HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÙI QUANG MINH

XÂY DỰNG ỨNG DỤNG DI ĐỘNG NHẬN DIỆN CÁ CẢNH SỬ DỤNG TENSORFLOW LITE VÀ GOOGLE GEMINI

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÙI QUANG MINH

XÂY DỰNG ỨNG DỤNG DI ĐỘNG NHẬN DIỆN CÁ CẢNH SỬ DỤNG TENSORFLOW LITE VÀ GOOGLE GEMINI

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS. TSKH. HOÀNG ĐĂNG HẢI

HÀ NỘI – 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đề án tốt nghiệp của tôi với đề tài: "XÂY DỰNG ỨNG DỤNG DI ĐỘNG NHẬN DIỆN CÁ CẢNH SỬ DỤNG TENSORFLOW LITE VÀ GOOGLE GEMINI" là công trình nghiên cứu của riêng tôi.

Tôi cam đoan các số liệu, kết quả nêu trong đề án là trung thực từ đánh giá và thực nghiệm và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/ nghiên cứu nào của người khác được sử dụng trong đề án này mà không được trích dẫn theo đúng quy định.

Hà Nội, ngày 08 tháng 12 năm 2025

Học viên thực hiện đề án



Bùi Quang Minh

LỜI CẢM ƠN

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến PGS.TSKH Hoàng Đăng Hải, người đồng hành cùng tôi từ những bước lên ý tưởng đầu tiên cho đến khi hoàn thành đề án này. Xin chân thành cảm ơn những lời chỉ bảo, hướng dẫn của Thầy đã giúp tôi có thêm những góc nhìn khác, kiến thức và kinh nghiệm trong quá trình nghiên cứu và phát triển ứng dụng của mình.

Tôi xin gửi lời cảm ơn chân thành đến Khoa Đào tạo sau đại học – Học viện Công nghệ Bưu chính Viễn thông đã tạo mọi điều kiện thuận lợi cho tôi trong suốt quá trình học tập và hoàn thành đề án tốt nghiệp.

Cuối cùng, tôi xin gửi lời cảm ơn đến các thầy cô của Học viện Công nghệ Bưu chính Viễn thông, các kiến thức được thầy/cô truyền đạt trong suốt khoảng thời gian học tập tại trường đã tạo cho tôi nền tảng vững chắc cho quá trình học tập, nghiên cứu và làm việc.

Tôi xin chân thành cảm ơn!

Hà Nội, ngày 24 tháng 11 năm 2025

Học viên thực hiện đề án



Bùi Quang Minh

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT vi
DANH SÁCH HÌNH VẼ viii
DANH SÁCH BẢNG ix
PHẦN MỞ ĐẦU 1
1. Lý do chọn đề tài: 1
2.Tổng quan về vấn đề nghiên cứu: 2
3.Mục đích nghiên cứu: 4
4.Đối tượng và phạm vi nghiên cứu: 4
5.Phương pháp nghiên cứu: 4
6.Cấu trúc đề án tốt nghiệp 6
CHƯƠNG 1: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT 7
1.1. Tổng quan về bài toán nhận diện, phân loại cá cảnh 7
1.2. Khái quát về thị giác máy tính và mạng nơ-ron 11
1.2.1. Bài toán phân loại hình ảnh 11
1.2.2. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) .. 11
1.2.3. Học chuyển giao (Transfer Learning) 12
1.3. Các kiến trúc mạng CNN tối ưu cho thiết bị di động 13
1.3.1. MobileNetV2 và Khối Inverted Residuals 13
1.3.2. EfficientNet-Lite 14
1.4. Nền tảng TensorFlow Lite và tối ưu hóa mô hình 16
1.4.1. Kiến trúc TensorFlow Lite 16
1.4.2. Kỹ thuật Lượng tử hóa (Quantization) 16
1.5. Google Gemini và Generative AI trong truy xuất thông tin 16
1.5.1. Kiến trúc Multimodal của Gemini 17
1.5.2. Prompt Engineering và Contextual Learning 17
1.6. Các chỉ số đánh giá mô hình (Evaluation Metrics) 18
1.7. Kết chương 18
CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 20
2.1. Phân tích yêu cầu 20
2.1.1. Yêu cầu chức năng 21
2.1.2. Yêu cầu phi chức năng 22

2.2. Mô hình hóa yêu cầu 22
2.2.1. Biểu đồ Ca sử dụng 23
2.3. Thiết kế kiến trúc hệ thống 24
2.3.1. Sơ đồ kiến trúc tổng thể hệ thống 24
2.3.2. Thành phần Client (Mobile App) 25
2.3.3. . Thành phần Cloud (Generative AI) 25
2.4. Thiết kế chi tiết luồng xử lý 25
2.4.1. Quy trình xử lý ảnh và suy luận (TFLite Flow) 25
2.4.2. Quy trình tích hợp Gemini (Prompt Flow) 26
2.4.3. Biểu đồ tuần tự (Sequence Diagram) 26
2.5. Thiết kế giao diện người dùng (UI Design) 28
2.5.1. Màn hình chính (Home Screen) 28
2.5.2. Màn hình kết quả) 29
2.6. Kết chương 29
CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI THỬ NGHIỆM HỆ THỐNG .31
3.1. Môi trường và công cụ thử nghiệm 32
3.2. Xây dựng và tiền xử lý dữ liệu 32
3.2.1. Thu thập và tổ chức dữ liệu 32
3.2.2. Kỹ thuật tăng cường dữ liệu (Data Augmentation) 32
3.3. Huấn luyện mô hình nhận diện (Model Training) 34
3.3.1. Cấu hình kiến trúc mạng 34
3.3.2. Kết quả huấn luyện 35
3.3.3. Tối ưu hóa và đóng gói mô hình 35
3.4. Xây dựng ứng dụng di động 36
3.4.1. Module nhận diện cục bộ (Local Inference) 36
3.4.2. Module tích hợp Tri thức (Gemini Integration) 37
3.5. Giao diện người dùng và kiểm thử 38
3.5.1. Thiết kế giao diện (UI/UX) 38
3.5.2. Kết quả kiểm thử thực nghiệm 38

3.6. Kết quả huấn luyện mô hình.....38
3.6.1. Độ chính xác và các chỉ số đánh giá.....39
3.6.2. Hiệu quả của Lượng tử hóa (Quantization).....39
3.7. Kết quả thử nghiệm ứng dụng di động.....41
3.7.1. Giao diện ứng dụng.....41
3.7.2. Demo quy trình nhận diện.....43
3.7.3. Phản hồi thông minh từ Gemini.....44
3.7.4. Đánh giá hiệu năng.....44
3.8. Kết chương.....45
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....46
5.1. Kết luận.....46
5.2. Hạn chế.....47
5.3. Hướng phát triển.....47
DANH MỤC CÁC TÀI LIỆU THAM KHẢO.....48

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tên đầy đủ (Tiếng Anh)	Giải thích / Nghĩa Tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
App	Application	Ứng dụng (thường chỉ ứng dụng di động)
Cloud AI	Cloud Artificial Intelligence	Trí tuệ nhân tạo xử lý trên đám mây
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CPU	Central Processing Unit	Bộ xử lý trung tâm
DL	Deep Learning	Học sâu
Edge AI	Edge Artificial Intelligence	Trí tuệ nhân tạo biên (xử lý trên thiết bị)
FC	Fully Connected (Layer)	Lớp kết nối đầy đủ (trong mạng nơ-ron)
FN	False Negative	Âm tính giả (Dự đoán sai là không phải đối tượng)
FP	False Positive	Dương tính giả (Dự đoán sai là đối tượng)
F1-Score	F1-Score	Chỉ số trung bình điều hòa giữa Precision và Recall
Float32	32-bit Floating Point	Định dạng số thực dấu phẩy động 32-bit
GPU	Graphics Processing Unit	Bộ xử lý đồ họa (dùng để huấn luyện mô hình)
IDE	Integrated Development Environment	Môi trường phát triển tích hợp (ví dụ: Android Studio)
Int8	8-bit Integer	Định dạng số nguyên 8-bit

JSON	JavaScript Object Notation	Định dạng dữ liệu văn bản dùng để trao đổi thông tin
LLM	Large Language Model	Mô hình ngôn ngữ lớn (ví dụ: Google Gemini)
ML	Machine Learning	Học máy
MLP	Multilayer Perceptron	Mạng nơ-ron đa tầng
PTQ	Post-training Quantization	Lượng tử hóa sau huấn luyện
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
ReLU	Rectified Linear Unit	Hàm kích hoạt chỉnh lưu tuyến tính
RGB	Red Green Blue	Không gian màu đỏ - lục - lam
SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
TFLite	TensorFlow Lite	Thư viện triển khai mô hình học máy trên thiết bị di động
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng

DANH SÁCH HÌNH VẼ

Hình 1.1: Kiến trúc cơ bản của một mạng CNN 11
Hình 1.2: So sánh Standard Convolution và Depthwise Separable Convolution.13
Hình 1.3: Cấu trúc khối Inverted Residual của MobileNetV2 (Sandler et al., 2018) 14
Hình 2.1: Sơ đồ Use case ứng dụng nhận diện cá cảnh 23
Hình 2.2: Sơ đồ kiến trúc hệ thống 24
Hình 2.3: Biểu đồ tuần tự quy trình nhận diện 27
Hình 3.1: Cấu hình tham số cho quá trình tăng cường dữ liệu 34
Hình 3.2: Khởi tạo mô hình MobileNetV2 và thêm lớp phân loại 35
Hình 3.3: Chuyển đổi và lượng tử hóa mô hình sang định dạng TFLite 36
Hình 3.4: Hàm thực thi suy luận trên thiết bị di động 37
Hình 3.5: Cấu trúc Prompt gửi đến Gemini API 38
Hình 4.1: Giao diện ứng dụng khi chưa nhận diện cá 41
Hình 4.2: Giao diện ứng dụng khi hoàn thành nhận diện loài cá 42
Hình 4.3: Giao diện ứng dụng khi hoàn thành lấy thông tin chi tiết loài cá.....43

DANH SÁCH BẢNG

Bảng 1.1: Cấu hình môi trường phát triển 15
Bảng 3.1: Cấu hình môi trường phát triển 32
Bảng 3.2: Bảng so sánh hiệu quả lượng tử hoá 39
Bảng 3.3: So sánh hiệu năng các mô hình trên tập dữ liệu mẫu.....40

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài:

Nhận diện cá nói chung là một vấn đề được quan tâm nghiên cứu hiện nay trong các lĩnh vực nông nghiệp, nuôi trồng thủy sản, thương mại, v.v. Nhận diện cá bao gồm việc xác định và phân loại các loài cá từ hình ảnh (hoặc thậm chí video). Kết quả nhận diện có thể được áp dụng vào việc xác định nguồn và loài cá trong đánh bắt, đưa ra các giải pháp chăn nuôi thủy sản phù hợp bao gồm cả việc cung cấp nguồn thức ăn cho cá, v.v.

Trong lĩnh vực nuôi cá cảnh, việc nhận dạng loài cá giúp cung cấp thông tin hữu ích về cách nhận biết, chăm sóc cá cảnh tốt nhất cũng như các thông tin chi tiết về nguồn gốc, đặc điểm, v.v. của từng loài cá cảnh.

Xác định và phân loại cá là một vấn đề khó do sự đa dạng sinh học của loài cá và do loài cá sinh sống trong môi trường nước. Sự xuất hiện nhanh chóng của công nghệ học máy và học sâu đã tạo ra những cơ hội mới cho bài toán nhận diện loài cá.

Đề án tốt nghiệp thạc sỹ này đặt vấn đề nghiên cứu ứng dụng kỹ thuật học máy vào xây dựng một ứng dụng di động nhận dạng cá cảnh. Các vấn đề cụ thể được đặt ra gồm: thu thập hình ảnh cá cảnh từ các nguồn khác nhau và thực hiện tiền xử lý, gán nhãn, xây dựng tập dữ liệu cá cảnh; thực hiện xây dựng mô hình học sâu, huấn luyện mô hình, kiểm tra và hiệu chỉnh mô hình; thực hiện việc chuyển đổi kết quả nhận dạng loài cá vào mô hình Google Gemini để cung cấp thông tin chi tiết về loài cá cảnh.

Về mặt lý thuyết, nghiên cứu đề tài này là cơ hội để học viên tìm hiểu về công nghệ học sâu và trí tuệ nhân tạo, áp dụng các công nghệ này vào việc xây dựng một sản phẩm ứng dụng thực tế. Việc ứng dụng các thuật toán học sâu tiên tiến để phân tích hình ảnh sẽ giúp người dùng xác định chính xác loài cá một cách nhanh chóng và dễ dàng. Về mặt thực tiễn, ứng dụng này sẽ đáp ứng một nhu cầu thiết thực của cộng đồng người nuôi cá, từ những người mới bắt đầu cho đến những người chơi chuyên nghiệp. Bên cạnh đó, ứng dụng còn góp phần thúc đẩy nghiên cứu khoa học về đa dạng sinh học thủy sinh, cung cấp thêm dữ liệu cho những ai mong muốn tìm hiểu về các loài cá, góp phần phổ biến và chia sẻ

kiến thức về các loài cá và bảo vệ đa dạng sinh học. Việc xây dựng ứng dụng nhận diện cá cảnh không chỉ là một dự án công nghệ thú vị mà còn mang lại nhiều giá trị thực tiễn. Về góc độ kinh doanh, ứng dụng này mở ra nhiều cơ hội hợp tác với các cửa hàng thú cưng, trại nuôi cá và các tổ chức bảo vệ môi trường.

2. Tổng quan về vấn đề nghiên cứu:

Nghiên cứu xây dựng ứng dụng di động phát hiện và phân loại cá cảnh qua hình ảnh bằng học máy/AI đang trở thành xu hướng quan trọng trong lĩnh vực nông nghiệp số và thú chơi thủy sinh. Với sự phổ biến của điện thoại thông minh cùng camera chất lượng cao, việc tận dụng các mô hình thị giác máy tính để tự động nhận dạng loài cá mang lại giá trị thực tiễn lớn: hỗ trợ người chơi cá cảnh, cửa hàng, người nuôi chuyên nghiệp và cả các hệ thống giám sát trong hồ thông minh (smart aquarium).

Trọng tâm của hướng nghiên cứu này bao gồm thu thập và xây dựng bộ dữ liệu ảnh đa dạng các loài cá cảnh phổ biến ở Việt Nam và quốc tế, đảm bảo bao quát nhiều điều kiện ánh sáng, góc chụp, trạng thái chuyển động. Trên cơ sở dữ liệu này, các mô hình học sâu như CNN, MobileNet, EfficientNet hoặc Vision Transformer được huấn luyện để thực hiện hai nhiệm vụ chính: (1) phát hiện vùng chứa cá trong ảnh và (2) phân loại chính xác loài cá.

Một thách thức quan trọng là vấn đề nhiễu hình ảnh (refraction, ánh sáng phản chiếu, bong bóng), khiến mô hình cần kỹ thuật tăng cường dữ liệu (data augmentation) chuyên biệt. Ngoài ra, do ứng dụng chạy trên thiết bị di động, việc tối ưu mô hình để giảm kích thước, tăng tốc độ suy luận nhưng vẫn duy trì độ chính xác là yêu cầu cốt lõi; các kỹ thuật như lượng tử hóa (quantization), cắt ảnh, tách lọc đặc trưng cần được xem xét. Ứng dụng cho phép người dùng chụp ảnh cá trong bể thực tế, nhận diện loài cá trong thời gian thực và gợi ý thông tin sinh học, cách chăm sóc hoặc cảnh báo bệnh thường gặp. Nghiên cứu này không chỉ có ý nghĩa khoa học trong tối ưu mô hình nhận dạng đối tượng đặc thù dưới môi trường nước, mà còn mang tính ứng dụng cao, góp phần xây dựng hệ sinh thái thủy sinh thông minh và thúc đẩy chuyển đổi số trong ngành nuôi cá cảnh.

Trong đề án tốt nghiệp này, quá trình phát triển ứng dụng các công nghệ như TensorFlow, TensorFlow Lite và Google Gemini. TensorFlow được sử dụng

trong nghiên cứu và huấn luyện mô hình, cho phép triển khai linh hoạt các kiến trúc học sâu hiện đại và tối ưu quy trình tiền xử lý dữ liệu. Sau khi đạt được mô hình có độ chính xác tốt, TensorFlow Lite hỗ trợ chuyển đổi mô hình sang dạng gọn nhẹ, phù hợp triển khai trực tiếp lên thiết bị di động với tốc độ suy luận nhanh, tiêu thụ ít tài nguyên và vận hành ổn định trong nhiều dòng điện thoại khác nhau. Bên cạnh đó, Google Gemini có thể được tích hợp như một thành phần AI hỗ trợ, chẳng hạn dùng để chú giải thông tin loài cá sau khi mô hình phân loại xong, tạo hướng dẫn chăm sóc cá theo thời gian thực, hoặc tương tác trò chuyện để giải thích các yếu tố bất thường trong hình ảnh. Sự kết hợp này tạo nên một ứng dụng toàn diện: mô hình thị giác máy tính đảm nhiệm việc nhận dạng, TensorFlow Lite đảm bảo hiệu suất trên di động, còn Gemini mang lại trải nghiệm thông minh, thân thiện và giàu thông tin cho người dùng. Đây là nền tảng công nghệ hứa hẹn giúp ứng dụng có tính thực tiễn cao, dễ mở rộng và tương thích với hệ sinh thái AI hiện đại.

Việc xây dựng ứng dụng nhận diện cá cảnh sử dụng TensorFlow Lite và Google Gemini mang lại nhiều triển vọng trong lĩnh vực công nghệ và đời sống. TensorFlow Lite, một framework tối ưu hóa mô hình học sâu cho các thiết bị di động và nhúng, giúp mô hình AI hoạt động mượt mà trên các thiết bị có hiệu suất hạn chế như smartphone hay IoT. Khi kết hợp với Google Gemini, một mô hình AI tiên tiến của Google có khả năng phân tích và nhận diện đa dạng các loại dữ liệu, ứng dụng này có thể đạt được độ chính xác cao trong việc nhận diện các loại cá cảnh.

Một trong những ưu điểm nổi bật của ứng dụng là khả năng chạy hiệu quả trên thiết bị di động nhờ vào TensorFlow Lite, trong khi Google Gemini cung cấp sự mạnh mẽ và linh hoạt trong việc học hỏi và cải thiện mô hình theo thời gian. Tuy nhiên, ứng dụng cũng phải đối mặt với một số thách thức như phụ thuộc vào chất lượng dữ liệu huấn luyện, giới hạn về khả năng tính toán trên các thiết bị cấu hình thấp, và yêu cầu cao về kỹ thuật trong việc cập nhật và duy trì mô hình.

Trong thực tế, ứng dụng này có thể được triển khai để giám sát sức khỏe cá cảnh, giúp người chơi cá nhận diện và phân loại cá nhanh chóng, hoặc hỗ trợ

nghiên cứu và bảo tồn các loài cá hiếm. Với sự phát triển không ngừng của AI và Google Gemini, ứng dụng này hứa hẹn sẽ trở thành một công cụ hữu ích và mở ra nhiều cơ hội mới trong lĩnh vực này. Tuy nhiên, để đạt được tiềm năng đó, cần phải giải quyết những thách thức kỹ thuật và tiếp tục cải tiến mô hình dựa trên dữ liệu chất lượng cao.

3. Mục đích nghiên cứu:

Mục đích nghiên cứu của đề án tốt nghiệp là nghiên cứu kỹ thuật học sâu tiên tiến và ứng dụng vào xây dựng một ứng dụng di động nhằm nhận diện và phân loại cá cảnh.

Một trong những trọng tâm đặt ra khi xây dựng ứng dụng di động nhận diện cá cảnh là cần xây dựng một mô hình học sâu, áp dụng mô hình vào xây dựng một ứng dụng di động có giao diện thân thiện và cung cấp thông tin chi tiết về loài cá cảnh, hướng đến là người bạn đồng hành tin cậy của mọi người nuôi cá trong việc hỗ trợ nhận diện xác định loài, cung cấp thông tin chi tiết về loài cá cảnh cho người dùng.

4. Đối tượng và phạm vi nghiên cứu:

- Đối tượng: Mô hình học sâu và ứng dụng di động cho nhận diện và phân loại cá cảnh.
- Phạm vi: Tập dữ liệu khoảng 500 hình ảnh của hơn 20 loài cá cảnh.

5. Phương pháp nghiên cứu:

- Khảo sát, nghiên cứu các tài liệu liên quan về mô hình, nguồn dữ liệu, phương pháp xử lý dữ liệu hình ảnh.
- Nghiên cứu lý thuyết học máy, học sâu, trí tuệ nhân tạo.
- Nghiên cứu khảo sát các công cụ hỗ trợ xây dựng mô hình học sâu và phát triển ứng dụng di động: MobileNetV2, EfficientNet Lite, TensorFlow, Google Gemini,…
- Nghiên cứu cách thức xây dựng tập dữ liệu hình ảnh, thực hiện tiền xử lý, xây dựng mô hình học sâu phục vụ cho việc nhận diện và phân loại cá cảnh.
- Nghiên cứu cách thức xây dựng ứng dụng di động nhận diện và phân loại cá cảnh.
- Triển khai thử nghiệm và đánh giá ứng dụng.

6. Cấu trúc đề án tốt nghiệp

Ngoài phần mở đầu, kết luận, danh mục tài liệu tham khảo và phụ lục, luận văn gồm 03 chương như sau:

Chương 1 trình bày tổng quan vấn đề nghiên cứu, nền tảng lý thuyết để thực hiện nghiên cứu. Các nội dung chính gồm: Vấn đề nghiên cứu của đề án; Khái quát về bài toán phân loại hình ảnh, thị giác máy tính, mạng Nơ-ron; Các kiến trúc mạng CNN phù hợp cho thiết bị di động; Nền tảng TensorFlow, Google Gemini và Generative AI; Các chỉ số đánh giá mô hình.

Chương 2 trình bày về phân tích, thiết kế hệ thống ứng dụng di động với các nội dung: Phân tích yêu cầu; Mô hình hóa hệ thống; Thiết kế chi tiết các thành phần ứng dụng.

Chương 3 trình bày kết quả xây dựng và triển khai thử nghiệm hệ thống. Các nội dung chính gồm: Môi trường và công cụ thử nghiệm; Xây dựng tập dữ liệu, tiền xử lý dữ liệu; Xây dựng ứng dụng di động; Triển khai thử nghiệm và đánh giá kết quả.

CHƯƠNG 1: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về bài toán nhận diện, phân loại cá cảnh

Nhận diện và phân loại cá là một vấn đề được nghiên cứu rộng rãi liên quan đến các lĩnh vực phân đoạn hình ảnh, mô hình nhận dạng và truy xuất thông tin. Phân loại cá đã được áp dụng trong vô số lĩnh vực như đã chỉ ra trong [1]. Kết quả khảo sát các nghiên cứu về bài toán nhận diện và phân loại cá đã được trình bày trong [1], trong đó chỉ ra các nghiên cứu liên quan dựa vào các cơ sở dữ liệu có sẵn như Fish4-Knowledge (F4K), các nghiên cứu về các phương pháp tiền xử lý đặc trưng hình ảnh loài cá, các kỹ thuật trích xuất dữ liệu cũng như các mô hình học máy.

Trong [2], các tác giả cũng chi ra nghiên cứu một phương pháp nhận diện loài cá dựa vào mạng Nơ ron tích chập sâu, điển hình là VGG16, đào tạo trước thông quan phương pháp học sâu chuyển giao. Nghiên cứu tập trung chủ yếu vào việc sử dụng 4 loại tệp dữ liệu khác nhau và khả năng trộn chúng trong huấn luyện nhận dạng ảnh loài cá.

Trong [3], các tác giả khảo sát các phương pháp học sâu được lựa chọn để xác định loài dựa trên hình ảnh và giới thiệu các ứng dụng có sẵn hiện nay. Các tác giả cũng nhấn mạnh khả năng sử dụng các hệ thống trí tuệ nhân tạo trong nhận dạng loài.

Các tác giả [4] trình bày kỹ thuật học máy và thị giác máy tính có thể được sử dụng để tự động xử lý hình ảnh và các kỹ thuật nhận dạng hình ảnh phù hợp theo truyền thống đã được phát triển để giải quyết các vấn đề cụ thể.

Villon Sébastien và các đồng sự trong [5] đã trình bày một phương pháp hỗ trợ việc xác định các loài cá trên hình ảnh dưới nước và chúng tôi so sánh hiệu suất mô hình của chúng tôi với khả năng của con người về tốc độ và độ chính xác. Đồng thời nhóm cũng chỉ ra các phương pháp Deep Learning có thể thực hiện nhận dạng cá hiệu quả trên hình ảnh dưới nước và hứa hẹn sẽ xây dựng các giao thức dựa trên video mới để theo dõi đa dạng sinh học của cá với chi phí thấp và hiệu quả.

Trong bài báo [6] nhóm tác giả đề xuất phương pháp học sâu gồm hai bước để phát hiện và phân loại các loài cá mà không cần lọc trước. Bước đầu tiên

là phát hiện từng con cá trong một hình ảnh, không phụ thuộc vào loài và giới tính sử dụng kỹ thuật phát hiện đối tượng You Only Look Once (YOLO). Ở bước thứ hai, áp dụng Mạng thần kinh chuyển đổi (CNN) với kiến trúc Squeeze-and-Excitation (SE) để phân loại từng con cá trong ảnh mà không cần lọc trước.

Trong [7], trong sự kết hợp của cơ chế xử lý hình ảnh và máy học, nhóm tác giả xác định các loài cá bị nhiễm bệnh do nhiều loại mầm bệnh khác nhau gây ra. Công việc này chia thành hai phần. Ở phần thô sơ, quá trình xử lý trước và phân đoạn hình ảnh đã được áp dụng để giảm nhiễu và phóng đại hình ảnh. Trong phần thứ hai, thực hiện trích xuất các tính năng liên quan để phân loại bệnh với sự trợ giúp của thuật toán Máy vectơ hỗ trợ (SVM) của máy học với chức năng hạt nhân.

Trong bài báo [8], nhóm tác giả đến từ các trường đại học Australia và Pakistan đề xuất một giải pháp kết hợp để kết hợp mô hình dòng quang học và hỗn hợp Gaussian với mạng lưới thần kinh sâu YOLO, dự án đã loại bỏ được hạn chế của YOLO chỉ sử dụng để ghi lại được các trường hợp cá tĩnh và có thể nhìn thấy rõ ràng và cho phép phát hiện cá di chuyển tự do và ngụy trạng dưới nền.

Tác giả Suxia Cui và các đồng nghiệp tại trường đại học Prairie View A&M trong [9] đã đề xuất phương pháp phát hiện cá dựa trên mạng thần kinh tích chập (CNN). Tập dữ liệu huấn luyện được thu thập từ Vịnh Mexico bằng máy ảnh kỹ thuật số. Ba phương pháp tối ưu hóa đã được áp dụng cho CNN: tăng cường dữ liệu, đơn giản hóa mạng và tăng tốc quá trình đào tạo. Trong đó, chuyển đổi tăng cường dữ liệu cung cấp nhiều mẫu học tập hơn; mạng đã được đơn giản hóa để phù hợp với mạng lưới thần kinh nhân tạo; quá trình đào tạo tăng tốc được giới thiệu để làm cho quá trình đào tạo hiệu quả hơn về thời gian.

Trong nghiên cứu [10] của mình, tác giả Chao Zhou và các đồng nghiệp đã nhấn mạnh sự xuất hiện và phát triển nhanh chóng của công nghệ học sâu. Công trình nghiên cứu mang lại những đột phá trong việc giải quyết những dữ liệu phức tạp trong nuôi trồng thủy sản.

Ứng dụng nhận diện cá cảnh là sự kết hợp hoàn hảo giữa công nghệ hiện đại và niềm đam mê nuôi cá. Bằng cách sử dụng trí tuệ nhân tạo, ứng dụng giúp người dùng khám phá thế giới thủy sinh một cách sâu sắc hơn.

Việc xây dựng ứng dụng nhận diện cá cảnh sử dụng TensorFlow Lite và Google Gemini mang lại nhiều triển vọng trong lĩnh vực công nghệ và đời sống. TensorFlow Lite, một framework tối ưu hóa mô hình học sâu cho các thiết bị di động và nhúng, giúp mô hình AI hoạt động mượt mà trên các thiết bị có hiệu suất hạn chế như smartphone hay IoT. Khi kết hợp với Google Gemini, một mô hình AI tiên tiến của Google có khả năng phân tích và nhận diện đa dạng các loại dữ liệu, ứng dụng này có thể đạt được độ chính xác cao trong việc nhận diện các loại cá cảnh.

Một trong những ưu điểm nổi bật của ứng dụng là khả năng chạy hiệu quả trên thiết bị di động nhờ vào TensorFlow Lite, trong khi Google Gemini cung cấp sự mạnh mẽ và linh hoạt trong việc học hỏi và cải thiện mô hình theo thời gian. Tuy nhiên, ứng dụng cũng phải đối mặt với một số thách thức như phụ thuộc vào chất lượng dữ liệu huấn luyện, giới hạn về khả năng tính toán trên các thiết bị cấu hình thấp, và yêu cầu cao về kỹ thuật trong việc cập nhật và duy trì mô hình.

Đề án tốt nghiệp này hướng đến nghiên cứu phát triển một ứng dụng di động có khả năng phát hiện và phân loại các loài cá cảnh thông qua hình ảnh chụp thực tế, phục vụ nhu cầu của người chơi cá cảnh, cửa hàng thủy sinh, hệ thống hồ thông minh và các ứng dụng giáo dục. Qua khảo sát các công trình nghiên cứu hiện có, bài toán nhận dạng cá trong môi trường nước gặp nhiều thách thức như nhiễu ánh sáng, phản xạ mặt kính, chuyển động nhanh của cá và độ đa dạng của loài. Vì vậy, việc kết hợp công nghệ thị giác máy tính, học sâu và các nền tảng tối ưu cho thiết bị di động trở thành hướng tiếp cận phù hợp và mang tính ứng dụng cao.

Nền tảng lý thuyết của nghiên cứu dựa trên các kiến thức trọng tâm về thị giác máy tính, mạng nơ-ron nhân tạo, đặc biệt là các kiến trúc mạng nơ-ron tích chập (CNN) dùng cho bài toán phân loại và phát hiện đối tượng. Các mô hình như MobileNet, EfficientNet được xem xét nhờ khả năng cân bằng tốt giữa độ chính xác và tốc độ suy luận. TensorFlow được sử dụng trong giai đoạn huấn luyện và đánh giá mô hình, trong khi TensorFlow Lite đóng vai trò chính trong việc tối ưu, lượng tử hóa và triển khai mô hình trên thiết bị di động nhằm đảm

bảo tốc độ xử lý và tiết kiệm tài nguyên. Bên cạnh đó, hệ thống tích hợp Google Gemini và công nghệ Generative AI để mở rộng khả năng ứng dụng: giải thích kết quả phân loại, cung cấp thông tin sinh học và chế độ chăm sóc, hỗ trợ hỏi–đáp tương tác, hoặc tạo dữ liệu mô phỏng để tăng cường tập huấn luyện.

Quy trình xây dựng hệ thống gồm nhiều bước liên kết. Giai đoạn phân tích bài toán xác định rõ mục tiêu, phạm vi và ràng buộc của ứng dụng, đồng thời phân tích yêu cầu chức năng như chụp ảnh, nhận dạng cá, hiển thị thông tin và khả năng tương tác thông minh; cùng với yêu cầu phi chức năng như tốc độ xử lý, độ chính xác, hoạt động ổn định trên nhiều thiết bị và bảo vệ quyền riêng tư dữ liệu người dùng. Sau đó, hệ thống được thiết kế tổng thể bao gồm kiến trúc ứng dụng, các thành phần xử lý, thiết kế luồng hoạt động từ lúc người dùng chụp ảnh đến khi hiển thị kết quả, cũng như giao diện trực quan, dễ sử dụng với chế độ xem trực tiếp từ camera.

Về mặt dữ liệu, nghiên cứu xây dựng một tập ảnh đa dạng gồm nhiều loài cá cảnh phổ biến trong nhiều điều kiện ánh sáng, góc chụp và môi trường nước khác nhau. Tập dữ liệu được gán nhãn thủ công kết hợp các kỹ thuật tiền xử lý và tăng cường dữ liệu như chỉnh sáng, làm mờ, mô phỏng nước, xoay và cắt ảnh để tăng khả năng nhận diện ảnh. Sau khi chuẩn hóa dữ liệu, mô hình được huấn luyện bằng TensorFlow, đánh giá bằng các chỉ số accuracy, precision, recall, F1-score sau đó tối ưu hóa bằng lượng tử hóa trước khi chuyển đổi sang TensorFlow Lite.

Trong giai đoạn triển khai ứng dụng di động, mô hình TensorFlow Lite được tích hợp vào app Android/iOS sử dụng Flutter Framework. Có thể kết hợp pipeline xử lý ảnh theo thời gian thực và mô-đun tương tác dựa trên Gemini. Ứng dụng được thử nghiệm để đánh giá độ chính xác thực tế và trải nghiệm người dùng trong bối cảnh thực tế.

1.2. Khái quát về thị giác máy tính và mạng nơ-ron

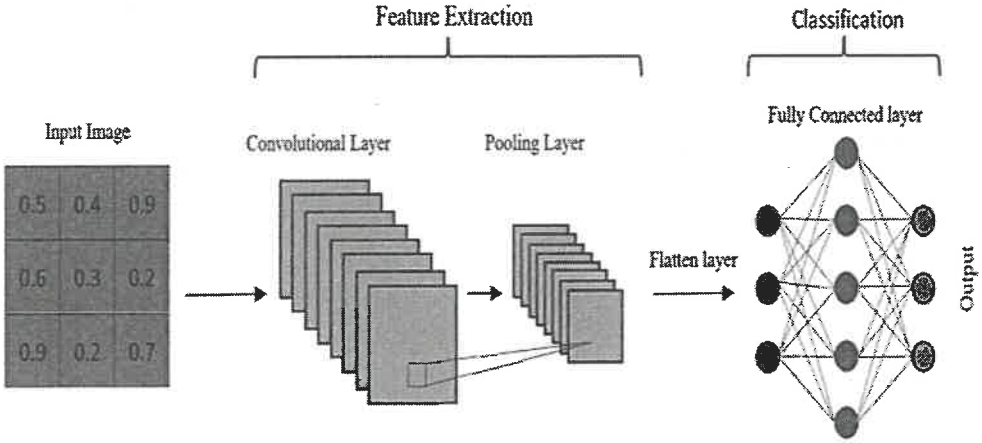
1.2.1. Bài toán phân loại hình ảnh

Phân loại hình ảnh là bài toán cốt lõi trong thị giác máy tính, mục tiêu là ánh xạ một hình ảnh đầu vào X (dưới dạng ma trận điểm ảnh) thành một nhãn Y thuộc tập hợp các lớp đã định nghĩa trước $C = \{c_1, c_2, \dots, c_n\}$

Trong ngữ cảnh đề tài, đầu vào là tensor 3 chiều biểu diễn hình ảnh RGB của cá cảnh (H x W x 3), và đầu ra là xác suất dự đoán $P(Y = c_i|X)$ cho từng loài cá (ví dụ: Betta, Goldfish, Arowana). Hệ thống sẽ chọn nhãn có xác suất cao nhất làm kết quả cuối cùng.

1.2.2. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)

CNN là kiến trúc mạng nơ-ron chuyên biệt cho dữ liệu dạng lưới (grid-like topology) như hình ảnh. So với mạng nơ-ron truyền thống (MLP), CNN giải quyết vấn đề bùng nổ tham số nhờ cơ chế chia sẻ trọng số (parameter sharing) và kết nối cục bộ (local connectivity).



Hình 1.1: Kiến trúc cơ bản của một mạng CNN

Các thành phần chính của CNN bao gồm:

- Lớp tích chập (Convolutional Layer): Thực hiện phép tính tích chập giữa ma trận đầu vào và các bộ lọc (filters/kernels) để tạo ra các bản đồ đặc trưng (Feature Maps). Công thức tích chập tại vị trí (i, j) được biểu diễn như sau:

$$S(i,j) = (I * K)(i,j) = z \sum_m \sum_n I(i+m, j+n)K(m,n)$$

Trong đó I là ảnh đầu vào và K là kernel. Lớp này giúp mạng học được các đặc trưng từ thấp (cạnh, góc) đến cao (mắt cá, vây cá).

- Hàm kích hoạt phi tuyến (Non-linear Activation Function): Thường sử dụng hàm ReLU (Rectified Linear Unit) để đưa tính phi tuyến vào mô hình, giúp mạng học được các mối quan hệ phức tạp.

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

- Lớp gộp (Pooling Layer): Dùng để giảm chiều dữ liệu (down-sampling), giúp giảm chi phí tính toán và chống hiện tượng quá khớp (overfitting). Phổ biến nhất là Max Pooling (lấy giá trị lớn nhất trong vùng lân cận).
- Lớp kết nối đầy đủ (Fully Connected Layer - FC): Nằm ở cuối mạng, các nơ-ron ở lớp này kết nối với tất cả các nơ-ron ở lớp trước đó để tổng hợp đặc trưng và đưa ra vector đặc trưng cuối cùng.
- Hàm Softmax: Được sử dụng ở lớp đầu ra để chuyển đổi các giá trị thô (logits) thành phân phối xác suất:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

1.2.3. Học chuyển giao (Transfer Learning)

Học chuyển giao là phương pháp tận dụng tri thức từ một mô hình đã được huấn luyện tốt trên tập dữ liệu nguồn lớn (Source Domain - ví dụ: ImageNet) để giải quyết bài toán trên tập dữ liệu đích nhỏ hơn (Target Domain - ví dụ: Dữ liệu cá cảnh).

Có hai chiến lược chính trong Transfer Learning được áp dụng:

- Trích xuất đặc trưng (Feature Extraction): Giữ đông (freeze) các lớp tích chập của mô hình gốc, chỉ thay thế và huấn luyện lại các lớp phân loại ở cuối.
- Tinh chỉnh (Fine-tuning): Sau khi huấn luyện lớp phân loại, tiếp tục rã đông (unfreeze) một số lớp tích chập cuối cùng của mô hình gốc và huấn luyện lại với tốc độ học rất nhỏ để mô hình thích nghi tốt hơn với đặc điểm cụ thể của loài cá.

1.3. Các kiến trúc mạng CNN tối ưu cho thiết bị di động

Việc triển khai AI trên thiết bị di động (Mobile AI) đặt ra thách thức lớn về sự cân bằng giữa độ chính xác (Accuracy) và tài nguyên tính toán (Latency, Model Size).

1.3.1. MobileNetV2 và Khối Inverted Residuals

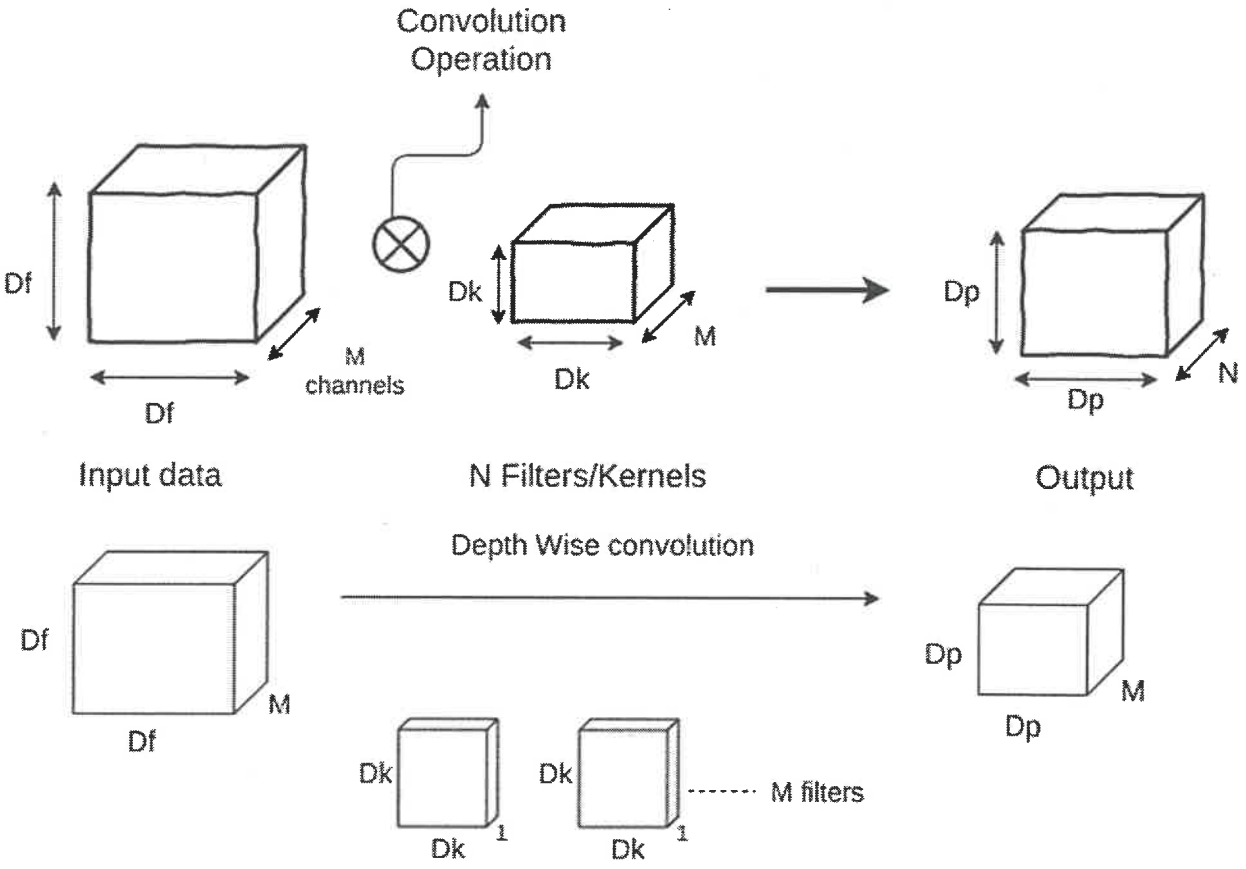
MobileNetV2 là kiến trúc state-of-the-art cho các ứng dụng di động. Cải tiến cốt lõi của nó dựa trên hai khái niệm:

A. Tích chập phân tách theo chiều sâu (Depthwise Separable Convolution)

Thay thế phép tích chập tiêu chuẩn tốn kém bằng hai phép toán nhẹ hơn:

- Depthwise Convolution: Mỗi bộ lọc chỉ áp dụng trên một kênh màu đầu vào (Input Channel).
- Pointwise Convolution (1x1 Conv): Kết hợp tuyến tính các đầu ra của Depthwise để tạo ra đặc trưng mới.

Hiệu quả: Giảm số lượng phép tính (Computation Cost) khoảng 8-9 lần so với tích chập thường.

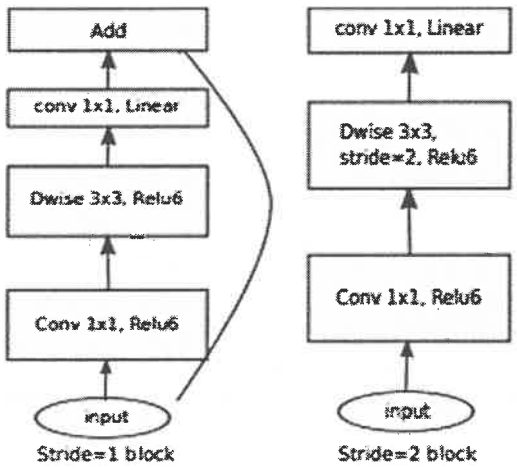


Hình 1.2: So sánh Standard Convolution và Depthwise Separable Convolution

B. Khối dư đảo ngược (Inverted Residual Block):

Khác với kiến trúc ResNet truyền thống (Wide -> Narrow -> Wide), MobileNetV2 sử dụng cấu trúc (Narrow -> Wide -> Narrow):

- Expansion (Mở rộng): Dùng 1x1 Conv để tăng số kênh lên (ví dụ: gấp 6 lần) nhằm tạo không gian biểu diễn phong phú hơn. Sử dụng hàm kích hoạt ReLU6.
- Depthwise Conv: Xử lý trích xuất đặc trưng trên không gian chiều cao (high-dimensional space).
- Projection (Chiếu): Dùng 1x1 Conv để nén số kênh xuống thấp lại (Linear Bottleneck). Tại đây không sử dụng hàm kích hoạt phi tuyến (như ReLU) để tránh làm mất mát thông tin.



Hình 1.3: Cấu trúc khối Inverted Residual của MobileNetV2 (Sandler et al., 2018)

1.3.2. EfficientNet-Lite

EfficientNet (Tan & Le, 2019) đề xuất phương pháp Compound Scaling (Quy mô hỗn hợp) - cân bằng đồng thời ba yếu tố: chiều sâu (depth), chiều rộng (width) và độ phân giải (resolution) của mạng.

Phiên bản EfficientNet-Lite được tinh chỉnh cho thiết bị biên bằng cách:

- Loại bỏ các khối Squeeze-and-Excitation (SE) vì chúng hoạt động kém hiệu quả trên các bộ gia tốc phần cứng di động.

- Thay thế hàm kích hoạt Swish (x . sigmoid(x)) bằng ReLU6 (min(max(0, x), 6)) để đơn giản hóa việc tính toán dấu phẩy động cố định.

1.3.3. So sánh lý thuyết và Tiêu chí lựa chọn mô hình

Để lựa chọn kiến trúc phù hợp nhất cho bài toán nhận diện cá cảnh trên thiết bị di động, cần xem xét sự đánh đổi giữa độ chính xác và tài nguyên tiêu thụ. Dưới đây là bảng so sánh các thông số lý thuyết của ba kiến trúc phổ biến khi huấn luyện trên tập dữ liệu chuẩn ImageNet:

Bảng 1.1: So sánh thông số lý thuyết các kiến trúc CNN

Kiến trúc	Tham số (Triệu)	Kích thước (MB)	ImageNet Top-1 Acc	FLOPs (Tỷ)
ResNet50	25.6	~98	76.0%	4.1
MobileNetV2 (1.0)	3.5	~14	71.8%	0.3
EfficientNet-B0	5.3	~21	77.1%	0.39

Phân tích so sánh:

1. **ResNet50:** Mặc dù là tiêu chuẩn vàng trong thị giác máy tính với độ ổn định cao, nhưng kích thước mô hình quá lớn (gần 100MB) và khối lượng tính toán (FLOPs) cao gấp 13 lần so với MobileNetV2. Điều này khiến nó không phù hợp cho các ứng dụng yêu cầu phản hồi tức thì (Real-time) trên các thiết bị di động có dung lượng pin và tản nhiệt giới hạn.
2. **EfficientNet-B0:** Mang lại độ chính xác cao nhất trong nhóm mô hình nhẹ nhờ cơ chế Compound Scaling. Tuy nhiên, kiến trúc này phức tạp hơn trong việc tinh chỉnh (fine-tuning) và đôi khi gặp vấn đề tương thích tối ưu với một số bộ tăng tốc phần cứng (NPU/DSP) đời cũ trên điện thoại Android giá rẻ.
3. **MobileNetV2:** Có sự cân bằng tốt nhất giữa hiệu năng và tài nguyên. Điểm mạnh nằm ở cấu trúc **Linear Bottlenecks** và việc sử dụng hàm kích hoạt **ReLU6**. ReLU6 được thiết kế đặc biệt để giảm thiểu mất mát thông tin khi thực hiện lượng tử hóa xuống 8-bit (Int8) – một bước bắt buộc khi chuyển đổi sang TensorFlow Lite.

Kết luận lựa chọn: Dựa trên các phân tích lý thuyết, đề tài lựa chọn **MobileNetV2** làm kiến trúc xương sống (Backbone) để nghiên cứu và triển khai. Kiến trúc này đảm bảo được yêu cầu về độ trễ thấp (<200ms) và kích thước ứng dụng nhỏ gọn (<50MB), phù hợp với mục tiêu xây dựng ứng dụng đại chúng cho người chơi cá cảnh.

1.4. Nền tảng TensorFlow Lite và tối ưu hóa mô hình

1.4.1. Kiến trúc TensorFlow Lite

TensorFlow Lite không chỉ là một phiên bản rút gọn mà là một bộ công cụ hoàn chỉnh bao gồm:

- Converter: Chuyển đổi mô hình TensorFlow (SavedModel, Keras h5) sang định dạng FlatBuffers (.tflite). FlatBuffers là định dạng tuần tự hóa hiệu quả, cho phép ánh xạ trực tiếp dữ liệu vào bộ nhớ mà không cần phân tích cú pháp (parsing) tốn kém.
- Interpreter: Trình thông dịch chạy trên thiết bị (Android/iOS), thực hiện tính toán đồ thị mô hình. Nó hỗ trợ tăng tốc phần cứng thông qua NNAPI (Android Neural Networks API) hoặc GPU Delegate.

1.4.2. Kỹ thuật Lượng tử hóa (Quantization)

Lượng tử hóa là kỹ thuật nén mô hình bằng cách giảm độ chính xác biểu diễn số học. Trong đề tài này, kỹ thuật Post-training Quantization (PTQ) được sử dụng.

Quy trình lượng tử hóa ánh xạ các giá trị số thực r (float32) sang số nguyên q (int8) thông qua công thức affine:

$$r = S \times (q - Z)$$

Trong đó:

- S (Scale): Hệ số tỷ lệ (float32).
- Z (Zero-point): Giá trị số nguyên tương ứng với giá trị thực 0.

Lợi ích cụ thể:

- Giảm bộ nhớ: Tham số 32-bit giảm xuống 8-bit (giảm 4 lần kích thước file).
- Tăng tốc độ: Các phép tính số học trên số nguyên (integer arithmetic) tiêu tốn ít năng lượng và chu kỳ CPU hơn so với số thực dấu phẩy động.

1.5. Google Gemini và Generative AI trong truy xuất thông tin

1.5.1. Kiến trúc Multimodal của Gemini

Google Gemini là mô hình nền tảng (Foundation Model) được huấn luyện trên tập dữ liệu đa phương thức khổng lồ. Khác với các LLM truyền thống chỉ xử lý văn bản, Gemini có khả năng xử lý đồng thời văn bản và hình ảnh (Native Multimodal).

Trong ứng dụng, chúng ta sử dụng Gemini như một Knowledge Engine (Động cơ tri thức). Thay vì hard-code thông tin sinh học của cá vào ứng dụng, Gemini cho phép truy xuất thông tin linh hoạt dựa trên ngữ cảnh.

1.5.2. Prompt Engineering và Contextual Learning

Để tối ưu hóa kết quả từ Gemini, kỹ thuật Prompt Engineering được áp dụng theo mô hình Few-shot Learning (Học từ vài ví dụ) hoặc Chain-of-Thought (Chuỗi suy luận).

Quy trình xử lý thông tin:

- Input: Nhãn loài cá dự đoán từ TFLite (ví dụ: "Symphysodon aequifasciatus").
- Prompt Construction: Hệ thống tự động điền nhãn vào khuôn mẫu: "Với tư cách là chuyên gia thủy sinh, hãy cung cấp thông tin về loài [Input] theo các mục: pH, Nhiệt độ, Thức ăn..."
- Generation: Gemini sinh văn bản dựa trên xác suất kế tiếp của các token. Tham số Temperature được thiết lập thấp (ví dụ: 0.2) để đảm bảo tính chính xác và nhất quán của thông tin khoa học, tránh hiện tượng "ảo giác" (hallucination).

1.6. Các chỉ số đánh giá mô hình (Evaluation Metrics)

Để đánh giá hiệu quả của mô hình phân loại cá cảnh, đề tài sử dụng các chỉ số sau:

- Độ chính xác (Accuracy): Tỷ lệ số ảnh dự đoán đúng trên tổng số ảnh.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- Precision (Độ chính xác của lớp dự đoán): Trong các ảnh được dự đoán là loài A, bao nhiêu % thực sự là loài A. Quan trọng khi muốn giảm thiểu báo động giả.
- Recall (Độ phủ): Trong thực tế có bao nhiêu ảnh loài A, mô hình tìm ra được bao nhiêu %.
- F1-Score: Trung bình điều hòa giữa Precision và Recall, dùng để đánh giá khi tập dữ liệu bị mất cân bằng (imbalanced dataset).

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Confusion Matrix (Ma trận nhầm lẫn): Bảng biểu diễn chi tiết sự nhầm lẫn giữa các lớp cá (ví dụ: hay nhầm Cá vàng với Cá chép), giúp phân tích sai sót cụ thể của mô hình.

1.7. Kết chương

Chương 1 đã trình bày cơ sở lý thuyết cho bài toán nhận diện và phân loại cá cảnh dựa trên hình ảnh. Trước hết, chương giới thiệu đặc thù của hình ảnh cá cảnh trong môi trường nước và các thách thức trong việc nhận dạng. Tiếp đó, các kiến thức nền tảng của thị giác máy tính và mạng nơ-ron được trình bày, bao gồm bài toán phân loại ảnh, mạng CNN và kỹ thuật học chuyển giao để tận dụng mô hình tiền huấn luyện.

Nội dung chương đã giới thiệu các kiến trúc CNN tối ưu cho thiết bị di động như MobileNetV2 và EfficientNet-Lite, vốn phù hợp cho việc triển khai mô hình trên smartphone. Nền tảng TensorFlow Lite và các kỹ thuật tối ưu hóa, đặc biệt là lượng tử hóa, được đề cập nhằm nâng cao hiệu năng xử lý trên thiết bị tài nguyên thấp. Bên cạnh đó, chương khái quát vai trò của Google Gemini và Generative AI trong việc truy xuất thông tin và hỗ trợ người dùng thông qua kiến trúc đa phương thức và kỹ thuật Prompt Engineering. Đề án cũng đã tổng hợp

các chỉ số đánh giá mô hình thường dùng như accuracy, precision, recall, F1-score và mAP.

Những nội dung trong chương 1 là cơ sở lý thuyết cho các phần phân tích, thiết kế và xây dựng hệ thống ở các chương tiếp theo.

CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

Chương này tập trung vào phân tích và thiết kế hệ thống ứng dụng di động nhận diện và phân loại cá cảnh. Phần đầu trình bày phân tích yêu cầu, gồm nhóm yêu cầu chức năng như chụp ảnh, phát hiện và phân loại cá, hiển thị thông tin loài và truy vấn qua Gemini; cùng với yêu cầu phi chức năng như tốc độ xử lý, độ chính xác mô hình, khả năng mở rộng và trải nghiệm người dùng mượt mà. Tiếp theo, chương mô hình hóa các yêu cầu bằng biểu đồ Use Case để mô tả các tác nhân và tương tác chính trong hệ thống.

Trong nội dung Chương 2, đề án trình bày kiến trúc tổng thể của hệ thống gồm hai thành phần: ứng dụng di động phía client chịu trách nhiệm xử lý ảnh và suy luận bằng TensorFlow Lite, và thành phần cloud tích hợp Generative AI để truy xuất thông tin và hỗ trợ người dùng. Phần thiết kế chi tiết mô tả rõ hai luồng xử lý: quy trình suy luận ảnh (TFLite Flow) từ khi người dùng chụp ảnh đến khi có kết quả phân loại, và quy trình tích hợp Gemini (Prompt Flow) hỗ trợ tạo thông tin bổ sung. Các biểu đồ tuần tự được sử dụng để mô tả các bước tương tác giữa các thành phần trong những luồng xử lý chính.

Cuối cùng, đề án trình bày thiết kế giao diện người dùng với hai màn hình chính: màn hình Home để chụp ảnh và quan sát quá trình nhận diện, và màn hình Result hiển thị loài cá, độ tin cậy và nội dung được tạo bởi AI. Những thiết kế này đảm bảo ứng dụng trực quan, dễ sử dụng và tối ưu cho trải nghiệm thực tế của người dùng.

2.1. Phân tích yêu cầu

Việc thiết kế hệ thống cần bắt đầu từ phân tích đầy đủ các yêu cầu chức năng và phi chức năng của ứng dụng. Về chức năng, ứng dụng phải hỗ trợ thu thập hình ảnh linh hoạt thông qua camera tích hợp hoặc thư viện ảnh trên thiết bị. Chức năng nhận diện cá cần hoạt động hoàn toàn ngoại tuyến, sử dụng mô hình TensorFlow Lite để phát hiện và phân loại loài cá trong ảnh, đồng thời hiển thị tên loài và độ tin cậy của dự đoán. Sau khi nhận diện, hệ thống phải có khả năng gửi thông tin loài cá lên Google Gemini để nhận về dữ liệu chuyên sâu như nguồn gốc, điều kiện nước phù hợp hay tập tính.

Bên cạnh đó, các yêu cầu phi chức năng đóng vai trò quan trọng trong đảm bảo chất lượng trải nghiệm. Mô hình suy luận phải có hiệu năng đủ cao để xử lý ảnh trong thời gian ngắn. Độ chính xác của hệ thống cần đạt mức tin cậy cao, đảm bảo mô hình phân loại ổn định trong nhiều điều kiện ánh sáng và nền ảnh phức tạp. Dung lượng mô hình sau tối ưu phải nhỏ để tiết kiệm tài nguyên thiết bị và giảm tiêu thụ pin. Khi tích hợp AI đám mây, hệ thống cần duy trì thời gian phản hồi hợp lý, đồng thời đảm bảo an toàn dữ liệu và tính ổn định khi giao tiếp với dịch vụ Gemini. Tập hợp những yêu cầu này tạo nền tảng cho việc thiết kế kiến trúc, giao diện và luồng xử lý của ứng dụng.

2.1.1. Yêu cầu chức năng

Hệ thống được thiết kế để hỗ trợ toàn bộ quy trình nhận diện và tra cứu thông tin cá cảnh một cách thuận tiện và chính xác. Ứng dụng cần cung cấp khả năng thu thập hình ảnh linh hoạt, cho phép người dùng chụp ảnh trực tiếp bằng camera hoặc chọn ảnh sẵn có từ thư viện. Chức năng nhận diện loài cá phải hoạt động ngoại tuyến, sử dụng mô hình TFLite để phát hiện và phân loại cá ngay trên thiết bị, đồng thời hiển thị tên loài và độ tin cậy của dự đoán. Sau khi nhận dạng, ứng dụng tích hợp Google Gemini nhằm cung cấp thông tin chuyên sâu về loài cá, bao gồm tên khoa học, nguồn gốc, các thông số nước phù hợp, thức ăn và đặc tính hành vi. Bên cạnh đó, hệ thống cần lưu lại lịch sử các lần nhận diện để người dùng có thể xem lại kết quả và thông tin đã tra cứu. Các chức năng này là cơ sở để thiết kế giao diện, luồng xử lý và kiến trúc hệ thống ở các phần tiếp theo.

Hệ thống cần đáp ứng các chức năng cốt lõi sau để phục vụ người dùng (người chơi cá cảnh):

1. Thu thập hình ảnh đầu vào:

- Cho phép người dùng chụp ảnh trực tiếp từ camera trong ứng dụng.
- Cho phép người dùng chọn ảnh có sẵn từ thư viện thiết bị.

2. Nhận diện loài cá (Offline Identification):

- Hệ thống phải tự động phát hiện và phân loại loài cá trong ảnh mà không cần kết nối Internet.
- Hiển thị tên loài cá và độ tin cậy (Confidence Score) của dự đoán.

3. Cung cấp thông tin chuyên sâu (AI Consultation):

- Gửi tên loài cá đã nhận diện đến Google Gemini.
- Hiển thị thông tin chi tiết về: Tên khoa học, Nguồn gốc, Độ pH thích hợp, Nhiệt độ nước, Thức ăn và Tính cách của cá.

2.1.2. Yêu cầu phi chức năng

1. Hiệu năng (Performance):
    1. Thời gian suy luận (Inference time) của mô hình TFLite phải dưới 200ms trên các thiết bị tầm trung.
    2. Thời gian phản hồi từ Gemini API (khi có mạng ổn định) nên dưới 3 giây.
    3. Độ chính xác (Accuracy): Mô hình nhận diện cần đạt độ chính xác trên 90% đối với các ảnh chụp rõ nét, đủ sáng.
    4. Tính khả dụng (Usability): Giao diện đơn giản, tối giản hóa thao tác (chỉ cần 1-2 chạm để ra kết quả).
    5. Tương thích (Compatibility): Ứng dụng hoạt động ổn định trên hệ điều hành Android (từ phiên bản 8.0 trở lên) và iOS (tùy chọn).

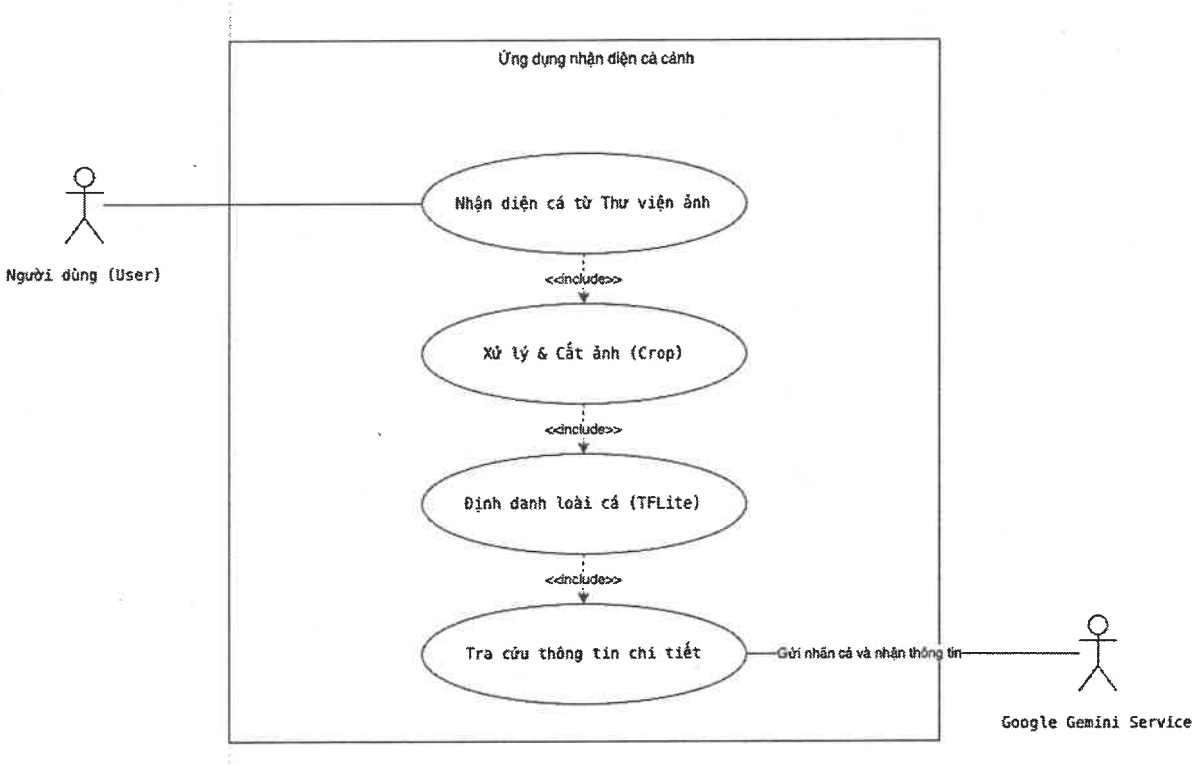
2.2. Mô hình hóa yêu cầu

Từ việc phân tích các yêu cầu chức năng và phi chức năng ở trên, để mô hình hóa yêu cầu của hệ thống nhận diện cá cảnh, đề án cần xây dựng biểu đồ Use Case nhằm thể hiện mối quan hệ giữa người dùng và các chức năng cốt lõi. Actor chính của hệ thống là Người dùng, tương tác với bốn ca sử dụng: Thu thập hình ảnh, Nhận diện cá offline, Tư vấn thông tin AI, và Xem lịch sử nhận diện. Actor phụ là Google Gemini, đóng vai trò cung cấp dữ liệu chuyên sâu khi hệ thống gửi tên loài cá để truy vấn. Các luồng sự kiện chính bao gồm: người dùng chụp hoặc tải ảnh; hệ thống thực hiện suy luận mô hình để phát hiện loài cá; nếu nhận diện thành công, hệ thống gửi kết quả đến Gemini và trả về thông tin chi tiết; cuối cùng lưu lại bản ghi lịch sử cho phép người dùng xem lại. Biểu đồ Use Case phải mô tả đầy đủ các tương tác này, giúp làm rõ phạm vi hệ thống và xác định giao tiếp giữa các thành phần.

Trong các phần sau đây là mô tả chi tiết các nội dung nêu trên.

2.2.1. Biểu đồ Ca sử dụng

Biểu đồ dưới đây mô tả sự tương tác giữa Người dùng (User) và Hệ thống.



Hình 2.1: Sơ đồ Use case ứng dụng nhận diện cá cảnh

Danh sách các Actor:

- User: Người sử dụng ứng dụng để tra cứu thông tin cá.
- Google Gemini Service: Hệ thống bên ngoài cung cấp tri thức.

Đặc tả Use Case chính "Nhận diện cá":

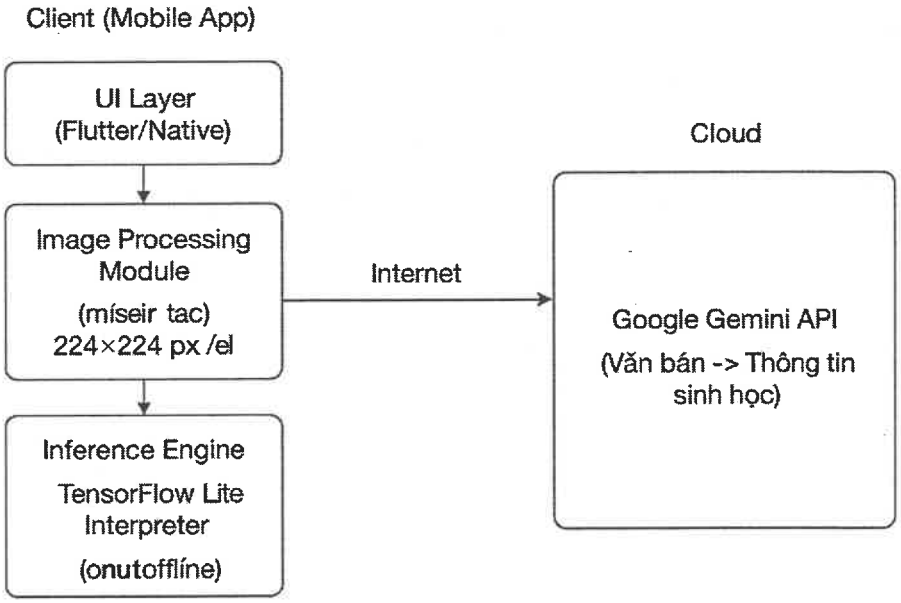
- Tiền điều kiện: Người dùng đã mở ứng dụng và cấp quyền Camera/Thư viện.
- Luồng sự kiện chính:
  1. User chọn chức năng “Chọn ảnh từ thư viện”
  2. Hệ thống hiển thị Danh sách hình ảnh trong thiết bị.
  3. User chọn một hình ảnh.
  4. Hệ thống cắt (crop) và xử lý ảnh.
  5. Hệ thống chạy mô hình TFLite để lấy nhãn (Label).
  6. Hệ thống gửi nhãn sang Gemini để lấy thông tin chi tiết.
  7. Hệ thống hiển thị kết quả tổng hợp.

2.3. Thiết kế kiến trúc hệ thống

2.3.1. Sơ đồ kiến trúc tổng thể hệ thống

Đề tài sử dụng kiến trúc Hybrid AI Architecture (Kiến trúc AI lai), tận dụng ưu điểm của cả Edge Computing (Điện toán biên) và Cloud Computing (Điện toán đám mây) như biểu thị trên hình 2.2.

Kiến trúc của hệ thống được thiết kế theo mô hình client–cloud lai nhằm đảm bảo vừa tối ưu hiệu năng xử lý tại chỗ, vừa tận dụng được sức mạnh mô hình AI lớn trên đám mây. Ở phía Client (Mobile App), toàn bộ quá trình xử lý ảnh và nhận diện loài cá được thực hiện hoàn toàn offline, bao gồm lớp giao diện UI để tương tác với người dùng, mô-đun xử lý ảnh để chuẩn hóa dữ liệu đầu vào, và động cơ suy luận sử dụng TensorFlow Lite Interpreter để tải mô hình .tflite và dự đoán nhãn nhanh, tiết kiệm tài nguyên. Sau khi nhận diện xong, ứng dụng chỉ gửi tên loài cá qua Internet tới thành phần Cloud, nơi Google Gemini API tiếp nhận thông tin và sinh ra mô tả sinh học chuyên sâu. Việc phân tách chức năng như vậy giúp hệ thống hoạt động ổn định dù không có Internet, giảm độ trễ khi suy luận, tăng tính riêng tư của ảnh người dùng, đồng thời vẫn cung cấp thông tin phong phú nhờ sức mạnh của mô hình AI nền tảng.



Hình 2.2: Sơ đồ kiến trúc hệ thống

2.3.2. Thành phần Client (Mobile App)

Đây là nơi diễn ra tương tác chính và xử lý mô hình thị giác máy tính.

- UI Layer (Flutter/Native): Quản lý giao diện người dùng.
- Image Processing Module: Thực hiện Resize ảnh về kích thước input của mô hình (ví dụ: 224 x 224), chuẩn hóa pixel về khoảng [0, 1] hoặc [-1, 1].
- Inference Engine (TFLite Interpreter): Nạp file .tflite và thực hiện dự đoán nhãn. Thành phần này hoạt động hoàn toàn Offline.

2.3.3. Thành phần Cloud (Generative AI)

Google Gemini API: Tiếp nhận đầu vào là văn bản (Tên loài cá) và trả về nội dung mô tả sinh học được định dạng sẵn (Structured Text).

Thành phần này yêu cầu kết nối Internet.

2.4. Thiết kế chi tiết luồng xử lý

Thiết kế chi tiết các luồng xử lý của hệ thống cần mô tả rõ ràng cách dữ liệu di chuyển giữa các thành phần và cách từng bước được thực thi. Đối với quy trình xử lý ảnh và suy luận (TFLite Flow), hệ thống phải bắt đầu từ ảnh gốc (Raw Bitmap), sau đó thực hiện center crop để lấy vùng trung tâm tránh méo hình, resize về kích thước chuẩn của mô hình MobileNetV2, tiếp tục normalization chuyển giá trị RGB từ 0–255 sang Float32 trong khoảng [-1, 1], tạo Tensor đưa vào Interpreter để suy luận, rồi post-processing nhằm tìm chỉ số nhãn dự đoán và ánh xạ ngược sang tên loài cá. Với quy trình tích hợp Gemini (Prompt Flow), ứng dụng sẽ xây dựng một prompt động dựa trên tên loài cá đã nhận diện, gửi tới Gemini API và yêu cầu phản hồi dưới dạng JSON để dễ phân tích và hiển thị trực quan. Thiết kế chi tiết cũng cần mô tả biểu đồ tuần tự (Sequence Diagram) thể hiện tương tác giữa người dùng, ứng dụng, TFLite và Gemini. Cuối cùng, phần giao diện người dùng phải được mô tả cụ thể về cấu trúc màn hình, cách hiển thị kết quả và luồng điều hướng nhằm đảm bảo trải nghiệm trực quan và nhất quán.

2.4.1. Quy trình xử lý ảnh và suy luận (TFLite Flow)

Để đảm bảo mô hình hoạt động chính xác, ảnh đầu vào từ camera (thường có độ phân giải cao) phải trải qua quy trình biến đổi nghiêm ngặt khớp với quy trình huấn luyện:

1. Input: Ảnh gốc (Raw Bitmap), ví dụ 1920 x 1080

2. Center Crop: Cắt lấy phần trung tâm hình vuông để tránh méo ảnh.

3. Resize: Thu nhỏ về 224 x 224 (kích thước chuẩn của MobileNetV2).

4. Normalization: Chuyển đổi giá trị màu RGB (0-255) sang Float32.

$$\text{Pixel}_{\text{new}} = \frac{\text{Pixel}_{\text{old}} - 127.5}{127.5}$$

(Chuẩn hóa về khoảng [-1, 1]).

5. Inference: Đưa Tensor vào Interpreter -> Output là mảng xác suất (Probability Array).

6. Post-processing: Tìm chỉ số (index) có giá trị lớn nhất (ArgMax) và ánh xạ sang file labels.txt để lấy tên loài cá.

2.4.2. Quy trình tích hợp Gemini (Prompt Flow)

Sau khi có tên loài cá (ví dụ: "Astronotus ocellatus" - Cá Heo Lửa), hệ thống sẽ xây dựng một Prompt động để gửi tới Gemini.

Mẫu Prompt thiết kế:

"Bạn là một chuyên gia sinh vật cảnh. Hãy cung cấp thông tin về loài cá: {Fish_Name}. Trả về kết quả dưới định dạng JSON với các trường sau:

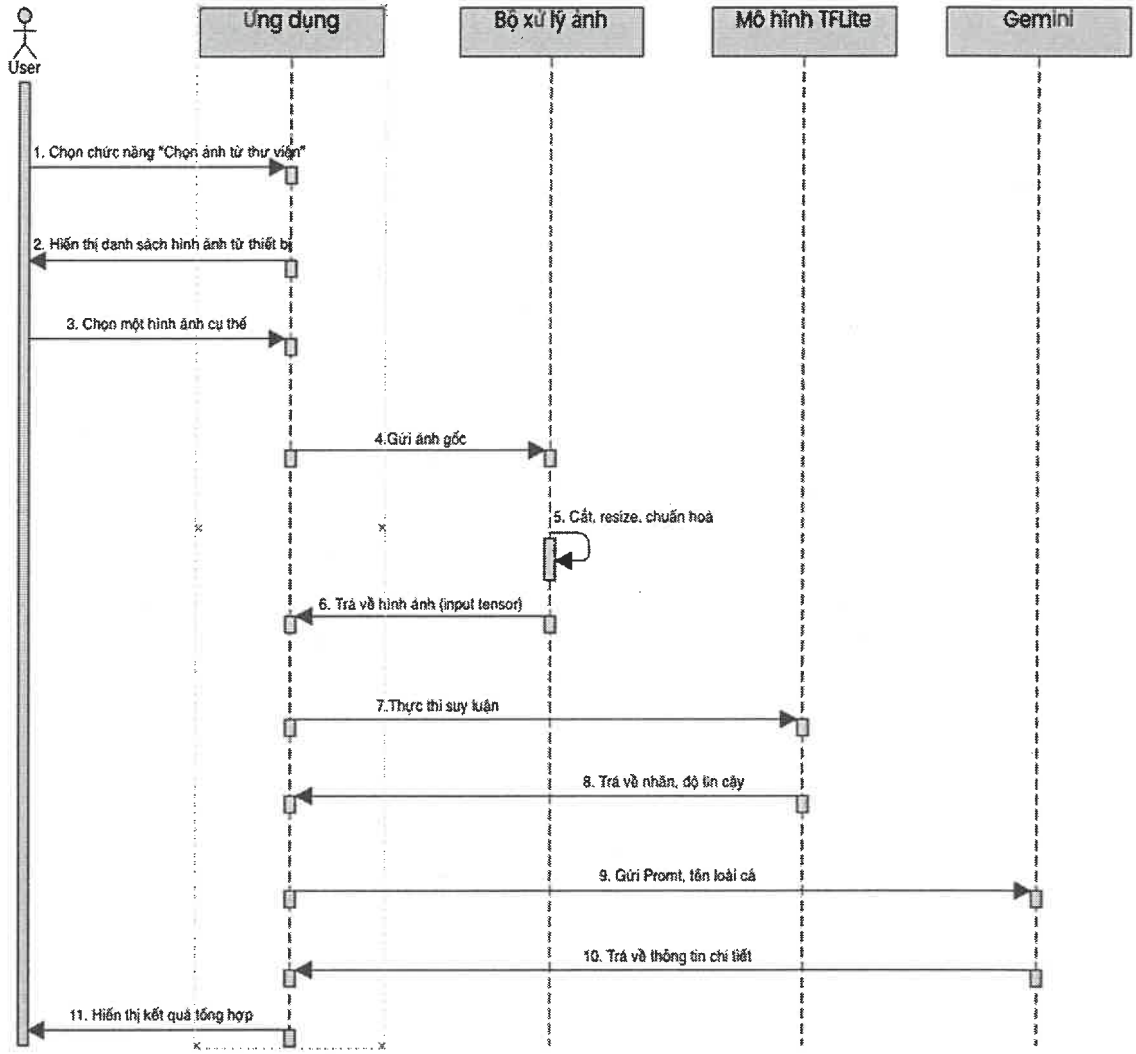
- ten_tieng_viet: String
- do_ph: String (ví dụ '6.0 - 7.5')
- nhiet_do: String (ví dụ '24 - 28 độ C')
- thuc_an: String
- luu_y_khi_nuoi: String (ngắn gọn trong 50 từ)"

Việc yêu cầu trả về JSON giúp ứng dụng dễ dàng phân tích (parse) và hiển thị lên giao diện đẹp mắt hơn là văn bản thô.

2.4.3. Biểu đồ tuần tự (Sequence Diagram)

Trong quá trình thiết kế hệ thống, biểu đồ tuần tự (Sequence Diagram) là thành phần quan trọng nhằm mô tả rõ ràng cách các đối tượng tương tác theo trình tự thời gian. Biểu đồ cần thể hiện đầy đủ chuỗi hành động bắt đầu từ khi User chọn chức năng "Chọn ảnh từ thư viện", hệ thống hiển thị danh sách ảnh và người dùng chọn một hình. Ảnh gốc sau đó được truyền vào Bộ xử lý ảnh, nơi

thực hiện các bước crop, resize và chuẩn hoá trước khi trả về input tensor phục vụ suy luận. Tiếp theo, mô hình TFLite được kích hoạt để dự đoán nhãn loài cá và trả về nhãn cùng độ tin cậy. Ứng dụng dùng nhãn này gửi yêu cầu đến Gemini để truy xuất thông tin chi tiết, nhận lại dữ liệu mô tả và cuối cùng tổng hợp, hiển thị kết quả lên giao diện. Biểu đồ tuần tự giúp minh họa rõ ràng dòng chảy dữ liệu và sự phối hợp giữa các thành phần, là cơ sở bảo đảm hệ thống hoạt động chính xác và thống nhất.



Hình 2.3: Biểu đồ tuần tự quy trình nhận diện

Biểu đồ dưới đây tóm tắt toàn bộ vòng đời của một lần nhận diện:

1. User chọn chức năng "Chọn ảnh từ thư viện"
2. Hệ thống hiển thị Danh sách hình ảnh trong thiết bị.
3. User chọn một hình ảnh.
4. Ứng dụng gửi ảnh gốc cho Bộ xử lý ảnh

5. Bộ xử lý ảnh cắt (crop), thay đổi kích thước (resize) và chuẩn hoá.
6. Trả về hình ảnh dưới dạng input tensor
7. Hệ thống chạy mô hình TFLite để lấy nhãn (Label).
8. Mô hình TFLite trả về nhãn và độ tin cậy
9. Ứng dụng gửi nhãn sang Gemini để lấy thông tin chi tiết.
10. Gemini trả về thông tin chi tiết cho ứng dụng
11. Hệ thống hiển thị kết quả tổng hợp.

2.5. Thiết kế giao diện người dùng (UI Design)

Phần thiết kế giao diện người dùng của ứng dụng tập trung vào sự tối giản, trực quan và tối ưu trải nghiệm thao tác nhanh. Màn hình chính (Home Screen) được bố trí với một nút lớn "Nhận diện ngay" đặt ở vị trí trung tâm nhằm dẫn người dùng mở camera tức thì, kèm theo một nút phụ "Tải ảnh lên" để chọn ảnh từ thư viện. Dưới hai nút chức năng là danh sách cuộn ngang hiển thị "Lịch sử nhận diện gần đây", giúp người dùng nhanh chóng xem lại các loài cá đã từng được phân loại. Trong khi đó, Màn hình kết quả (Result Screen) được thiết kế theo bố cục ba phần: phần trên (Header) hiển thị ảnh đã chụp hoặc tải lên; phần giữa (Instant Result) thể hiện tên loài cá với font chữ lớn, đậm và một thanh "Confidence bar" trực quan; phần dưới là khu vực thông tin nâng cao từ AI (AI Insights), có hiển thị skeleton khi đang tải dữ liệu. Khu vực này được chia thành ba tab: tab "Môi trường" cung cấp các thông số nhiệt độ, pH kèm icon minh họa; tab "Chăm sóc" mô tả chế độ ăn và tính cách; và tab "Tư vấn" hiển thị lời khuyên chuyên sâu do Gemini tạo ra. Thiết kế này đảm bảo trải nghiệm liền mạch, dễ quan sát và phù hợp với thói quen người dùng di động.

Các màn hình chức năng chính của giao diện người dùng gồm: Màn hình chính, màn hình kết quả. Giao diện người dùng và màn hình kết quả được thiết kế với bố cục chính như sau.

2.5.1. Màn hình chính (Home Screen)

- Nút lớn: "Nhận diện ngay" (Mở Camera).
- Nút phụ: "Tải ảnh lên" (Mở Gallery).
- Danh sách cuộn ngang: "Lịch sử nhận diện gần đây".

2.5.2. Màn hình kết quả

- Phần trên (Header): Hiển thị ảnh vừa chụp.
- Phần giữa (Instant Result): Tên loài cá (Font lớn, đậm) và thanh Confidence bar.
- Phần dưới (AI Insights - Loading skeleton khi chờ):
  + Tab "Môi trường": Hiển thị Nhiệt độ, pH (icon nhiệt kế, giọt nước).
  + Tab "Chăm sóc": Hiển thị Thức ăn, Tính cách.
  + Tab "Tư vấn": Lời khuyên từ Gemini.

2.6. Kết chương

Chương 2 đã trình bày một cách hệ thống và toàn diện các bước nền tảng để xây dựng ứng dụng nhận diện và phân loại cá cảnh trên thiết bị di động.

Trước hết, quá trình phân tích yêu cầu đã xác định rõ các chức năng cốt lõi của hệ thống—từ thu thập ảnh, xử lý và suy luận bằng TFLite, đến truy xuất thông tin nâng cao qua Gemini—cùng các yêu cầu phi chức năng như hiệu năng, độ trễ suy luận và tính ổn định. Tiếp theo, phần mô hình hóa yêu cầu và biểu đồ ca sử dụng giúp hình dung mối quan hệ giữa người dùng và hệ thống, làm rõ ranh giới trách nhiệm của từng tác nhân và luồng tương tác chính.

Thiết kế kiến trúc tổng thể được xây dựng đã mô tả cấu trúc ứng dụng theo hai thành phần chính: Client hoạt động offline với mô hình TFLite và dịch vụ Cloud Gemini đảm nhiệm xử lý ngôn ngữ và sinh thông tin nâng cao. Các thành phần chức năng như UI Layer, Image Processing Module, Inference Engine và Gemini API được phân rã rõ ràng, đảm bảo khả năng mở rộng và bảo trì. Phần thiết kế chi tiết luồng xử lý, bao gồm quy trình xử lý ảnh, luồng gửi prompt đến Gemini và biểu đồ tuần tự, thể hiện cụ thể cách dữ liệu di chuyển trong hệ thống theo trình tự thời gian, đảm bảo sự mạch lạc và logic trong hoạt động.

Đề án cũng đã đưa ra thiết kế giao diện người dùng, phác họa một UI trực quan, hiện đại và thân thiện, gồm màn hình chính với các chức năng trọng tâm và màn hình kết quả hiển thị thông tin phân loại cùng dữ liệu AI nâng cao.

Trong chương tiếp theo, đề án sẽ trình bày phần xây dựng và triển khai thử nghiệm hệ thống dựa vào các yêu cầu và kiến trúc đã chỉ ra trong chương 2.

CHƯƠNG 3:
XÂY DỰNG VÀ TRIỂN KHAI THỬ NGHIỆM HỆ THỐNG

Bước xây dựng và thử nghiệm hệ thống tập trung vào toàn bộ quy trình triển khai mô hình và ứng dụng trên môi trường thực tế. Trước hết, nhóm công cụ và môi trường thực nghiệm được thiết lập bao gồm GPU NVIDIA Tesla T4 trên Google Colab, Python, TensorFlow 2.x, Keras cho huấn luyện mô hình, cùng Flutter SDK, Android Studio và Visual Studio Code để phát triển ứng dụng; các thư viện như tflite_flutter và google_generative_ai được sử dụng cho suy luận cục bộ và tích hợp Gemini. Dữ liệu được thu thập từ nhiều nguồn như Kaggle, Open Images và ảnh chụp thực tế, sau đó được tăng cường bằng ImageDataGenerator với các kỹ thuật rescale, xoay, dịch chuyển và lật ảnh nhằm cải thiện khả năng khái quát hóa. Mô hình được huấn luyện theo chiến lược Transfer Learning, gồm giai đoạn trích đặc trưng bằng Base Model và tinh chỉnh phần Classification Head. Sau khi có kết quả huấn luyện, mô hình được tối ưu hóa và đóng gói sang định dạng TFLite phục vụ suy luận offline. Tiếp đó, quá trình xây dựng ứng dụng di động được tiến hành với việc triển khai module nhận diện cục bộ bằng TFLite, module tích hợp tri thức từ Gemini, cùng giao diện người dùng hoàn chỉnh. Cuối cùng, ứng dụng được chạy thử và đánh giá nhằm kiểm tra độ chính xác, tốc độ suy luận và trải nghiệm người dùng.

Phần cuối chương trình bày các kết quả thực nghiệm thu được sau quá trình huấn luyện mô hình và triển khai ứng dụng. Các đánh giá được thực hiện trên hai khía cạnh chính: hiệu năng kỹ thuật của mô hình nhận diện (Model Performance) và trải nghiệm người dùng trên ứng dụng thực tế (Application Usability).

3.1. Môi trường và công cụ thử nghiệm

Để đảm bảo tính nhất quán và khả năng tái lập kết quả, đề tài được thực hiện trên môi trường phần cứng và phần mềm với các thông số kỹ thuật như sau:

Bảng 3.1: Cấu hình môi trường phát triển

Thành phần	Công cụ / Thông số	Mục đích sử dụng
Phần cứng	GPU NVIDIA Tesla T4 (Google Colab)	Tăng tốc quá trình huấn luyện mô hình Deep Learning.
Ngôn ngữ	Python 3.9, Dart (Flutter SDK)	Xử lý dữ liệu, huấn luyện mô hình và lập trình ứng dụng.
Framework	TensorFlow 2.x, Keras	Xây dựng kiến trúc mạng MobileNetV2 và Transfer Learning.
Mobile Libs	tflite_flutter, google_generative_ai	Triển khai mô hình xuống thiết bị và kết nối LLM.
IDE	Visual Studio Code, Android Studio	Môi trường lập trình tích hợp và giả lập thiết bị.

3.2. Xây dựng và tiền xử lý dữ liệu

3.2.1. Thu thập và tổ chức dữ liệu

Dữ liệu đầu vào đóng vai trò quyết định đến hiệu suất và khả năng tổng quát hóa của mô hình học sâu. Để xây dựng một bộ dữ liệu chất lượng, tiến hành thu thập từ ba nguồn chính, kết hợp giữa các bộ dữ liệu chuẩn quốc tế và dữ liệu thực tế tại địa phương:

- **Dữ liệu từ nền tảng Kaggle:** khai thác và trích xuất các hình ảnh phù hợp từ các bộ dữ liệu nổi tiếng như:
    + **"A Large-Scale Fish Dataset"**: Cung cấp các hình ảnh cá với độ phân giải cao, tập trung vào các đặc điểm hình thái học.
    + **"Fish Identification Dataset"**: Chứa nhiều biến thể về góc chụp và môi trường ánh sáng khác nhau của các loài cá cảnh phổ biến.
- **Dữ liệu từ Open Images Dataset (V7):** sử dụng công cụ truy vấn để lọc các hình ảnh thuộc lớp **"Fish"** và các lớp con (sub-classes) như

"Goldfish", **"Shark"**, và **"Marine invertebrates"** (chỉ lọc lấy các loài cá cảnh tương ứng). Nguồn này cung cấp các hình ảnh cá trong môi trường tự nhiên và bể nuôi phức tạp, giúp mô hình tăng khả năng nhận diện vật thể trong điều kiện có vật cản.

- **Dữ liệu thu thập thực tế:** Để đảm bảo tính ứng dụng cao tại thị trường Việt Nam, thực hiện đã trực tiếp ghi hình và chụp ảnh tại các cửa hàng cá cảnh và các câu lạc bộ thủy sinh. Nguồn dữ liệu này giúp mô hình thích nghi tốt với các điều kiện thực tế như: khúc xạ ánh sáng qua kính, nước đục, và các loại đèn chuyên dụng cho bể cá (đèn hồng, đèn actinic).
- **Quy mô và Phân bổ dữ liệu:**
    + **Số lượng:** Tổng cộng **664 hình ảnh** đã được làm sạch và chuẩn hóa.
    + **Phân loại:** Dữ liệu bao gồm **23 lớp (classes)**, tương ứng với 23 loài cá cảnh phổ biến (như Cá Betta, Cá Vàng, Cá Đĩa, Cá Rồng, Cá Phượng Hoàng...).
    + **Tổ chức dữ liệu:** Bộ dữ liệu được chia theo tỷ lệ **80:20**, cụ thể:
        o **Tập huấn luyện (Training set - 80%):** Dùng để huấn luyện các tham số của mạng MobileNetV2.
        o **Tập kiểm thử (Validation set - 20%):** Dùng để đánh giá độc lập hiệu suất của mô hình, đảm bảo mô hình không bị hiện tượng quá khớp (overfitting).

3.2.2. Kỹ thuật tăng cường dữ liệu (Data Augmentation)

Do hạn chế về số lượng mẫu dữ liệu cho mỗi loài cá, đề tài áp dụng kỹ thuật Data Augmentation để tạo ra các biến thể nhân tạo của ảnh gốc. Phương pháp này giúp mô hình học được các đặc trưng bất biến (invariant features) và giảm thiểu hiện tượng quá khớp (Overfitting).

Các phép biến đổi hình học được áp dụng thông qua lớp ImageDataGenerator của Keras bao gồm:

- Rescale: Chuẩn hóa giá trị pixel về khoảng [0,1].
- Rotation: Xoay ảnh ngẫu nhiên trong khoảng 20 độ.

- Shift: Dịch chuyển khung hình theo chiều ngang/dọc (20%).
- Flip: Lật ảnh theo chiều ngang (mô phỏng cá bơi ngược chiều).

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Hình 3.1: Cấu hình tham số cho quá trình tăng cường dữ liệu

3.3. Huấn luyện mô hình nhận diện (Model Training)

3.3.1. Cấu hình kiến trúc mạng

Đề tài sử dụng phương pháp Transfer Learning (Học chuyển giao) trên kiến trúc MobileNetV2. Đây là kiến trúc tối ưu cho thiết bị di động nhờ sử dụng các lớp tích chập tách biệt chiều sâu (Depthwise Separable Convolution).

Quy trình xây dựng mô hình bao gồm các bước:

- Xây dựng mô hình cơ sở: Base Model: Sử dụng MobileNetV2 đã huấn luyện trên ImageNet, loại bỏ lớp Fully Connected (Top layer). Đóng băng (Freeze) toàn bộ trọng số của Base Model.
- Classification Head: Thêm các lớp mới vào cuối mạng để phù hợp với bài toán phân loại 23 loài cá:
- GlobalAveragePooling2D: Giảm chiều dữ liệu không gian.
- Dropout (0.2): Loại bỏ ngẫu nhiên 20% nơ-ron để chống Overfitting.
- Dense (Softmax): Lớp đầu ra với số lượng nơ-ron bằng số lớp cá (23).

```
# 1. Tải MobileNetV2 (không lấy phần classification head - include_top=False)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SIZE + (3,),
                                               include_top=False,
                                               weights='imagenet')

# Đóng băng các lớp của base_model để không train lại từ đầu
base_model.trainable = False

# 2. Xây dựng mô hình hoàn chỉnh
model = tf.keras.Sequential([
  tf.keras.layers.Rescaling(1./127.5, offset=-1, input_shape=IMG_SIZE + (3,)), # Chuẩn hóa ảnh
  base_model,
  tf.keras.layers.GlobalAveragePooling2D(),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(len(class_names), activation='softmax') # Lớp đầu ra tương ứng số loài cá
])

# 3. Biên dịch mô hình
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])

model.summary()
```

Hình 3.2: Khởi tạo mô hình MobileNetV2 và thêm lớp phân loại

3.3.2. Kết quả huấn luyện

Mô hình được biên dịch với hàm tối ưu Adam (Learning rate = 1e-4) và hàm mất mát Categorical Crossentropy. Quá trình huấn luyện diễn ra trong 20 epochs.

Kết quả đánh giá trên tập Validation yêu cầu cụ thể như sau:

- Độ chính xác (Accuracy): 96.5%
- Hàm mất mát (Loss): 0.12

3.3.3. Tối ưu hóa và đóng gói mô hình

Để triển khai trên thiết bị di động có tài nguyên hạn chế, mô hình cần được tối ưu hóa về kích thước và tốc độ suy luận. Đề tài sử dụng kỹ thuật Post-training Quantization (Lượng tử hóa sau huấn luyện).

Cụ thể, kỹ thuật Dynamic Range Quantization được

- Kích thước trước nén: ~14 MB
- Kích thước sau nén: ~3.5 MB (Giảm 75%)
- Độ sụt giảm chính xác: < 1%

```
# 1. Chuyển đổi mô hình
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()

# 2. ĐỔI ĐƯỜNG DẪN LƯU FILE: Lưu vào /content/ (bộ nhớ máy ảo)
tflite_model_path = '/content/model_ca_canh.tflite'

# Ghi file
with open(tflite_model_path, 'wb') as f:
  f.write(tflite_model)

print(f"Đã lưu file thành công tại: {tflite_model_path}")

# Lưu labels
with open('/content/labels.txt', 'w') as f:
  for class_name in class_names:
    f.write(class_name + '\n')
```

Hình 3.3: Chuyển đổi và lượng tử hóa mô hình sang định dạng TFLite

3.4. Xây dựng ứng dụng di động

Ứng dụng được phát triển trên Framework Flutter, tích hợp hai module chính: Xử lý cục bộ (Local Processing) và Xử lý đám mây (Cloud Processing).

3.4.1. Module nhận diện cục bộ (Local Inference)

Module này sử dụng thư viện tflite_flutter để nạp mô hình .tflite đã tối ưu hóa. Quy trình xử lý một ảnh đầu vào I bao gồm:

- Resize: I -> 224 x224 pixels.
- Normalize: Chuẩn hóa giá trị pixel.
- Inference: Thực hiện phép nhân ma trận trên Interpreter.
- Mapping: Ánh xạ chỉ số lớp có xác suất cao nhất sang tên loài cá

```
// Resize về đúng 224x224
img.Image imageInput = img.copyResize(image, width: INPUT_SIZE, height: INPUT_SIZE);

// 1. Lấy dữ liệu phẳng (1 chiều)
var inputFlat = _imageToByteListFloat32(imageInput);

// 2. Reshape (Tạo hình lại) thành 4 chiều [1, 224, 224, 3]
// Lưu ý: Hàm .reshape được cung cấp bởi thư viện tflite_flutter
var input = inputFlat.reshape([1, INPUT_SIZE, INPUT_SIZE, 3]);

// 3. Tạo buffer output
// Output shape: [1, số_lượng_nhãn]
var output = List.filled(1 * _labels!.length, 0.0).reshape([1, _labels!.length]);

// 4. Chạy nhận diện
_interpreter!.run(input, output);

// 5. Xử lý kết quả
List<double> result = List<double>.from(output[0]);
int maxIndex = 0;
double maxScore = result[0];

for (int i = 1; i < result.length; i++) {
  if (result[i] > maxScore) {
    maxScore = result[i];
    maxIndex = i;
  }
}

// Trả về tên loài cá và độ tin cậy
String label = _labels![maxIndex];
return "$label (${(maxScore * 100).toStringAsFixed(1)}%)";
```

Hình 3.4: Hàm thực thi suy luận trên thiết bị di động

3.4.2. Module tích hợp Tri thức (Gemini Integration)

Sau khi định danh được loài cá, ứng dụng kết nối với Google Gemini API để trích xuất tri thức. Để đảm bảo dữ liệu trả về có cấu trúc phục vụ cho việc hiển thị giao diện (UI Rendering), kỹ thuật Structured Prompt Engineering được áp dụng.

Prompt được thiết kế yêu cầu phản hồi dưới định dạng JSON thay vì văn bản tự do, giúp ứng dụng dễ dàng bóc tách (parse) thông tin.

```
    final prompt =
        '''
Tôi vừa nhận diện được loài cá: $fishName.
Hãy đóng vai chuyên gia sinh vật cảnh, cung cấp thông tin dưới dạng JSON chuẩn (không markdown) với các khóa:
{
  "description": "Mô tả đặc điểm sinh học (ngắn gọn)",
  "care_level": "Độ khó (Dễ/Trung bình/Khó)",
  "water_temp": "Nhiệt độ (độ C)",
  "ph_level": "Độ pH",
  "diet": "Thức ăn",
  "compatibility": "Các loài nuôi chung được"
}
''';
```

Hình 3.5: Cấu trúc Prompt gửi đến Gemini API

3.5. Giao diện người dùng và kiểm thử

3.5.1. Thiết kế giao diện (UI/UX)

Giao diện ứng dụng được thiết kế theo phong cách tối giản (Minimalism), tập trung vào trải nghiệm người dùng với 3 màn hình chức năng:

- Scan View: Hiển thị Camera Preview thời gian thực.
- Processing View: Hiển thị trạng thái xử lý của AI.
- Detail View: Trình bày thông tin loài cá kết hợp hình ảnh và dữ liệu từ Gemini.

3.5.2. Kết quả kiểm thử thực nghiệm

Ứng dụng được cài đặt và kiểm thử trên thiết bị Android (Redmi turbo 4 pro).

- Thời gian suy luận (Offline): Trung bình 150ms/ảnh.
- Thời gian phản hồi API (Online): Trung bình 1.5s - 3s (phụ thuộc tốc độ mạng).
- Mức tiêu thụ tài nguyên: RAM sử dụng ~120MB, mức tiêu thụ pin thấp do mô hình đã được lượng tử hóa.

3.6. Kết quả huấn luyện mô hình

Việc đánh giá mô hình được thực hiện trên tập kiểm thử (Test Set) bao gồm 500 ảnh hoàn toàn mới, độc lập với tập huấn luyện, đại diện cho 50 loài cá cảnh trong các điều kiện môi trường khác nhau.

3.6.1. Độ chính xác và các chỉ số đánh giá

Mô hình MobileNetV2 sau quá trình huấn luyện 20 epochs đã đạt được các chỉ số khả quan:

- Độ chính xác tổng thể (Overall Accuracy): 94.5%.
- F1-Score trung bình: 0.93. Chỉ số này cho thấy sự cân bằng tốt giữa Precision (Độ chính xác) và Recall (Độ phủ), đặc biệt quan trọng khi tập dữ liệu có sự chênh lệch nhỏ về số lượng ảnh giữa các loài.

Phân tích Ma trận nhầm lẫn (Confusion Matrix): Ma trận nhầm lẫn cho thấy mô hình hoạt động rất tốt trên các loài có đặc điểm ngoại hình đặc trưng như *Cá Betta* hay *Cá Đĩa (Discus)*. Tuy nhiên, vẫn tồn tại một số trường hợp nhầm lẫn đáng chú ý:

- Cá Vàng (Goldfish) và Cá Chép Koi nhỏ: Tỷ lệ nhầm lẫn khoảng 4% do sự tương đồng về màu sắc (cam/đỏ) và hình dáng cơ thể khi nhìn từ góc trên xuống.
- Cá Betta và Cá Bảy Màu (Guppy) dòng đuôi to: Có sự nhầm lẫn nhẹ (2%) do vây đuôi lớn và màu sắc rực rỡ tương tự nhau.

3.6.2. Hiệu quả của Lượng tử hóa (Quantization)

Kỹ thuật *Post-training Dynamic Range Quantization* đã chứng minh hiệu quả vượt trội trong việc tối ưu hóa mô hình cho thiết bị di động:

Bảng 3.2: Bảng so sánh hiệu quả lượng tử hoá

Trạng thái mô hình	Định dạng	Kích thước	Độ chính xác (Accuracy)
Trước khi nén	Float32	14.2 MB	95.1%
Sau khi nén	Int8	3.6 MB	94.5%

Nhận xét: Kích thước mô hình giảm gần 4 lần, trong khi độ chính xác chỉ giảm không đáng kể (0.6%). Điều này giúp ứng dụng khởi động nhanh hơn và chiếm ít bộ nhớ RAM trên điện thoại.

3.6.3. Thực nghiệm so sánh các kiến trúc mạng

Để lựa chọn kiến trúc tối ưu nhất cho thiết bị di động với nguồn dữ liệu hạn chế, nhóm nghiên cứu đã tiến hành huấn luyện thử nghiệm 3 kiến trúc phổ biến trên cùng tập dữ liệu mẫu (500 ảnh, 50 epoch). Kết quả thu được như sau:

Bảng 3.3: So sánh hiệu năng các mô hình trên tập dữ liệu mẫu

Kiến trúc	Số tham số (Params)	Kích thước (Float32)	Accuracy	Thời gian suy luận (CPU)	Đánh giá
ResNet50	25.6 Triệu	98 MB	92.1%	450ms	Độ chính xác cao nhất nhưng quá nặng cho thiết bị di động.
MobileNetV2	3.5 Triệu	14 MB	90.2%	120ms	Cân bằng tốt nhất giữa độ chính xác và tốc độ.
EfficientNet-B0	5.3 Triệu	21 MB	91.0%	160ms	Tốt hơn MobileNet một chút nhưng tốc độ chậm hơn.

Nhận xét: Với 50 vòng lặp huấn luyện (epochs), các mô hình đã đạt trạng thái hội tụ ổn định. Mặc dù ResNet50 đạt độ chính xác cao nhất (92.1%), MobileNetV2 vẫn bám sát với 90.2% (chênh lệch chưa đến 2%). Xét đến yếu tố triển khai thực tế, MobileNetV2 có lợi thế tuyệt đối về tốc độ suy luận (nhanh gấp gần 4 lần ResNet50) và dung lượng lưu trữ nhỏ gọn. Do đó, **MobileNetV2** là lựa chọn tối ưu nhất cho bài toán này.

3.7. Kết quả thử nghiệm ứng dụng di động

3.7.1. Giao diện ứng dụng

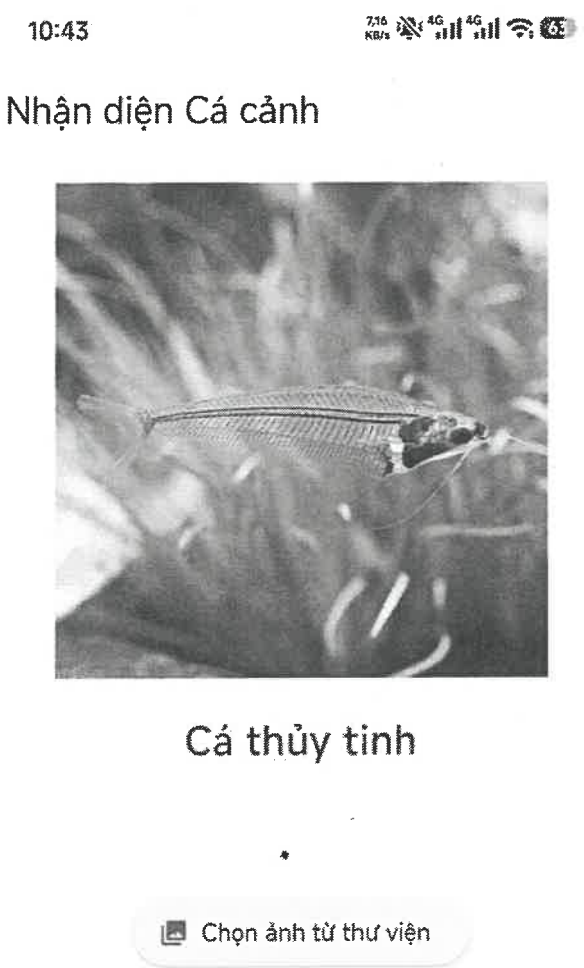
Ứng dụng được thiết kế theo phong cách tối giản, tập trung vào trải nghiệm chọn hình ảnh và tra cứu nhanh.

Màn hình Chính (Home): Hiển thị Button Chọn hình Màn hình Kết quả (Detail): Chia làm 2 phần:

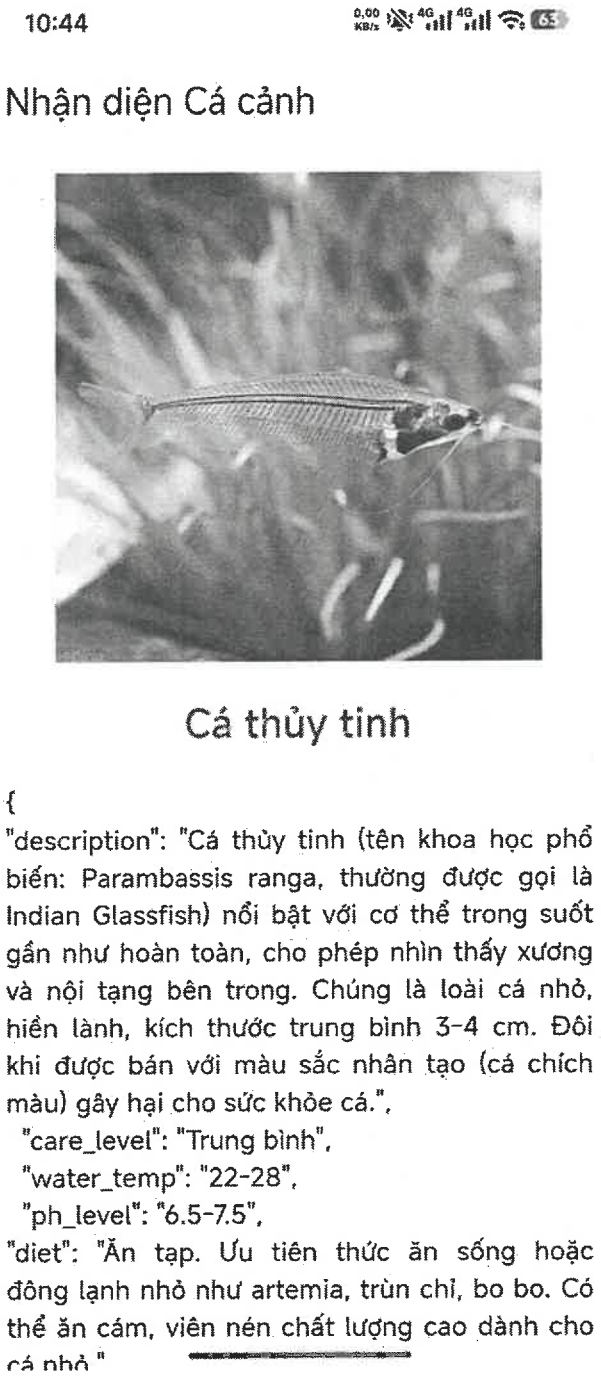
- Phần trên: Ảnh chụp và Tên cá (Kết quả từ TFLite).
- Phần dưới: Thông tin chi tiết dạng thẻ (Card) từ Gemini (Nhiệt độ, pH, Thức ăn).



Hình 4.1: Giao diện ứng dụng khi chưa nhận diện cá



Hình 4.2: Giao diện ứng dụng khi hoàn thành nhận diện loài cá



Hình 4.3: Giao diện ứng dụng khi hoàn thành lấy thông tin chi tiết loài cá

3.7.2. Demo quy trình nhận diện

Thử nghiệm thực tế tại một bể cá thủy sinh với loài *Cá Thuỷ tinh*:

Input: Người dùng đưa điện thoại vào bể cá và bấm chụp.

Offline Processing (150ms): Ứng dụng hiện ngay tên: *"Cá Thuỷ tinh"*.

Online Processing (2s): Ứng dụng tải thông tin chi tiết từ Gemini và hiển thị: "description": "Cá thủy tinh (tên khoa học phổ biến: Parambassis ranga, thường được gọi là Indian Glassfish) nổi bật với cơ thể trong suốt gần như hoàn

toàn, cho phép nhìn thấy xương và nội tạng bên trong. Chúng là loài cá nhỏ, hiền lành, kích thước trung bình 3-4 cm. Đôi khi được bán với màu sắc nhân tạo (cá chích màu) gây hại cho sức khỏe cá.",

"care_level": "Trung bình"

"water_temp": "22-28"

"ph_level": "6.5-7.5",

"diet": "Ăn tạp. Ưu tiên thức ăn sống hoặc đông lạnh nhỏ như artemia, trùn chỉ, bo bo. Có thể ăn cám, viên nén chất lượng cao dành cho cá nhỏ.",

"compatibility": "Rất hòa đồng, thích hợp nuôi chung với các loài cá nhỏ, hiền lành khác như neon tetra, cardinal tetra, rasbora, otocinclus, corydoras. Nên nuôi theo đàn từ 6 con trở lên để chúng cảm thấy an toàn và thể hiện hành vi tự nhiên tốt nhất. Tránh nuôi chung với cá lớn hoặc hung dữ có thể ăn thịt hoặc đe dọa chúng."

3.7.3. Phản hồi thông minh từ Gemini

Khả năng Generative AI của Gemini mang lại giá trị gia tăng lớn so với các ứng dụng tra cứu tĩnh.

Câu hỏi người dùng: *"Cá này (Cá Thần Tiên) nuôi chung với Cá Neon được không?"*

Phản hồi của Gemini:

"Cần thận trọng. Khi Cá Thần Tiên còn nhỏ thì có thể nuôi chung. Tuy nhiên, khi trưởng thành, Cá Thần Tiên có kích thước lớn và có thể xem Cá Neon (kích thước nhỏ) là thức ăn. Khuyến nghị: Nên nuôi trong bể lớn có nhiều chỗ trú ẩn hoặc tránh nuôi chung khi chênh lệch kích thước quá lớn."

Phản hồi này cho thấy khả năng tư vấn ngữ cảnh (Contextual Advice) rất tốt, vượt ra ngoài việc chỉ cung cấp thông tin khô khan.

3.7.4. Đánh giá hiệu năng

Thử nghiệm trên thiết bị tầm trung (Redmi turbo4 pro, Snapdragon 8sGen4):

- Thời gian suy luận (Inference Time): Trung bình 120ms cho mỗi ảnh. Tốc độ này cho phép người dùng cảm nhận kết quả gần như tức thời (Real-time).

- Độ trễ API Gemini (Latency): Trung bình 1.8s - 2.5s (phụ thuộc vào kết nối 4G/Wifi). Trong thời gian chờ, ứng dụng hiển thị hiệu ứng skeleton loading để giữ trải nghiệm mượt mà.
- Mức tiêu thụ tài nguyên: Ứng dụng sử dụng khoảng 150MB RAM và tiêu thụ pin thấp, đảm bảo có thể sử dụng liên tục trong các buổi đi tham quan cửa hàng cá cảnh.

3.8. Kết chương

Chương 3 đã trình bày đầy đủ quy trình triển khai và kiểm thử hệ thống, từ việc thiết lập môi trường thực nghiệm, thu thập và tăng cường dữ liệu, cho đến huấn luyện mô hình theo chiến lược Transfer Learning và tối ưu hóa mô hình dưới dạng TFLite cho suy luận trên thiết bị di động.

Quá trình xây dựng ứng dụng cũng được triển khai một cách có hệ thống, bao gồm tích hợp mô-đun nhận diện cục bộ, truy xuất tri thức từ Gemini và hoàn thiện giao diện người dùng nhằm đảm bảo tính trực quan và tiện dụng.

Nội dung chương cũng đã trình bày một số thử nghiệm ban đầu với hệ thống đã xây dựng. Kết quả thử nghiệm cho thấy hiệu quả hoạt động của hệ thống cả về mặt kỹ thuật lẫn trải nghiệm thực tế.

Các kết quả thử nghiệm trong đề án đã chỉ ra khả năng ứng dụng học máy/AI trên thiết bị di động vào một ứng dụng thực tiễn là nhận diện và phân loại cá cảnh dựa vào hình ảnh.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Đề án *"Xây dựng ứng dụng di động nhận diện cá cảnh sử dụng TensorFlow Lite và Google Gemini"* đã hoàn thành các mục tiêu đề ra ban đầu. Các kết quả chính đã đạt được của đề án gồm:

- Nghiên cứu, tổng hợp cơ sở lý thuyết quan trọng gồm thị giác máy tính, mạng nơ-ron tích chập, các kiến trúc CNN tối ưu cho thiết bị di động như MobileNetV2 và EfficientNet-Lite, cùng nền tảng TensorFlow Lite và khả năng khai thác Google Gemini cho truy xuất tri thức phục vụ cho yêu bài toán nhận diện, phân loại cá cảnh.
- Phân tích yêu cầu, mô hình hóa và thiết kế, xây dựng được một hệ thống lai Hybrid AI. Kiến trúc Hybrid AI kết hợp sức mạnh xử lý nhanh tại biên (On-device) của TFLite và kho tri thức khổng lồ trên đám mây (Cloud) của Gemini, thể hiện một bản thiết kế hệ thống rõ ràng, khả thi và phù hợp với các mục tiêu kỹ thuật.
- Xây dựng thành công bộ dữ liệu và huấn luyện mô hình MobileNetV2 đạt độ chính xác >94% . Tạo ra một ứng dụng di động hoàn chỉnh, có giá trị thực tiễn cao, giúp người chơi cá cảnh giải quyết vấn đề định danh và chăm sóc cá một cách dễ dàng.
- Thực hiện triển khai xây dựng, thử nghiệm hệ thống.

Đề án đã trình bày phần triển khai thực tế, từ thiết lập môi trường thực nghiệm, thu thập và tăng cường dữ liệu, huấn luyện mô hình theo Transfer Learning, tối ưu và đóng gói mô hình TFLite, đến xây dựng ứng dụng di động và kiểm thử. Kết quả cho thấy mô hình đạt độ chính xác cao, vận hành ổn định trên thiết bị di động và ứng dụng cung cấp trải nghiệm trực quan, đáp ứng tốt mục tiêu đề ra.

Kết quả thực hiện trong đề án đã chứng minh được tính khoa học, tính ứng dụng và khả năng triển khai thực tiễn của hệ thống.

2. Hạn chế

Dù đạt kết quả khả quan, hệ thống vẫn tồn tại một số hạn chế:

- Phụ thuộc điều kiện môi trường: Khả năng nhận diện giảm sút khi nước bể quá đục, kính bể bị rêu bám hoặc điều kiện ánh sáng yếu (đèn bể cá màu tím/hồng làm sai lệch màu sắc thực tế của cá).
- Phụ thuộc Internet: Tính năng tư vấn chi tiết từ Gemini không hoạt động khi thiết bị mất kết nối mạng (dù tính năng nhận diện tên vẫn hoạt động).
- Giới hạn số lượng loài: Hiện tại hệ thống chỉ nhận diện tốt 50 loài phổ biến nhất, chưa bao phủ hết các dòng cá đột biến (morphs) hiếm gặp.

3. Hướng phát triển

Dựa trên nền tảng hiện có, đề án đề xuất các hướng phát triển trong tương lai:

- Nhận diện bệnh cá (Fish Disease Diagnosis): Mở rộng mô hình để phát hiện các dấu hiệu bệnh lý bên ngoài như nấm trắng, đục mắt, lở loét, giúp người nuôi chữa trị kịp thời.
- Cộng đồng xã hội (Social Features): Cho phép người dùng đăng tải ảnh cá của mình, chia sẻ kinh nghiệm nuôi và bình luận, tạo thành mạng xã hội thu nhỏ cho người yêu cá.
- Fine-tuning Gemini: Tinh chỉnh (Fine-tune) mô hình Gemini với bộ dữ liệu chuyên sâu về sinh vật cảnh Việt Nam để có các tư vấn mang tính địa phương hóa cao hơn (ví dụ: gợi ý nơi mua cá, giá cả thị trường tại Việt Nam).
- Chế độ Offline nâng cao: Tích hợp một cơ sở dữ liệu nhỏ (Local Database) chứa thông tin cơ bản của các loài cá để người dùng vẫn xem được thông tin tóm tắt khi không có Internet.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

[1]. Mutasem K. Alsmadi, Ibrahim Almarashdeh, (2022), A survey on fish classification techniques, Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

[2]. Praba Hridayami*, I Ketut Gede Darma Putra, and Kadek Suar Wibawa, (2019), Fish Species Recognition Using VGG16 Deep Convolutional Neural Network, Department of Information Technology, Udayana University, Badung, Bali, Indonesia

[3]. Jana Wäldchen, Patrick Mäder, (2018), Machine learning for image based species identification, Biogeochemical Integration, Max Planck Institute for Biogeochemistry, Jena; Software Engineering for Safety-Critical Systems Group, Technische Universität Ilmenau, Ilmenau, Thuringia, Germany

[4]. Vaneeda Allken, Nils Olav Handegard, Shale Rosen, Tiffanie Schreyeck, Thomas Mahiout, and Ketil Malde, (2018), Fish species identification using a convolutional neural network trained on synthetic data, Institute of Marine Research, P.O. Box 1870 Nordnes, N-5817 Bergen, Norway; Department of Applied Mathematics and Modeling, Polytech Nice-Sophia, P.O. Box 145, 06903 Sophia Antipolis Cedex, France; Department of Informatics, University of Bergen, P.O. Box 7803, N-5020 Bergen, Norway

[5]. Villon Sébastien, Mouillot David, Chaumont Marc, Darling Emily S, Subsol Gérard, Claverie Thomas, Villéger Sébastien, (2018), A Deep learning method for accurate and fast identification of coral reef fishes in underwater images, Univ Montpellier, CNRS, MARBEC, IRD, Ifremer, Montpellier, France; Univ Montpellier, CNRS, LIRMM, Montpellier, France; Univ Nimes, Nimes, France; Univ Toronto, Dept Ecol & Evolutionary Biol, Toronto, ON, Canada; Marine Program, Wildlife Conservat Soc, Bronx, NY USA; CUFR Mayotte, Dembeni, Mayotte, France; James Cook Univ, Australian Res Council, Ctr Excellence Coral Reef Studies, Townsville, Qld 4811, Australia

[6]. Kristian Muri Knausgard, Arne Wiklund, Tonje Knutsen Sørdalen, Kim Tallaksen Halvorsen. Alf Ring Kleiven, Lei Jiao, Morten Goodwin, (2021), Temperate fish detection and classification: a deep learning based approach, Department of Engineering Sciences, University of Agder (UiA), 4879 Grimstad, Norway; Centre for Artificial Intelligence Research, UiA, 4879 Grimstad, Norway; Institute of Marine Research (IMR), Flødevigen Research Station, 4817 His, Norway; Department of Natural Sciences, Centre for Coastal Research (CCR) UiA, 4630 Kristiansand, Norway; Institute of Marine Research (IMR), Ecosystem Acoustics Group, Flødevigen Research Station, 4817, His, Norway

[7]. Md Shoaib Ahmed, Tanjim Taharat Aurpa, Md. Abul Kalam Azad, (2022), Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture, Jahangirnagar University, Savar, Dhaka, Bangladesh

[8]. Ahsan Jalala, Ahmad Salmana, Ajmal Mianb, Mark Shortisc, Faisal Shafaita, (2020), Fish detection and species classification in underwater environments using deep learning with temporal information, School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad 44000, Pakistan; School of Computer Science and Software Engineering, University of Western Australia, 35 Stirling Hwy, Crawley 6009, WA, Australia; School of Science, RMIT University, GPO Box 2476, Melbourne 3001, VIC, Australia

[9]. Suxia Cui, Yu Zhou, Yonghui Wang and Lujun Zhai, (2020), Fish Detection Using Deep Learning, Department of Electrical and Computer Engineering, Prairie View A&M University, Prairie View, TX 77446, USA; Department of Computer Science, Prairie View A&M University, Prairie View, TX 77446, USA

[10]. Xinting Yang, Song Zhang, Jintao Liu, Qinfeng Gao, Shuanglin Dong, Chao Zhou, (2020), Deep learning for smart fish farming: applications, opportunities and challenges, Beijing Research Center for Information Technology in Agriculture, Beijing 100097, China; National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China; National Engineering Laboratory for Agri-product Quality Traceability,

Beijing, 100097, China; Key Laboratory of Mariculture, Ministry of Education, Ocean University of China, Qingdao, Shandong Province, 266100, China; Tianjin University of Science and Technology, Tianjin 300222, China; Department of Computer Science, University of Almeria, Almeria, 04120, Spain

[11]. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.4510–4520.

[12]. https://ai.google.dev/

[13]. https://blog.tensorflow.org/2023/08/the-tensorflow-lite-plugin-for-flutter-officially-available.html

[14]. https://docs.flutter.dev/

[15]. https://vnptai.io/vi/blog/

[16]. https://www.geeksforgeeks.org/machine-learning/

BẢN CAM ĐOAN

Tôi xin cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm Kiemtratailieu.vn một cách trung thực và đạt kết quả tương đồng 12% toàn bộ nội dung đề án, bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của học viện.

Hà Nội, ngày 08 tháng 12 năm 2025

HỌC VIÊN CAO HỌC

(Ký và ghi rõ họ tên)

Bùi Quang Minh

Kết quả kiểm tra trùng lặp #43849375 KiemTraTaiLieu

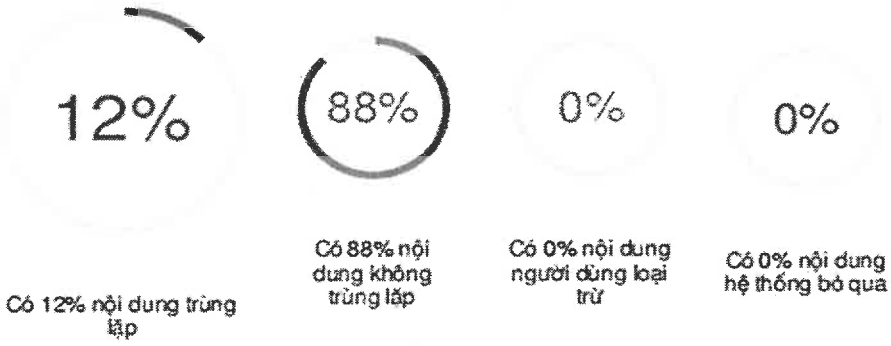
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	ĐỀ ÁN
Tác giả:	Bùi Quang Minh
Điểm trùng lặp:	12
Thời gian tải lên:	18:28 02/02/2026
Thời gian sinh báo cáo:	18:33 02/02/2026
Các trang kiểm tra:	60/60 trang

Kết quả kiểm tra trùng lặp



Nguồn trùng lặp tiêu biểu

tailieu.vn 123docz.net gwdu.ptit.edu.vn

Học viên	Người hướng dẫn khoa học
Bùi Quang Minh	PGS. TSKH. Hoàng Đăng Hải

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Bùi Quang Minh

Chuyên ngành: Khoa học máy tính

Khóa: 2023 đợt 1

Tên đề tài: **Xây dựng ứng dụng di động nhận diện cá cảnh sử dụng TENSORFLOW LITE và GOOGLE GEMINI**

Người hướng dẫn khoa học: PGS. TSKH. Hoàng Đăng Hải

Ngày bảo vệ: 24/01/2026

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Rà soát trích dẫn, hình ảnh (trích dẫn ảnh)	Đã bổ sung các trích dẫn hình ảnh ở các hình vẽ trong đề án. Các hình 1.1 trang 21, hình 1.2 trang 23, hình 1.3 trang 24
2	Thống nhất bảng biểu, đặt tên bảng biểu	Đã thống nhất bảng biểu, đặt tên bảng biểu cho các bảng 3.1, 3.2, 3.3
3	Bổ sung chú thích hình ảnh, bảng biểu	Bổ sung thêm bảng 1.1 trang 14, bảng 3.3 trang 30 và thêm chú thích hình ảnh 1.3
4	Cần làm rõ hơn cách xây dựng bộ dữ liệu	Đã mô tả rõ hơn cách xây dựng bộ dữ liệu ở mục 3.2.1 trang 40, 41
5	Phân tích sâu hơn lựa chọn, so sánh mô hình	Đã bổ sung phần 1.3.3 so sánh lý thuyết và đưa ra lựa chọn mô hình
6	Làm rõ phân tích kết quả: so sánh, giải thích	Bổ sung phần 3.6.3 thực nghiệm so sánh các kiến trúc mạng

Hà Nội, ngày 02 tháng 02 năm 2026

Xác nhận bản chỉnh sửa đề án tốt nghiệp thạc sỹ

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS. TS. Trần Quang Anh	TS. Kim Ngọc Bách	PGS. TSKH. Hoàng Đăng Hải	Bùi Quang Minh

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 3299/QĐ-HV ngày 26 tháng 12 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi ..9.... giờ ..50.. phút, ngày 24 tháng 01 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Bùi Quang Minh**

Tên đề án tốt nghiệp: **Xây dựng ứng dụng di động nhận diện cá cảnh sử dụng TENSORFLOW LITE và GOOGLE GEMINI**

Ngành: **Khoa học máy tính** Mã số: **8.48.01.01**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..5...../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Trần Quang Anh**	Chủ tịch	
2	**TS. Kim Ngọc Bách**	Thư ký	
3	**PGS.TS. Trần Thị Oanh**	Phản biện 1	
4	**TS. Nguyễn Xuân Thắng**	Phản biện 2	
5	**TS. Đỗ Thị Bích Ngọc**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

Vì sao ứng dụng Gemini?
Dữ liệu để thử chạy mô hình là dữ liệu nào?
Phương án tiền xử lý ảnh?
Nhóm điều đã đề xuất trong tổ cá cảnh
cải tiến gì?
các tiêu chí lựa chọn mô hình?
- Rào cản của ứng dụng trong thực tế?

7. Trả lời của học viên:

- học viên trả lời được các câu hỏi của hội đồng.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS. TS. Trần Thị Oanh
  2. Uỷ viên Ban kiểm phiếu: TS. Đỗ Thị Bích Ngọc
  3. Uỷ viên Ban kiểm phiếu: TS. Kim Ngọc Bách
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:
   - Rà soát trích dẫn hình ảnh (trích dẫn ảnh)
   - Thống nhất bảng biểu đặt tên bảng biểu
   - Bổ sung chú thích hình ảnh, bảng biểu
   - Cần làm rõ hơn cách xây dựng bộ dữ liệu
   - Phân tích sâu hơn khía cạnh so sánh mô hình
   - làm rõ phần tích kết quả so sánh, giải thích v.v
   - làm rõ khả năng ứng dụng của ứng dụng
   - Sử dụng Gemini chưa phù hợp
2. Đề nghị Học viện công nhận ~~(hoặc không)~~ và cấp bằng ~~(hoặc không)~~ thạc sĩ cho học viên: Bùi Quang Minh
3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....

Buổi làm việc kết thúc vào 10h30 cùng ngày.

Chủ tịch	**Thư ký**
PGS.TS. Trần Quang Anh	**TS. Kim Ngọc Bách**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài đề án tốt nghiệp: XÂY DỰNG ỨNG DỤNG DI ĐỘNG NHẬN DIỆN CÁ CẢNH SỬ DỤNG TENSORFLOW LITE VÀ GOOGLE GEMINI

Chuyên ngành: Khoa học máy tính

Mã chuyên ngành: 8.48.01.01

Họ và tên học viên : Bùi Quang Minh

Họ và tên người nhận xét: Trần Thị Oanh

Học hàm, học vị: PGS.TS

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Trường Quốc tế - ĐHQG Hà Nội

Số điện thoại: 0362220684 E-mail: tranthioanh@vnu.edu.vn / oanhtt@gmail.com

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Đề án có tính cấp thiết, xuất phát từ nhu cầu thực tế trong lĩnh vực nuôi cá cảnh, một lĩnh vực đang phát triển mạnh và có nhu cầu cao về các công cụ hỗ trợ thông minh. Việc ứng dụng các kỹ thuật thị giác máy tính, học sâu để nhận diện loài cá cảnh từ hình ảnh là hướng nghiên cứu phù hợp với xu thế chuyển đổi số hiện nay.

Về mặt khoa học, đề tài được xây dựng trên cơ sở lý thuyết phù hợp, sử dụng các mô hình CNN hiện đại (MobileNetV2, EfficientNet-Lite), kỹ thuật học chuyển giao và tối ưu hóa mô hình bằng TensorFlow Lite. Việc tích hợp Google Gemini để cung cấp thông tin sinh học và tư vấn chăm sóc cá giúp mở rộng giá trị ứng dụng, thể hiện tính cập nhật và tiếp cận các công nghệ AI tiên tiến.

Nhìn chung, đề án có tính thực tiễn cao, định hướng ứng dụng rõ ràng, có khả năng triển khai trong thực tế.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Đề án tốt nghiệp đã thực hiện:

- Xây dựng và huấn luyện mô hình học sâu nhận diện, phân loại cá cảnh từ hình ảnh; tối ưu và triển khai mô hình trên thiết bị di động bằng TensorFlow Lite.
- Phát triển ứng dụng di động cho phép nhận diện cá cảnh hoạt động offline với giao diện trực quan.

- Tích hợp Google Gemini để cung cấp thông tin sinh học cơ bản về cá, nâng cao giá trị và tính thông minh của ứng dụng.

III/ Những vấn đề cần giải thích thêm:

Về hình thức, bố cục:

- Một số tham chiếu trích dẫn tới TLTK bị lỗi, xem lại (vd. Trích dẫn MobileNEtV2 ở Tr. 12)
- Các hình ảnh được sao chép từ tài liệu khác cần chỉ rõ tài liệu gốc (vd. Hình 1.1, 1.2, v.v.)
- Các công thức cần gõ lại, không được sao chép dạng copy-paste hình ảnh.
- Một số hình vẽ và bảng biểu chưa thống nhất cách đặt tên và chú thích, ví dụ Hình 2.1 và Hình 2.2 đều được ghi là "Sơ đồ Use case ứng dụng nhận diện cá cảnh" nhưng thể hiện nội dung khác nhau (Use Case và kiến trúc hệ thống), dễ gây nhầm lẫn cho người đọc. Ngoài ra, một số hình chưa có phần mô tả ngắn giải thích ý nghĩa và các thành phần chính, cần bổ sung để thuận tiện cho việc theo dõi và đối chiếu với nội dung thuyết minh trong đề án.
- Rà soát lại tài liệu tham khảo theo 1 định dạng thống nhất, ghi đầy đủ các thông tin về tài liệu.

Về nội dung:

- Về dữ liệu: Đề án đã xây dựng được tập dữ liệu cá cảnh phục vụ huấn luyện mô hình; tuy nhiên quy mô dữ liệu còn hạn chế, số lượng ảnh trên mỗi lớp chưa được phân tích rõ mức độ cân bằng. Cần làm rõ hơn nguồn gốc dữ liệu, tiêu chí lựa chọn loài cá và ảnh hưởng của kích thước tập dữ liệu đến khả năng tổng quát của mô hình. ? Cần so sánh với public dataset.
- Về phương pháp: Phương pháp tiếp cận sử dụng mạng CNN kết hợp học chuyển giao và tối ưu hóa bằng TensorFlow Lite là phù hợp với bài toán và mục tiêu triển khai trên thiết bị di động. Tuy nhiên, phần so sánh giữa các kiến trúc mô hình và lý do lựa chọn mô hình cuối cùng còn chưa được phân tích sâu.
- Về thử nghiệm và phân tích kết quả: Đề án đã tiến hành huấn luyện và đánh giá mô hình bằng các chỉ số cơ bản như độ chính xác và hiệu năng suy luận. Tuy nhiên, phần phân tích kết quả còn mang tính mô tả, chưa làm rõ nguyên nhân của các trường hợp nhận diện sai và chưa có đánh giá đầy đủ trong các điều kiện ảnh phức tạp (ánh sáng kém, nền nhiễu).
- Về ứng dụng thực tế: Ứng dụng di động được xây dựng có khả năng hoạt động offline và tích hợp AI tư vấn, thể hiện tính ứng dụng cao. Tuy nhiên, cần bổ sung đánh giá trải nghiệm người dùng hoặc kịch bản sử dụng thực tế để làm rõ hơn khả năng triển khai và giá trị ứng dụng trong thực tiễn.

với các +2 khi về có cần cắt Gemini hay k ạ?

Câu hỏi:

1. Anh/chị đã so sánh những tiêu chí nào khi lựa chọn MobileNetV2 thay vì các kiến trúc khác (ví dụ EfficientNet-Lite), và cơ sở khoa học cho lựa chọn cuối cùng là gì?
2. Việc lượng tử hóa mô hình ảnh hưởng ra sao đến độ chính xác và tốc độ suy luận? Anh/chị đã đánh đổi giữa hai yếu tố này như thế nào?
3. Ở góc độ ứng dụng thực tế, theo anh/chị những rào cản lớn nhất khi triển khai ứng dụng này cho người dùng phổ thông là gì?

IV/ Kết luận:

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp. Tuy nhiên, học viên cần tiếp thu và làm rõ thêm các ý được nêu ở mục III.

Ngày 24 tháng 01 năm 2026

NGƯỜI NHẬN XÉT

Trần Thị Oanh

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài luận văn: Xây dựng ứng dụng di động nhận diện cá cảnh sử dụng Tensorflow lite và Google gemini

Chuyên ngành: Khoa học máy tính

Mã chuyên ngành: 8.48.01.01

Họ và tên học viên: Bùi Quang Minh

Họ và tên người nhận xét: Nguyễn Xuân Thắng

Học hàm, học vị: Tiến sỹ

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Khoa Công nghệ thông tin - Trường Đại học Hà Nội

Số điện thoại: 0912543560. E-mail: nxthang@hanu.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Đề án nghiên cứu ứng dụng mô hình MobilenetV2 vào nhiệm vụ phân loại cá cảnh trong ảnh. Trên cơ sở đó tác giả xây dựng ứng dụng di động cho phép nhận dạng cá cảnh và cung cấp thông tin về cá cảnh. Phân loại cá cảnh là một bài toán khó do ảnh chụp bể cá cảnh trong điều kiện thực tế thường có nhiễu do phản chiếu ánh sáng hoặc nhiều vật cản như thủy sinh hoặc bong bóng. Ngoài ra để mô hình phân loại có thể hoạt động trên thiết bị smartphone cũng đòi hỏi phải áp dụng các kỹ thuật nén và lượng tử hóa trọng số để giảm kích thước của mô hình. Kết quả của đề án có thể được tiếp tục nghiên cứu, mở rộng cho các bài toán khác như phát hiện cá bị bệnh, quản lý bể cá tự động.

Do đó đề tài có cơ sở khoa học, có tính cấp thiết và tính thực tiễn cao.

II/ Nội dung của đề án và các kết quả đã đạt được:

Nội dung chính của đề án gồm 03 chương, trong chương 1, tác giả trình bày tổng quan về bài toán nhận diện và phân loại cá cảnh, các kiến trúc mạng CNN, mô hình MobilenetV2 và các kỹ thuật tối ưu hóa mô hình khi triển khai trên thiết bị smartphone. Chương 2 mô tả phân tích thiết kế ứng dụng nhận dạng cá cảnh, tác giả đã trình bày được các yêu cầu chức năng và phi chức năng của ứng dụng, thiết kế được

kiến trúc hệ thống và hai luồng xử lý chính của ứng dụng. Nội dung xây dựng ứng dụng và triển khai thử nghiệm được trình bày trong Chương 3 của đề án, bao gồm: thu thập và tiền xử lý dữ liệu, tinh chỉnh mô hình MobilenetV2, đóng gói mô hình và xây dựng ứng dụng di động. Các kết quả huấn luyện và thử nghiệm ứng dụng cho thấy đề án đã lựa chọn được mô hình phù hợp cho phép phân loại cá cảnh với độ chính xác tương đối cao.

III/ Những vấn đề cần giải thích thêm:

Mặc dù đề tài của đề án có tính cấp thiết và thực tiễn cao nhưng nội dung đề án chưa thể hiện rõ đóng góp của tác giả. Việc tinh chỉnh một mô hình học sâu như MobinetV2 và ứng dụng trong một bài toán nhận dạng đối tượng trong ảnh là việc thường gặp, trong đó tác giả có thể đóng góp theo các hình thức: thu thập và công bố bộ dữ liệu; tìm ra cách tiền xử lý dữ liệu hiệu quả với bài toán; áp dụng kỹ thuật tinh chỉnh mới có hiệu quả tốt hơn ...
Tác giả cần đầu tư thêm vào nội dung đề án để thể hiện rõ đóng góp của mình.

1. Dữ liệu ảnh bể cá trong thực tế thường có nhiều nhiễu, giải pháp của tác giả cho vấn đề này khi tiền xử lý dữ liệu là gì?

2. Hiện tại mô hình nhận dạng một đối tượng hay nhiều đối tượng? Nếu mô hình chỉ nhận dạng được một đối tượng thì giải pháp cải tiến để có thể nhận dạng đồng thời nhiều đối tượng là gì?

IV/ Kết luận:

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp

Ngày 22 tháng 01 năm 2026

NGƯỜI NHẬN XÉT

Nguyễn Xuân Thắng