

BÁO CÁO XỬ LÝ ẢNH

MÔ HÌNH MẠNG NEURAL NHẬN DẠNG HÌNH HỌC TOÁN
HỌC

Nhóm thực hiện: 19

Thành viên: Đỗ Nhật Anh, Nguyễn Như Thuật, Vũ Hồng Linh

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 điều, hình bình hành, hình chữ nhật, hình thoi, hình vuông, hình thang và hình tam giác.

Phương pháp

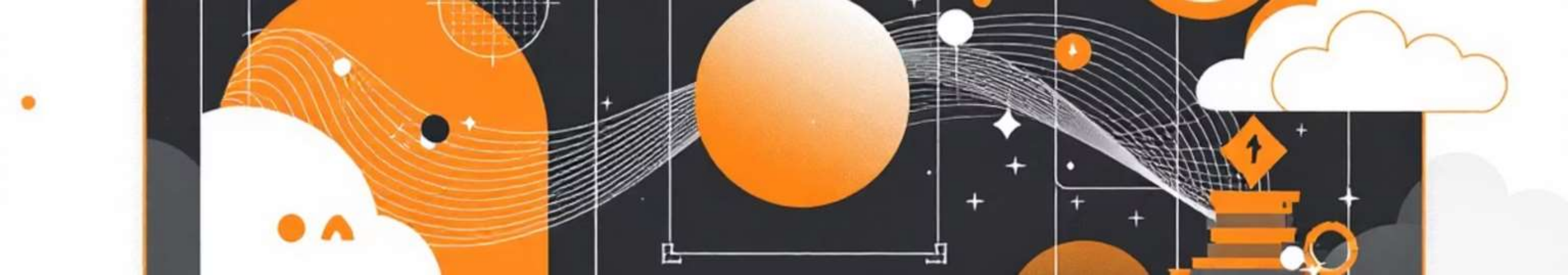
Sử dụng kiến trúc Mạng nơ-ron tích chập (CNN) trên nền tảng TensorFlow/Keras.

Dữ liệu

Thu thập từ "Geometric Shapes Mathematics" trên Kaggle, ảnh đơn sắc và đa sắc trên nền trắng.

Tiền xử lý

Chuẩn hóa pixel $[0, 1]$, thay đổi kích thước 224×224 , tăng cường dữ liệu (xoay, dịch chuyển, biến dạng, lật).



Kiến trúc Mô hình & Kết quả



Kiến trúc Mạng CNN

Gồm 4 khối trích xuất đặc trưng, lớp tích chập, chuẩn hóa theo lô, gộp cực đại, Dropout.



Tối ưu hóa

Thuật toán Adam (tốc độ học 0.001), hàm mất mát Categorical Crossentropy.



Kết quả

Mô hình đạt hiệu suất cao, phân biệt tốt các hình học tương đồng.

Nghiên cứu tích hợp mô hình đã huấn luyện vào ứng dụng cho phép người dùng tải ảnh và nhận kết quả dự đoán cùng độ tin cậy trong thời gian thực.

1. Giới thiệu

1.1. Bối cảnh

Thị giác máy tính và Học sâu đã đạt nhiều tiến bộ, đặc biệt trong giáo dục và số hóa tài liệu. Nhu cầu tự động nhận dạng và phân loại hình học từ hình ảnh rất quan trọng để xây dựng hệ thống giải toán tự động, công cụ giảng dạy trực quan và số hóa bản vẽ kỹ thuật.



1.2. Phát biểu bài toán

Xây dựng hệ thống phân loại đa lớp cho 8 hình học phẳng cơ bản: Hình tròn, Hình điều, Hình bình hành, Hình chữ nhật, Hình thoi, Hình vuông, Hình thang và Hình tam giác.

Thách thức: sự tương đồng cao giữa các lớp và biến đổi không gian (xoay, tịnh tiến, tỷ lệ).



Phương pháp tiếp cận & Mục tiêu

1.3. Phương pháp tiếp cận

Sử dụng Mạng nơ-ron tích chập (CNN) vì khả năng trích xuất đặc trưng tự động từ hình ảnh.

Sử dụng TensorFlow và Keras để thiết kế và huấn luyện mô hình với các kỹ thuật hiện đại như Chuẩn hóa theo lô và Dropout.



1.4. Mục tiêu và Đóng góp

- Xây dựng quy trình xử lý dữ liệu và tăng cường dữ liệu.
- Tối ưu hóa kiến trúc mô hình CNN để đạt độ chính xác cao.
- Đánh giá thực nghiệm hiệu năng mô hình.
- Triển khai ứng dụng tương tác thực tiễn.





2. Dữ liệu và Tiền xử lý



Tổng quan về Bộ dữ liệu

"Geometric Shapes Mathematics" (Kaggle) gồm 8 lớp hình học: Circle, Kite, Parallelogram, Rectangle, Rhombus, Square, Trapezoid, Triangle.

Dữ liệu chia thành 3 tập: Training, Validation, Test để đánh giá khách quan.



Tiền xử lý dữ liệu

- Thay đổi kích thước: 224x224 pixels.
- Chuẩn hóa giá trị: pixel [0, 255] về [0, 1]
$$x_{norm} = \frac{x_{original}}{255}$$
- Mã hóa nhãn: Vector one-hot cho phân loại đa lớp.



Tăng cường dữ liệu

- Xoay ảnh: $\pm 20^\circ$.
- Dịch chuyển: $\pm 20\%$ kích thước ảnh.
- Biến dạng cắt: Cường độ 0.2.
- Phóng to/Thu nhỏ: $\pm 20\%$.
- Lật ngang.
- Lấy đầy điểm ảnh: nearest.

3. Kiến trúc Mô hình

Mô hình dựa trên kiến trúc CNN sử dụng API Sequential của TensorFlow/Keras, gồm Bộ trích xuất đặc trưng và Bộ phân loại.

1

Bộ trích xuất đặc trưng

4 khối tuần tự, số lượng bộ lọc tăng dần ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$). Mỗi khối gồm Conv2D (3x3, ReLU), Batch Normalization, MaxPooling2D (2x2), Dropout (0.25).

2

Bộ phân loại

Sau khi trích xuất, bản đồ đặc trưng 3D được Flatten thành vector 1D. Tiếp theo là các lớp Dense (512, 256 nơ-ron, ReLU) với BatchNormalization và Dropout (0.5). Lớp Output (8 nơ-ron) sử dụng Softmax.





Cấu hình Huấn luyện và Tối ưu hóa

Hàm mất mát

Categorical Crossentropy, đo lường sự khác biệt giữa phân phối xác suất dự đoán và nhãn thực tế.

Thuật toán tối ưu

Adam với tốc độ học khởi tạo 0.001, tự điều chỉnh tốc độ học cho từng tham số, giúp hội tụ nhanh và ổn định.

Độ đo

Dựa trên Accuracy và Top-k Categorical Accuracy (nhấn đúng trong k dự đoán hàng đầu).

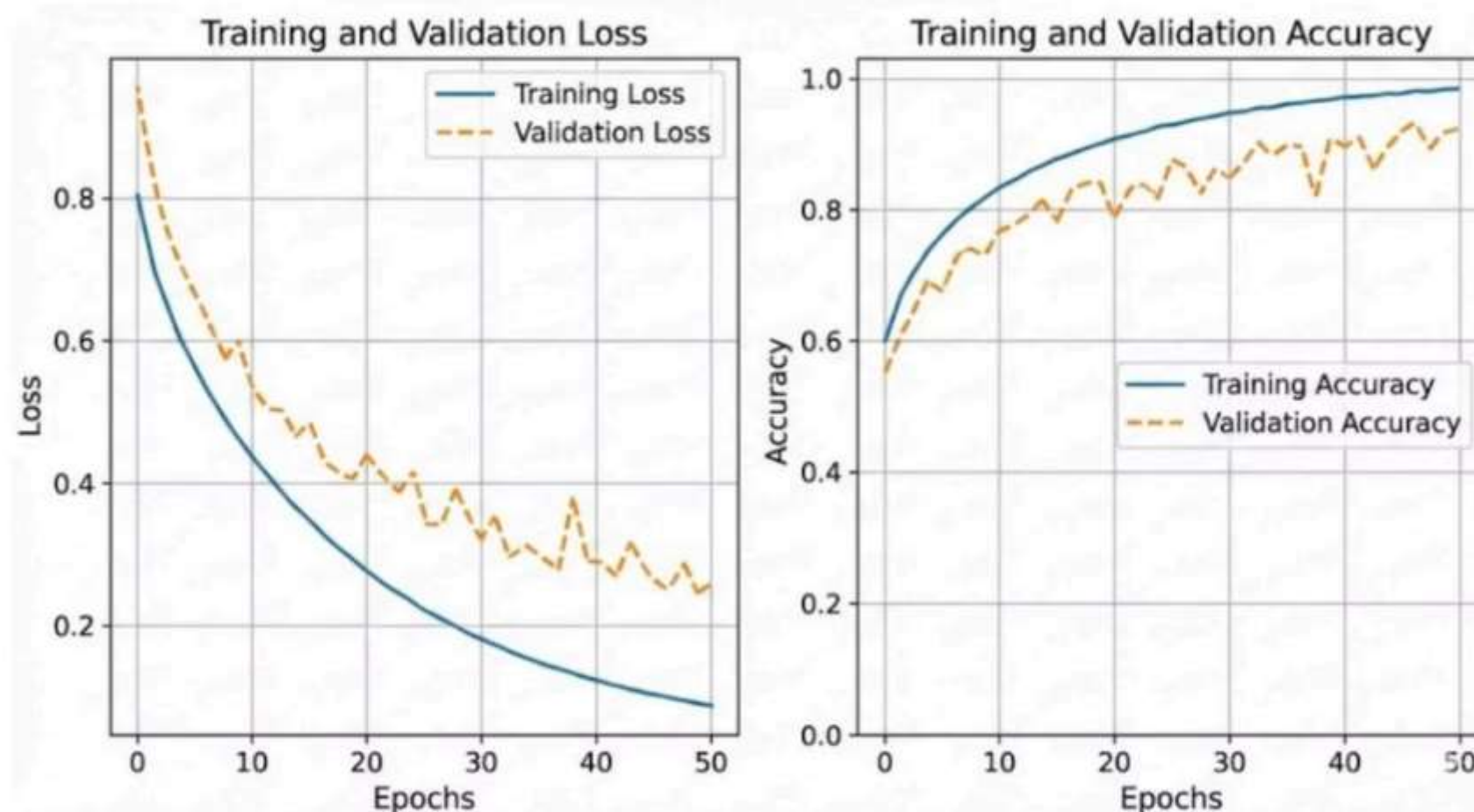
Tổng số tham số của mô hình là hàng triệu, đòi hỏi GPU để huấn luyện hiệu quả trong 50 epochs.

4. Thiết lập Thực nghiệm và kết quả

4.1. Môi trường triển khai

Sử dụng Python, TensorFlow ($\geq 2.10.0$), Keras, NumPy, Matplotlib. Quá trình tính toán được thực hiện trên môi trường có hỗ trợ GPU.

4.2. Phân tích diễn biến quá trình huấn luyện



Hàm mất mát (Loss)

- Training Loss (xanh): Giảm đơn điệu từ 0.8 xuống dưới 0.1.
- Validation Loss (cam): Giảm tổng thể từ 0.95 xuống 0.25, có dao động mạnh do batch size nhỏ và đa dạng tập validation.

Độ chính xác (Accuracy)

- Training Accuracy: Tăng ổn định, tiệm cận 1.0 (100%).
- Validation Accuracy: Tăng nhanh ban đầu, bão hòa quanh 0.90 – 0.92 (90–92%) sau epoch 30, có dao động răng cưa.

Kết luận và Hướng phát triển

5.1. Kết luận

- **Hiệu quả Kiến trúc:** Mô hình CNN đạt độ chính xác ~92% trên tập kiểm định, giải quyết tốt bài toán phân loại.
- **Chiến lược Huấn luyện:** Kết hợp Data Augmentation, Batch Normalization, Dropout kiểm soát hiệu quả quá khớp.
- **Tính ứng dụng:** Ứng dụng web Gradio hoạt động ổn định, dự đoán thời gian thực.

5.2. Hạn chế

- Dao động trong Validation Loss.
- Độ nhạy với dữ liệu nhiễu (hình vẽ tay, nền phức tạp).

5.3. Hướng phát triển

- **Cải thiện Dữ liệu:** Mở rộng với hình vẽ tay, áp dụng MixUp/CutMix.
- **Tối ưu hóa Kiến trúc:** Thử nghiệm Transfer Learning (MobileNetV2, EfficientNet), tinh chỉnh siêu tham số tự động.
- **Mở rộng Bài toán:** Phát triển Object Detection (YOLO, SSD), tích hợp tính năng tính chu vi/diện tích.



CẢM ƠN THẦY VÀ CÁC BẠN
ĐÃ LẮNG NGHE