PREDICT STEERING ANGLES IN SELF-DRIVING CARS USING INNOVATION CONVOLUTIONAL NEURAL NETWORK

Luong Thi Thao Hieu*, Pham Thi Thuy
University of Economic and Technical Industries

ARTICLE INFO

Received: 24/02/2022

Revised: 12/5/2022

Published: 16/5/2022

KEYWORDS

Self-driving car
CNN
Deep learning
Steering Angles
VGG16

ABSTRACT

Now a day, artificial intelligence and deep learning have emerged as evidence of the industrial revolution 4.0. Convolutional Neural Network (CNN) is one of the most popular Deep Learning network models, capable of recognizing and classifying images with high accuracy, even better than humans in many cases. This model has been applied to large image processing systems as Facebook, Google or Amazon... In this paper, we focus on studying some advanced CNN network models (VGG-16), based on VGG-16 architecture, we build new model, by increasing network depth, interleaved kernel 3x3, 1x1 increasing number of convolutional blocks, using Exponential Linear Unit (ELU) activation function after each convolution layer. Apply a new model to predict steering angles in autonomous driving based on image data obtained from Udacity self-driving car simulation software. Evaluation, experimentation, and research results show that the steering angle prediction in new model is really effective.

DỰ ĐOÁN GÓC LÁI XE TỰ HÀNH SỬ DỤNG MẠNG NORON TÍCH CHẬP TIÊN TIẾN

Lương Thị Thảo Hiếu*, Phạm Thị Thùy
Trường Đại học Kinh tế Kỹ thuật Công nghiệp

THÔNG TIN BÀI BÁO

Ngày nhận bài: 24/02/2022

Ngày hoàn thiện: 12/5/2022

Ngày đăng: 16/5/2022

TỪ KHÓA

Self driving car
CNN
Deep learning
Steering Angles
VGG16

TÓM TẮT

Những năm gần đây, trí tuệ nhân tạo và cụ thể hơn là học sâu nổi lên như một bằng chứng của cuộc cách mạng 4.0. Mạng noron tích chập (CNN) là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác cao, thậm chí tốt hơn con người trong nhiều trường hợp. Mô hình này đang được ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon… Mục tiêu của bài báo, nghiên cứu lý thuyết về mô hình mạng noron tích chập tiên tiến (VGG-16), dựa trên kiến trúc VGG-16, chúng tôi xây dựng mô hình mới, bằng cách tăng cường độ sâu mạng, xen kẽ kích thước bộ lọc 3x3, 1x1, tăng số lượng khối tích chập, sử dụng hàm kích hoạt ELU sau mỗi lớp tích chập, tinh chỉnh các siêu tham số. Sau đó, thực nghiệm áp dụng mô hình mới vào dự đoán góc lái xe tự hành dựa trên dữ liệu hình ảnh thu được từ phần mềm mô phỏng xe tự lái Udacity. Thực hiện đánh giá, so sánh, kết quả nghiên cứu cho thấy mô hình mới dự đoán góc lái thực sự hiệu quả.

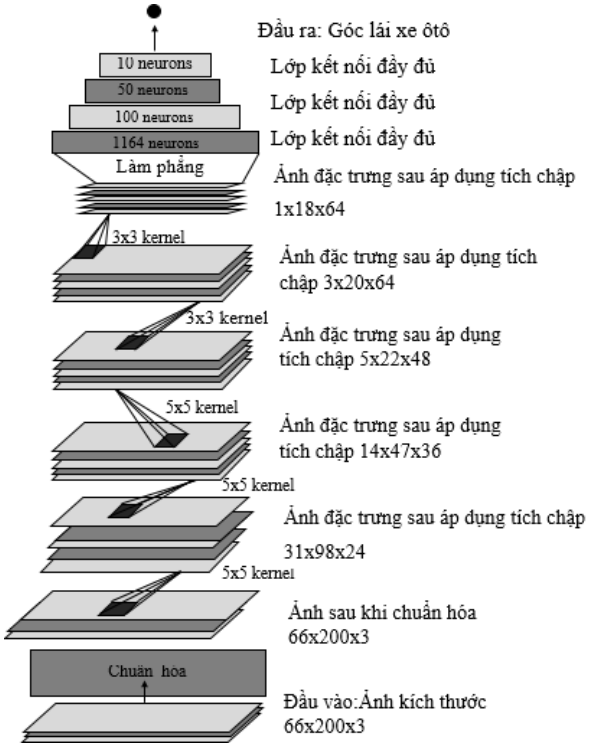
DOI: https://doi.org/10.34238/tnu-jst.5585

* Corresponding author. *Email: ltthieu@uneti.edu.vn*

1. Giới thiệu

Cùng với sự phát triển của công nghệ trí tuệ nhân tạo (AI), các phương tiện xe tự hành (tự lái) tăng lên đáng kể trong những năm gần đây. Một trong các bộ phận quan trọng tích hợp trong xe tự lái là phần mềm AI, chức năng quan trọng của AI dùng để dự đoán góc lái của xe ở đoạn đường phía trước [1]. Để dự đoán góc lái xe tự hành, sử dụng dữ liệu huấn luyện học giám sát, góc lái sẽ được dự đoán bởi một mô hình mạng noron nhân tạo sử dụng đầu vào là các pixel ảnh [2], [3]. Khi đó mô hình học tự động dự đoán góc lái không cần sự can thiệp của con người. Với sự gia tăng của khả năng tính toán cho phép huấn luyện các mạng noron tích chập (CNN) đạt kết quả tốt trong phân lớp hình ảnh [4]. Các thuật toán học sâu CNN ban đầu được sử dụng cho các tác vụ nhận diện với kiến trúc đơn giản như LeNet, Alexnet [5], hiệu năng của các thuật toán học sâu dựa vào kiến trúc thiết kế và các tham số huấn luyện [6]. Trong bài báo này, chúng tôi nghiên cứu mô hình mạng CNN tiên tiến VGG-16 [7], sau đó dựa trên nguyên lý xây dựng VGG-16, xây dựng mô hình có kiến trúc tương tự VGG-16, thực hiện thay đổi độ xen kẽ bộ lọc 3x3, 1x1, tăng cường số lớp tích chập, sử dụng hàm kích hoạt Exponential Linear Units (ELU) thay **Rectified Linear Activation** (ReLu), sử dụng thuật toán tối ưu nadam, thực hiện biến đổi một số siêu tham số tại các lớp phù hợp với dự đoán góc lái của xe tự hành. Thực nghiệm đánh giá mô hình mới trên bộ dữ liệu thu được từ Udacity [8], kết quả cho thấy mô hình mới thực sự hiệu quả. Việc nghiên cứu đem lại kết quả như sau: khai thác hiệu năng mạng CNN, chỉ sử dụng tín hiệu huấn luyện là góc lái, mạng học sâu có thể tự động trích xuất đặc điểm từ các ảnh để học được vị trí của ô tô trên đường và đưa ra góc lái tương ứng.

2. Các nghiên cứu liên quan



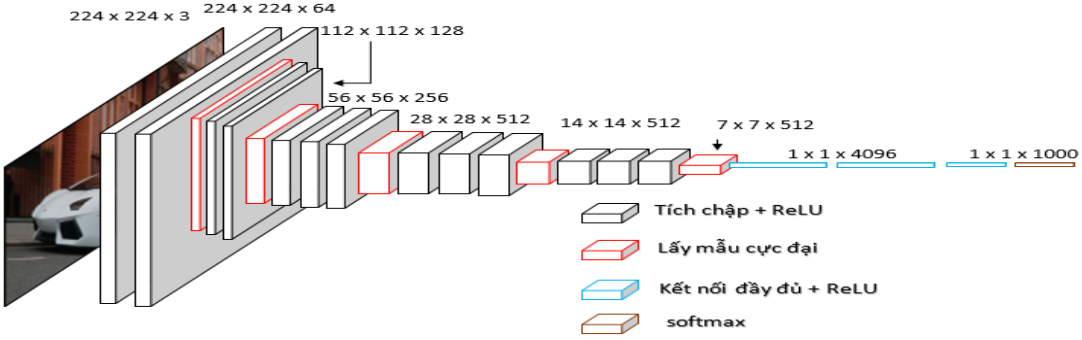
Hình 1. *Kiến trúc NVIDIA*

Thiết kế kiến trúc mạng và tinh chỉnh siêu tham số trong mạng CNN để đạt kết quả tối ưu là vấn đề đang được nhiều nhà nghiên cứu khoa học quan tâm [9]. Năm 2016, nhóm NVIDIA đã nghiên cứu thiết kế mô hình xe tự lái, nhóm huấn luyện sử dụng kiến trúc mạng LeNet, bổ sung hàm kích hoạt, chứa 9 lớp bao gồm 1 lớp chuẩn hóa, 5 lớp tích chập và 3 lớp kết nối đầy đủ [10].

Mô hình của NVIDIA minh họa trong hình 1, sử dụng 252.219 tham số, hàm kích hoạt ReLu, nhân tích chập kích thước 5x5, ánh xạ các pixel ảnh thu được từ camera giữa của ô tô để dự đoán góc lái. Kết quả áp dụng mô hình dự đoán góc lái đưa ra kết quả chính xác ngạc nhiên. Trong những năm gần đây, nhiều kiến trúc CNN được sử dụng bởi các nhà nghiên cứu để dự đoán góc lái xe tự hành [11].

3. Mô hình mạng VGG-16

VGG-16 được phát triển năm 2014, quan điểm xây dựng VGG-16 là một mạng noron sâu hơn sẽ giúp cải thiện độ chính xác của mô hình tốt hơn, cụ thể VGG-16 có độ sâu và số lượng tham số lên tới 138 triệu, đây là một trong những mạng có số lượng tham số lớn nhất. Hình mẫu chung cho các mạng CNN trong các tác vụ học có giám sát trong xử lý ảnh sử dụng các khối VGG dạng $[Conv2D * n + Max\ pooling]$. Một khối VGG gồm một chuỗi các lớp CNN, sau mỗi lớp CNN là một lớp kích hoạt ReLU, tiếp nối bởi một tầng max pooling, để giảm chiều không gian. Cấu trúc VGG-16 gồm 5 khối VGG, 13 lớp tích chập với kích thước 3x3, đầu vào là ảnh kích thước 224x224x3, với 3 là kênh màu R, G, B. Ảnh được truyền qua khối đầu tiên với 2 lớp tích chập, mỗi lớp tích chập chứa 64 bộ lọc kích thước 3x3, theo sau lớp tích chập là hàm kích hoạt ReLU. Sau khi được kích hoạt, đầu ra sẽ được truyền qua lớp max pooling với kích thước cửa sổ 2x2. Thông tin được lan truyền tiếp tục qua khối tích chập thứ 2, sử dụng 128 bộ lọc, kết quả cho ra ảnh kích thước 56x56x128, quá trình được thực hiện tương tự qua các khối tích chập tiếp theo. Sau các khối tích chập là ba fully connected layer, trong đó hai lớp đầu tiên có 4096 noron, và fully connected layer cuối cùng có 1000 noron. Kiến trúc VGG-16 được minh họa trong hình 2.



Hình 2. *Mô hình mạng VGG-16*

3.1. Lớp tích chập

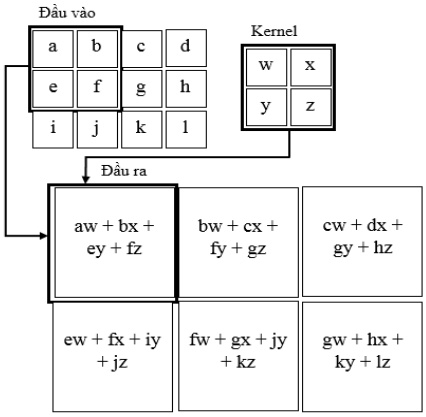
Đây là thành phần quan trọng nhất, nhiệm vụ của lớp tích chập là phát hiện liên kết cục bộ của các đặc điểm trong lớp trước và ánh xạ sang bản đồ đặc trưng. Giá trị điểm ảnh mới được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc có kích thước nhỏ. Về mặt toán học, phép tích chập rời rạc giữa hai hàm f và g được định nghĩa như sau:

$$(f * g)(x) = \sum_{t} f(t)\, g(x + t) \tag{1}$$

Với dữ liệu ảnh hai chiều, sử dụng phép tích chập hai chiều:

$$(K * I)(i, j) = \sum_{m,n} K(m, n)\, I(i + n, j + m) \tag{2}$$

với K là nhân tích chập áp dụng lên ảnh hai chiều I.

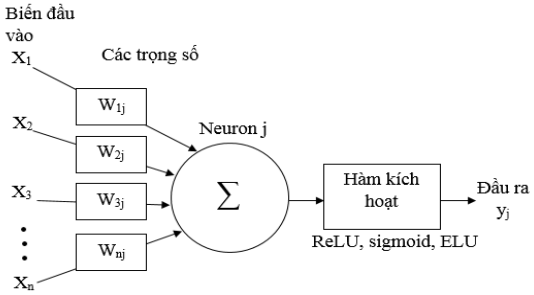


Hình 3. *Bộ lọc tích chập sử dụng trên ma trận điểm ảnh*

Trong hình 3, sử dụng bộ lọc là ma trận kích thước 2x2, nguyên lý của phép tích chập 2 chiều như sau: dịch chuyển nhân tích chập trên toàn bộ ảnh, tại mỗi vị trí tính tích chập giữa nhân và phần hình ảnh đang quét, sau đó nhân (kernel) sẽ dịch chuyển *s* pixel, *s* gọi là bước nhảy (strike).

3.2. Lớp kích hoạt phi tuyến

Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện, cho phép mô hình có thể học các tổ hợp phi tuyến của các tín hiệu đầu vào. Lớp kích hoạt phi tuyến sử dụng các hàm kích hoạt như ReLU, ELU, sigmoid, hoặc tanh… để kích hoạt các trọng số trong các node. Ở mỗi lớp CNN, sau khi được các hàm kích hoạt tác động sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Lớp kế tiếp là kết quả tích chập từ lớp trước đó, từ đó thu được các kết nối cục bộ. Sử dụng ReLU trong CNN có lợi thế không xảy ra lỗi lan truyền ngược, thời gian huấn luyện nhanh hơn nhiều lần so với sử dụng sigmoid, hoặc tanh.



Hình 4. *Áp dụng hàm kích hoạt lên noron j*

Hình 4, mô tả lược đồ áp dụng hàm kích hoạt tác động lên một noron $\sum = (w_j,\ x) + b_j$

Công thức tính toán của hàm ReLU chuyển tất cả các giá trị âm thành giá trị 0:

$$f(x) = \max(0, x) \quad (3)$$

Khi sử dụng ReLU, đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ. Sử dụng ReLU mặc dù được lợi thế tính toán, nhưng có thiếu sót, đó là hiện tượng Dying ReLU (các noron ReLU không hoạt động cho dù cung cấp bất cứ đầu vào nào).

Hàm kích hoạt ELU: Được sử dụng tăng tốc độ học, hàm ELU cho độ chính xác tốt hơn ReLU và hội tụ nhanh hơn. Công thức tổng quát:

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \le 0 \end{cases} \quad (4)$$

Tham số α thường chọn là 1. Hàm ELU liên tục tại mọi điểm, đạo hàm của hàm $f(x)$ bằng 1 với $x > 0$ và $\alpha * e^x$ *với* $x < 0$. Sử dụng ELU không gặp phải vấn đề triệt tiêu và bùng nổ đạo

hàm và cũng không xảy ra hiện tượng noron bất hoạt, hàm hội tụ nhanh dẫn đến thời gian huấn luyện thấp, đồng thời đem lại độ chính xác cao hơn so với ReLU.

3.3. Lớp lấy mẫu

Lớp lấy mẫu (Pooling), được đặt sau lớp tích chập và lớp kích hoạt để giảm kích thước ảnh đầu ra trong khi vẫn giữ được 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 pooling sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh tương tự lớp tích chập và thực hiện phép lấy mẫu bằng cách lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó. Như vậy, với mỗi ảnh đầu vào, qua quá trình lấy mẫu, thu được 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 sau này.

3.4. Lớp kết nối đầy đủ (fully connected layer)

Sau một vài lớp tích chập và lấy mẫu, CNN thường kết thúc bởi lớp kết nối đầy đủ được thiết kế tương tự như trong mạng noron truyền thống, thực chất là một perceptron nhiều lớp. So với mạng noron truyền thống, các ảnh đầu vào của lớp này đã có kích thước giảm rất nhiều, tuy nhiên vẫn đảm bảo giữ được các thông tin quan trọng cho việc nhận dạng.

4. Chuẩn bị dữ liệu

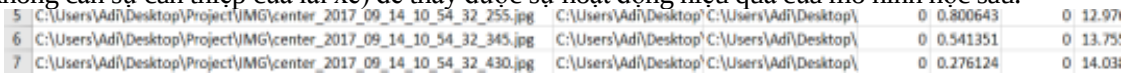
4.1. Phần mềm mô phỏng xe tự lái

Dữ liệu thu được từ phần mềm mã nguồn mở được phát triển bởi Udacity, đây là phần mềm mô phỏng xe tự lái thời gian thực trong các điều kiện giao thông khác nhau, sử dụng cho các cuộc thi "thử thách điều khiển xe tự lái". Thử thách yêu cầu bắt chước hành vi lái xe của con người trên trình mô phỏng với sự trợ giúp của một mô hình mạng noron học sâu. Trình mô phỏng chứa hai làn đường, một làn đường đơn giản (ít cung đường cong và dễ điều khiển) sử dụng cho chế độ huấn luyện và một làn đường phức tạp (có độ dốc, góc cua, góc nhìn bị che khuất) sử dụng cho chế độ tự lái (Hình 5). Dữ liệu được tạo ra từ trình mô phỏng bởi người dùng thực hiện điều khiển xe trên làn đường đơn giản, hình ảnh thu được liên tục từ 3 camera ở giữa, bên phải, bên trái [12].

(a) (b)

Hình 5. *Làn đường huấn luyện (a), làn đường tự lái (b)*

Thực hiện điều khiển xe khoảng 20 phút, thu được 30.000 ảnh. Luồng hình ảnh này được lưu trữ trên ổ đĩa với định dạng file driving_log.csv (Hình 6), cột 1,2,3 chứa đường dẫn đến ảnh thu được từ camera giữa, trái, phải, cột 4 chứa góc lái tương ứng: 0 - đi thẳng, âm - rẽ trái, dương - rẽ phải. Dữ liệu này sau đó được thực nghiệm trong chế độ tự lái (xe tự di chuyển trên địa hình phức tạp mà không cần sự can thiệp của lái xe) để thấy được sự hoạt động hiệu quả của mô hình học sâu.



Hình 6. *File driving_log.csv*

4.2. Một số kỹ thuật tăng cường ảnh

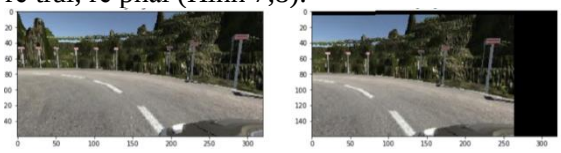
Một mạng CNN có thể xử lý lên tới hàng triệu tham số, việc điều chỉnh các tham số cần hàng triệu các trường hợp dữ liệu huấn luyện. Trong trường hợp dữ liệu huấn luyện quá ít có thể dẫn đến hiện tượng quá khớp, để tránh hiện tượng này chúng tôi sử dụng kỹ thuật tăng cường ảnh. Để dữ liệu tổng quát hơn, cần có hình ảnh ô tô di chuyển trong các điều kiện thời tiết, ánh sáng, đường xá giao thông khác nhau, do đó chúng tôi đã tạo ra hàng nghìn phiên bản mới của ảnh trong thời gian thực bằng cách sử dụng một số kỹ thuật tăng cường như sau:

4.2.1. Dịch chuyển ngang và dọc ảnh

Để mô phỏng ôtô khi di chuyển trong các vị trí khác nhau trên đường, chúng tôi đã dịch chuyển hình ảnh camera theo chiều ngang và thêm độ lệch tương ứng với sự dịch chuyển vào góc lái. Để mô phỏng quá trình lên và xuống dốc, thực hiện dịch chuyển ảnh theo chiều dọc.

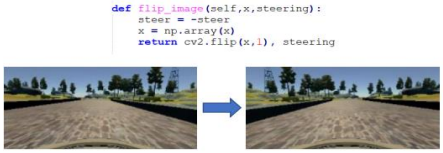
4.2.2. Lật ảnh theo chiều ngang

Hình ảnh được lật ngược theo chiều ngang bằng cách đảo ngược góc lái để mô phỏng ôtô khi rẽ trái, rẽ phải (Hình 7,8).



(a) (b)

Hình 7. *Ảnh gốc: Góc lái = -0,75 (a) Ảnh dịch chuyển ngang: Góc lái = -0,946 (b)*



Hình 8. *Lật ảnh*

4.2.3. Chỉnh độ sáng

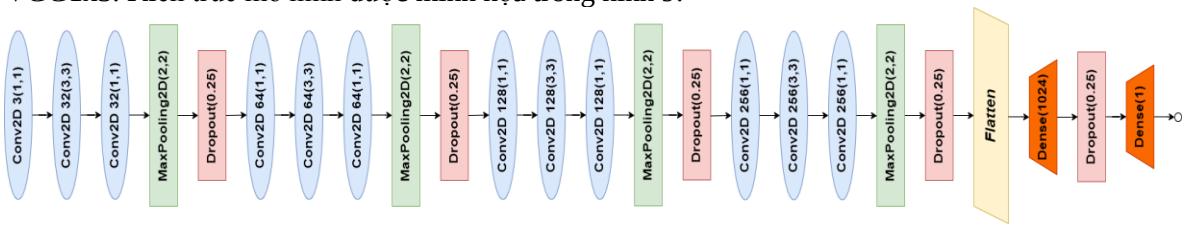
Việc tăng cường độ sáng của ảnh giúp mô phỏng ảnh hưởng của các điều kiện ánh sáng khác nhau như ô tô di chuyển ban ngày hay ban đêm. Ngoài ra còn áp dụng một số kỹ thuật khác như: thêm nhiễu vào ảnh, làm mờ ảnh...

Sau toàn bộ các bước tăng cường, ảnh đầu vào có kích thước 160x230x3, sử dụng lớp lambda trong keras để cắt ảnh theo chiều dọc thành 88x230x3. Sau đó cường độ ảnh được chuẩn hóa nằm giữa (-5) và 5 và được tiếp tục thu nhỏ thành 66x200x3.

5. Xây dựng mô hình mạng noron dựa trên kiến trúc VGG-16

5.1. Xây dựng mô hình

Dựa trên nghiên cứu về kiến trúc xây dựng mạng VGG-16, chúng tôi xây dựng mô hình VGG1x3. Kiến trúc mô hình được minh họa trong hình 9.



Hình 9. *Mô hình VGG1x3*

Mô hình này với số lượng tham số 6.151.405, kiến trúc mạng có 4 khối VGG, tổng 12 lớp tích chập. Ảnh đầu vào được truyền qua mạng, các lớp tích chập được thiết kế để thực hiện trích xuất đặc trưng ảnh, chúng tôi đã thực hiện nhiều thực nghiệm trên các bộ lọc kích thước khác nhau như 1x1, 3x3 và 5x5, và cuối cùng lựa chọn kết hợp xen kẽ tích chập 1x1 và 3x3, sau các lớp tích chập là một lớp Max pooling làm giảm số chiều của ảnh nhưng vẫn giữ được đặc trưng của ảnh giảm bớt số lượng noron và theo sau là lớp Dropout giảm bớt số lượng tham số trùng nhau, cuối

cùng cần thêm một lớp fully connected layer đủ để chuyển đầu ra từ lớp phía trước thành ma trận có số chiều bằng 1, đây chính là dự đoán giá trị góc lái.

5.2. Tinh chỉnh siêu tham số

Sau khi xây dựng xong mô hình, tiếp theo chúng tôi tinh chỉnh các siêu tham số. Tinh chỉnh tham số là kỹ thuật cần thiết để tìm tập tham số phù hợp nhất để xây dựng mô hình từ tập dữ liệu sao cho kết quả dự đoán chính xác. Các tham số cần tinh chỉnh là: hàm kích hoạt, hàm tối ưu, tốc độ học, batch size, epoch. Trong mô hình mới sử dụng hàm kích hoạt ELU giúp giảm thời gian huấn luyện, đồng thời đem lại độ chính xác cao hơn so với ReLu. Để đánh giá mức độ hiệu quả của mô hình, chúng tôi lựa chọn hàm sai số root mean squared error (RMSE). RMSE được tính bằng căn bậc hai của trung bình của sự sai khác giữa kết quả dự đoán và giá trị thực tế. RMSE càng nhỏ tức là sai số (loss) càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất, khi đó giá trị dự đoán gần sát với giá trị thực.

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y_i})^2} \quad (5)$$

Để huấn luyện mô hình cần sử dụng thuật toán tối ưu, quá trình tối ưu hóa có nhiệm vụ thay đổi tốc độ học (*learning rate*) và trọng số của các noron trong mạng để đạt được *loss* tối thiểu. Trình tối ưu hóa sẽ giúp tối ưu các tham số có trong mô hình đồng thời tự điều chỉnh *learning rate* phù hợp giúp mô hình hội tụ được, *learning rate* cao làm cho mô hình học nhanh hơn, tuy nhiên dẫn đến tình trạng có thể bỏ lỡ trường hợp l*oss* tối thiểu, *learning rate* thấp mang lại cơ hội tìm kiếm loss tối thiểu, tuy nhiên cần nhiều tài nguyên bộ nhớ và tốn thời gian. Với mô hình xây dựng, nhóm đã thực hiện nhiều thí nghiệm với nhiều thuật toán tối ưu khác nhau như adam, nadam. Kết quả cho thấy sử dụng nadam cho tốc độ hội tụ nhanh nhất. Chúng tôi đã sử dụng keras để triển khai nadam với $learning\ rate = 1\text{e} - 6, \beta_1 = 0{,}9, \beta_2 = 0{,}999, \varepsilon = 1e - 10$.

Số epoch: Nếu số epoch quá nhỏ dẫn đến hiện tượng underfitting, do mạng không đủ dữ liệu để học, ngược lại nếu số epoch quá lớn dẫn đến hiện tượng overfitting, tức là mô hình có thể dự đoán tốt dữ liệu trên tập train nhưng lại không đoán đúng dữ liệu trên tập validation nên cần tinh chỉnh số epoch mang lại kết quả tối ưu. Trong thực nghiệm với bộ dữ liệu 30.000 ảnh từ xe tự lái chúng tôi chọn thực nghiệm với 40 epoch. Để giúp mô hình học nhanh hơn, chọn batch size (số lượng mẫu huấn luyện cho mỗi lần input) là 2000 ảnh.

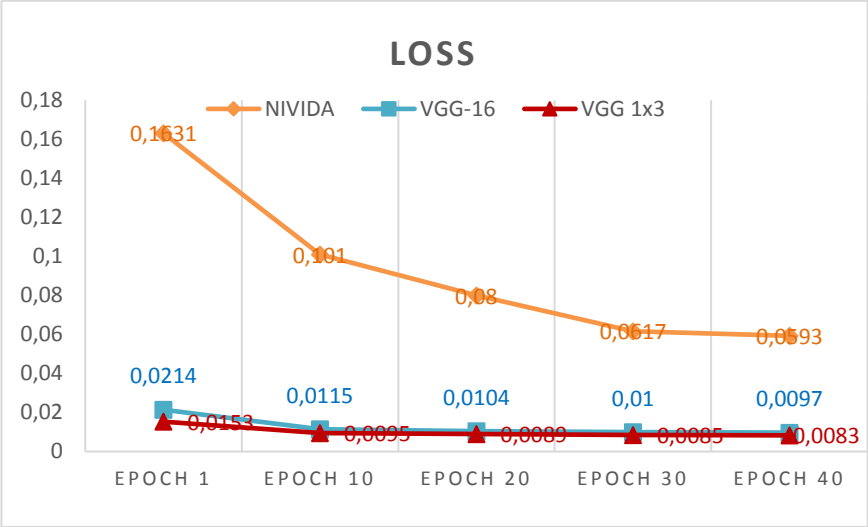
5.3. Thực nghiệm và kết quả

Sau khi xây dựng hoàn chỉnh model, chúng tôi xử lý tệp .csv chứa tên tệp 30.000 ảnh và góc lái tương ứng, dữ liệu được chia ngẫu nhiên 80% ảnh sử dụng để train, 20% sử dụng để validation, thực nghiệm và đánh giá mô hình, sử dụng server có card GPU 64GB. Thông số trong 3 mô hình minh họa qua bảng 1.

Bảng 1. *Thông số của ba mô hình áp dụng trên xe tự lái*

Mô hình NIVIDA	Mô hình VGG-16	Mô hình VGG1x3
• 5 lớp tích chập • Kernel 5x5 • Thuật toán tối ưu: Adam • Hàm kích hoạt ReLu • Sử dụng sub sample • Số tham số: 252.219	• 7 lớp tích chập (3 khối VGG) • Kernek 3x3 • Thuật toán tối ưu: Adam • Hàm kích hoạt ReLu • Sử dụng max pooling • Số tham số: 5.826.445	• 12 lớp tích chập (4 khối VGG) • Kernel 1x1 xen kẽ 3x3 • Thuật toán tối ưu: Nadam • Hàm kích hoạt ELU • Sử dụng drop out,max pooling • Số tham số: 6.151.405

Từ biểu đồ hình 10 cho thấy, mô hình VGG1x3 mang kết quả dự đoán vượt trội so với mô hình NIVIDA và mô hình VGG-16 ban đầu. Giá trị sai số (loss) RMSE đánh giá trên tập train của mô hình VGG1x3 đều thấp hơn VGG-16 ban đầu và thấp hơn rất nhiều so với NIVIDA. Trong trường hợp tốt nhất: Loss của NIVIDA là 0,0593, của VGG-16 ban đầu : 0,0097, của VGG1x3: 0,0083.



Hình 10. *So sánh loss giữa 3 model áp dụng trên xe tự lái*

6. Kết luận

Dự đoán góc lái của xe tự hành luôn là vấn đề thú vị và thu hút nhiều nghiên cứu, một trong những thách thức gặp phải đó là huấn luyện model học sâu để thực hiện dự đoán góc lái, điều khiển ô tô di chuyển trong tình trạng giao thông khác nhau. Ngoài việc thiết kế một model hiệu quả, cần có thêm dữ liệu và thời gian training. Dựa trên các nghiên cứu về VGG-16, chúng tôi đã thiết kế model VGG1x3, áp dụng vào dự đoán góc lái xe tự hành. Mô hình mới sử dụng xen kẽ các lớp tích chập 3x3, 1x1, sử dụng các lớp max pooling để giảm chiều dữ liệu, giúp tối ưu hóa các tham số, sử dụng thuật toán tối ưu nadam để điều chỉnh learning rate, sử dụng các hàm dropout để giảm bớt số lượng các tham số trùng lặp tránh overfiting. Kết hợp với các kỹ thuật tăng cường ảnh, tạo thêm hình ảnh khi đang di chuyển giúp model tổng quát hơn và cho kết quả dự đoán khá tốt khi so sánh với mô hình trước đó. Trong thời gian tới, chúng tôi nghiên cứu một số mô hình học sâu như ResNet, RNN, GAN để điều khiển xe phù hợp với các điều kiện trong thế giới thực với sai số thấp nhất.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] D. Wang, J. Wen, Y. Wang, X. Huang, and F. Pei, "End-to-end self-driving using deep neural network with multi-auxilary tasks," *Automotive Innovation*, vol. II, no. 2, pp. 127-136, 2019.

[2] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, and S. I. Popoola, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, vol. VIII, pp. 163797-163817, 2020.

[3] X. Galorot and Y. Bengio, "Understanding difficulty of traning feedforward neural networks," *In Proc. AISTATS*, vol. IX, pp. 249-256, 2010.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classsification with deep convolutional neural networks," *Communications of the ACM*, vol. I, no. 60, pp. 84-90, 2012.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jacket, "Backpropopagation applied to handwritten zip code recognition," *Neural Computation*, vol. I, no. 4, pp. 541-551, 1989.

[6] A. Bakhshi, N. Norman, Z. Chen, M. Zamani, and S. Chalup, "Fast automatic optimisation of cnn archi-tectures for image classification using genetic algorithm," in IEEE Congress on Evolutionary Computation (CEC) Conf.Proc., Wellington, New Zealand, 2019.

[7] Zisserman, K. Simonyan, and Andrew, "Very deep convolutional network for large-scale image recognition," The 3rd International Conference on Learning Representations(ICLR2015), 2015.

[8] M. V. Smolyakov, A. I. Frolov, V. N. Volkov, and I. V. Stelmashchuk, "Self-driving car steering anglepredicition based on deep neural network an example of carND udacity simulator," in IEEE 12th Int. Conf.on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 2018.

[9] H. Saleem, F. Riaz, L. Mostarda, M. A. Niazi, and A. Rafiqet, "Steering angle prediction techniques forautonomous ground vehicles: A review," *IEEE Access*, vol. IX, pp. 78567-78585, 2021.

[10] M. Bojarski, D. W. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. J. Muller, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," *ArXiv*, vol. abs/1604.07316., 2016.

[11] V. Rausch, A. Hansen, E. Solowjow, C. Liu, and E. Kreuzer, "Learning a deep neural net policy for end-to-end control of autonomous vehicles," in *American Control Conf. (ACC)*, Seattle, USA, 2017, pp. 4914-4919.

[12] S. Lade, P. Shrivastav, S. Waghmare, S. Hon, S. Waghmode, and S. Teli, "Simulation of Self Driving Car Using Deep Learning," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021.