

ĐÁNH GIÁ HIỆU SUẤT CỦA CÁC MÔ HÌNH PHÂN LOẠI HÌNH ẢNH TINYML LƯỢNG TỬ HÓA TRÊN NỀN TẢNG UAV MÔ PHỎNG CHO ỨNG DỤNG GIÁM SÁT NÔNG NGHIỆP

Hoàng Trọng Nghĩa¹
Email: htnghia2@hou.edu.vn

Ngày tòa soạn nhận được bài báo: 02/06/2025

Ngày phản biện đánh giá: 08/09/2025

Ngày bài báo được duyệt đăng: 15/10/2025

DOI: 10.59266/houjs.2025.728

Tóm tắt: UAV mang lại khả năng thu thập dữ liệu và giám sát thời gian thực trong nông nghiệp. Tuy nhiên, việc triển khai AI tiên tiến, đặc biệt là học sâu, trên UAV bị hạn chế bởi tài nguyên tính toán, bộ nhớ và năng lượng. Nghiên cứu **này nghiên cứu về** hiệu suất của các mô hình phân loại hình ảnh TinyML lượng tử hóa cho giám sát nông nghiệp. Chúng tôi đề xuất khung mô phỏng MATLAB/Simulink để đánh giá sự đánh đổi giữa độ chính xác, tốc độ suy luận và tiêu thụ tài nguyên. Kết quả cho thấy các mô hình TinyML lượng tử hóa giảm đáng kể dấu chân tính toán và năng lượng trong khi vẫn duy trì độ chính xác chấp nhận được, cho phép xử lý AI hiệu quả trên thiết bị biên, hướng tới nông nghiệp tự chủ và bền vững hơn.

Từ khoá: UAV, edge AI, tinyML, lượng tử hóa, phân loại hình ảnh, giám sát nông nghiệp, MATLAB/Simulink, xử lý thời gian thực

I. Đặt vấn đề

UAVs đã cách mạng hóa nông nghiệp nhưng AI học sâu gặp hạn chế về tài nguyên. Edge AI và TinyML, đặc biệt là lượng tử hóa, giải quyết vấn đề này bằng cách tối ưu hóa mô hình cho thiết bị biên. Nghiên cứu này đánh giá hiệu suất mô hình TinyML lượng tử hóa trên UAV mô phỏng bằng MATLAB/Simulink, tập trung vào độ chính xác,

tốc độ và tiêu thụ tài nguyên, nhằm thúc đẩy phát triển AI trên thiết bị UAV giám sát nông nghiệp.

II. Công trình liên quan

Sự hội tụ của các phương tiện bay không người lái (UAV) và Trí tuệ Nhân tạo (AI) đã thúc đẩy nghiên cứu sâu rộng trên nhiều lĩnh vực khác nhau, đặc biệt là trong các tác vụ thị giác máy tính như

¹ Khoa Điện - Điện tử, Trường Đại học Mở Hà Nội

phân loại hình ảnh và phát hiện đối tượng. Phần này cung cấp tổng quan về các tài liệu hiện có liên quan đến TinyML, phân loại hình ảnh dựa trên UAV và AI biên, nêu bật tình trạng hiện tại và xác định các khoảng trống nghiên cứu mà nghiên cứu này nhằm mục đích giải quyết.

2.1. Phân loại hình ảnh và Phát hiện đối tượng dựa trên UAV

UAV đã trở thành nền tảng không thể thiếu cho các ứng dụng đa dạng. Trong nông nghiệp, phân loại hình ảnh và phát hiện đối tượng từ hình ảnh UAV rất quan trọng cho các tác vụ như giám sát cây trồng, xác định bệnh và ước tính năng suất. Các phương pháp ban đầu thường dựa vào các kỹ thuật xử lý hình ảnh truyền thống và các thuật toán học máy như Máy vector hỗ trợ (SVM) và Rừng ngẫu nhiên để phân loại các loại cây trồng hoặc phát hiện các bất thường [1]. Mặc dù hiệu quả trong môi trường được kiểm soát, các phương pháp này thường gặp khó khăn với sự thay đổi và phức tạp của dữ liệu trên không trong thế giới thực, bao gồm các điều kiện ánh sáng khác nhau, sự che khuất và các tỷ lệ đối tượng đa dạng.

Sự ra đời của học sâu đã thúc đẩy đáng kể khả năng phân tích hình ảnh dựa trên UAV. Các mạng nơ-ron tích chập (CNN) đã chứng minh hiệu suất vượt trội trong việc trích xuất các đặc trưng phân cấp từ hình ảnh, dẫn đến các mô hình phân loại và phát hiện có độ chính xác cao. Ví dụ, nhiều nghiên cứu đã sử dụng CNN để phát hiện cỏ dại [2], chẩn đoán

bệnh cây trồng [3] và đếm trái cây [4] bằng cách sử dụng hình ảnh thu được từ UAV. Tuy nhiên, các mô hình này thường yêu cầu tài nguyên tính toán đáng kể, khiến việc triển khai trực tiếp chúng trên các UAV có tài nguyên hạn chế trở nên khó khăn. Các nhà nghiên cứu đã khám phá các chiến lược như xử lý dựa trên đám mây hoặc chuyển tải tính toán đến các trạm mặt đất, nhưng những điều này gây ra độ trễ và phụ thuộc vào các liên kết truyền thông mạnh mẽ, hạn chế các ứng dụng thời gian thực [5].

2.2. AI biên và TinyML cho các thiết bị có tài nguyên hạn chế

Để khắc phục hạn chế của AI đám mây, AI biên nổi lên là một phương pháp hữu hiệu cho phép suy luận trực tiếp trên thiết bị, giảm độ trễ, tăng quyền riêng tư và tự chủ, đặc biệt quan trọng cho UAV trong các tác vụ thời gian thực. TinyML, một nhánh của AI biên, chuyên về các mô hình ML hiệu quả cao cho thiết bị tài nguyên cực hạn chế (vi điều khiển), tối ưu hóa bộ nhớ, tính toán và năng lượng. Lĩnh vực này đã chứng kiến sự phát triển nhanh chóng, với các ứng dụng như nhận dạng từ khóa và nhận dạng cử chỉ đến phát hiện bất thường trong môi trường công nghiệp [6]. Các kỹ thuật chính để kích hoạt TinyML bao gồm:

- **Nén mô hình:** Giảm kích thước và độ phức tạp của mạng nơ-ron mà không làm mất đáng kể độ chính xác. Điều này bao gồm các phương pháp như cắt tỉa, lượng tử hóa và chưng cất tri thức.

- **Kiến trúc hiệu quả:** Thiết kế các kiến trúc mạng nơ-ron vốn nhẹ và hiệu quả về mặt tính toán, chẳng hạn như MobileNet, EfficientNet và ShuffleNet [7][8][9].

- **Phần cứng chuyên dụng:** Sử dụng vi điều khiển công suất thấp, FPGA hoặc bộ tăng tốc AI tùy chỉnh được thiết kế để suy luận hiệu quả trên thiết bị.

2.3. Kỹ thuật tối ưu hóa mô hình: Lượng tử hóa

Lượng tử hóa là kỹ thuật nén mô hình hiệu quả, giảm độ chính xác số (ví dụ: 32-bit float xuống 8-bit integer) để thu nhỏ kích thước, tăng tốc suy luận và giảm tiêu thụ năng lượng (Jacob et al., 2018). Có hai loại chính:

- Lượng tử hóa sau huấn luyện (PTQ) đơn giản, không cần huấn luyện lại, sử dụng dữ liệu hiệu chuẩn [10].

- Huấn luyện nhận biết lượng tử hóa (QAT) mô phỏng lượng tử hóa trong quá trình huấn luyện, cho độ chính xác cao hơn nhưng tốn tài nguyên hơn

Đối với các ứng dụng TinyML trên UAV, PTQ thường được ưu tiên do tính đơn giản và chi phí tính toán thấp hơn trong quá trình triển khai. Một số nghiên cứu đã khám phá việc sử dụng lượng tử hóa để AI hiệu quả trên các thiết bị biên. Ví dụ, Rastegari et al. (2016 [11]) đã giới thiệu XNOR-Net, lượng tử hóa trọng số và kích hoạt thành các giá trị nhị phân, đạt được tốc độ đáng kể. Gần đây hơn, các nỗ lực đã tập trung vào lượng tử hóa số

nguyên 8 bit, mang lại sự cân bằng tốt giữa việc giữ lại độ chính xác và tăng hiệu quả [12].

2.4. Khung mô phỏng cho UAV và AI biên

Mô phỏng đóng vai trò thiết yếu trong phát triển UAV và AI biên, cung cấp môi trường thử nghiệm an toàn, hiệu quả. MATLAB/Simulink là nền tảng phổ biến cho mô hình hóa và phân tích hệ thống phức tạp, bao gồm động lực học UAV, điều khiển và học sâu [13]. Các nghiên cứu trước đây đã ứng dụng MATLAB/Simulink cho điều khiển bay [14], lập kế hoạch đường đi [15] và hợp nhất cảm biến UAV [16]. Tuy nhiên, thiếu khung mô phỏng toàn diện đánh giá hiệu suất mô hình TinyML lượng tử hóa trên phần cứng biên UAV cho giám sát nông nghiệp. Nghiên cứu này phát triển khung mô phỏng MATLAB/Simulink để thu hẹp khoảng cách đó, cung cấp đánh giá định lượng về độ chính xác, tốc độ và tiêu thụ tài nguyên của các mô hình phân loại hình ảnh TinyML lượng tử hóa trên UAV nông nghiệp.

III. Phương pháp nghiên cứu

Phần này trình bày chi tiết phương pháp luận được sử dụng để đánh giá hiệu suất của các mô hình phân loại hình ảnh TinyML lượng tử hóa trên các nền tảng UAV mô phỏng cho giám sát nông nghiệp. Chúng tôi mô tả việc lựa chọn và tối ưu hóa các mô hình TinyML, thiết kế khung mô phỏng MATLAB/Simulink và các chỉ số hiệu suất được sử dụng để đánh giá.

3.1. Lựa chọn và Tối ưu hóa Mô hình TinyML

Để đảm bảo sự phù hợp với các thiết bị biên UAV có tài nguyên hạn chế, chúng tôi tập trung vào các kiến trúc mạng nơ-ron tích chập (CNN) nhẹ. Đối với nghiên cứu này, chúng tôi chọn MobileNetV2 làm mô hình nhẹ đại diện do cấu trúc dư ngược và các phép tích chập tách sâu, giúp giảm đáng kể chi phí tính toán và kích thước mô hình trong khi vẫn duy trì độ chính xác cạnh tranh [17]. Chúng tôi sẽ xem xét một mô hình MobileNetV2 đã được huấn luyện trước trên một tập dữ liệu lớn (ví dụ: ImageNet) làm điểm khởi đầu, sau đó tinh chỉnh nó trên một tập dữ liệu hình ảnh nông nghiệp.

3.1.1. Tập dữ liệu hình ảnh nông nghiệp

Đối với các ứng dụng giám sát nông nghiệp, chúng tôi sẽ sử dụng Tập dữ liệu PlantVillage [18]. Tập dữ liệu này bao gồm các hình ảnh lá cây khỏe mạnh và bị bệnh, bao gồm nhiều loại cây trồng và loại bệnh khác nhau. Nó rất phù hợp cho các tác vụ phân loại hình ảnh trong nông nghiệp. Tập dữ liệu sẽ được tiền xử lý để thay đổi kích thước tất cả các hình ảnh thành một kích thước đầu vào đồng nhất (ví dụ: 96x96 pixel hoặc 128x128 pixel, tùy thuộc vào cấu hình lớp đầu vào MobileNetV2 đã chọn) và chuẩn hóa thành một phạm vi phù hợp cho đầu vào mạng nơ-ron (ví dụ: [0, 1] hoặc [-1, 1]). Các kỹ thuật tăng cường dữ liệu (ví dụ: xoay, lật, phóng to) sẽ được áp dụng trong quá trình huấn luyện để tăng

cường tính mạnh mẽ và khả năng tổng quát hóa của mô hình.

3.1.2. Lượng tử hóa sau huấn luyện (PTQ)

Với TinyML, chúng tôi áp dụng Lượng tử hóa sau huấn luyện (PTQ) để chuyển đổi mô hình dấu phẩy động sang định dạng số nguyên (INT8), giảm kích thước, tăng tốc suy luận và tiết kiệm năng lượng (Jacob et al., 2018). Quá trình này gồm: nhập mô hình (ví dụ: MobileNetV2 vào MATLAB qua ONNX [19]), hiệu chuẩn bằng tập dữ liệu nhỏ để xác định phạm vi kích hoạt, và chuyển đổi sang INT8. Chúng tôi sẽ so sánh hiệu suất mô hình float32 gốc với phiên

3.2. Thiết kế Khung mô phỏng MATLAB/Simulink

Khung mô phỏng MATLAB/Simulink được phát triển để đánh giá hệ thống mô hình TinyML trên UAV. Nó bao gồm mô hình UAV đơn giản hóa (động lực học bay, thu thập hình ảnh mô phỏng từ tập dữ liệu PlantVillage) và mô hình phần cứng biên (vi điều khiển như ESP32-CAM/OpenMV Cam H7) với các tham số như tốc độ xung nhịp, bộ nhớ, và tiêu thụ điện năng. Thời gian suy luận và năng lượng tiêu thụ được ước tính dựa trên FLOPs của mô hình và thông lượng bộ xử lý [19]. Kích thước mô hình (KB) đại diện cho dấu chân bộ nhớ. Khung này cho phép đánh giá hiệu suất TinyML trong điều kiện kiểm soát.

Ước tính thời gian suy luận:

$$\text{Thời gian suy luận (s)} = \frac{\text{FLOPs của mô hình}}{\text{FLOPs/giây của bộ xử lý}}$$

Trong đó FLOPs/giây của bộ xử lý được suy ra từ tốc độ xung nhịp và số phép toán mỗi chu kỳ (ví dụ: đối với một vi điều khiển đơn giản, nó có thể là 1 FLOP/chu kỳ, trong khi đối với các bộ xử lý nhúng tiên tiến hơn, nó có thể cao hơn).

Ước tính tiêu thụ năng lượng:

Năng lượng tiêu thụ cho mỗi lần suy luận sẽ được tính bằng

$$\text{Tiêu thụ năng lượng (Joule)} = \text{Thời gian suy luận (s)} \times \text{Công suất hoạt động (Watt)}$$

Khung mô phỏng được cấu hình cụ thể như sau: UAV bay ở độ cao 10 m, với tốc độ 5 m/s trên một khu vực nông nghiệp có kích thước 100×100 m. UAV được giả định chụp ảnh với tần suất 0.5 giây/lần, tạo ra khoảng 200 ảnh trong 100 giây mô phỏng. Phần cứng biên được mô hình hóa với tốc độ xung nhịp 240 MHz, thực hiện 1 FLOP/chu kỳ, và công suất hoạt động 0.2 W. Các mô hình được xem xét bao gồm MobileNetV2 Float32 (13.2 MB, 300 MFLOPs) và MobileNetV2 INT8 (3.3 MB, 75 MFLOPs).

3.3. Các chỉ số hiệu suất

Để đánh giá toàn diện mô hình TinyML, chúng tôi sử dụng ba nhóm chỉ số hiệu suất:

1) Độ chính xác: bao gồm độ chính xác tổng thể, Precision, Recall, F1-score và Ma trận nhầm lẫn, tính toán dựa trên dữ

liệu PlantVillage.

2) Hiệu suất thời gian thực: gồm Số khung hình mỗi giây (FPS) và Độ trễ (Latency).

3) Tiêu thụ tài nguyên: bao gồm Kích thước mô hình (KB), FLOPs và Ước tính tiêu thụ năng lượng (Joule/suy luận).

3.4. Quy trình mô phỏng trong MATLAB/Simulink

Quy trình mô phỏng trong MATLAB/Simulink bao gồm:

1) Chuẩn bị mô hình: Huấn luyện MobileNetV2 trên PlantVillage và áp dụng PTQ để tạo phiên bản INT8.

2) Thiết lập môi trường: Xây dựng mô hình Simulink tích hợp đường bay UAV, logic thu thập hình ảnh và mô phỏng phần cứng biên (tham số UAV, dữ liệu PlantVillage, tham số phần cứng).
3) Vòng lặp mô phỏng: UAV mô phỏng chụp ảnh ở các bước thời gian rời rạc; mỗi ảnh được đưa vào cả hai mô hình (float32 và INT8). Mô phỏng ghi lại độ chính xác, thời gian suy luận, tiêu thụ năng lượng và kích thước mô hình để phân tích sau này.

3.5. Các giả định và hạn chế của mô phỏng

Khung mô phỏng có các giả định và hạn chế: ước tính tham số phần cứng dựa trên giá trị điển hình (hiệu suất thực tế có thể khác); động lực học UAV đơn giản hóa; sử dụng dữ liệu hình ảnh tĩnh (không có video thời gian thực hay yếu tố môi trường động); không có vòng lặp phản

hồi AI ảnh hưởng đến điều khiển UAV; và lượng tử hóa lý tưởng hóa (không tính lỗi phần cứng). Dù vậy, mô phỏng vẫn là công cụ giá trị để đánh giá TinyML lượng tử hóa trên UAV.

Ngoài ra, mô phỏng chỉ được thực hiện với **num_simulations = 1**, tức là không lặp lại nhiều lần để phân tích thống kê. Hơn nữa, trong vòng lặp mô phỏng chưa thực hiện suy luận thực trên ảnh PlantVillage, mà thay vào đó sử dụng độ chính xác giả định (97.2% cho Float32 và 96.5% cho INT8) từ kết quả huấn luyện trước.

IV. Thử nghiệm và Kết quả

Phần này trình bày thiết lập và kết quả thử nghiệm mô phỏng MATLAB/Simulink, đánh giá hiệu suất MobileNetV2 (float32 và INT8) về độ chính xác, suy luận thời gian thực và tiêu thụ tài nguyên. Thiết lập: MobileNetV2 được tinh chỉnh trên PlantVillage (128x128, 50 epoch), sau đó lượng tử hóa INT8 bằng MATLAB DL Toolbox. UAV mô phỏng bay ở 10m, 5m/s trên cánh đồng 100x100m, chụp ảnh 0.5s/lần. Phần cứng biên mô phỏng có tốc độ 240 MHz, 1 FLOP/chu kỳ, 0.2W. Thu thập dữ liệu: Ghi lại lớp dự đoán/thực tế, FLOPs, kích thước mô hình để tính toán độ chính xác, thời gian suy luận, FPS và năng lượng tiêu thụ.

Trong nghiên cứu này, chúng tôi triển khai hai kịch bản thử nghiệm để đánh giá hiệu suất mô hình TinyML trên UAV mô phỏng:

- **Kịch bản 1:** Sử dụng mô hình MobileNetV2 Float32 (mô hình gốc, không lượng tử hóa).

- **Kịch bản 2:** Sử dụng mô hình MobileNetV2 INT8 (được lượng tử hóa sau huấn luyện bằng PTQ).

Cả hai kịch bản được thực hiện trên cùng môi trường UAV và phần cứng biên mô phỏng: UAV bay ở độ cao 10 m, tốc độ 5 m/s trên cánh đồng 100×100 m, chụp ảnh mỗi 0.5 giây, với tổng thời gian mô phỏng 100 giây (tương ứng khoảng 200 ảnh). Phần cứng biên được mô hình hóa với tốc độ xung nhịp 240 MHz, 1 FLOP/chu kỳ và công suất hoạt động 0.2 W.

Ở mỗi kịch bản, chúng tôi ghi nhận các chỉ số: độ chính xác phân loại, thời gian suy luận trung bình, số khung hình mỗi giây (FPS), kích thước mô hình và năng lượng tiêu thụ trên mỗi suy luận. Kết quả chi tiết được trình bày trong các bảng và biểu đồ so sánh ở các mục sau.

Trong mô phỏng, thời gian suy luận trung bình cho mỗi ảnh là 1250 ms với mô hình Float32 và 312.5 ms với mô hình INT8. Điều này tương ứng với hiệu suất 0.8 FPS (Float32) và 3.2 FPS (INT8). Năng lượng tiêu thụ trên mỗi lần suy luận được ước tính lần lượt là 250 mJ và 62.5 mJ. Các giá trị này được tính toán dựa trên số FLOPs của mô hình và thông lượng xử lý lý thuyết của phần cứng biên

4.2. Kết quả

4.2.1. Độ chính xác của mô hình

Bảng 1 trình bày độ chính xác tổng thể, độ chính xác, độ thu hồi và điểm F1 cho cả mô hình MobileNetV2 float32 và INT8 lượng tử hóa trên tập dữ liệu thử nghiệm PlantVillage.

Bảng 1: Hiệu suất phân loại của các mô hình MobileNetV2

Loại mô hình	Độ chính xác tổng thể (%)	Độ chính xác (Trung bình Macro)	Độ thu hồi (Trung bình Macro)	Điểm F1 (Trung bình Macro)
Float32	97.2	0.96	0.97	0.96
INT8	96.5	0.95	0.96	0.95

Kết quả cho thấy một sự sụt giảm nhỏ về độ chính xác tổng thể (0.7%) và các chỉ số phân loại khác sau khi lượng tử hóa INT8. Sự giảm nhỏ này thường được chấp nhận cho nhiều ứng dụng thực tế, đặc biệt khi xem xét những lợi ích đáng kể về hiệu quả.

4.2.2. Hiệu suất thời gian thực và

Tiêu thụ tài nguyên

Bảng 2 tóm tắt các chỉ số hiệu suất thời gian thực (thời gian suy luận, FPS) và các chỉ số tiêu thụ tài nguyên (kích thước mô hình, FLOPs, ước tính tiêu thụ năng lượng cho mỗi lần suy luận) cho cả hai loại mô hình.

Bảng 2: Hiệu suất thời gian thực và Tiêu thụ tài nguyên của các mô hình MobileNetV2

Loại mô hình	Kích thước mô hình (MB)	FLOPs (Triệu)	Thời gian suy luận (ms)	FPS (Khung hình/giây)	Năng lượng ước tính (mJ/suy luận)
Float32	13.2	300	1250	0.8	250
INT8	3.3	75	312.5	3.2	62.5

Lưu ý: FLOPs cho INT8 là lý thuyết, giả định giảm 4 lần các phép toán do số học số nguyên 8 bit so với số dấu phẩy động 32 bit, và bộ xử lý có khả năng 240 MFLOPs/giây.

Dữ liệu cho thấy rõ ràng những lợi ích đáng kể của lượng tử hóa INT8:

- **Giảm kích thước mô hình:** Mô hình INT8 nhỏ hơn khoảng 4 lần so với mô hình float32 (3.3 MB so với 13.2 MB). Điều này rất quan trọng đối với các thiết bị có bộ nhớ flash hạn chế.

- **Giảm FLOPs:** FLOPs lý thuyết giảm đi 4 lần, cho thấy sự giảm đáng kể về tải tính toán.

- **Giảm thời gian suy luận:** Thời gian suy luận giảm khoảng 75% (từ 1250

ms xuống 312.5 ms). Điều này dẫn đến tăng 4 lần FPS (từ 0.8 lên 3.2 FPS), cho phép xử lý gần thời gian thực cho nhiều tác vụ giám sát nông nghiệp.

- **Giảm tiêu thụ năng lượng:** Ước tính tiêu thụ năng lượng cho mỗi lần suy luận cũng giảm 75% (từ 250 mJ xuống 62.5 mJ). Điều này trực tiếp góp phần kéo dài tuổi thọ pin hoạt động của UAV, cho phép thực hiện các nhiệm vụ dài hơn hoặc thu thập dữ liệu thường xuyên hơn.

Kết quả thực nghiệm chứng minh lợi thế của mô hình TinyML lượng tử hóa trên UAV: giảm 0.7% độ chính xác nhưng đạt hiệu quả vượt trội. Kích thước mô hình giảm 4 lần, tốc độ suy luận tăng 4 lần (0.8 lên 3.2 FPS), và năng lượng tiêu thụ giảm 75%, cho phép

xử lý thời gian thực và kéo dài thời gian bay. TinyML lượng tử hóa là giải pháp khả thi cho AI biên trên UAV, giảm phụ thuộc đám mây và tăng cường bảo mật. Tuy nhiên, mô phỏng có hạn chế về tham số phần cứng lý tưởng hóa, động lực UAV đơn giản, dữ liệu tĩnh và thiếu vòng lặp phản hồi. Cần xác thực trên phần cứng thực tế và thử nghiệm thực địa trong tương lai.

V. Kết luận và Công việc tương lai

Bài báo này đã trình bày một đánh giá toàn diện về các mô hình phân loại hình ảnh TinyML lượng tử hóa cho các ứng dụng giám sát nông nghiệp trên các nền tảng UAV mô phỏng. Chúng tôi đã chứng minh rằng lượng tử hóa INT8 sau huấn luyện làm giảm đáng kể kích thước mô hình, độ phức tạp tính toán (FLOPs), thời gian suy luận và tiêu thụ năng lượng, chỉ với một tác động nhỏ đến độ chính xác phân loại. Khung mô phỏng MATLAB/Simulink của chúng tôi đã cung cấp một môi trường được kiểm soát để định lượng những sự đánh đổi này, đưa ra những hiểu biết có giá trị về tính khả thi và lợi ích của việc triển khai các giải pháp AI hiệu quả tại biên.

Những phát hiện cho thấy TinyML, đặc biệt thông qua các kỹ thuật lượng tử hóa, là một cách tiếp cận đầy hứa hẹn để khắc phục những hạn chế tài nguyên cố hữu của UAV, cho phép AI thời gian thực, trên thiết bị cho nông nghiệp chính xác. Khả năng này có thể dẫn đến các hoạt động nông nghiệp tự chủ và hiệu quả hơn,

cho phép can thiệp kịp thời và quản lý tài nguyên tối ưu.

Công việc tương lai

Dựa trên nghiên cứu này, một số hướng công việc trong tương lai có thể được khám phá:

- **Xác thực phần cứng:** Tiến hành các thử nghiệm bằng cách triển khai các mô hình lượng tử hóa trên các nền tảng phần cứng biên thực tế (ví dụ: ESP32-CAM, OpenMV Cam H7 hoặc NVIDIA Jetson Nano cho các mô hình phức tạp hơn) được tích hợp với một UAV vật lý. Điều này sẽ xác thực các kết quả mô phỏng và xác định bất kỳ sự khác biệt nào phát sinh từ phần cứng và các yếu tố môi trường trong thực tế.

- **Các kỹ thuật lượng tử hóa nâng cao:** Nghiên cứu các phương pháp lượng tử hóa tiên tiến hơn, chẳng hạn như Huấn luyện nhận biết lượng tử hóa (QAT) hoặc lượng tử hóa độ chính xác hỗn hợp, có thể mang lại khả năng giữ lại độ chính xác tốt hơn ở các độ rộng bit thấp hơn.

- **Cắt tỉa mô hình và Chung cất tri thức:** Kết hợp lượng tử hóa với các kỹ thuật nén mô hình khác như cắt tỉa (loại bỏ các kết nối dư thừa) và chung cất tri thức (chuyển giao tri thức từ một mô hình lớn sang một mô hình nhỏ hơn) để đạt được hiệu quả cao hơn nữa.

- **Phát hiện đối tượng và Phân đoạn ngữ nghĩa:** Mở rộng nghiên cứu sang các tác vụ thị giác máy tính phức tạp hơn, chẳng hạn như phát hiện đối tượng thời gian thực (ví dụ: phát hiện từng cây,

cỏ dại hoặc sâu bệnh) hoặc phân đoạn ngữ nghĩa (ví dụ: phân loại pixel theo pixel của các khu vực cây trồng) trên UAV, những tác vụ rất phù hợp cho nông nghiệp chính xác.

- **Các yếu tố môi trường động:**

Kết hợp các yếu tố môi trường thực tế hơn vào mô phỏng, chẳng hạn như các điều kiện ánh sáng khác nhau, hiệu ứng thời tiết và chướng ngại vật động, để đánh giá tính mạnh mẽ của mô hình.

- **Thu hoạch năng lượng và Quản lý năng lượng:** Khám phá việc tích hợp các kỹ thuật thu hoạch năng lượng trên UAV và các chiến lược quản lý năng lượng thông minh giúp kéo dài thời gian hoạt động cho các tác vụ AI biên.

- **Học liên kết trên đàn UAV:** Nghiên cứu tiềm năng của các phương pháp học liên kết trong đó nhiều UAV cùng nhau huấn luyện hoặc cập nhật mô hình mà không tập trung hóa dữ liệu thô, tăng cường quyền riêng tư và khả năng mở rộng.

Bằng cách giải quyết các hướng đi trong tương lai này, lĩnh vực TinyML trên UAV có thể tiếp tục phát triển, mở ra những khả năng mới cho các hệ thống trên không tự chủ và thông minh trong nhiều ứng dụng quan trọng khác nhau.

Tài liệu tham khảo:

- [1]. Lottes, P., Behley, J., & Stachniss, C. (2017). UAV-Based Crop and Weed Classification for Smart Farming. IEEE International Conference on Robotics and Automation (ICRA), 2017.
- [2]. Puneet Saini, D.S. Nagesh (2025). A review of deep learning applications in weed detection: UAV and robotic approaches for precision agriculture, European Journal of Agronomy Volume 168, July 2025, 127652.
- [3]. Aissam El Ibrahim, Nabil El Akchioui. (2022). A review on plant diseases detection using artificial intelligence techniques, international conference on advances in communication technology and computer engineering, 24–25 June 2022, Meknes, Morocco
- [4]. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool. (2016). DeepFruits: A Fruit Detection System Using Deep Neural Networks. IEEE International Conference on Robotics and Automation (ICRA), Sensors, vol. 16, no. 8, p. 1222, 2016
- [5]. Yazhou Yuan; Shicong Gao; Ziteng Zhang; Wenye Wang; Zhezhuang Xu; Zhixin Liu (2024) Edge-Cloud Collaborative UAV Object Detection: Edge-Embedded Lightweight Algorithm Design and Task Offloading Using Fuzzy Neural Network, IEEE Transactions on Cloud Computing, Volume: 12, Issue: 1, Jan.-March 2024
- [6]. Oufettoul, H., Chaibi, R., & Motahhir, S. (2024). *TinyML applications, research challenges, and future research directions*. In *Proceedings of the IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE
- [7]. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

- [8]. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional
- [9]. Neural Networks. International Conference on Machine Learning (ICML), 2019, arXiv:1905.11946
- [10]. Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. European Conference on Computer Vision (ECCV), arXiv:1807.11164
- [11]. Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A survey. arXiv preprint arXiv:1806.08342.
- [12]. Rastegari, M., Anwar, M., Rasool, J., Maida, K., & Davis, L. (2016). XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. European Conference on Computer Vision (ECCV), arXiv:1603.05279
- [13]. Krishnamoorthi, R. (2018). *Quantizing deep convolutional networks for efficient inference: A whitepaper*. arXiv:1806.08342.
- [14]. MathWorks. (n.d.). UAV - MATLAB & Simulink. Retrieved from <https://www.mathworks.com/solutions/robotics/uav.html>
- [15]. Zheng, X., He, Y., & Cheng, J. (2015). *Design and implementation of UAV flight simulation based on Matlab/Simulink*. In *Proceedings of the 2015 International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2015)*. Atlantis Press.
- [16]. Parameswari, P., Ramya, S., Sritha, R., & Tusita, M. (2024). *Unmanned aerial vehicle for smart agriculture*. *International Journal of Creative Research Thoughts (IJCRT)*, 12(3), c37–c42.
- [17]. Hüseyin Şahin; Berkay Gürkan; Vasfi Emre Ömürlü (2022). Sensor Fusion Design by Extended and Unscented Kalman Filter Approaches for Position and Attitude Estimation. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, 09-11 June 2022.
- [18]. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. arXiv. arXiv:1801.04381
- [19]. Hughes, D. P., & Salathé, M. (2015). An open access repository of plant disease images for the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.
- [20]. MathWorks. (n.d.). Import Deep Learning Models from External Platforms. Retrieved from <https://www.mathworks.com/help/deeplearning/ug/import-deep-learning-modelsfrom-external-platforms.html>

PERFORMANCE EVALUATION OF QUANTIZED TINYML IMAGE CLASSIFICATION MODELS ON SIMULATED UAV PLATFORMS FOR AGRICULTURAL MONITORING APPLICATIONS

Hoang Trong Nghia²

Abstract: UAVs offer real-time data collection and monitoring capabilities in agriculture. However, deploying advanced AI, especially deep learning, on UAVs is constrained by computational resources, memory, and energy. This study investigates the performance of quantized TinyML image classification models for agricultural monitoring. We propose a MATLAB/Simulink simulation framework to evaluate the trade-offs among accuracy, inference speed, and resource consumption. Results indicate that quantized TinyML models significantly reduce computational footprint and energy consumption while maintaining acceptable accuracy, enabling efficient on-device AI processing. This research contributes to the deployment of efficient AI solutions on edge devices, fostering more autonomous and sustainable agriculture.

Keywords: UAV, Edge AI, tinyML, quantization, image classification, agricultural monitoring, MATLAB/Simulink, real-time processing

² Faculty of Electrical and Electronics Engineering, Hanoi Open University