chính xác 84.7%. Với độ chính xác và thời gian truy xuất nhanh ứng dụng đáp ứng được các yêu cầu đề ra và có thể áp dụng thực tế.

Với những kết quả đạt được như trên, mô hình ứng dụng vẫn còn một số mặt hạn chế do thời gian nghiên cứu ngắn, một số phương pháp mới trong lĩnh vực phân loại ảnh chưa được áp dụng triệt để. Về mặt dữ liệu huấn luyện và thử nghiệm chưa đa dạng và phong phú do phần lớn ảnh phải thao tác thủ công và tốn nhiều thời gian. Với bộ dữ liệu huấn luyện hiện tại, ứng dụng không hỗ trợ các ảnh được chụp từ camera điện thoại.

Kết quả đạt được là tiền đề cho việc trợ giúp về cơ sở lí thuyết và mô hình ứng dụng đối với giảng viên và sinh viên trong hoạt động giảng dạy, nghiên cứu và triển khai ứng dụng trong lĩnh vực lập trình trò chơi, đặc biệt là các trò chơi trí tuệ trong ngành giáo dục.

❖ ***Tuyên bố về quyền lợi:*** *Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.*

TÀI LIỆU THAM KHẢO

Alzu'bi, A., Amira, A., & Ramzan, N. (2017). Content-based image retrieval with compact deep convolutional features. *Neurocomputing*, *249*, 95-105. https://doi.org/10.1016/j.neucom.2017.03.072

Da, K. (2014). A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

Mohamed, O., Mohammed, O., & Brahim, A. (2017). *Content-based image retrieval using convolutional neural networks*. Paper presented at the First International Conference on Real Time Intelligent Systems. https://doi.org/10.1007/978-3-319-91337-7_41

Kolisnik, B., Hogan, I., & Zulkernine, F. (2021). Condition-CNN: A hierarchical multi-label fashion image classification model. *Expert Systems with Applications*, *182*, Article 115195. https://doi.org/10.1016/j.eswa.2021.115195

Li, X., Yang, J., & Ma, J. (2021). Recent developments of content-based image retrieval (CBIR). *Neurocomputing*, *452*, 675-689.

Lin, K., Yang, H.-F., Hsiao, J.-H., & Chen, C.-S. (2015). Deep learning of binary hash codes for fast image retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (pp. 27-35). https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W03/html/Lin_Deep_Learning_of_2015_CVPR_paper.html

Seo, Y., & Shin, K. (2019). Hierarchical convolutional neural networks for fashion image classification. *Expert Systems with Applications*, *116*, 328-339.

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (arXiv:1409.1556). arXiv. http://arxiv.org/abs/1409.1556

Tran, H., Le, T., Le, T., & Nguyen, T. (2016). Burn Image Classification Using One-Class Support Vector Machine. In P. C. Vinh & V. Alagar (Eds.), *Context-Aware Systems and Applications* (Vol. 165, pp. 233-242). Springer International Publishing. https://doi.org/10.1007/978-3-319-29236-6_23

Tran, H. S., Le, T. H., & Nguyen, T. T. (2016). The degree of skin burns images recognition using convolutional neural network. *Indian Journal of Science and Technology*. *Indian J. Sci. Technol*, *9*(45), 1-6.

Tran, S. H, Le, M. T., Le, H. T., & Thuy, N. T. (2017). Real Time Burning Image Classification Using Support Vector Machine. *EAI Endorsed Transactions on Context-Aware Systems and Applications*, *4*(12), Article 152760. https://doi.org/10.4108/eai.6-7-2017.152760

Vo, A. T., Tran, H. S., & Le, T. H. (2017). Advertisement image classification using convolutional neural network. *2017 9th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 197-202). https://ieeexplore.ieee.org/abstract/document/8119458/

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. (2014). Deep Learning for Content-Based Image Retrieval: A Comprehensive Study. *Proceedings of the 22nd ACM International Conference on Multimedia* (pp. 157-166). https://doi.org/10.1145/2647868.2654948

Xu, H., Liu, B., Shu, L., & Yu, P. (2019). Open-world Learning and Application to Product Classification. *The World Wide Web Conference*, 3413-3419. https://doi.org/10.1145/3308558.3313644

DEVELOPING A DEEP LEARNING MODEL FOR SEARCHING ITEM INFORMATION IN THE GAME USING IMAGES

Trinh Huy Hoang[1*], Tran Son Hai[1], Le Hong Thuy Vu[2]

[1]Ho Chi Minh City University of Education, Vietnam

[2]Ho Chi Minh City University of Foreign Languages – Information Technology, Vietnam

[*]Corresponding author: Trinh Huy Hoang – Email: hoangth@hcmue.edu.vn

Received: October 18, 2022; Revised: October 25, 2022; Accepted: March 04, 2024

ABSTRACT

The strong development of information technology in the world has promoted the gaming industry to become popular and diverse, bringing strong attraction to many people of different ages. Many types of games are developed in a new way, bringing many interesting feelings, as well as high entertainment. Moreover, some games integrate advertising activities with services that bring great profits such as exchanging and trading items in good simulation games or role-play. Today, thanks to deep learning, item recognition has achieved positive results and holds an important position in the field of computer vision and artificial intelligence. The study proposes a deep learning model MILU_MODEL_1 that supports looking up game item information using images with high accuracy if there is no noise, and improves the model to MILU_MODEL_2 that responds to identifying items with noise. The application runs on Tensorflow's Keras, one of the most powerful platforms today. The training is based on a separately collected data set with symbols and specific information extracted from the MILU game project of the company Grateful Days.

Keywords: deep learning; game items; game programming; image query