



PHÁT HIỆN CHÁY RỪNG BẰNG MẠNG NƠ RON HỌC SÂU, DỰA TRÊN KHÓI VÀ LỬA THU NHẬN ĐƯỢC TỪ CAMERA GIÁM SÁT

Đoàn Hồng Quang

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

Ngày tòa soạn nhận được bài báo: 15/03/2020

Ngày phản biện đánh giá và sửa chữa: 05/06/2020

Ngày bài báo được duyệt đăng: 15/06/2020

Tóm tắt:

Mạng nơ ron học sâu được ứng dụng trong nhận diện hình ảnh, nhận diện giọng nói, xử lý ngôn ngữ tự nhiên. Hiện nay rất nhiều các bài toán nhận dạng sử dụng mạng nơ ron học sâu, vì nó có thể giải quyết các bài toán với số lượng lớn các biến, tham số kích thước đầu vào lớn với hiệu năng cũng như độ chính xác vượt trội so với các phương pháp phân lớp truyền thống, xây dựng những hệ thống thông minh với độ chính xác cao. Trong bài báo này, các tác giả nghiên cứu mạng nơ ron học sâu (CNN - Convolutional Neural Network)[1] cho bài toán phát hiện lửa cũng như khói bằng việc xử lý dữ liệu từ camera giám sát nhằm phát hiện, cảnh báo cháy rừng.

Từ khóa: *Mạng nơ ron học sâu, mạng nơ ron tích chập, phát hiện cháy rừng.*

1. Giới thiệu

Hỏa hoạn gây ra những thiệt hại rất nghiêm trọng như phá hoại cơ sở vật chất, thiêu rụi các nhà dân, khu rừng chỉ trong một thời gian ngắn. Không những thế, hỏa hoạn còn gây ra nhiều thiệt hại về người, hủy hoại môi trường sống của con người và sinh vật, gây ô nhiễm môi trường..., nếu không được kiểm soát kịp thời thì hậu quả rất nặng nề. Vì vậy, việc phòng chống hỏa hoạn đóng một vai trò hết sức quan trọng và luôn được chú trọng. Tuy nhiên, việc phòng chống hỏa hoạn là một công việc hết sức khó khăn vì nguyên nhân phát sinh của hỏa hoạn là rất nhiều và khó đoán trước. Để giải quyết vấn đề này thì việc phát hiện cháy sớm, có biện pháp kịp thời ứng phó được xem là khả thi nhất.

Với sự phát triển nhanh chóng của công nghệ, nhiều hệ thống đã ra đời để đưa ra cảnh báo sớm cho cháy rừng. Có thể kể đến như: hệ thống cảnh báo cháy rừng nhờ cảm biến (cảm biến nhiệt độ, cảm biến khói...); hệ thống cảnh báo cháy rừng thông qua vệ tinh; hệ thống cảnh báo cháy rừng phát hiện qua video, hình ảnh từ camera quan trắc. Tuy nhiên, qua thực tế sử dụng thì hệ thống dùng phương pháp phân tích video mang lại nhiều ưu điểm và chính xác hơn. Với một camera có độ phân giải tốt, hoàn toàn có thể giám sát được một vùng rừng rộng lớn với độ chính xác cao. Trong khi đó, hệ thống dùng

cảm biến có nhược điểm là vùng giám sát rất nhỏ, và vấn đề năng lượng cung cấp cho các nút cảm biến cũng như tuổi thọ của các nút này bị ảnh hưởng rất nhiều bởi môi trường và điều kiện địa lý. Do đó, độ chính xác của hệ thống dùng cảm biến cũng không cao. Đối với hệ thống cảnh báo dùng vệ tinh thì có thể bao quát vùng rộng, tuy nhiên lại có một số hạn chế như: độ phân giải của ảnh không cao; thời gian thực không được đáp ứng, chi phí cao; ảnh hưởng của thời tiết làm giảm độ chính xác. Vậy nên dùng xử lý video hình ảnh được xem là một hệ thống tối ưu cho đến thời điểm này.

CNN là một trong những mô hình mạng nơ ron học sâu phổ biến nhất hiện nay [1,3], có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon... cho các mục đích khác nhau, như các thuật toán gán thẻ tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.

Sự ra đời của mạng CNN là dựa trên ý tưởng cải tiến cách thức các mạng nơ ron nhân tạo truyền thống [4-7] học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng nơ ron nhân tạo truyền thẳng (Feedforward

Neural Network) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh, kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra, sự liên kết đầy đủ này cũng là sự dư thừa với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. Mạng CNN với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ ron truyền thẳng.

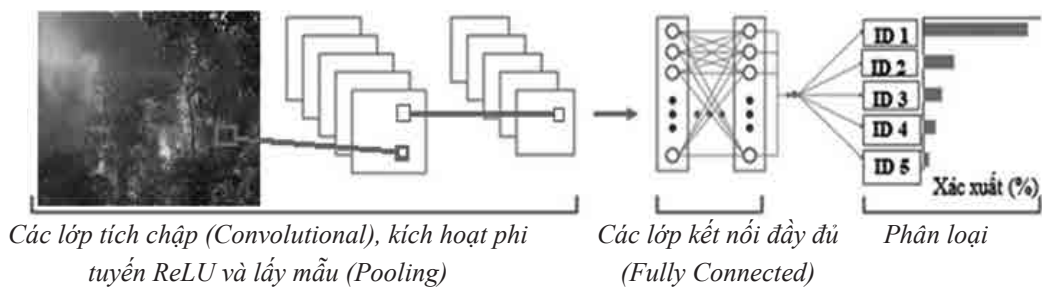
Trong bài viết này, chúng tôi nghiên cứu về

mạng CNN [8-10] sử dụng mô hình VGG16 ứng dụng trong việc xây dựng hệ thống nhận dạng và phát hiện cháy tự động từ camera [11-15].

2. Mạng nơ ron CNN - VGG16

2.1. Kiến trúc mạng CNN

Hình 1 trình bày một kiến trúc mạng CNN, các lớp cơ bản trong một mạng CNN bao gồm: lớp tích chập (Convolutional); lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit); lớp lấy mẫu (Pooling); lớp kết nối đầy đủ (Fully Connected) được thay đổi về số lượng và cách sắp xếp để tạo ra các mô hình huấn luyện phù hợp cho từng bài toán khác nhau.



Hình 1. Kiến trúc cơ bản của một mạng CNN.

- Lớp tích chập: đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước nhỏ.

Trong Hình 2 (a), bộ lọc được sử dụng là một ma trận có kích thước 3x3, bộ lọc này dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức 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 trống được thêm ở viền bức ảnh gốc và được tính theo công thức sau:

$$O = \frac{i + 2 * p - k}{s} + 1 \quad (1)$$

Trong đó:

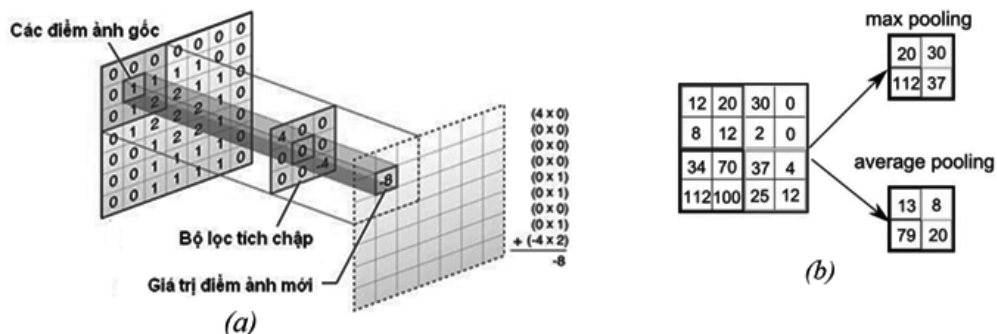
O : kích thước ảnh đầu ra;

i : kích thước ảnh đầu vào;

p : kích thước khoảng trống phía ngoài viền của ảnh gốc;

k : kích thước bộ lọc;

s : bước trượt của bộ lọc.



Hình 2. (a) Bộ lọc tích chập được sử dụng trên ma trận điểm ảnh. (b) Phương thức Average Pooling và Max Pooling.

Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp tích chập nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cập nhật trong quá trình huấn luyện.

- Lớp kích hoạt phi tuyến ReLU: được xây dựng để đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã 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 hoặc sigmoid, tanh... để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.

Lớp ReLU được áp dụng ngay phía 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 trừ các giá trị âm đã bị loại bỏ.

$$f(x) = \max(0, x) \quad (2)$$

- Lớp lấy mẫu: được đặt sau lớp tích chập và lớp ReLU để làm giảm kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh như lớp tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập, sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hình 2 (b) thể hiện các phương thức lấy mẫu thường được sử dụng nhất hiện nay, đó là Max Pooling (lấy giá trị điểm ảnh lớn nhất) và Average

Pooling (lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ).

Như vậy, với mỗi ảnh đầu vào được đưa qua lấy mẫu sẽ thu được một ảnh đầu ra tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán và nhận dạng.

- Lớp kết nối đầy đủ: được thiết kế tương tự như trong mạng nơ ron truyền thống, tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo.

So với mạng nơ ron truyền thống [4], các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng của ảnh cho việc nhận dạng, 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.

2.2. Xây dựng mô hình mạng

Hình 3 trình bày một cấu trúc VGG16 [1,3] ứng dụng vào bài toán phát hiện cháy rừng từ camera giám sát.

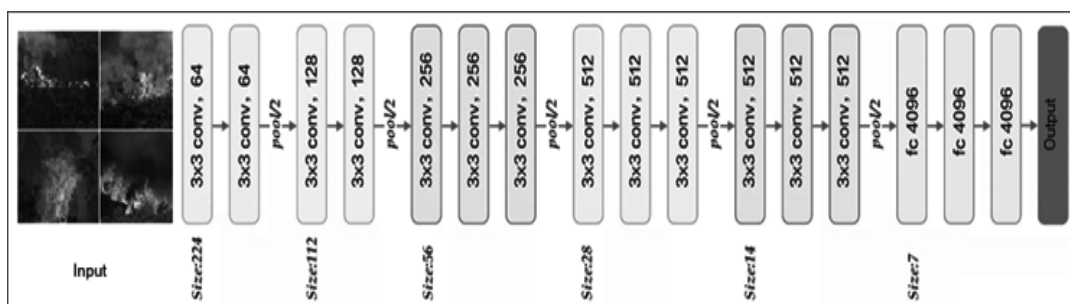
Tổng tham số trong mô hình là 138.357.544, các tham số và bộ nhớ trong mỗi lớp của mô hình mạng, như sau:

* Ảnh đầu vào:

Đầu vào: ảnh với kích thước $224 \times 224 \times 3 = 150K$ (3 tương ứng với 3 màu: đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường).

* Lớp 1 (Tích chập):

- Số bộ lọc: 64
- Kích thước bộ lọc: $3 \times 3 \times 64$
- Bộ nhớ: $224 \times 224 \times 64 = 3,2M$
- Số lượng tham số: $(3 \times 3 \times 3) \times 64 = 1.728$



Hình 3. Kiến trúc mạng VGG16.

*** Lớp 2 (Tích chập):**

- Đầu vào: $224 \times 224 \times 64$
- Số bộ lọc: 64
- Kích thước bộ lọc: $3 \times 3 \times 64$
- Bộ nhớ: $224 \times 224 \times 64 = 3,2M$
- Số lượng tham số: $(3 \times 3 \times 64) \times 64 = 36.864$

*** Lớp chuyển tiếp sang lớp 3 (Lấy mẫu):**

- Kích thước = (2,2)
- Khoảng cách thay đổi = 2
- Kích thước đệm = 0
- Bộ nhớ: $112 \times 112 \times 64 = 800K$

Kích thước đầu ra của dữ liệu giảm 1/2 từ ($224 \times 224 \times 3$) xuống ($112 \times 112 \times 3$), và chiều sâu được giữ nguyên

*** Lớp 3 (Tích chập):**

- Đầu vào: $112 \times 112 \times 3$
- Số bộ lọc: 128
- Kích thước bộ lọc: $3 \times 3 \times 128$
- Bộ nhớ: $112 \times 112 \times 128 = 1,6M$
- Số lượng tham số: $(3 \times 3 \times 64) \times 128 = 73.728$

*** Lớp 4 (Tích chập):**

- Đầu vào: $112 \times 112 \times 3$
- Số bộ lọc: 128
- Kích thước bộ lọc: $3 \times 3 \times 128$
- Bộ nhớ: $112 \times 112 \times 128 = 1,6M$
- Số lượng tham số: $(3 \times 3 \times 128) \times 128 = 147.456$

*** Lớp chuyển tiếp sang lớp 5 (Lấy mẫu):**

- Kích thước = (2,2)
- Khoảng cách thay đổi = 2
- Kích thước đệm = 0
- Bộ nhớ: $56 \times 56 \times 128 = 400K$

Kích thước đầu ra của dữ liệu giảm 1/2 từ ($112 \times 112 \times 3$) xuống ($56 \times 56 \times 3$), và chiều sâu được giữ nguyên

*** Lớp 5 (Tích chập):**

- Đầu vào: $56 \times 56 \times 3$
- Số bộ lọc: 256
- Kích thước bộ lọc: $3 \times 3 \times 256$
- Bộ nhớ: $56 \times 56 \times 256 = 800K$
- Số lượng tham số: $(3 \times 3 \times 128) \times 256 = 294.912$

*** Lớp 6 (Tích chập):**

- Đầu vào: $56 \times 56 \times 3$
- Số bộ lọc: 256
- Kích thước bộ lọc: $3 \times 3 \times 256$
- Bộ nhớ: $56 \times 56 \times 256 = 800K$
- Số lượng tham số: $(3 \times 3 \times 256) \times 256 = 589.824$

*** Lớp 7 (Tích chập):**

- Đầu vào: $56 \times 56 \times 3$

- Số bộ lọc: 256

- Kích thước bộ lọc: $3 \times 3 \times 256$

- Bộ nhớ: $56 \times 56 \times 256 = 800K$

- Số lượng tham số: $(3 \times 3 \times 256) \times 256 = 589.824$

*** Lớp chuyển tiếp sang lớp 8 (Lấy mẫu):**

- Kích thước = (2,2)

- Khoảng cách thay đổi = 2

- Kích thước đệm = 0

- Bộ nhớ: $28 \times 28 \times 256 = 200K$

Kích thước đầu ra của dữ liệu giảm 1/2 từ ($56 \times 56 \times 3$) xuống ($28 \times 28 \times 3$), và chiều sâu được giữ nguyên

*** Lớp 8 (Tích chập):**

- Đầu vào: $28 \times 28 \times 3$

- Số bộ lọc: 512

- Kích thước bộ lọc: $3 \times 3 \times 512$

- Bộ nhớ: $28 \times 28 \times 512 = 400K$

- Số lượng tham số: $(3 \times 3 \times 256) \times 512 = 1.179.648$

*** Lớp 9 (Tích chập):**

- Đầu vào: $28 \times 28 \times 3$

- Số bộ lọc: 512

- Kích thước bộ lọc: $3 \times 3 \times 512$

- Bộ nhớ: $28 \times 28 \times 512 = 400K$

- Số lượng tham số: $(3 \times 3 \times 512) \times 512 = 2.359.296$

*** Lớp 10 (Tích chập):**

- Đầu vào: $28 \times 28 \times 3$

- Số bộ lọc: 512

- Kích thước bộ lọc: $3 \times 3 \times 512$

- Bộ nhớ: $28 \times 28 \times 512 = 400K$

- Số lượng tham số: $(3 \times 3 \times 512) \times 512 = 2.359.296$

*** Lớp chuyển tiếp sang lớp 11 (Lấy mẫu):**

- Kích thước = (2,2)

- Khoảng cách thay đổi = 2

- Kích thước đệm = 0

- Bộ nhớ: $14 \times 14 \times 512 = 100K$

Kích thước đầu ra của dữ liệu giảm 1/2 từ ($28 \times 28 \times 3$) xuống ($14 \times 14 \times 3$), và chiều sâu được giữ nguyên

*** Lớp 11 (Tích chập):**

- Đầu vào: $14 \times 14 \times 3$

- Số bộ lọc: 512

- Kích thước bộ lọc: $3 \times 3 \times 512$

- Bộ nhớ: $14 \times 14 \times 512 = 100K$

- Số lượng tham số: $(3 \times 3 \times 512) \times 512 = 2.359.296$

*** Lớp 12 (Tích chập):**

- Đầu vào: $14 \times 14 \times 3$

- Số bộ lọc: 512
- Kích thước bộ lọc: $3 \times 3 \times 512$
- Bộ nhớ: $14 \times 14 \times 512 = 100K$
- Số lượng tham số: $(3 \times 3 \times 512) \times 512 = 2.359.296$

*** Lớp 13 (Tích chập):**

- Đầu vào: $14 \times 14 \times 3$
- Số bộ lọc: 512
- Kích thước bộ lọc: $3 \times 3 \times 512$
- Bộ nhớ: $14 \times 14 \times 512 = 100K$
- Số lượng tham số: $(3 \times 3 \times 512) \times 512 = 2.359.296$

*** Lớp chuyển tiếp sang lớp 14 (Lấy mẫu):**

- Kích thước = (2,2)
- Khoảng cách thay đổi = 2
- Kích thước đệm = 0
- Bộ nhớ: $7 \times 7 \times 512 = 25K$

Kích thước đầu ra của dữ liệu giảm $1/2$ từ $(14 \times 14 \times 3)$ xuống $(7 \times 7 \times 3)$, và chiều sâu được giữ nguyên

*** Lớp 14 (Kết nối đầy đủ):**

- Đầu vào: $1 \times 1 \times 4.096$
- Bộ nhớ: 4096K
- Số lượng tham số: $7 \times 7 \times 512 \times 4096 = 102.760.448$

*** Lớp 15 (Kết nối đầy đủ):**

- Đầu vào: $1 \times 1 \times 4096$
- Bộ nhớ: 4096K
- Số lượng tham số: $4.096 \times 4.096 = 16.777.216$

*** Lớp 16 (Kết nối đầy đủ):**

- Đầu vào: $1 \times 1 \times 4.096$
- Bộ nhớ: 1000K
- Số lượng tham số: $4.096 \times 1.000 = 4.096.000$

2.3. Học chuyển giao và tinh chỉnh mô hình huấn luyện [2]

Là quá trình khai thác, tái sử dụng các tri thức đã được học bởi một mô hình huấn luyện trước đó vào giải quyết một bài toán mới mà không phải xây dựng mô hình huấn luyện khác từ đầu. Hiện nay, phương pháp phổ biến thường được áp dụng khi huấn luyện mô hình với một bộ cơ sở dữ liệu (CSDL) tương đối nhỏ là sử dụng học chuyển giao để tận dụng một mạng đã được huấn luyện trước.

CNN đã được huấn luyện trước đó với bộ dữ liệu rất lớn như ImageNet (1,2 triệu ảnh với 1.000 nhãn đánh dấu). Phương pháp này sử dụng mạng CNN theo hai cách chính như sau:

Cách 1: Mạng CNN này chỉ được sử dụng như một bộ trích chọn đặc trưng cho bộ CSDL huấn

luyện mới, bằng cách thay thế các lớp kết nối đầy đủ ở cuối mô hình mạng và giữ cố định các tham số cho toàn bộ các lớp còn lại của mô hình. Thực hiện tối ưu, tinh chỉnh (Fine-Tune) một vài hoặc tất cả các lớp trong mô hình mạng.

Cách 2: Việc tái sử dụng mạng CNN là dựa trên các đặc trưng được học trong các lớp đầu của mạng là các đặc trưng chung nhất với phần lớn bài toán (ví dụ đặc trưng về cạnh, hình khối hay các khối màu khói, đám mây, lá cây,...). Các lớp sau đó của mạng CNN sẽ nâng dần độ cụ thể, riêng biệt của các chi tiết phục vụ cho bài toán nhận dạng cần giải quyết. Do đó, hoàn toàn có thể tái sử dụng lại các lớp đầu của mạng CNN mà không phải mất nhiều thời gian và công sức huấn luyện mô hình mạng từ đầu.

Có 2 loại học chuyển giao:

Feature Extractor: sau khi lấy ra các đặc điểm (khối, lửa, cảnh, lá...) của ảnh bằng việc sử dụng ConvNet của mô hình được huấn luyện trước, sẽ dùng phân loại tuyến tính (Linear SVM, Softmax Classifier...) để phân loại ảnh.

Fine-Tuning: sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng CNN của mô hình được huấn luyện trước, thì sẽ coi đây là đầu vào của CNN mới bằng cách thêm các lớp tích chập và lớp kết nối đầy đủ.

3. Huấn luyện mạng

* Nguồn dữ liệu dùng huấn luyện mô hình thử nghiệm được thu thập trên Internet:

- Tổng số mẫu ảnh cháy: 5000
- Tổng số mẫu ảnh khói: 2000

* Khởi tạo các thông số để huấn luyện mạng, được thiết lập như trong Bảng 1

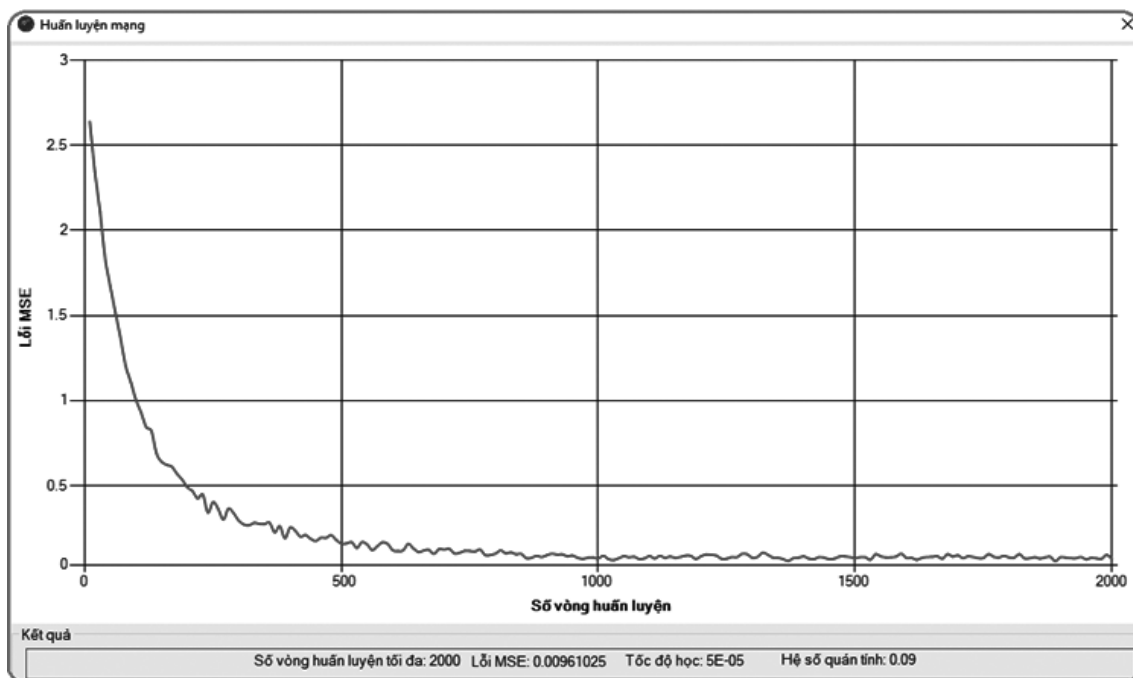
* Tham số huấn luyện mạng

- Tốc độ học: 0,00005
- Hệ số quán tính: 0,09
- Sai số cực tiểu: 0,00001
- Số lần học tối đa: 2000 vòng

* Huấn luyện mạng:

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

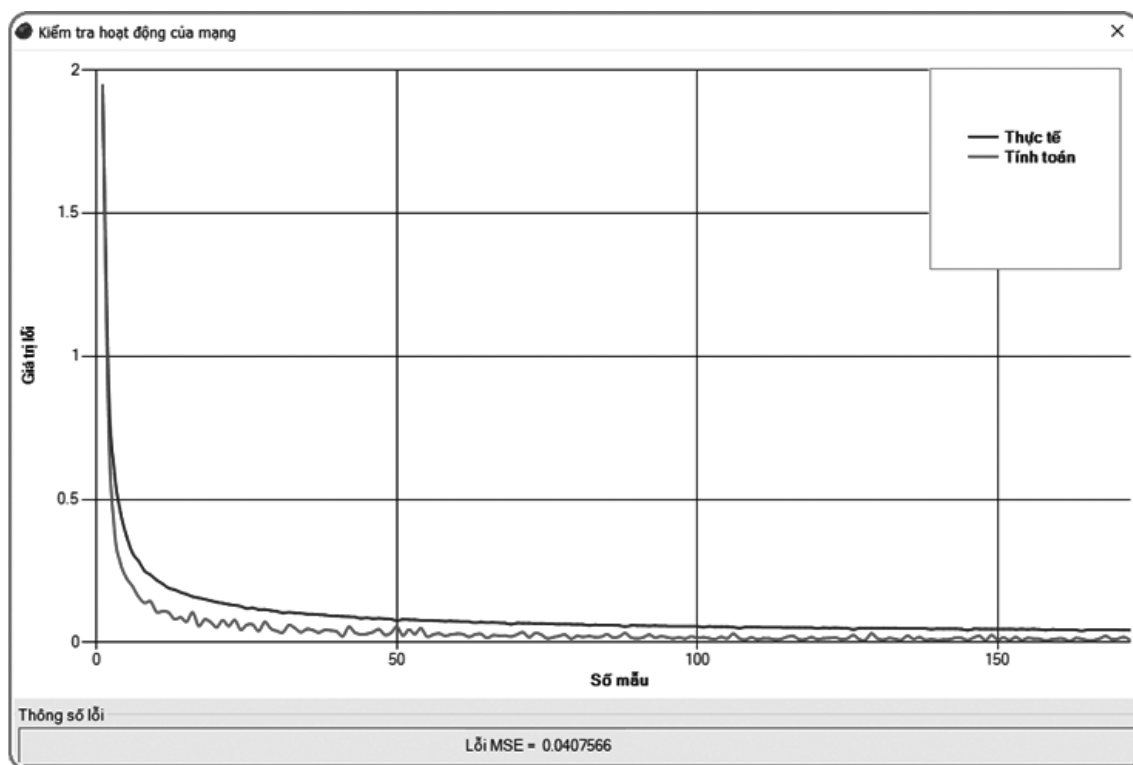
Huấn luyện mạng: lấy 80% mẫu dữ liệu để huấn luyện mạng, kết quả huấn luyện thể hiện ở Hình 4 với sai số MSE là 0,002 qua 2000 vòng huấn luyện.



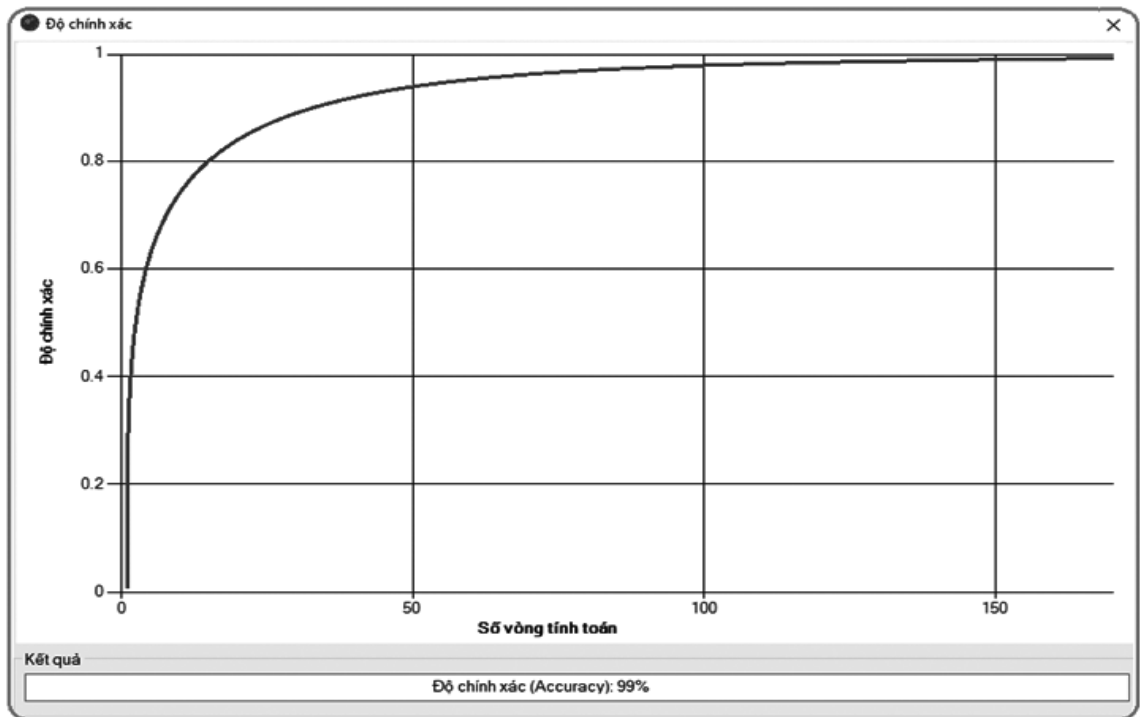
Hình 4. Đồ thị lỗi huấn luyện với mẫu dữ liệu

* Kiểm tra hoạt động của mạng: lấy 20% mẫu dữ liệu còn lại để kiểm tra mạng với các bộ trọng số đã được huấn luyện, kết quả kiểm tra với lỗi MSE

là 0.0407566, kết quả này được thể hiện trên đồ thị kiểm tra hoạt động của mạng trong Hình 5 và độ chính xác của mô hình trong Hình 6.



Hình 5. Đồ thị lỗi kiểm tra



Hình 6. Đồ thị kiểm tra độ chính xác của mô hình

4. 4. Kết quả phát hiện cháy rừng bằng mô hình VGG16

Dữ liệu video thử nghiệm dùng để phát hiện cháy, được lấy trên Youtube (độ phân giải 1902x1080). Kết quả thử nghiệm mô hình được trình bày trong Bảng 1.

Bảng 1. Kết quả thử nghiệm phát hiện cháy từ tệp video

Id	Tên ảnh kết quả	Thời gian	Từ tệp video	Độ chính xác (%)
240	Chay_09-04-2020-11-15-30.jpg	11:15:30	Video_Chay_Rung_1	98,18
241	Chay_09-04-2020-11-15-31.jpg	11:15:31	Video_Chay_Rung_1	97,45
242	Chay_09-04-2020-11-15-32.jpg	11:15:32	Video_Chay_Rung_1	96,95
243	Chay_09-04-2020-11-15-33.jpg	11:15:33	Video_Chay_Rung_1	98,23
244	Chay_09-04-2020-11-15-36.jpg	11:15:36	Video_Chay_Rung_1	98,22
245	Chay_09-04-2020-11-15-37.jpg	11:15:37	Video_Chay_Rung_1	99,63
246	Chay_09-04-2020-11-15-39.jpg	11:15:39	Video_Chay_Rung_1	97,72

5. Kết luận

Mô hình mạng VGG16 với kiến trúc thay đổi, khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ ron truyền thẳng, làm tăng khả năng xử lý và đạt tỷ lệ cao trong phân loại ảnh.

Độ chính xác phát hiện của mạng trong điều kiện lý tưởng đã đạt hoặc vượt qua khả năng phát hiện của con người. Tuy nhiên, do các yếu tố khác nhau như ánh sáng, thời tiết, làm giảm độ chính xác của quá trình nhận dạng và phát hiện cháy. Trong thời gian tới, các tác giả sẽ tập trung vào việc xây dựng và bổ sung các tập thuộc tính để nâng cao độ chính xác của quá trình phát hiện cháy.

Từ những kết quả đã thử nghiệm của mô hình cho thấy, có thể xây dựng các ứng dụng dựa trên phân loại và cảnh báo cháy, như: hệ thống cảnh báo cháy trong các khu công nghiệp, nhà kho, và các hệ thống cảnh báo cháy rừng.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi đề tài: “Nghiên cứu, ứng dụng công nghệ cao nhằm nâng cao hiệu quả quản lý và phát triển bền vững Khu di tích lịch sử Quốc gia đặc biệt Đền Hùng”, mã số: ĐTĐL.CN-35/17.

Tài liệu tham khảo

- [1]. Đoàn Hồng Quang, Lê Hồng Minh, Thái Doãn Nguyên, “Nhận dạng khuôn mặt trong video bằng mạng nơ ron tích chập”. *Tạp chí Khoa học và Công nghệ Việt Nam*, Bộ Khoa học và Công nghệ, **62(1)**, tr. 8-12, 1.2020.
- [2]. Đoàn Hồng Quang, Lê Hồng Minh, “Học chuyển giao và tinh chỉnh mô hình mạng nơ ron học sâu ứng dụng trong nhận dạng khuôn mặt”. *Hội thảo “Ứng dụng công nghệ cao trong phát triển kinh tế - xã hội”*, Viện Ứng dụng Công nghệ, Bộ Khoa học và Công nghệ, tr. 50-61, 2019.
- [3]. Đoàn Hồng Quang, Nguyễn Huy Công, “Phân loại hoa quả bằng mạng nơ ron học sâu”. *Hội thảo “Ứng dụng công nghệ cao trong phát triển kinh tế - xã hội”*, Viện Ứng dụng Công nghệ, Bộ Khoa học và Công nghệ, tr. 132-142, 2019.
- [4]. Đoàn Hồng Quang, Lê Hồng Minh, Chu Anh Tuấn, “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”*, Viện Ứng dụng Công nghệ, Bộ Khoa học và Công nghệ, tr. 70-79, 2015.
- [5]. Đoàn Hồng Quang, Lê Hồng Minh, “Dùng RFNN kết hợp khứ mùa và khứ 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”*, Viện Ứng dụng Công nghệ, Bộ Khoa học và Công nghệ, tr. 126-136, 2014.
- [6]. Nguyễn Quang Hoan, Đoàn Hồng Quang, “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, 2014.
- [7]. Nguyễn Quang Hoan, Dương Thu Trang, Đoàn Hồng Quang, “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, 2018.
- [8]. A. Canziani, A. Paszke and E. Culurciello, “An Analysis of Deep neural network models for Practical applications”, arXiv preprint arXiv:1605.07678, 2016.
- [9]. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell Caffe (2014), “Convolutional Architecture for Fast Feature”, Embedding arXiv:1408.5093.
- [10]. Y. Sun, X. Wang, and X. Tang (2014), “Deep Learning Face Representation by Joint Identification-Verification”, *CoRR*, abs/1406.4773.
- [11]. P. Piccinini, S. Calderara, and R. Cucchiara, “Reliable Smoke Detection System in the Domains of Image Energy and Color”, *15th International Conf. Image Processing*, pp. 1376-1379, 2008.
- [12]. B. U Toreyin, Y. Dedeoglu, and A. E. Cetin, “Wavelet Based Realtime Smoke Detection in Video”. *Signal Processing: Image Comm. EURASIP*, **vol. 20**, pp. 255-256, 2005.
- [13]. D. K. Kim and Y. F. Wang, “Smoke Detection in Video”. *WRI World Congress on Computer Science and Information Engineering*, pp. 759763, 2009.
- [14]. R. Yasmin, “Detection of Smoke Propagation Direction Using Color Video Sequences”, *International J. Soft Computing*, **vol. 4, no. 1**, pp. 45-48, 2009.
- [15]. D. Krstinic, D. Stipanicev, and T. Jakovcevic, “Histogram-Based Smoke Segmentation in Forest Fire Fetection System”, *Inform. Technol. and Control*, **no. 38, vol. 3**, pp.237-244, 2009.

**DETECT FOREST FIRES USING DEEP LEARNING, BASED ON SMOKE AND FIRE
RECEIVED FROM SURVEILLANCE CAMERA**

Abstract:

Deep learning is applied in image recognition, vocal recognition and natural language processing. Nowadays, a variety of applications in recognition are using Deep learning since it can work with an enormous number of parameters, variables and input size while having outstanding performance and precision in comparison with classical methods. In this paper, the authors do research on deep neural network (CNN - Convolutional Neural Network) [1] for detecting fire and smoke by analyzing data from surveillance cameras in order to early detecting and warning forest fire.

Keywords: Deep Learning, Convolutional Neural Network, Forest Fire Detection.