

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

NHẬN DẠNG ẢNH DƯA TRÊN ỨNG DỤNG MÔ HÌNH GOOGLNET

Bài Báo Cáo Cơ Sở Ngành

Giảng viên hướng dẫn: Nguyễn Mộng Hiền

Sinh viên thực hiện: Nguyễn Minh Thuận

MSSV: 110123241 | Email: 110123241@st.tvu.edu.vn

Nội dung báo cáo

Phần 1: Tổng quan

- Giới thiệu đề tài & Mục tiêu nghiên cứu
- Cơ sở lý thuyết: Học máy & Học sâu
- Mạng Nơ-ron Tích chập (CNN)

Phần 2: Thực hiện

- Kiến trúc GoogleNet & MobileNetV2
- Phương pháp & Quy trình thực hiện
- Thực nghiệm & Đánh giá kết quả
- Kết luận & Hướng phát triển

Giới thiệu & Mục tiêu

Lý do chọn đề tài

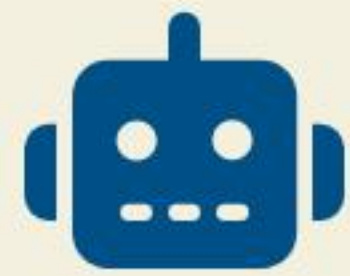
Trong kỷ nguyên 4.0, **Deep Learning** và đặc biệt là **CNN** đã tạo ra bước đột phá trong thị giác máy tính. Việc tự xây dựng mô hình từ đầu tốn kém tài nguyên, do đó việc ứng dụng các mô hình tiên tiến của Google là hướng đi tối ưu.

Mục tiêu nghiên cứu

- Nghiên cứu lý thuyết mạng nơ-ron sâu (CNN).
- Khai thác mô hình GoogleNet/MobileNetV2.
- Ứng dụng Transfer Learning để nhận dạng bộ dữ liệu **Animal-10**.



Cơ sở lý thuyết



Machine Learning

Lĩnh vực nghiên cứu giúp máy tính có khả năng "học" tự động từ dữ liệu để giải quyết vấn đề mà không cần lập trình cụ thể cho từng tác vụ.



Deep Learning

Nhánh của ML sử dụng mạng nơ-ron đa lớp (ANN) mô phỏng não người để học các đặc trưng phức tạp từ dữ liệu lớn.



CNN

Mạng nơ-ron tích chập: Mô hình chuyên biệt cho xử lý ảnh, có khả năng tự động trích xuất đặc trưng không gian (cạnh, góc, hình dạng).

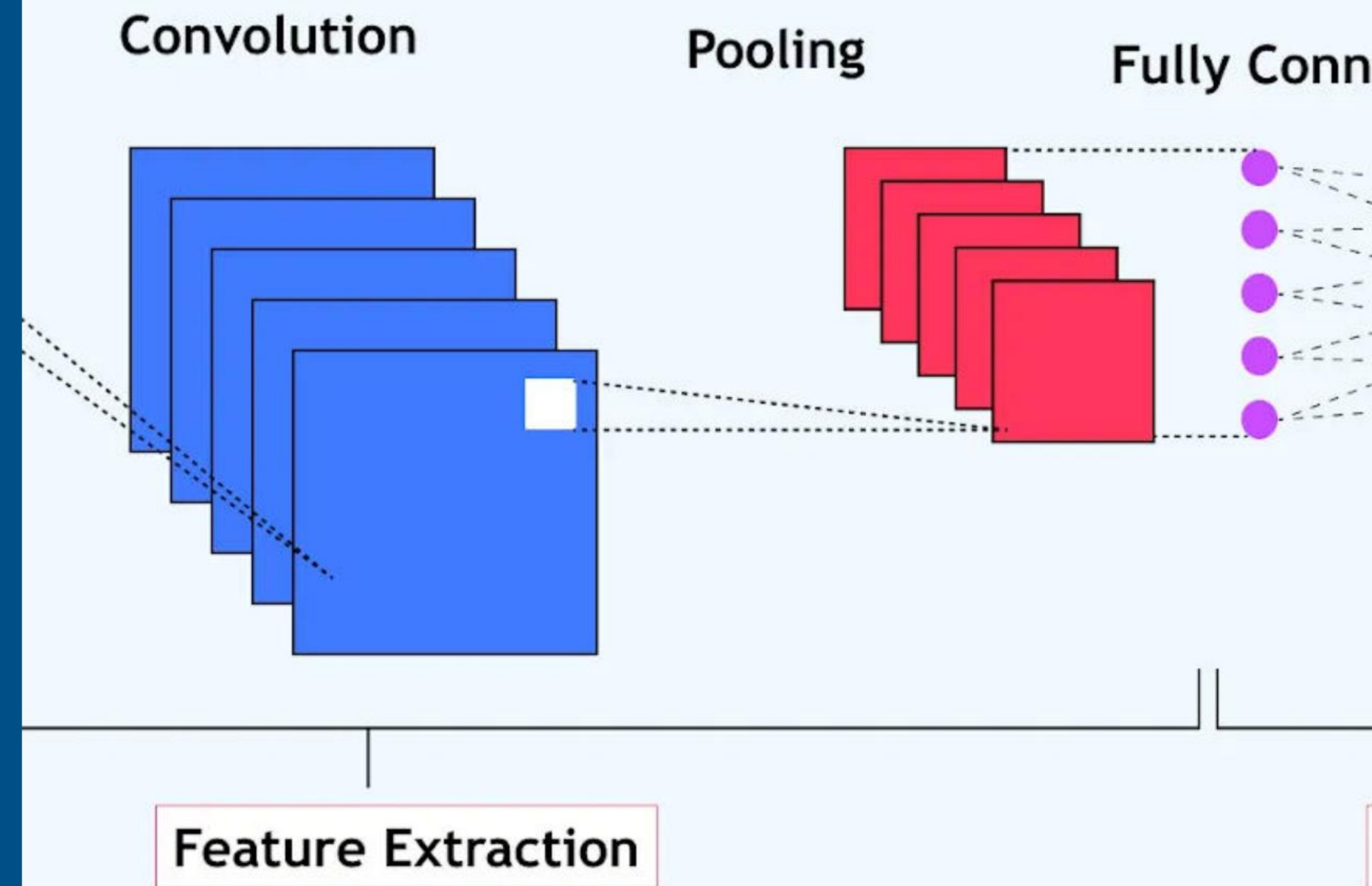
Mạng Nơ-ron Tích chập

CNN hoạt động thông qua việc luân chuyển dữ liệu qua các lớp chuyên biệt:

- **Convolutional Layer:** Dùng bộ lọc (kernel) để trích xuất đặc trưng ảnh.
- **Pooling Layer:** Giảm kích thước dữ liệu, giữ lại đặc trưng quan trọng nhất.
- **Fully-Connected Layer:** Phân loại dựa trên đặc trưng đã học.

Ưu điểm: Tự học đặc trưng, không cần tiền xử lý phức tạp.

Architecture of Convolutional Neural Network



GoogleNet & Inception Module

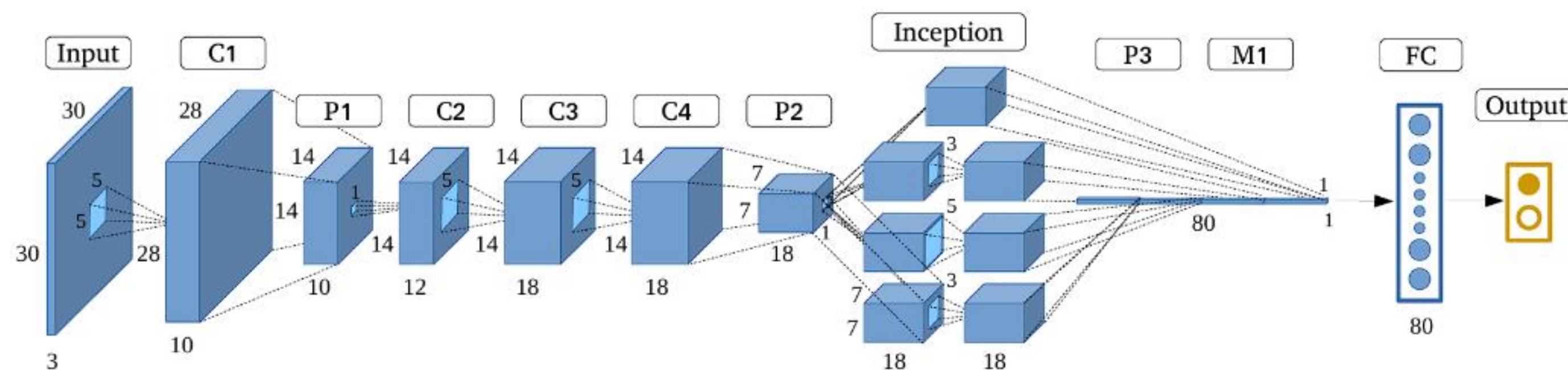
Kiến trúc Đột phá

Thay vì chọn cố định kích thước bộ lọc (3x3 hay 5x5), **Inception Module** kết hợp các phép tích chập song song.

Cơ chế hoạt động:

- Nhánh 1x1: Giảm số chiều dữ liệu.
- Nhánh 3x3 & 5x5: Trích xuất đặc trưng ở các phạm vi khác nhau.
- Max Pooling: Giữ lại đặc trưng nổi bật.

Kết quả: Tăng độ sâu (22 lớp) nhưng vẫn đảm bảo hiệu quả tính toán.

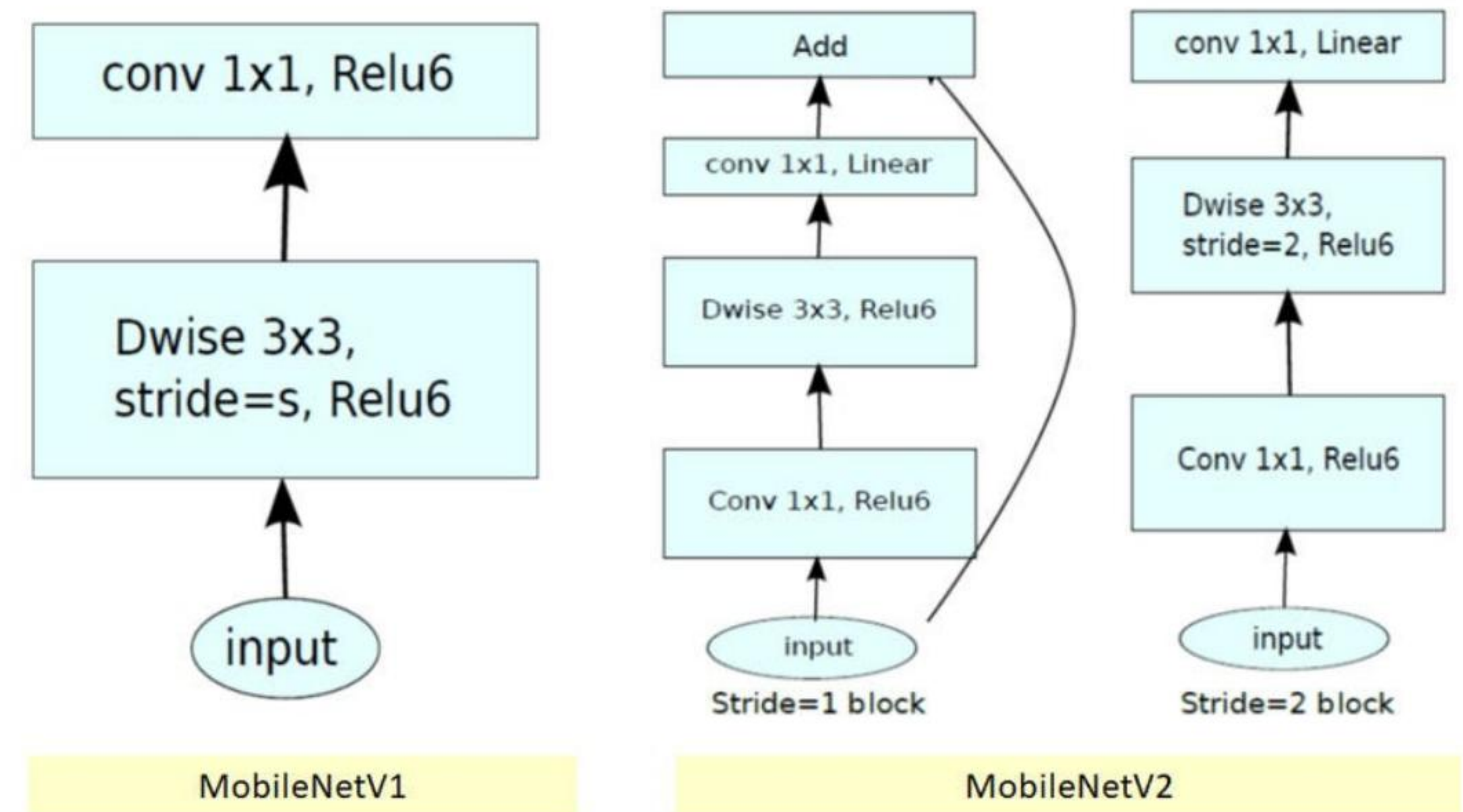


MobileNetV2: Tối ưu hóa

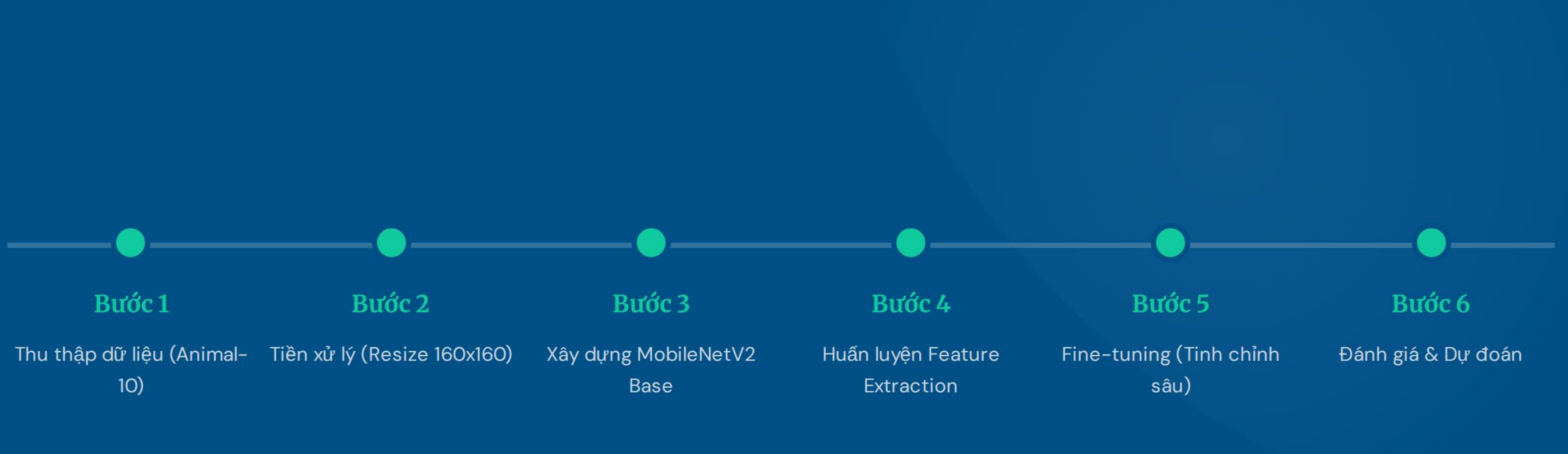
Tại sao chọn MobileNetV2?

Đây là kiến trúc CNN "nhẹ", thiết kế riêng cho các thiết bị di động và nhúng với tài nguyên hạn chế.

- **Depthwise Separable Convolution:** Tách phép tích chập thành 2 bước, giảm khối lượng tính toán cực lớn.
- **Inverted Residual Block:** Mở rộng số kênh ở giữa để giữ thông tin, nén lại ở hai đầu.
- **Hiệu quả:** Cân bằng tuyệt vời giữa kích thước mô hình, tốc độ và độ chính xác (~96% trên tập ImageNet).



Quy trình thực hiện



Dữ liệu & Công cụ thực nghiệm



Bộ dữ liệu Animal-10

Tập dữ liệu đa dạng bao gồm khoảng 28,000 ảnh thực tế.

10 Lớp đối tượng:

- ✓ Chó (Cane)
- ✓ Voi (Elefante)
- ✓ Nhện (Ragno)
- ✓ Mèo (Gatto)
- ✓ Ngựa (Cavallo)
- ✓ Bướm (Farfalla)



Môi trường & Công cụ

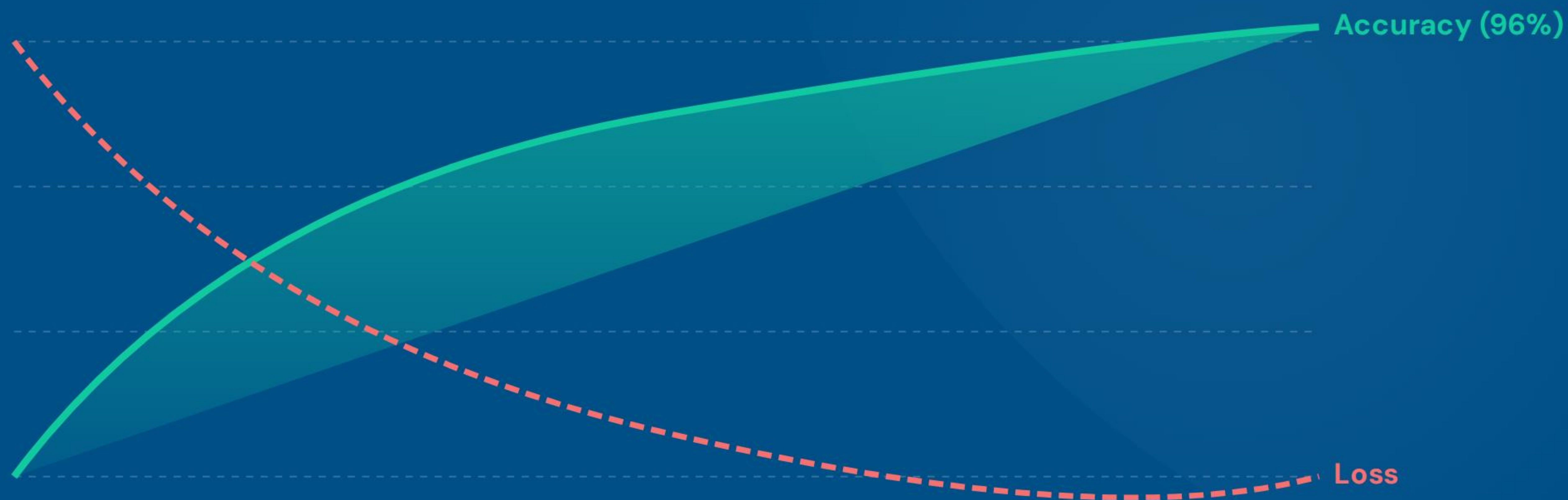
Ngôn ngữ: Python 3.x

Framework: TensorFlow / Keras

Môi trường: Kaggle Kernel / Google Colab (GPU T4)

Thư viện hỗ trợ: NumPy, Pandas, Matplotlib

Kết quả Huấn luyện (70 Epochs)



Biểu đồ cho thấy sự hội tụ ổn định. Sau Epoch 50 (giai đoạn Fine-tuning), độ chính xác tiếp tục được cải thiện, đạt ~96% trên tập kiểm thử.

Chi tiết Đánh giá (Confusion Matrix Data)

Class (Lớp)	Precision	Recall	F1-Score
Cane (Chó)	0.98	0.95	0.96
Ragno (Nhện)	0.99	0.97	0.98
Elefante (Voi)	0.86	0.99	0.92
Farfalla (Bướm)	0.96	0.99	0.97
Trung bình (Weighted Avg)	0.96	0.96	0.96

Nhận xét: Lớp **Ragno** đạt kết quả tốt nhất ($F1=0.98$). Lớp Voi có Recall rất cao (0.99) nhưng Precision thấp hơn, cho thấy mô hình hiếm khi bỏ sót voi nhưng đôi khi nhầm lẫn.

So sánh Hiệu suất F1-Score



Mô hình hoạt động đồng đều trên hầu hết các lớp, với độ chính xác vượt trội ở các đối tượng có đặc trưng hình học rõ ràng như Nhện và Sóc.

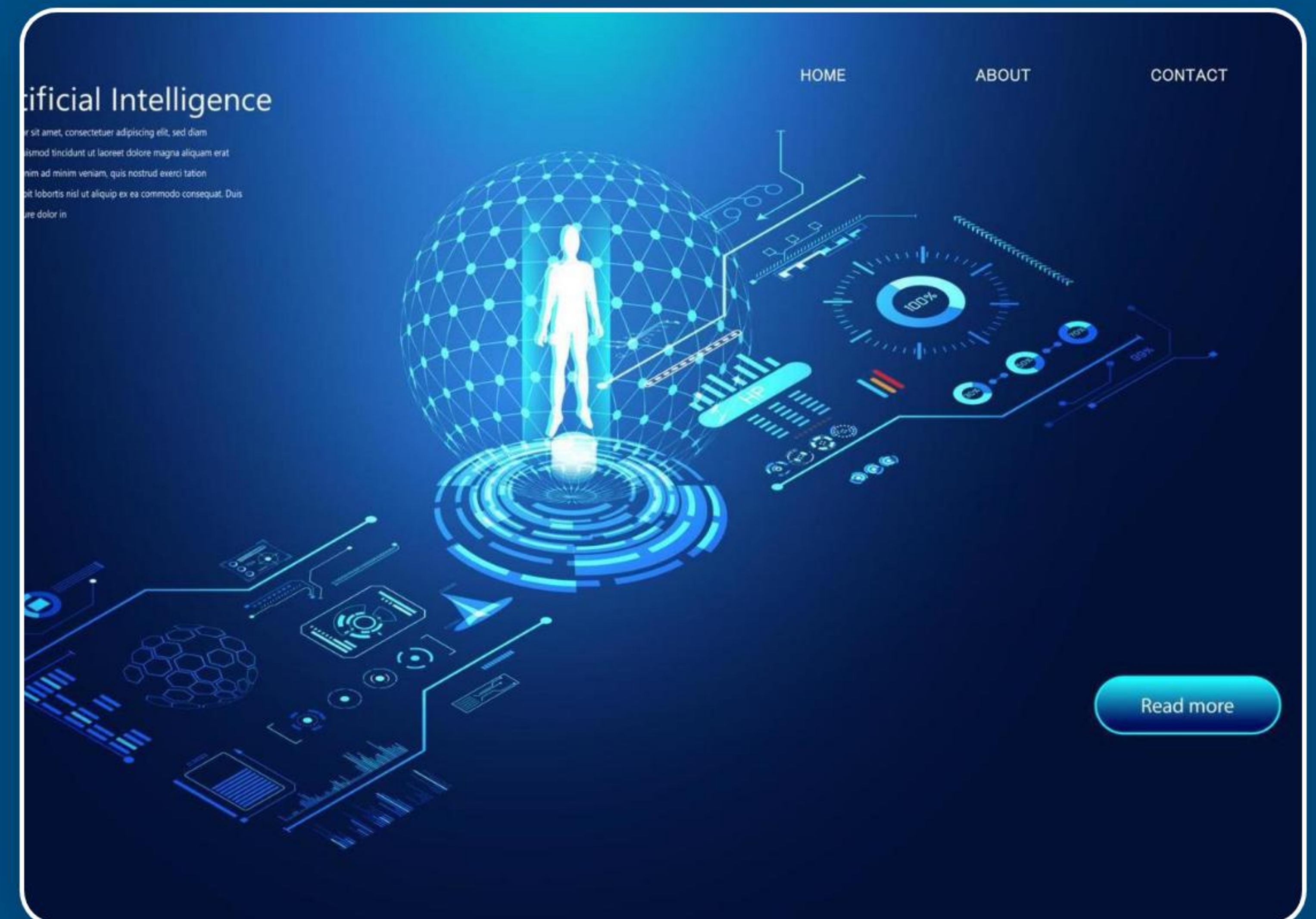
Kết luận & Hướng phát triển

Kết luận

- Mô hình **MobileNetV2** là lựa chọn tối ưu, cân bằng giữa tốc độ và độ chính xác (96%).
- Chiến lược huấn luyện 2 giai đoạn (Feature Extraction + Fine-tuning) ngăn chặn Overfitting hiệu quả.

Hướng phát triển

- → Thử nghiệm với EfficientNet hoặc Vision Transformer.
- → Triển khai lên mobile dùng **TensorFlow Lite** (Real-time).
- → Mở rộng số lớp động vật để tạo ứng dụng tra cứu sinh học.



Questions?

Cảm ơn Thầy và Hội đồng đã lắng nghe.

Sinh viên: Nguyễn Minh Thuận

Image Sources



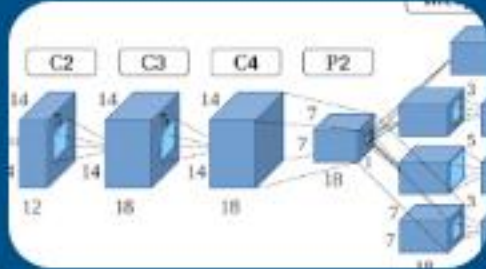
https://img.freepik.com/premium-photo/isometric-illumination-robotic-human-brain-with-detailed-circuits-concept-art-artificial-intelligence-machine-learning-brain-power-energy_778980-610.jpg

Source: www.freepik.com



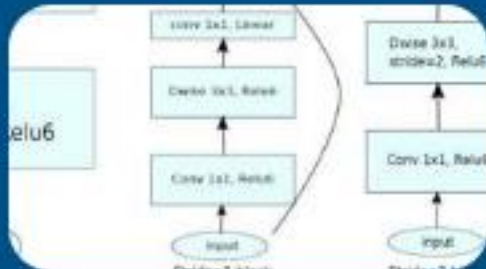
https://ik.imagekit.io/upgrad1/abroad-images/imageCompo/images/unnamed8PDPDZ_1_1ZBHR.webp

Source: www.upgrad.com



<https://viso.ai/wp-content/uploads/2024/04/google-net-like.png>

Source: viso.ai



https://miro.medium.com/1*_eOCb_RHISIUTIKZpZDRcA.jpeg

Source: [medium.com](https://medium.com/@michaelmiller1987/when-is-a-bug-a-feature-when-is-a-feature-a-bug-400000000000)



https://static.vecteezy.com/system/resources/previews/005/922/773/non_2x/abstract-technology-ai-interface-computing-concept-working-data-of-artificial-intelligence-and-futuristic-digital-for-future-on-dark-blue-background-vector.jpg

Source: www.vecteezy.com