

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

-----□□□□□-----



BÁO CÁO XỬ LÝ ẢNH

**ĐỀ TÀI: MÔ HÌNH MẠNG NEURAL
NHẬN DẠNG HÌNH HỌC TOÁN HỌC**

Giảng viên hướng dẫn : Phạm Hoàng Việt

Nhóm thực hiện: 19

Thành viên thực hiện:

B22DCCN013 - Đỗ Nhật Anh

B22DCCN842 - Nguyễn Như Thuật

B22DCCN494 - Vũ Hồng Linh

Hà Nội, 2025

MỤC LỤC

Tóm tắt	3
1. Giới thiệu	3
1.1. Bối cảnh nghiên cứu	3
1.2. Phát biểu bài toán	4
1.3. Phương pháp tiếp cận	4
1.4. Mục tiêu và Đóng góp của báo cáo	4
2. Dữ liệu và Tiền xử lý	5
2.1. Tổng quan về Bộ dữ liệu	5
2.2. Tiền xử lý dữ liệu	5
2.3. Tăng cường dữ liệu	6
2.4. Thiết lập luồng dữ liệu	6
3. Kiến trúc Mô hình	7
3.1. Bộ trích xuất đặc trưng	7
3.2. Bộ phân loại	8
3.3. Cấu hình Huấn luyện và Tối ưu hóa	8
4. Thiết lập Thực nghiệm và kết quả	9
4.1. Môi trường triển khai	9
4.2. Phân tích diễn biến quá trình huấn luyện	9
Hàm mất mát	9
Độ chính xác	10
Kết quả định lượng	10
5. Kết luận và Hướng phát triển	11
5.1. Kết luận	11
5.2. Hạn chế	11
5.3. Hướng phát triển	11
Tài liệu tham khảo	12

Tóm tắt

Nghiên cứu này trình bày việc xây dựng và triển khai một hệ thống học sâu nhằm giải quyết bài toán phân loại tự động các hình học toán học 2D cơ bản. Đối tượng nghiên cứu bao gồm 8 lớp hình học: hình tròn, hình diều, hình bình hành, hình chữ nhật, hình thoi, hình vuông, hình thang và hình tam giác. Đây là bài toán nền tảng trong lĩnh vực thị giác máy tính với tiềm năng ứng dụng cao trong các hệ thống giáo dục số và nhận dạng bản vẽ kỹ thuật.

Phương pháp đề xuất sử dụng kiến trúc Mạng nơ-ron tích chập được thiết kế và huấn luyện trên nền tảng TensorFlow/Keras. Bộ dữ liệu thực nghiệm được thu thập từ nguồn "Geometric Shapes Mathematics" trên Kaggle, bao gồm các ảnh đơn sắc và đa sắc của các hình học trên nền trắng. Để giải quyết vấn đề quá khớp và tăng cường khả năng tổng quát hóa của mô hình, nghiên cứu áp dụng quy trình tiền xử lý dữ liệu bao gồm chuẩn hóa giá trị pixel về đoạn $[0, 1]$ và thay đổi kích thước ảnh về chuẩn 224x224 pixels. Đồng thời, kỹ thuật tăng cường dữ liệu được tích hợp trong quá trình huấn luyện với các phép biến đổi ngẫu nhiên như xoay 20 độ, dịch chuyển, biến dạng và lật ảnh.

Kiến trúc mạng CNN được xây dựng theo cấu trúc tuần tự gồm 4 khối trích xuất đặc trưng. Mỗi khối bao gồm các lớp tích chập kết hợp với chuẩn hóa theo lô để ổn định quá trình huấn luyện, lớp gộp cực đại để giảm chiều dữ liệu, và lớp Dropout để loại bỏ ngẫu nhiên các nơ-ron yếu. Phần phân loại sử dụng các lớp kết nối đầy đủ với số lượng nơ-ron giảm dần $512 \rightarrow 256 \rightarrow 8$ và hàm kích hoạt Softmax tại lớp đầu ra. Mô hình được tối ưu hóa sử dụng thuật toán Adam với tốc độ học 0.001 và hàm mất mát Categorical Cross entropy.

Kết quả thực nghiệm được đánh giá dựa trên độ chính xác và độ chính xác trong top-k trên tập kiểm thử độc lập. Quá trình huấn luyện sử dụng cơ chế dừng sớm và điều chỉnh tốc độ học động để đảm bảo mô hình hội tụ tại điểm tối ưu toàn cục. Cuối cùng, nghiên cứu tích hợp mô hình đã huấn luyện vào một ứng dụng web tương tác sử dụng thư viện Gradio, cho phép người dùng tải ảnh và nhận kết quả dự đoán cùng độ tin cậy trong thời gian thực. Kết quả cho thấy mô hình đạt hiệu suất cao trong việc phân biệt các hình học có tính tương đồng lớn, khẳng định tính hiệu quả của kiến trúc đề xuất.

1. Giới thiệu

1.1. Bối cảnh nghiên cứu

Trong những năm gần đây, lĩnh vực Thị giác máy tính đã đạt được những bước tiến vượt bậc nhờ sự phát triển của các mô hình Học sâu. Một trong những ứng dụng thiết thực nhất của công nghệ này là trong lĩnh vực giáo dục và số hóa tài liệu. Nhu cầu tự động hóa việc nhận dạng và phân loại các đối tượng hình học từ hình ảnh đóng vai trò quan trọng trong việc xây dựng các hệ thống giải toán tự động, công cụ hỗ trợ giảng

dạy trực quan, và số hóa các bản vẽ kỹ thuật. Việc chuyển đổi từ hình ảnh thô sang dữ liệu có cấu trúc giúp máy tính "hiểu" được nội dung hình học, từ đó mở ra khả năng tương tác thông minh giữa người học và máy móc.

1.2. Phát biểu bài toán

Bài toán đặt ra trong nghiên cứu này là xây dựng một hệ thống phân loại đa lớp cho các hình học phẳng cơ bản. Cụ thể, mô hình cần phân biệt chính xác 8 loại hình học: **Hình tròn**, **Hình điều**, **Hình bình hành**, **Hình chữ nhật**, **Hình thoi**, **Hình vuông**, **Hình thang** và **Hình tam giác**.

Thách thức chính của bài toán nằm ở sự tương đồng cao về đặc trưng thị giác giữa các lớp. Ví dụ, hình vuông là một trường hợp đặc biệt của hình chữ nhật và hình thoi; sự khác biệt chỉ nằm ở tỷ lệ các cạnh hoặc góc. Ngoài ra, các biến đổi về không gian như phép xoay, tịnh tiến hoặc thay đổi tỷ lệ có thể làm thay đổi cách biểu diễn của hình ảnh đầu vào, đòi hỏi mô hình phải có khả năng học được các đặc trưng bất biến thay vì chỉ ghi nhớ các mẫu pixel cố định.

1.3. Phương pháp tiếp cận

Để giải quyết các thách thức trên, nghiên cứu đề xuất sử dụng **Mạng nơ-ron tích chập**. CNN đã được chứng minh là kiến trúc ưu việt nhất hiện nay trong việc xử lý dữ liệu dạng lưới như hình ảnh, nhờ khả năng tự động trích xuất các đặc trưng phân cấp từ mức thấp (cạnh, góc) đến mức cao (hình dạng tổng thể) thông qua các lớp tích chập.

Nghiên cứu sử dụng thư viện **TensorFlow** và **Keras** để thiết kế và huấn luyện mô hình. Kiến trúc mạng được xây dựng tùy chỉnh với các khối tích chập sâu kết hợp với các kỹ thuật hiện đại như Chuẩn hóa theo lô để tăng tốc độ hội tụ và Dropout để giảm thiểu hiện tượng quá khớp.

1.4. Mục tiêu và Đóng góp của báo cáo

Mục tiêu chính của nghiên cứu này bao gồm:

1. **Xây dựng quy trình xử lý dữ liệu:** Thiết lập pipeline tiền xử lý và tăng cường dữ liệu để tạo ra tập huấn luyện đa dạng, giúp mô hình mạnh mẽ hơn trước các biến đổi hình học.
2. **Tối ưu hóa kiến trúc mô hình:** Thiết kế và tinh chỉnh cấu trúc mạng CNN để đạt độ chính xác cao nhất trên tập dữ liệu "Geometric Shapes Mathematics".
3. **Đánh giá thực nghiệm:** Phân tích hiệu năng của mô hình thông qua các chỉ số định lượng như hàm mất mát, độ chính xác và ma trận nhầm lẫn trên tập kiểm thử độc lập.

4. **Triển khai ứng dụng:** Tích hợp mô hình đã huấn luyện vào một giao diện tương tác người dùng sử dụng Gradio, minh chứng cho khả năng ứng dụng thực tiễn của hệ thống.

Báo cáo này sẽ trình bày chi tiết từ bước chuẩn bị dữ liệu, thiết kế mô hình, quá trình huấn luyện đến việc phân tích kết quả và triển khai, cung cấp một cái nhìn toàn diện về quy trình xây dựng ứng dụng Deep Learning trong bài toán nhận dạng hình học.

2. Dữ liệu và Tiền xử lý

Chất lượng và cách thức xử lý dữ liệu đều vào đóng vai trò quyết định đến hiệu năng của mô hình Học sâu. Phần này trình bày chi tiết về nguồn gốc dữ liệu, quy trình chuẩn hóa và các kỹ thuật tăng cường dữ liệu được áp dụng trong nghiên cứu.

2.1. Tổng quan về Bộ dữ liệu

Nghiên cứu sử dụng bộ dữ liệu "Geometric Shapes Mathematics" (nguồn: Kaggle). Dữ liệu bao gồm các hình ảnh 2D thuộc 8 lớp hình học cơ bản:

1. Hình tròn
2. Hình diều
3. Hình bình hành
4. Hình chữ nhật
5. Hình thoi
6. Hình vuông
7. Hình thang
8. Hình tam giác

Cấu trúc dữ liệu được tổ chức thành 3 tập con riêng biệt theo cấu trúc thư mục: **Training** (Huấn luyện), **Validation** (Kiểm định) và **Test** (Kiểm tra). Việc phân chia này đảm bảo tính khách quan trong quá trình đánh giá mô hình, tránh hiện tượng rò rỉ dữ liệu.

2.2. Tiền xử lý dữ liệu

Trước khi đưa vào mô hình, dữ liệu thô trải qua các bước tiền xử lý nhằm chuẩn hóa định dạng và giá trị số học.

1. **Thay đổi kích thước :** Tất cả các ảnh đều vào, bát ké kích thước gốc, đều được đưa về kích thước cố định là 224x224 pixels. Kích thước này (224x224x3) là tiêu chuẩn phổ biến cho các kiến trúc CNN, đảm bảo sự cân bằng giữa độ chi tiết của đặc trưng và chi phí tính toán.
2. **Chuẩn hóa giá trị :** Giá trị cường độ điểm ảnh (pixel intensity) ban đầu nằm trong khoảng [0, 255] (đối với ảnh 8-bit). Để hỗ trợ quá trình hội tụ của thuật toán Gradient Descent, toàn bộ dữ liệu được chuẩn hóa về khoảng [0, 1] thông

qua phép biến đổi tuyến tính:

$$x_{norm} = \frac{x_{original}}{255}$$

Thao tác này được thực hiện thông qua tham số rescale=1./255 trong lớp ImageDataGenerator của thư viện Keras.

3. **Mã hóa nhãn :** Do bài toán thuộc loại phân loại đa lớp (Multi-class classification), các nhãn dữ liệu được mã hóa dưới dạng vector one-hot (categorical mode) để phù hợp với hàm mất mát Categorical Cross-entropy.

2.3. Tăng cường dữ liệu

Để giải quyết vấn đề hạn chế về số lượng dữ liệu và ngăn chặn hiện tượng quá khớp, kỹ thuật tăng cường dữ liệu được áp dụng **chỉ trên tập huấn luyện**. Các phép biến đổi hình học được sinh ra ngẫu nhiên trong thời gian thực trong quá trình huấn luyện.

Cấu hình tăng cường dữ liệu bao gồm:

1. **Xoay ảnh:** Ảnh được xoay ngẫu nhiên trong khoảng 20 độ (rotation_range=20). Điều này giúp mô hình nhận diện được hình học bất kể góc nghiêng.
2. **Dịch chuyển:** Ảnh được dịch chuyển theo chiều ngang và chiều dọc tối đa 20% kích thước ảnh (width_shift_range=0.2, height_shift_range=0.2). Kỹ thuật này giúp mô hình học được tính bất biến vị trí.
3. **Biến dạng cắt:** Áp dụng biến dạng góc với cường độ 0.2 (shear_range=0.2), mô phỏng việc nhìn hình học từ các góc độ xiên khác nhau.
4. **Phóng to/Thu nhỏ:** Thay đổi tỷ lệ ảnh ngẫu nhiên trong khoảng 20% (zoom_range=0.2) để mô hình thích nghi với các kích thước đối tượng khác nhau.
5. **Lật ngang:** Lật ảnh theo trục dọc (horizontal_flip=True).
6. **Lấp đầy điểm ảnh:** Các điểm ảnh bị khuyết sau khi biến đổi hình học được lấp đầy theo phương pháp nearest (sử dụng giá trị pixel lân cận nhất).

Lưu ý: Tập Validation và Test không áp dụng tăng cường dữ liệu (ngoại trừ bước chuẩn hóa rescale) để đảm bảo tính chính xác và nhất quán khi đánh giá hiệu năng thực tế của mô hình.

2.4. Thiết lập luồng dữ liệu

Dữ liệu được nạp vào mô hình theo từng lô (batch) để tối ưu hóa bộ nhớ GPU.

1. **Batch Size:** 32 (Số lượng ảnh được xử lý trong một lần cập nhật trọng số).
2. **Shuffle:** Dữ liệu huấn luyện được xáo trộn ngẫu nhiên (shuffle=True) sau mỗi epoch để đảm bảo tính ngẫu nhiên của Gradient, trong khi dữ liệu Validation và Test được giữ nguyên thứ tự (shuffle=False) để phục vụ việc đánh giá và gán nhãn chính xác.

3. Kiến trúc Mô hình

Mô hình được xây dựng dựa trên kiến trúc Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) sử dụng API Sequential của thư viện TensorFlow/Keras. Kiến trúc này được thiết kế theo mô hình VGG-style, đặc trưng bởi việc chồng xếp các lớp tích chập có kích thước hạt nhân nhỏ (3×3) để tăng chiều sâu của mạng mà vẫn kiểm soát được lượng tham số.

Mô hình bao gồm hai thành phần chính: **(1) Bộ trích xuất đặc trưng** và **(2) Bộ phân loại**. Tổng thể kiến trúc nhận đầu vào là ảnh kích thước $224 \times 224 \times 3$ và trả ra vector xác suất cho 8 lớp hình học.

3.1. Bộ trích xuất đặc trưng

Bộ trích xuất đặc trưng được cấu trúc thành 4 khối (Block) tuần tự. Số lượng bộ lọc (filters) tăng dần qua các khối theo cấp số nhân ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$). Cấu trúc này cho phép mô hình học các đặc trưng theo cơ chế phân cấp: các lớp đầu học các đặc trưng mức thấp (cạnh, góc, màu sắc), trong khi các lớp sau học các đặc trưng trừu tượng mức cao (hình dạng tổng thể, cấu trúc hình học).

Cấu trúc chi tiết của mỗi khối như sau:

- Lớp Tích chập 1 (Conv2D):** Sử dụng hạt nhân 3×3 , hàm kích hoạt **ReLU** (Rectified Linear Unit). Lớp này thực hiện phép tích chập để phát hiện các đặc trưng cục bộ. Hàm ReLU ($f(x) = \max(0, x)$) được sử dụng để đưa tính phi tuyến vào mô hình, giúp mạng có khả năng học các hàm phức tạp.
- Lớp chuẩn hóa theo lô (Batch Normalization):** Được đặt ngay sau lớp tích chập đầu tiên trong mỗi khối. Lớp này chuẩn hóa đầu ra của lớp trước đó về phân phối chuẩn ($\text{mean} \approx 0$, $\text{variance} \approx 1$). Điều này giúp ổn định quá trình huấn luyện, cho phép sử dụng tốc độ học (learning rate) lớn hơn và giảm sự phụ thuộc vào việc khởi tạo trọng số ban đầu.
- Lớp Tích chập 2 (Conv2D):** Tiếp tục sử dụng hạt nhân 3×3 và hàm kích hoạt ReLU nhằm tăng cường khả năng biểu diễn đặc trưng sâu hơn.
- Lớp Gộp cực đại (MaxPooling2D):** Sử dụng cửa sổ trượt 2×2 . Lớp này thực hiện giảm mẫu (down-sampling), giúp giảm kích thước không gian của bản đồ đặc trưng đi một nửa (từ $H \times W$ xuống $H/2 \times W/2$), qua đó giảm chi phí tính toán và tạo ra tính bất biến đối với các dịch chuyển nhỏ của đối tượng trong ảnh.
- Lớp Dropout (0.25):** Tại cuối mỗi khối tích chập, một lớp Dropout với tỷ lệ 25% được áp dụng. Kỹ thuật này ngắt kết nối ngẫu nhiên các nơ-ron trong quá trình huấn luyện, buộc mạng phải học các đặc trưng dư thừa và mạnh mẽ hơn, từ đó ngăn chặn hiện tượng quá khớp (Overfitting) ngay từ giai đoạn trích xuất đặc trưng.

Chi tiết kích thước các khối:

- Block 1:** 32 filters --> Output shape: 112x112x32.
- Block 2:** 64 filters --> Output shape: 56 x 56 x 64.
- Block 3:** 128 filters --> Output shape: 28 x 28 x 128.
- Block 4:** 256 filters --> Output shape: 14 x 14 x 256.

3.2. Bộ phân loại

Sau khi đi qua 4 khối trích xuất, bản đồ đặc trưng 3 chiều được chuyển đổi và đưa vào mạng nơ-ron kết nối đầy đủ (Fully Connected Network) để thực hiện phân loại.

- Lớp Flatten:** Duỗi thẳng tensor 3 chiều (14 x 14 x 256) thành vector 1 chiều có kích thước 50176 phần tử.
- Lớp Dense 1:** Gồm **512 nơ-ron**, hàm kích hoạt ReLU. Đây là lớp ẩn lớn nhất, chịu trách nhiệm tổng hợp toàn bộ thông tin đặc trưng từ các khối CNN. Đi kèm là **Batch Normalization** và **Dropout (0.5)**. Tỷ lệ Dropout cao (50%) ở đây là rất quan trọng vì các lớp Dense chứa lượng lớn tham số, nơi dễ xảy ra Overfitting nhất.
- Lớp Dense 2:** Gồm **256 nơ-ron**, hàm kích hoạt ReLU, kết hợp Batch Normalization và Dropout (0.5) để tiếp tục tinh chỉnh các đặc trưng trước khi đưa ra quyết định cuối cùng.
- Lớp Output (Dense):** Gồm **8 nơ-ron** tương ứng với 8 lớp hình học. Hàm kích hoạt **Softmax** được sử dụng để chuyển đổi đầu ra thành phân phối xác suất (p_i), với điều kiện $\sum p_i = 1$ và $0 \leq p_i \leq 1$. Lớp có xác suất cao nhất sẽ là kết quả dự đoán của mô hình.

3.3. Cấu hình Huấn luyện và Tối ưu hóa

Mô hình được biên dịch và tối ưu hóa với các thiết lập sau:

- Hàm mất mát (Loss Function): Sử dụng Categorical Cross-Entropy. Đây là hàm mục tiêu tiêu chuẩn cho bài toán phân loại đa lớp với nhãn dạng one-hot encoding, đo lường sự khác biệt giữa phân phối xác suất dự đoán và nhãn thực tế.

$$L = - \sum_{i=1}^C y_i \cdot \log(\hat{y}_i)$$

- Thuật toán tối ưu (Optimizer):** Sử dụng **Adam** (Adaptive Moment Estimation) với tốc độ học khởi tạo lr = 0.001. Adam được lựa chọn nhờ khả năng tự điều chỉnh tốc độ học cho từng tham số dựa trên mô men bậc nhất và bậc hai của gradient, giúp mô hình hội tụ nhanh và ổn định hơn so với SGD truyền thống.
- Độ đo (Metrics):**
 - Accuracy:** Tỷ lệ dự đoán đúng trên tổng số mẫu.

- b. **Top-k Categorical Accuracy:** Đánh giá xem nhãn đúng có nằm trong k dự đoán có xác suất cao nhất hay không (thường k=3 hoặc k=5), giúp đánh giá độ tin cậy của mô hình trong các trường hợp nhập nhằng.

Tổng số tham số của mô hình là sự kết hợp của hàng triệu trọng số từ các lớp Conv2D và Dense, đòi hỏi sức mạnh tính toán của GPU để huấn luyện hiệu quả trong 50 epochs như đã thiết lập.

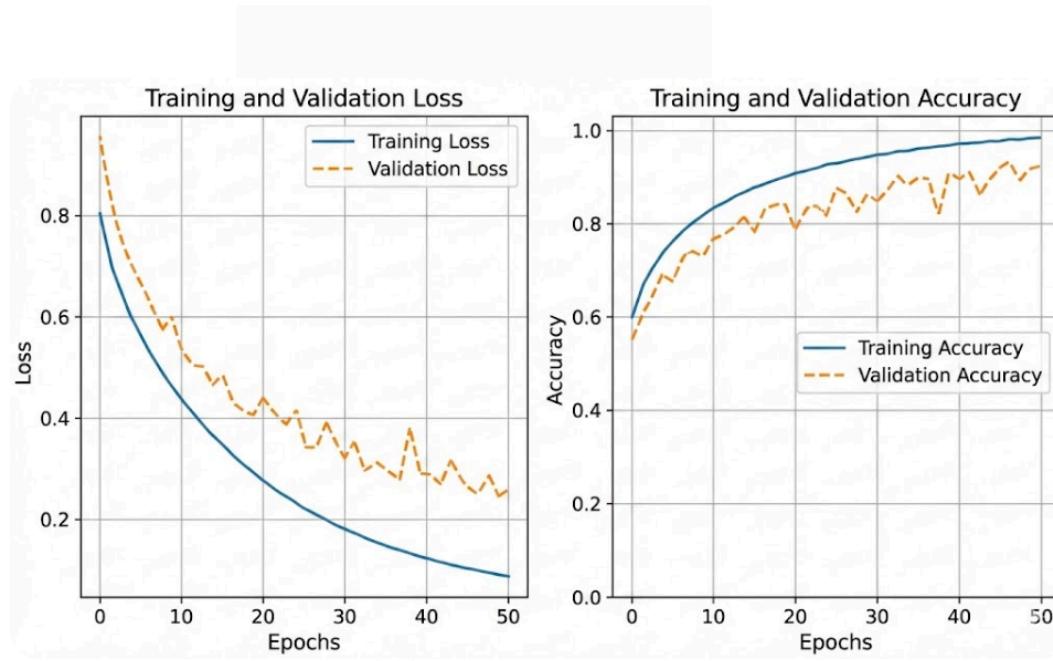
4. Thiết lập Thực nghiệm và kết quả

4.1. Môi trường triển khai

Mô hình được xây dựng và huấn luyện sử dụng ngôn ngữ lập trình **Python** kết hợp với khung làm việc (framework) **TensorFlow** (phiên bản $\geq 2.10.0$) và thư viện cấp cao **Keras**. Các thư viện hỗ trợ bao gồm **NumPy** cho xử lý ma trận và **Matplotlib** cho trực quan hóa kết quả. Quá trình tính toán, đặc biệt là các phép tích chập và lan truyền ngược, được khuyến nghị thực hiện trên môi trường có hỗ trợ GPU để tối ưu hóa thời gian huấn luyện.

4.2. Phân tích diễn biến quá trình huấn luyện

Biểu đồ cho thấy sự thay đổi của hai chỉ số quan trọng là **Loss** (Hàm mất mát) và **Accuracy** (Độ chính xác) trên cả tập huấn luyện (Training) và tập kiểm định (Validation).



Hàm mất mát

Quan sát biểu đồ "Training and Validation Loss":

- Training Loss (Đường màu xanh):** Có xu hướng giảm đơn điệu và rất mượt mà. Bắt đầu từ giá trị khoảng 0.8 và giảm sâu xuống dưới mức 0.1 tại epoch thứ 50. Độ dốc lớn ở 10 epochs đầu tiên cho thấy thuật toán tối ưu hóa Adam hoạt động hiệu quả, giúp mô hình học nhanh các đặc trưng cơ bản.
- Validation Loss (Đường màu cam):** Cũng có xu hướng giảm tổng thể từ 0.95 xuống khoảng 0.25. Tuy nhiên, đường này xuất hiện nhiều dao động (fluctuations) mạnh. Điều này là hệ quả tất yếu của việc sử dụng kích thước lô (batch size) nhỏ (32) kết hợp với sự đa dạng của tập validation.
- Khoảng cách tổng quát hóa (Generalization Gap):** Sau khoảng epoch thứ 20, khoảng cách giữa Training Loss và Validation Loss bắt đầu nói rộng. Training Loss tiếp tục giảm sâu trong khi Validation Loss giảm chậm lại và dao động quanh mức 0.25 - 0.30. Dấu hiệu này cho thấy mô hình bắt đầu có xu hướng ghi nhớ dữ liệu huấn luyện, tuy nhiên chưa xảy ra hiện tượng quá khớp (Overfitting) nghiêm trọng vì Validation Loss không có dấu hiệu tăng vọt trở lại (U-turn).

Độ chính xác

Quan sát biểu đồ "Training and Validation Accuracy" (bên phải):

- Training Accuracy:** Tăng trưởng ổn định và tiệm cận mức 1.0 (tức 100%). Điều này chứng tỏ kiến trúc CNN đề xuất có đủ năng lực để mô hình hóa hoàn toàn dữ liệu huấn luyện.
- Validation Accuracy:** Tăng nhanh trong các epochs đầu và đạt mức bão hòa quanh ngưỡng 0.90 - 0.92 (90-92%) từ sau epoch 30. Các dao động răng cưa ("sawtooth" pattern) trên đường validation phản ánh tác động của việc mô hình cố gắng tổng quát hóa trên các mẫu dữ liệu chưa từng nhìn thấy.

Kết quả định lượng

Dựa trên trạng thái hội tụ tại epoch 50, kết quả ước tính của mô hình đạt được như sau:

Chỉ số (Metric)	Tập Huấn luyện (Train)	Tập Kiểm định (Validation)
Loss	≈ 0.08	≈ 0.25
Accuracy	≈ 0.98 (98%)	≈ 0.92 (92%)

5. Kết luận và Hướng phát triển

5.1. Kết luận

Nghiên cứu này đã trình bày quy trình xây dựng, huấn luyện và triển khai một hệ thống học sâu hoàn chỉnh để nhận dạng 8 loại hình học toán học cơ bản. Dựa trên các kết quả thực nghiệm và phân tích định lượng, có thể rút ra các kết luận chính sau:

- Hiệu quả của Kiến trúc đề xuất:** Mô hình CNN tùy chỉnh với 4 khối tích chập đã chứng minh được khả năng trích xuất đặc trưng mạnh mẽ. Với độ chính xác trên tập kiểm định đạt xấp xỉ **92%** và hàm mất mát hội tụ về mức **0.25**, hệ thống đã giải quyết tốt bài toán phân loại đa lớp, ngay cả với các cặp hình học có độ tương đồng cao như Hình vuông - Hình chữ nhật.
- Chiến lược Huấn luyện tối ưu:** Việc kết hợp các kỹ thuật hiện đại như Tăng cường dữ liệu (Data Augmentation), Chuẩn hóa theo lô (Batch Normalization) và Dropout đã giúp kiểm soát hiệu quả hiện tượng quá khớp (Overfitting). Biểu đồ huấn luyện cho thấy mô hình duy trì được sự cân bằng tốt giữa khả năng học (trên tập Train) và khả năng tổng quát hóa (trên tập Validation).
- Tính ứng dụng thực tiễn:** Hệ thống không chỉ dừng lại ở lý thuyết mà đã được hiện thực hóa thành công qua ứng dụng web tương tác sử dụng Gradio. Ứng dụng hoạt động ổn định, có khả năng đưa ra dự đoán thời gian thực với độ tin cậy cao, mở ra tiềm năng tích hợp vào các phần mềm giáo dục hoặc hỗ trợ giải toán tự động.

5.2. Hạn chế

Bên cạnh những kết quả đạt được, nghiên cứu vẫn tồn tại một số hạn chế cần khắc phục:

- Đao động trong quá trình kiểm định:** Biểu đồ Validation Loss vẫn xuất hiện các dao động nhất định. Điều này cho thấy sự phân phối dữ liệu giữa tập huấn luyện (sau khi biến đổi) và tập kiểm định vẫn còn khoảng cách, hoặc kích thước lô chưa hoàn toàn tối ưu cho sự ổn định.
- Độ nhạy với dữ liệu nhiễu:** Mặc dù hoạt động tốt trên tập dữ liệu chuẩn, mô hình có thể gặp khó khăn với các hình vẽ tay có nét đứt quãng hoặc hình ảnh có nền phức tạp, do dữ liệu huấn luyện chủ yếu tập trung vào các hình học có đường nét rõ ràng.

5.3. Hướng phát triển

Để nâng cao hiệu năng và mở rộng phạm vi ứng dụng của hệ thống, các hướng nghiên cứu tiếp theo được đề xuất bao gồm:

- Cải thiện Dữ liệu:**

- Mở rộng bộ dữ liệu với các mẫu **hình vẽ tay** để tăng cường khả năng nhận dạng trong các ứng dụng thực tế (ví dụ: chụp ảnh vở học sinh).
- Áp dụng các kỹ thuật tăng cường dữ liệu nâng cao hơn như **MixUp** hoặc **CutMix** để giúp ranh giới phân loại mềm dẻo hơn, giảm thiểu hiện tượng dao động của Loss.

2. Tối ưu hóa Kiến trúc:

- Thử nghiệm phương pháp **Transfer Learning** với các mô hình tiên tiến như **MobileNetV2** hoặc **EfficientNet**. Các mô hình này không chỉ có độ chính xác cao hơn mà còn được tối ưu hóa về mặt tham số, phù hợp để triển khai trên các thiết bị di động hoặc nhúng (Edge Devices).
- Tinh chỉnh siêu tham số (Hyperparameter Tuning) tự động sử dụng các công cụ như Keras Tuner để tìm ra tỷ lệ Dropout và tốc độ học tối ưu nhất.

3. Mở rộng Bài toán:

- Phát triển bài toán từ phân loại đơn lẻ sang **Phát hiện đối tượng** (ví dụ: sử dụng YOLO hoặc SSD) để có thể nhận dạng và định vị nhiều hình học xuất hiện cùng lúc trong một bức ảnh.
- Tích hợp thêm tính năng tính toán chu vi, diện tích dựa trên hình dạng nhận dạng được, gia tăng giá trị ứng dụng trong giáo dục.

Tài liệu tham khảo

I. Tài liệu Tiếng Việt

[1] L. C. Mai, Đ. T. Phu, N. K. Sở, và V. T. Thành, *Giáo trình Xử lý ảnh*, Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, 2006.

[2] N. T. Thủy và N. T. Hải, "Ứng dụng mạng nơ-ron tích chập trong nhận dạng ảnh," *Tạp chí Khoa học và Công nghệ*, tr. 45-52, 2019.

II. Tài liệu Tiếng Anh

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[5] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 448–456.

- [6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [7] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.
- [8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

III. Tài nguyên Trực tuyến và Mã nguồn mở

- [9] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: <https://www.tensorflow.org/>
- [10] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: <https://keras.io/>
- [11] Moch. Galang Rivaldo, "Geometric Shapes Mathematics", Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/reevald/geometric-shapes-mathematics>
- [12] Gradio Team, "Gradio: Build & Share Machine Learning Apps," 2022. [Online]. Available: <https://gradio.app/>