 DOI: 10.31276/VJST.64(10DB).02-07

Phát hiện một số bệnh gia cầm dựa trên hình ảnh thu nhận từ camera bằng mạng nơ-ron tích chập

Đoàn Hồng Quang*, Nguyễn Huy Công, Lê Hồng Minh, Phạm Vũ Ban, Nguyễn Huy Hưng, Nguyễn Tuấn Hùng

Trung tâm Công nghệ Vi điện tử và Tin học, Viện Ứng dụng Công nghệ

Ngày nhận bài 11/7/2022; ngày chuyển phản biện 14/7/2022; ngày nhận phản biện 28/7/2022; ngày chấp nhận đăng 3/8/2022

Tóm tắt:

Bài viết giới thiệu một số kết quả trong việc nghiên cứu xây dựng hệ thống tự động phát hiện sớm một số bệnh gia cầm bằng công nghệ thị giác máy sử dụng trí tuệ nhân tạo (AI) để phân tích các hình ảnh từ camera giám sát chuồng nuôi. Hệ thống thử nghiệm sử dụng mạng nơ-ron học sâu RepVGG làm mô hình suy diễn phát hiện bệnh gia cầm. Dữ liệu để huấn luyện và đánh giá mô hình đối với 3 loại bệnh là ORT, CRD, MAREK gồm 500 ảnh cho mỗi bệnh, được thu thập từ camera giám sát chuồng nuôi và sưu tầm trên internet. Quá trình huấn luyện sử dụng 80% số ảnh mẫu, mô hình đạt sai số là 0,02 sau 5000 vòng huấn luyện. Đánh giá mô hình với 10% số mẫu còn lại đạt độ chính xác 99,94%. Trong thử nghiệm thực tế với camera giám sát chuồng nuôi (độ phân giải 1902x1080 pixels), hệ thống có thể phát hiện và phân loại đúng trên 90%, đặc biệt với bệnh có hình ảnh có nhiều đặc điểm phân biệt rõ thì độ chính xác đạt hơn 98%. Kết quả đạt được cho thấy, có thể sử dụng hệ thống để hỗ trợ cảnh báo sớm bệnh cho gia cầm nuôi. Hệ thống hoàn toàn có thể huấn luyện tiếp để phát hiện các bệnh cho gia cầm khác như: Newcastle, đậu gà (bánh trái gà), bạch lỵ, thương hàn...

Từ khóa: **mạng nơ-ron học sâu, mạng nơ-ron tích chập, phát hiện sớm bệnh gia cầm.**

Chỉ số phân loại: **1.2**

Mở đầu

Ngành chăn nuôi nói chung, chăn nuôi gia cầm nói riêng ngày càng phát triển cả về số lượng và chất lượng, góp phần tạo việc làm, tăng thu nhập cho người lao động, tạo thế mạnh kinh tế của nhiều địa phương. Tuy nhiên, sự phát triển về số lượng của chăn nuôi gia cầm hiện nay cũng gặp nhiều khó khăn trong quá trình chăm sóc, phòng bệnh, chữa bệnh, đặc biệt là phát hiện bệnh để điều trị kịp thời còn rất khó khăn.

Hiện nay, nhiều trại chăn nuôi, hộ chăn nuôi quy mô vừa và nhỏ chưa đủ nguồn lực để phát hiện sớm dịch bệnh trên đàn gia cầm, do đó dịch bệnh vẫn thường xuyên xảy ra. Dịch bệnh sẽ làm ảnh hưởng đến năng suất, chất lượng sản phẩm gia cầm, đặc biệt còn ảnh hưởng đến tính mạng người chăn nuôi và tiêu thụ sản phẩm gia cầm. Do thời tiết khắc nghiệt ở các miền trong cả nước, nóng lạnh bất thường ở miền Bắc, nắng nóng ở các tỉnh phía Nam, nguy cơ dịch bệnh trên gia cầm là rất cao. Phát hiện sớm một số bệnh trên gia cầm và thông báo kịp thời đến người chăn nuôi sẽ làm giảm nguy cơ lây nhiễm ra cả đàn, giảm nguy cơ chết, giảm chi phí điều trị, hạn chế thiệt hại kinh tế cho các hộ chăn nuôi, trại chăn nuôi.

Deep learning [1, 2] là một phương pháp của học máy, các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán, nó được xây dựng dựa trên các khái niệm phức tạp, chủ yếu hoạt động với các mạng nơ-ron nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người. Các hệ thống deep learning có thể giải quyết mọi nhu cầu nhận dạng mẫu (nhận diện hình ảnh, giọng nói, xử lý ngôn ngữ tự nhiên...) mà không cần đến sự can thiệp của con người với độ chính xác cao.

Mạng nơ-ron tích chập (Convolutional neural network - CNN) [3-5] là một trong những mô hình để nhận dạng và phân loại dựa trên quá trình phân tích hình ảnh đầu vào. Mỗi hình ảnh đầu vào sẽ được chuyển qua các lớp tích chập với các bộ lọc, được tổng hợp bởi các lớp kết nối đầy đủ và sử dụng hàm Softmax để nhận dạng, phân loại với độ chính xác cao hơn các mô hình học máy truyền thống [6-9].

Những mô hình mạng học sâu [3, 4, 10-12] hiện nay đã đạt được những thành công lớn trong lĩnh vực phân loại hình ảnh với kiến trúc gồm các lớp Conv, ReLU và Pooling. Những mô hình phức tạp này tuy mang lại độ chính xác cao nhưng cũng tồn tại một số bất lợi do sự phức tạp của thiết kế nhiều nhánh khiến cho việc xây dựng mô hình trở nên khó khăn hơn, đồng thời khiến cho việc suy luận chậm hơn và giảm sự tận dụng bộ nhớ hay một số mô hình làm tăng cost của việc truy cập bộ nhớ nên không phù hợp trên một số thiết bị. Trong khi đó, RepVGG [9] có lợi thế cấu trúc đơn giản (không có nhánh và đầu ra của lớp trước sẽ trở thành đầu vào của lớp sau), chỉ sử dụng lớp Conv3x3, lớp kích hoạt ReLU, không cần những thiết kế phức tạp như tìm kiếm tự động hay sàng lọc thủ công [3, 4].

Bài viết trình bày ứng dụng mô hình CNN - RepVGG để xây dựng hệ thống nhận dạng và tự động phát hiện một số bệnh trên gia cầm từ hình ảnh được thu nhận trực tiếp của các camera giám sát chuồng nuôi.

*Tác giả liên hệ: Email: daohaoquang@gmail.com

Identification of some avian diseases based on images obtained from a camera by convolutional neural networks

Hong Quang Doan*, Huy Cong Nguyen, Hong Minh Le, Vu Ban Pham, Huy Hung Nguyen, Tuan Hung Nguyen

Center for Microelectronics and Information Technology, National Center for Technological Progress

Received 11 July 2022; accepted 3 August 2022

Abstract:

The research paper introduced several results achieved in the research and construction of an automatic system for early detection of various poultry diseases by computer vision technology, using artificial intelligence to analyse images from surveillance cameras. The experimental system used the deep learning neural network RepVGG as an inference model for detecting poultry diseases. Training data set and model evaluation for 3 diseases ORT, CRD, MAREK, including 500 images for each disease, were directly collected from surveillance cameras and from the internet. The training process used 80% of the sample images; the model reached an error of 0.02 after 5000 rounds of training. Evaluation of the model with the remaining 10% of samples achieved an accuracy of 99.94%. In the actual test with the surveillance camera (resolution 1902x1080 pixels), the system can detect and classify correctly diseases over 90%, especially for diseases with images involving multiple distinguishing symptoms, the accuracy was more than 98%. The obtained results showed that the system can be used to support early warning of diseases for poultry. The system can be trained to detect other poultry diseases such as Newcastle disease, fowl pox, pullorum disease, typhoid...

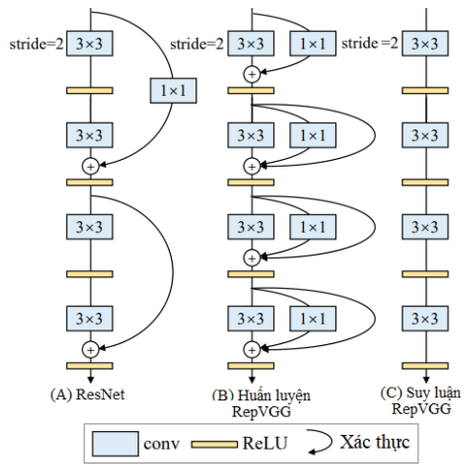
Keywords: **convolutional neural network, deep learning neural network, early detection for poultry disease.**

Classification number: **1.2**

Mạng nơ-ron - RepVGG

Kiến trúc mạng

Hình 1 trình bày một mô hình mạng RepVGG, với kiến trúc được tách thành hai phần riêng biệt đơn nhánh khi suy luận và đa nhánh khi huấn luyện mô hình, mô hình có 5 giai đoạn. Tất cả các khối đầu tiên mỗi giai đoạn đều có stride bằng 2 để giảm số mẫu, các khối khác thì có stride bằng 1. Mô hình chỉ sử dụng tích chập có kernel 3x3 và Relu (nhánh xác thực và 1x1 chỉ dùng khi huấn luyện mô hình), loại bỏ hoàn toàn lớp Pooling trong mô hình VGG.

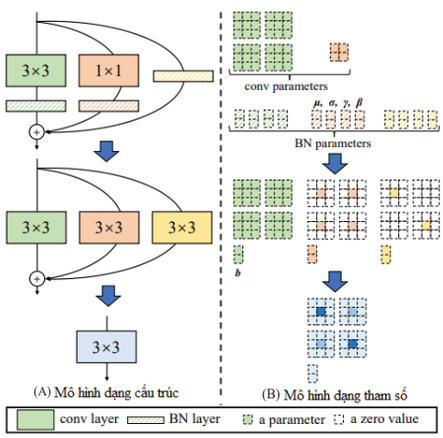


Hình 1. Mô hình mạng RepVGG. (A) Mạng ResNet; **(B)** Kiến trúc mạng RepVGG trong giai đoạn huấn luyện; **(C)** Mô tả mạng RepVGG trong giai đoạn suy luận.

Khi huấn luyện mô hình (hình 1B), đầu vào sẽ đi qua đồng thời 2 lớp Conv3x3 và Conv1x1 với stride=2 của một tiến trình; đầu ra của 2 lớp này sẽ được cộng lại với nhau và đi qua lớp kích hoạt ReLU.

Đầu ra của khối đầu tiên là đầu vào cho các khối tiếp theo, từ khối thứ 2 cấu trúc các khối trong mỗi tiến trình là giống nhau, bao gồm các nhánh: Conv3x3 với stride=1, Conv1x1 với stridde=1, là một nhánh xác thực sử dụng Batch normalization (BN) - phương pháp chuẩn hóa đầu ra của mỗi lớp sau khi đi qua lớp kích hoạt. Đầu ra của các nhánh được cộng lại với nhau trước khi đưa qua 1 lớp kích hoạt ReLU.

Trong suy luận (hình 1C), cấu trúc của RepVGG có sự thay đổi, đầu ra đi qua mô hình sẽ chỉ còn đi qua các lớp Conv3x3 và hàm kích hoạt ReLU liên tiếp. Đây là điểm phức tạp khi xây dựng mô hình, vì các Conv3x3 không chỉ đơn giản là một Conv3x3 với trọng số được lấy ngẫu nhiên. Trọng số của các Conv3x3 trong suy luận là tổng các trọng số của các nhánh Conv3x3, Conv1x1 và xác thực trong huấn luyện. Phương pháp này được gọi là Re-Parameterization (hình 2).



Hình 2. Mô hình dạng cấu trúc và tham số.

Đơn nhánh và đa nhánh

Trong huấn luyện, mô hình đa nhánh thường dễ hội tụ và đạt độ chính xác cao hơn so với đơn nhánh, nhưng trong quá trình suy luận thì lại có một số nhược điểm sau: tốc độ suy luận chậm hơn so với đơn nhánh; chiếm nhiều bộ nhớ (đa nhánh không hiệu quả trong sử dụng bộ nhớ do kết quả của tất cả các nhánh phải được lưu trữ cho đến khi thực hiện ghép các nhánh đó vào lại với nhau, còn với đơn nhánh thì bộ nhớ được giải phóng ngay khi tính toán xong, do đó tối ưu được bộ nhớ hơn); trong việc thiết kế kiến trúc mô hình, cấu trúc đa nhánh thường kém linh hoạt hơn so với mô hình đơn nhánh.

Để sử dụng được ưu điểm của cả hai mô hình (đa nhanh trong quá trình huấn luyện, đơn nhánh trong quá trình suy luận) thì cần đến một kỹ thuật để chuyển đổi các tham số của mô hình này sang mô hình khác, đây là kỹ thuật chính tạo nên sự khác biệt cho mô hình Re-Parameterization.

Chuyển đa nhánh sang đơn nhánh

Huấn luyện mô hình: Một khối của mô hình RepVGG sẽ có 3 nhánh gồm các convolution có kernel 3x3, 1x1 và nhánh xác thực.

Quá trình suy luận: Một khối RepVGG chỉ có 1 convolution 3x3, tất cả sau lớp convolution 3x3, 1x1 hay xác thực thì dùng thêm lớp BN để chuyển tham số của mô hình khi huấn luyện sang suy luận [9]. BN được tính như sau:

Với $\forall 1 \le i \le C_2$

$bn(M,\mu,\sigma,\gamma,\beta)_{:,I,:,:}$=$(M_{:,i,:,i,:} - \mu_i)\frac{\gamma_i}{\sigma_i} + \beta_i$

Đặt $W'_{i,:,:,:}$=$\frac{\gamma_i}{\sigma_i}W_{i,:,:,:}$; $b'_{i,:,:,:}$=$-\frac{\mu_i\gamma_i}{\sigma_i} + \beta_i$

Ta có:

$$M^{(2)} = bn(M * W,\mu,\sigma,\gamma,\beta)_{:,i,:,:}$$
$$= (M_{:,i,:,:}W_{:,i,:,:} - \mu_i)\frac{\gamma_i}{\sigma_i} + \beta_i$$
$$= M^{(1)} * W'_{:,i,:,:} + b'_i$$
$$= M^{(1)} * W' + b'$$

Quá trình huấn luyện: Đầu ra của khối RepVGG sẽ được tính như sau:

$$M^{(2)} = bn(M^{(1)} * W^{(3)},\mu^{(3)},\sigma^{(3)},\gamma^{(3)},\beta^{(3)}) \quad (1)$$
$$+ bn\left(M^{(1)} * W^{(1)},\mu^{(1)},\sigma^{(1)},\gamma^{(1)},\beta^{(1)}\right)$$
$$+ bn\left(M^{(1)},\mu^{(0)},\sigma^{(0)},\gamma^{(0)},\beta^{(0)}\right)M^{(2)}$$
$$= M(M^{(1)} * W'^{(3)})_{:,i,:,:} + b'^{(3)}_i \quad (2)$$
$$+(M^{(1)} * W'^{(1)})_{:,i,:,:} + b'^{(1)}_i$$
$$+(M^{(1)} * W'^{(0)})_{:,i,:,:} + b'^{(0)}_i$$
$$=> M^{(2)} = M^{(1)} * (W'^{(1)} + W'^{(3)}W'^{(0)})$$

Trong quá trình chuyển tham số, BN giữa hai kiến trúc đa nhánh và đơn nhánh phải tương đồng, do đó kích thước của $W'^{(1)}$, $W'^{(3)}$, $W'^{(0)}$ phải giống nhau để có thể thực hiện phép cộng tương ứng.

Do $W'^{(1)}$, $W'^{(3)}$, $W'^{(0)}$ có kích thước 1x1, 3x3, 1x1, để cùng kích thước cần phải gán các giá trị 0 vào $W'^{(1)}$, $W'^{(0)}$ cùng kích thước với $W'^{(3)}$. Cuối cùng, ta có:

Huấn luyện:

$$M^{(2)} = M^{(1)} * W'^{(1+3+0)} + b'^{(1+3+0)}) \quad (3)$$

Suy luận:

$$M^{(2)} = M^{(1)} * W' + b' \quad (4)$$
$$M^{(1)} = W'^{(1+3+0)} \text{ và } b' = b'^{(1+3+0)}$$

trong đó: C_1: kênh đầu vào; C_2: kênh đầu ra; $M^{(1)} \in R^{N*C_1*H_1*W_1}$: đầu vào; $M^{(2)} \in R^{N*C_2*H_2*W_2}$: đầu ra; $W^{(3)} \in R^{C_2*C_1*3*3}$: nhân của conv3x3; $W^{(1)} \in R^{C_2*C_1}$: nhân của conv1x1; $\mu^{(3)}$, $\delta^{(3)}$, $\gamma^{(3)}$, $\beta^{(3)}$: trung bình, độ lệch chuẩn, độ phóng đại, độ lệch của lớp BN sau lớp conv3x3; $\mu^{(1)}$, $\delta^{(1)}$, $\gamma^{(1)}$, $\beta^{(1)}$: trung bình, độ lệch chuẩn, độ phóng đại, độ lệch của lớp BN sau lớp conv1x1; $\mu^{(0)}$, $\delta^{(0)}$, $\gamma^{(0)}$, $\beta^{(0)}$: trung bình, độ lệch chuẩn, độ phóng đại, độ lệch của lớp BN của nhánh xác thực; [*]: phép tích chập; bn: Batch Normalization.

Xây dựng mô hình mạng

Cấu trúc RepVGG ứng dụng vào bài toán tự động nhận dạng một số bệnh gia cầm trong quá trình chăn nuôi được mô tả ở hình 3, hình ảnh gia cầm đưa vào mô hình được thu nhận trực tiếp từ camera trong chuồng nuôi.

Hình 3. Kiến trúc mạng RepVGG.

Kiến trúc mạng được thiết kế gồm 22 lớp tích chập và một lớp kết nối đầy đủ, với tổng tham số trong mô hình là 8.303.888 và tổng bộ nhớ 1.329.542 K. Tham số và bộ nhớ của từng lớp trong mô hình được mô tả như sau:

Đầu vào: ảnh với kích thước 224x224x3=147 K

- *Lớp conv1:*
+ Số bộ lọc: 48
+ Bộ lọc: (48,3,3,3)
+ Kích thước đầu ra: 48x112x112
+ Bộ nhớ: 15,876 K
+ Số lượng tham số: (3x3x3)x48 = 1.296
- *Lớp conv2:*
+ Số bộ lọc: 48
+ Bộ lọc: (48,48,3,3)
+ Kích thước đầu ra: 48x56x56
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (48x3x3)x48=20.736
- *Lớp conv3:*
+ Số bộ lọc: 48
+ Bộ lọc: (48,48,3,3)
+ Kích thước đầu ra: 48x56x56
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (48x3x3)x48=20.736
- *Lớp conv4:*
+ Số bộ lọc: 96
+ Bộ lọc: (48,3,3)
+ Kích thước đầu ra: 96x28x28
+ Bộ nhớ: 31,752 K
+ Số lượng tham số: (48x3x3)x96=41.472
- *Lớp conv5:*
+ Số bộ lọc: 96
+ Bộ lọc: (96,3,3)
+ Kích thước đầu ra: 96x28x28
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (96x3x3)x96=82.944
- *Lớp conv6:*
+ Số bộ lọc: 96
+ Bộ lọc: (96,3,3)
+ Kích thước đầu ra: 96x28x28
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (96x3x3)x96 =82.944
- *Lớp conv7:*
+ Số bộ lọc: 96
+ Bộ lọc: (96,3,3)
+ Kích thước đầu ra: 96x28x28
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (96x3x3)x96=82.944
- *Lớp conv8:*
+ Số bộ lọc: 192
+ Bộ lọc: (96,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 31,752 K
+ Số lượng tham số: (96x3x3)x192=165.888
- *Lớp conv9:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv10:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv11:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv12:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv13:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv14:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv15:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv16:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv17:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv18:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv19:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv20:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)

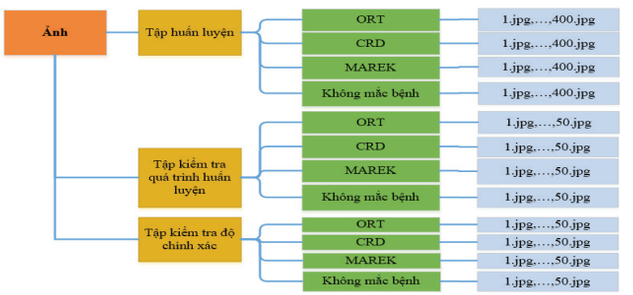
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv21:*
+ Số bộ lọc: 192
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 192x14x14
+ Bộ nhớ: 63,504 K
+ Số lượng tham số: (192x3x3)x192=331.776
- *Lớp conv22:*
+ Số bộ lọc: 1280
+ Bộ lọc: (192,3,3)
+ Kích thước đầu ra: 1280x7x7
+ Bộ nhớ: 105,840 K
+ Số lượng tham số: (192x3x3)x1280=2.211.840
- *Lớp FC (kết nối đầy đủ):*
+ Số bộ lọc: 21
+ Kích thước đầu ra: 1280
+ Bộ nhớ: 26 K
+ Số lượng tham số: 21x1280=26.88

Thu thập dữ liệu và huấn luyện mạng

Thu thập dữ liệu

Quá trình thu thập nguồn dữ liệu bệnh gia cầm dùng huấn luyện mô hình thử nghiệm được chụp bằng camera màu lắp trực tiếp trong chuồng nuôi gia cầm và thu thập trên internet.

Chúng tôi sử dụng một số phương pháp làm tăng dữ liệu cho mô hình (lật, cắt ngẫu nhiên, chuyển đổi màu, thay đổi độ tương phản...), với số tập mẫu: hình ảnh bệnh ORT (500 mẫu), CRD (500 mẫu), MAREK (500 mẫu) và hình ảnh gia cầm không bị bệnh (500 mẫu). Dữ liệu hình ảnh bệnh gia cầm được phân loại và gán nhãn, mỗi loại mẫu bệnh được đặt trong một thư mục loại bệnh tương ứng để sử dụng cho quá trình huấn luyện mô hình (hình 4).



Hình 4. Cấu trúc thư mục dữ liệu.

Huấn luyện

Nguồn dữ liệu hình ảnh bệnh gia cầm dùng huấn luyện mô hình thử nghiệm được thu thập từ các camera trong chuồng nuôi và trên internet.

Khởi tạo các tham số để huấn luyện:

+ net: "RepVGG.prototxt"	+ momentum: 0,9	+ gamma: 0,001
+ test_iter: 100	+ weight_decay: 0.0005	+ stepsize: 1000
+ test_interval: 100	+ snapshot: 500	+ display: 10
+ base_lr: 0,0001	+ snapshot_prefix: "RepVGG"	+ max_iter: 450000
+ learnRate: "step"	+ solver_mode: GPU	

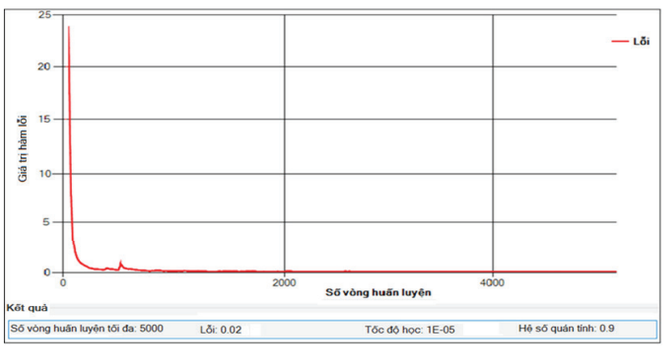
Môi trường được sử dụng để huấn luyện mô hình là Windows server 2012, ngôn ngữ Python phiên bản 3.7.1 với framework dùng cho huấn luyện mô hình là Caffe [13], card đồ họa Nvidia Quadro P2200, trong khoảng 5 ngày huấn luyện.

Huấn luyện mạng: Chúng tôi sử dụng 1600 mẫu bệnh (80% mẫu) để huấn luyện mô hình và sử dụng hàm lỗi "*cross-entropy loss*" trong công thức (5) để đánh giá sai số.

$$L_i = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{e^{\sum_j^n W_j^T x_i + b_j}} \quad (5)$$

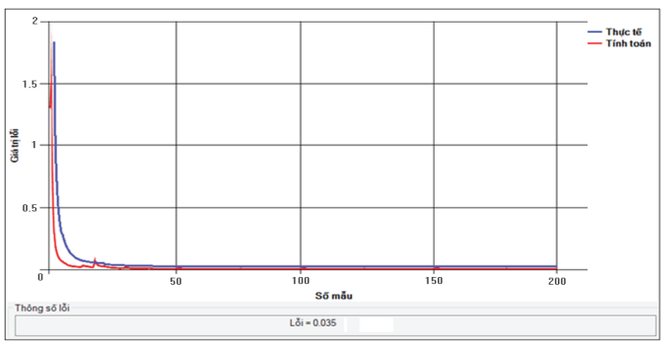
trong đó: x_i là vector thuộc $\mathbb{R}^d$; $W_j \in \mathbb{R}^d$ là cột thứ j của ma trận trọng số $W_j \in \mathbb{R}^{d\,x\,n}$; N và n là kích thước và số lớp trong tập dữ liệu.

Sau 5000 vòng huấn luyện bằng các tham số được khai báo với giá trị sai số là 0,02, kết quả huấn luyện mô hình được trình bày ở hình 5.



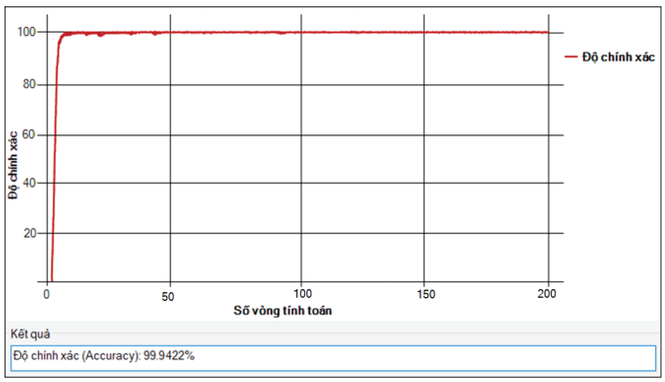
Hình 5. Đồ thị lỗi huấn luyện với mẫu dữ liệu.

Kiểm tra hoạt động của mô hình: Chúng tôi sử dụng 200 mẫu (10% mẫu) để kiểm tra mô hình mạng với các bộ trọng số đã được huấn luyện, kết quả kiểm tra với giá trị trung bình hàm lỗi là 0,035, kết quả này được thể hiện trên đồ thị kiểm tra hoạt động của mô hình ở hình 6.



Hình 6. Đồ thị kiểm tra hoạt động của mô hình.

Kiểm tra độ chính xác của mô hình: Chúng tôi sử dụng 200 mẫu còn lại (10% mẫu) để kiểm tra độ chính xác của mô hình mạng, kết quả kiểm tra trên bộ trọng số của mô hình với độ chính xác 99,94%, kết quả này được thể hiện trên đồ thị ở hình 7.



Hình 7. Đồ thị kiểm tra độ chính xác của mô hình.

Kết quả thực nghiệm

Hệ thống thử nghiệm trên máy tính với cấu hình CPU Core i7 2.9 Ghz, RAM 8 GB, GPU Nvidia Quadro P2200; camera màu với độ phân giải 1902x1080 pixel được lắp đặt trực tiếp trong chuồng nuôi gia cầm, với mỗi giây hệ thống phân tích và xử lý trung bình được 4 khung hình.

Mô hình được thử nghiệm nhận dạng 3 loại bệnh (ORT, CRD, MAREK), thời gian thử nghiệm từ 8/4/2022 đến 11/5/2022 trong chuồng nuôi 400 m^2 với số lượng gần 700 con gà đang có biểu hiện mắc một số bệnh. Trong ngày đầu tiên gia cầm mắc bệnh thường ít biểu hiện, các đặc trưng bên ngoài không được rõ, hệ thống khó phát hiện bệnh, những ngày sau đó các đặc trưng của bệnh rõ hơn, hệ thống phát hiện với độ chính xác cao hơn. Kết quả phát hiện bệnh và phân loại bệnh được trình bày ở bảng 1.

Bảng 1. Kết quả thực nghiệm.

TT	Loại bệnh	Ngày	Hệ thống phát hiện	Số ảnh phát hiện đúng	Độ chính xác (%)
1	ORT	8/4/2022	100	93	93,18
2	ORT	8/4/2022	150	143	95,56
3	ORT	8/4/2022	200	196	98,02
4	ORT	9/4/2022	210	207	98,78
5	ORT	10/4/2022	220	215	97,95
6	CRD	20/4/2022	215	211	98,32
7	CRD	21/4/2022	180	171	95,14
8	CRD	22/4/2022	150	147	98,16
9	CRD	30/4/2022	160	154	96,03
10	CRD	30/4/2022	170	167	98,08
11	CRD	30/4/2022	120	118	98,07
12	MAREK	8/5/2022	90	83	92,09
13	MAREK	8/5/2022	95	93	98,11
14	MAREK	9/5/2022	100	96	96,21
15	MAREK	10/5/2022	150	147	98,27
16	MAREK	11/5/2022	170	166	97,65

Kết luận

Một hệ thống chăn nuôi gia cầm công nghệ cao, ngoài các phân hệ quản lý giám sát môi trường, chế độ dinh dưỡng, quá trình sinh trưởng thì phân hệ phát hiện sớm bệnh của gia cầm nuôi là rất quan trọng. Bài viết đã trình bày kết quả đạt được về nội dung nghiên cứu xây dựng phân hệ "Tự động phát hiện sớm một số bệnh gia cầm bằng trí tuệ nhân tạo" - là một bộ phận trong hệ thống quản lý, giám sát nuôi gia cầm công nghệ cao cho quy mô vừa và nhỏ.

Hệ thống thử nghiệm sử dụng mạng RepVGG làm mô hình suy diễn phát hiện bệnh gia cầm. Dữ liệu để huấn luyện và đánh giá mô hình đối với 3 loại bệnh là ORT, CRD, MAREK gồm 500 ảnh cho mỗi bệnh, được trực tiếp thu thập từ camera giám sát chuồng nuôi và sưu tầm internet. Quá trình huấn luyện sử dụng 80% số ảnh mẫu; mô hình đã đạt đến sai số là 0,02 sau 5000 vòng huấn luyện. Đánh giá mô hình với 10% số mẫu còn lại đạt độ chính xác 99,94%. Trong thử nghiệm thực tế với camera giám sát chuồng nuôi (độ phân giải 1902x1080 pixels), hệ thống có thể phát hiện và phân loại đúng trên 90%, đặc biệt với bệnh có hình ảnh có nhiều đặc điểm phân biệt rõ thì độ chính xác đạt hơn 98%. Mô hình mạng RepVGG với kiến trúc đơn giản, tốc độ suy luận nhanh, tiết kiệm bộ nhớ và linh hoạt trong việc thiết kế kiến trúc là một lựa chọn hợp lý.

Kết quả đạt được cho thấy, có thể sử dụng hệ thống này để hỗ trợ cảnh báo sớm bệnh cho gia cầm nuôi. Hệ thống hoàn toàn có thể huấn luyện tiếp để phát hiện các bệnh khác cho gia cầm như: Newcastle, đậu gà (bánh trái gà), bạch lỵ, thương hàn...

TÀI LIỆU THAM KHẢO

[1] Y. LeCun, et al. (2015), "Deep learning", *Nature*, **521**, pp.436-444.

[2] A. Canziani, et al. (2016), "An analysis of deep neural network models for practical applications", *arXiv*, DOI: 10.48550/arXiv.1605.07678.

[3] Nguyễn Huy Công và cs (2021), "Phát hiện trứng gia cầm ứng dụng trong quy trình kiểm soát đóng gói tự động bằng mô hình mạng YOLO", *Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*, **33**, tr.41-47.

[4] Đoàn Hồng Quang (2020), "Phát hiện cháy rừng bằng mạng nơ-ron học sâu, dựa trên khói và lửa thu nhận được từ camera giám sát", *Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*, **26**, tr.92-99.

[5] Đoàn Hồng Quang, Lê Hồng Minh (2019), "Học chuyển giao và tinh chỉnh mô hình mạng nơ-ron học sâu ứng dụng trong nhận dạng khuôn mặt", *Kỷ yếu Hội thảo Ứng dụng công nghệ cao trong phát triển kinh tế - xã hội*, tr.50-61.

[6] Đoàn Hồng Quang, Lê Hồng Minh, Chu Anh Tuấn (2015a), "Nhận dạng bàn tay bằng mạng nơ-ron nhân tạo", *Tuyển tập báo cáo Diễn đàn "Đổi mới - Chìa khóa cho sự phát triển bền vững"*, tr.70-79.

[7] Đoàn Hồng Quang, Lê Hồng Minh (2015b), "Dùng RFNN kết hợp khử mùa và khử xu hướng để dự báo chỉ số giá vàng trên thị trường", *Tuyển tập báo cáo Diễn đàn "Đổi mới - Chìa khóa cho sự phát triển bền vững"*, tr.126-136.

[8] Nguyễn Quang Hoan, Đoàn Hồng Quang (2014b), "Dự báo chỉ số giá chứng khoán bằng RFNN", *Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*, **1**, tr.52-56.

[9] Nguyễn Quang Hoan, Dương Thu Trang, Đoàn Hồng Quang (2018), "Dự báo số học sinh nhập trường bằng mạng nơ-ron nhân tạo", *Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*, **18**, tr.1-8.

[10] Đoàn Hồng Quang, Lê Hồng Minh, Thái Doãn Nguyên (2020), "Nhận dạng khuôn mặt trong video bằng mạng nơ-ron tích chập", *Tạp chí Khoa học và Công nghệ Việt Nam*, **62(1)**, tr.8-12.

[11] Đoàn Hồng Quang, Nguyễn Huy Công (2019), "Phân loại hoa quả bằng mạng nơ-ron học sâu", *Kỷ yếu Hội thảo Ứng dụng công nghệ cao trong phát triển kinh tế - xã hội*, tr.132-142.

[12] C. Ding, D. Tao (2016), "A comprehensive survey on pose-invariant face recognition", *ACM Transactions on Intelligent Systems and Technology*, **7(3)**, DOI: 10.1145/2845089.

[13] Y. Jia, et al. (2014), "Caffe: Convolutional architecture for fast feature embedding", *arXiv*, DOI: 10.1145/2647868.2654889.