

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KỲ
HAND SIGN NUMBER RECOGNITION

Giảng viên giảng dạy: Thầy Nguyễn Đức Hoàng Hạ

Môn: Nhập môn lập trình điều khiển thiết bị thông minh

Họ và tên sinh viên: Nguyễn Minh Tuấn - 21120587

Lớp: CQ2021/23

Thành phố Hồ Chí Minh, tháng 1 năm 2025

Mục lục

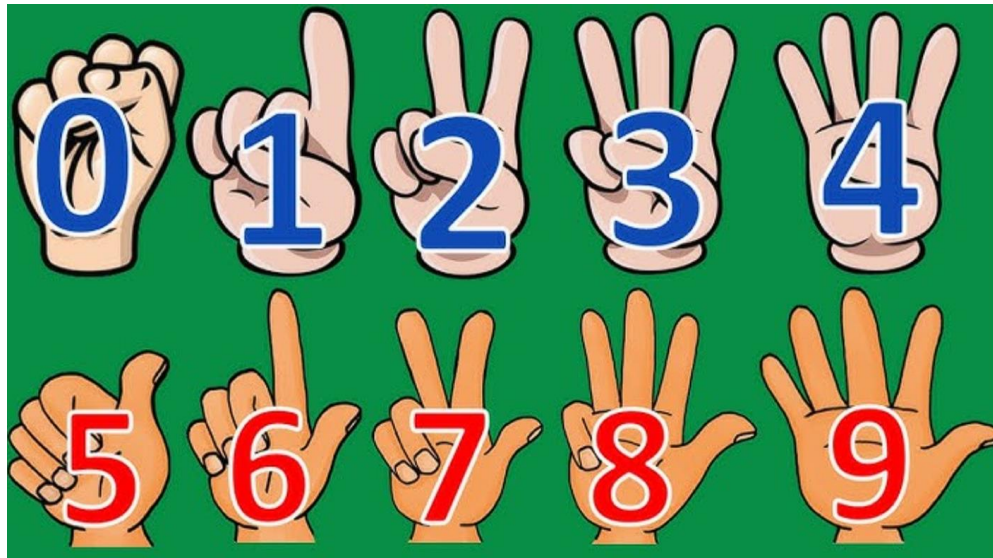
1. Giới thiệu chung:	3
1.1 Nội dung yêu cầu:	3
1.2 Phương pháp thực hiện:	4
2. Xây dựng mô hình:	5
2.1 Thu thập tập dữ liệu:	5
2.2 Kiến trúc mô hình và các tham số:	7
2.2.1 Khối Image:	8
2.2.2 Khối Object Detection:	9
2.3 Hiệu suất của mô hình:	12
3. Thực nghiệm trên thiết bị và kết quả:	14
3.1 Thực hiện deploy trên thiết bị ESP32:	14
3.2 Kết quả thu được:	17
3.3 Đánh giá kết quả:	22
4. Tài liệu tham khảo:	22

1. Giới thiệu chung:

1.1 Nội dung yêu cầu:

Mục tiêu đồ án:

Mục tiêu của đồ án này là huấn luyện một mạng nơ-ron học sâu để nhận diện các con số dựa trên cử chỉ tay. Sinh viên sẽ áp dụng các kỹ thuật học máy để tạo ra một mô hình có khả năng phân loại cử chỉ tay đại diện cho các số từ 0 đến 9.



Giai đoạn 1: Thu thập dữ liệu:

Sinh viên sẽ thu thập và chụp 5 hình ảnh cho mỗi con số, tổng cộng là 50 hình ảnh. Các hình ảnh này cần được gắn nhãn với đóng khung (bounding boxes) cho các nhãn từ zero/one/.../nine. Dữ liệu sau đó sẽ được xuất từ Edge Impulse và tải lên thư mục chung để có thể tải về.

Giai đoạn 2 – Xây dựng và hoàn thiện mô hình nhận dạng:

Sinh viên cần huấn luyện một mô hình để nhận diện 10 ký hiệu tay. Sau đó, mô hình phải được triển khai lên thiết bị ESP32.

1.2 Phương pháp thực hiện:

Trong dự án này, em sẽ sử dụng Edge Impulse để thu thập dữ liệu và xây dựng mô hình nhận diện. Những mô hình mà Edge Impulse hỗ trợ có đặc điểm kích thước nhỏ và phù hợp với việc deploy trên thiết bị ESP32 bao gồm:

MobileNetV2 SSD FPN-Lite (chỉ hỗ trợ độ phân giải 320x320)

Mô hình phát hiện đối tượng đã được huấn luyện sẵn, được thiết kế để định vị tối đa 10 đối tượng trong một hình ảnh và xuất ra bounding box cho mỗi đối tượng được phát hiện. Mô hình này có kích thước khoảng 3.7MB. Nó hỗ trợ đầu vào RGB ở độ phân giải 320x320 pixel.

FOMO (Faster Objects, More Objects) MobileNetV2 0.1

Mô hình phát hiện đối tượng dựa trên MobileNetV2 (alpha 0.1), được thiết kế để phân đoạn thô một hình ảnh thành lưới gồm nền và các đối tượng quan tâm. Các mô hình này được thiết kế với kích thước <100KB và hỗ trợ đầu vào dạng grayscale (đơn sắc) hoặc RGB ở bất kỳ độ phân giải nào.

FOMO (Faster Objects, More Objects) MobileNetV2 0.35

Mô hình phát hiện đối tượng dựa trên MobileNetV2 (alpha 0.35), được thiết kế để phân đoạn thô một hình ảnh thành lưới gồm nền và các đối tượng quan tâm. Các mô hình này được thiết kế với kích thước <100KB và hỗ trợ đầu vào dạng grayscale hoặc RGB ở bất kỳ độ phân giải nào.

Em sẽ thử và đánh giá các mô hình xem việc sử dụng mô hình nào là tối ưu. Quá trình triển khai mô hình lên thiết bị ESP32 sẽ được thực hiện thông qua Arduino IDE.

Mô tả quy trình xây dựng mô hình và deploy:

- Thu thập dữ liệu và tiền xử lý.
- Xây dựng và huấn luyện các mô hình và đánh giá trên Edge Impulse.
- Xuất mô hình và triển khai qua Arduino IDE.
- Kiểm thử và tối ưu hóa trên thiết bị.
- Quay Video kiểm thử và soạn tài liệu báo cáo nội dung thực hiện.

2. Xây dựng mô hình:



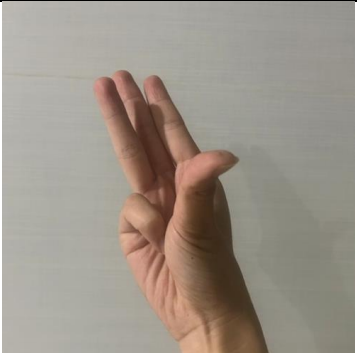
2.1 Thu thập tập dữ liệu:

Mô tả dữ liệu:

Bộ dữ liệu trong dự án bao gồm 632 hình ảnh các ký hiệu tay biểu diễn các số từ 0 đến 9. Tất cả hình ảnh được gán nhãn tương ứng với từng số để phục vụ quá trình huấn luyện và đánh giá mô hình.

Trong quá trình chuẩn bị dữ liệu, các hình ảnh kém chất lượng đã được loại bỏ, bao gồm những hình ảnh có góc chụp khó, không rõ nét, bị mất góc quan trọng hoặc nhiễu do ánh sáng và phong nền. Việc này nhằm đảm bảo chất lượng dữ liệu, giảm thiểu ảnh hưởng tiêu cực đến quá trình huấn luyện mô hình.

Ví dụ các mẫu cần loại bỏ:

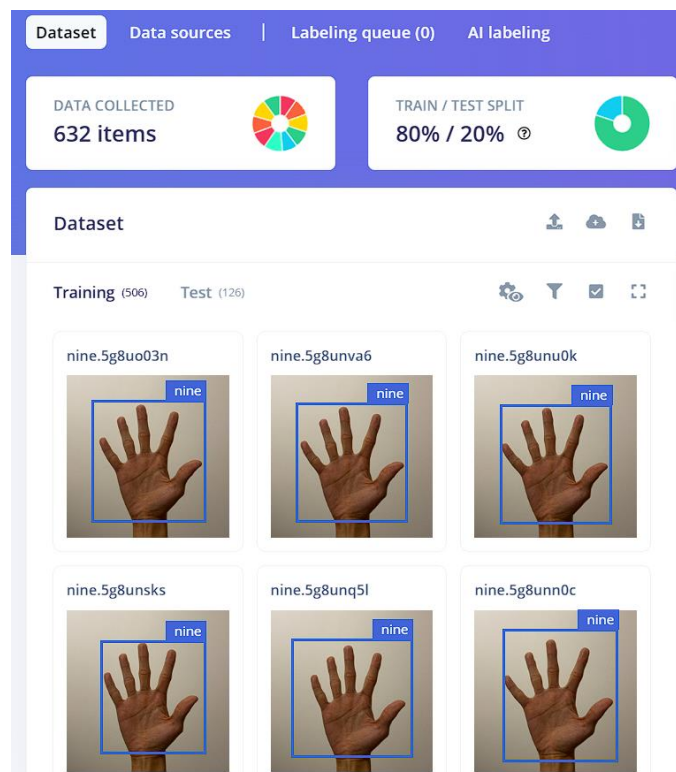
Mất góc, mờ, thiếu sáng	Dáng tay không rõ	Dáng tay không rõ
		

Ngoài ra, dữ liệu được bổ sung và thu thập với phong nền thực tế sẽ thực hiện deploy và sử dụng bàn tay cá nhân để tăng tính cá nhân hóa. Các hình ảnh này giúp mô hình có khả năng nhận diện chính xác hơn khi thử nghiệm trên các điều kiện thực tế, với phong nền và ánh sáng đã được huấn luyện. Ngoài ra còn làm dày tập dữ liệu. Dữ liệu được phân chia thành hai tập chính:

80% cho tập train (506 hình ảnh).

20% cho tập test (126 hình ảnh).

Tỉ lệ của các nhãn:



Sự cân bằng giữa các nhãn trong dữ liệu được duy trì, đảm bảo mỗi số từ 0 đến 9 đều có số lượng hình ảnh gần như đồng đều, giúp tránh hiện tượng thiên lệch nhãn trong quá trình huấn luyện. Những bước này giúp cải thiện chất lượng dữ liệu và tăng khả năng tổng quát hóa của mô hình, đảm bảo mô hình hoạt động ổn định khi triển khai trên thiết bị ESP32.

Đánh giá:

Bộ dữ liệu đã được xử lý và chuẩn bị khá tốt, phù hợp để huấn luyện mô hình trong phạm vi của đề án. Tuy nhiên, để cải thiện thêm, có thể mở rộng quy mô dữ liệu và tăng độ đa dạng về nguồn cung cấp cũng như môi trường thu thập. Việc này sẽ giúp mô hình hoạt động ổn định và chính xác hơn khi triển khai trên các điều kiện thực tế đa dạng.

Với tổng cộng 632 hình ảnh, bộ dữ liệu vẫn còn tương đối nhỏ, đặc biệt khi triển khai mô hình trên thực tế, có thể gặp các trường hợp ngoài dữ liệu huấn luyện. Vì vậy cần thu thập thêm nhiều dáng tay khác với đa dạng các trường hợp ứng với thực tế có thể sẽ xảy ra khi test.

2.2 Kiến trúc mô hình và các tham số:

Mô hình bao gồm hai khối chính: Khối Image để trích xuất đặc trưng từ hình ảnh và Object Detection sử dụng một trong các mô hình được đề xuất để phân loại ký hiệu tay.

The screenshot shows three panels in the Google AI Platform interface:

- Image panel (light blue):** Contains a 'Name' field with 'Image', 'Input axes (1)' with 'Image', and a 'Save Impulse' button.
- Object Detection (Images) panel (purple):** Contains a 'Name' field with 'Object detection', 'Input features' with a checked 'Image' checkbox, and 'Output features' with '10 (eight, five, four, nine, one, seven, six, three, two, zero)'.
- Output features panel (teal):** Contains the same 'Output features' text and a 'Save Impulse' button.

- Input:

- + 320 x 320: đối với mô hình MobileNetV2 SSD FPN-Lite (320x320 only).
- + 96 x 96: đối với mô hình FOMO (Faster Objects, More Objects) MobileNetV2 0.1 và FOMO (Faster Objects, More Objects) MobileNetV2 0.35.

The screenshot shows two panels in the Google AI Platform interface for 'Image data' configuration:

- Left panel (orange):** 'Input axes' is 'image'. 'Image width' and 'Image height' are both set to 320. 'Resize mode' is 'Fit shortest axis'.
- Right panel (red):** 'Input axes' is 'image'. 'Image width' and 'Image height' are both set to 96. 'Resize mode' is 'Fit shortest axis'.

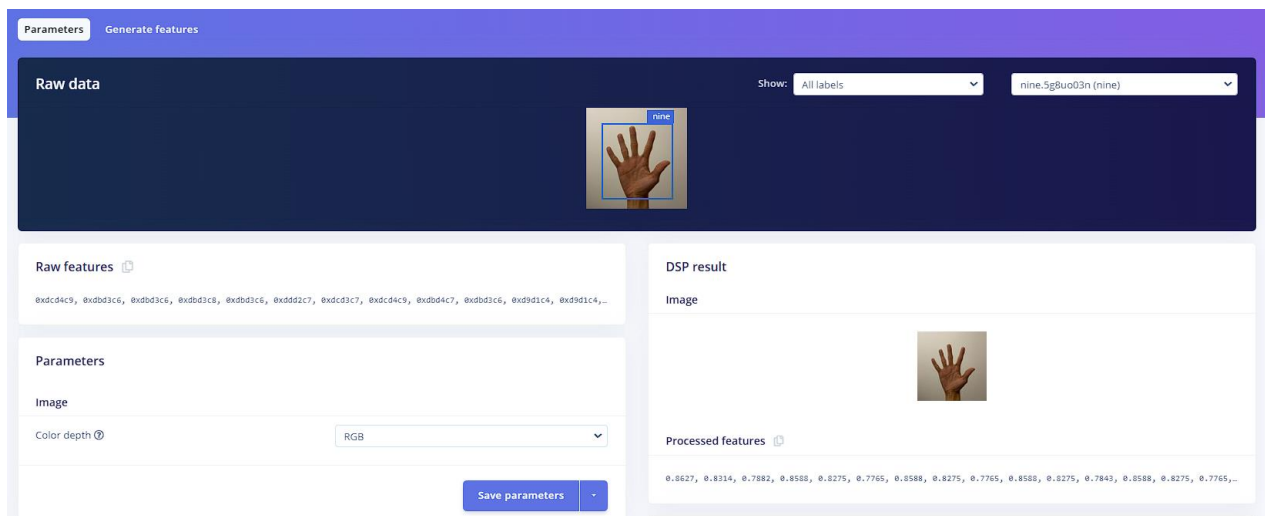
- Số lượng đầu ra (Output): 10 nhãn 'one', 'two', ..., 'nine'.

2.2.1 Khối Image:

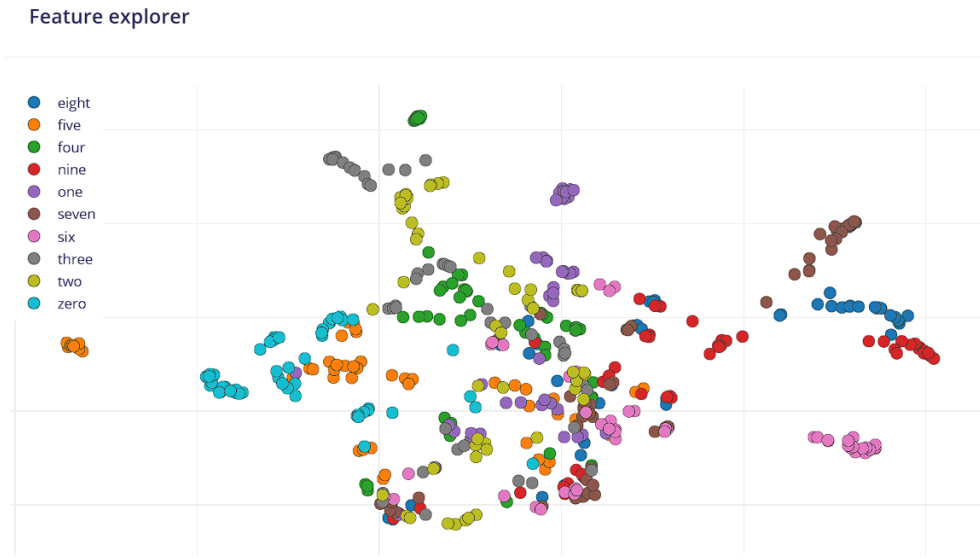
Chức năng: Khối Image trong mô hình đóng vai trò quan trọng trong việc trích xuất đặc trưng từ hình ảnh đầu vào, giúp chuẩn bị dữ liệu để phục vụ cho bước phân loại ký hiệu tay ở khối Object Detection.

Dữ liệu hình ảnh đầu vào được sử dụng bao gồm các hình ảnh màu (RGB), phản ánh đầy đủ các thông tin về màu sắc và đặc điểm hình dạng của ký hiệu tay. Hình ảnh được gán nhãn tương ứng với các số từ 0 đến 9, giúp mô hình học được mối liên hệ giữa đặc trưng hình ảnh và nhãn.

Ngoài ra, sử dụng độ sâu màu RGB, mỗi pixel trong hình ảnh được đại diện bởi ba kênh màu (Red, Green, Blue) có thể giúp nắm bắt đầy đủ thông tin về đặc điểm hình học và màu sắc.



Kết quả đầu ra của khối này sẽ là đầu vào cho khối phân loại để nhận diện từ khóa. Khối Image là bước đầu tiên trong pipeline xử lý, đảm bảo rằng các hình ảnh được chuẩn bị tốt để đưa vào khối Object Detection. Nó thực hiện nhiệm vụ chuyển đổi dữ liệu hình ảnh thô thành các đặc trưng trích xuất, dễ dàng sử dụng cho việc huấn luyện và suy diễn. Kết quả thu được như sau:



2.2.2 Khối Object Detection:

Chức năng:

Khối Object Detection trong dự án đóng vai trò quan trọng trong việc nhận diện và phân loại các ký hiệu tay từ hình ảnh đã được xử lý bởi khối Image. Đây là bước tiếp theo trong pipeline, nơi đặc trưng trích xuất từ hình ảnh có thể được sử dụng để xác định vị trí và phân loại các đối tượng.

Mô hình sử dụng:

- MobileNetV2 SSD FPN-Lite: Phát hiện đa đối tượng trên ảnh, cung cấp bounding box và nhãn cho tối đa 10 đối tượng. Kích thước mô hình lớn hơn nhưng phù hợp nếu cần độ chính xác cao và ảnh có độ phân giải cố định (320x320 pixel).
- FOMO MobileNetV2: Nhẹ và hiệu quả, tối ưu cho ESP32. Mô hình phân đoạn ảnh thành các ô lưới để phát hiện nền và đối tượng quan tâm, phù hợp với các ứng dụng yêu cầu tài nguyên hạn chế. ($\alpha = 0.1$ và 0.35).

a) MobileNetV2 SSD FPN-Lite

Đây là mô hình phát hiện đối tượng gọn nhẹ và hiệu quả, với kiến trúc được tối ưu cho phát hiện đa đối tượng.

Kiến trúc Mạng Nơ-ron Lớp đầu vào: Kích thước đầu vào: 320x320 RGB. Lớp đầu ra: 10 lớp (tương ứng với các số từ 0 đến 9).

Tham số huấn luyện:

Số chu kỳ huấn luyện (Training cycles): 28 (bản miễn phí có giới hạn thời gian train là 20 phút, kết quả là số epoch cũng bị rút ngắn không như mong muốn)

Tốc độ học (Learning rate): 0.02

Bộ xử lý huấn luyện (Training processor): CPU

Tỷ lệ tập xác thực (Validation set size): 20%

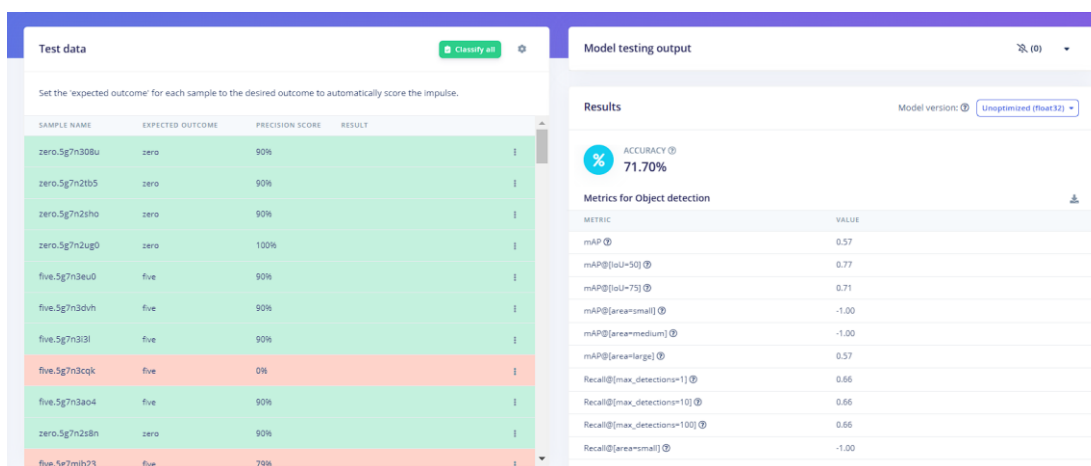
Batch size: 32

Kết quả huấn luyện:

Điểm Precision trên tập xác thực: 54.8% Điểm Precision đo lường độ chính xác của các dự đoán đúng so với tất cả các dự đoán của mô hình trên tập xác thực.

Kết quả kiểm tra mô hình:

Độ chính xác (Accuracy): 71.70% Độ chính xác cho thấy tỷ lệ dự đoán đúng của mô hình trên tập kiểm tra, thể hiện khả năng phân loại khá chính xác các ký hiệu tay. Đây là kết quả có thể chấp nhận được khi sử dụng mô hình gốc và chưa qua tinh chỉnh chi tiết mô hình



The screenshot displays a web-based model testing interface. On the left, the 'Test data' section contains a table with columns: SAMPLE NAME, EXPECTED OUTCOME, PRECISION SCORE, and RESULT. It lists 12 samples, mostly with 'zero' or 'five' as expected outcomes and 90% or 100% precision scores. One sample, 'five_5g7n3cqk', shows a 0% precision score. On the right, the 'Model testing output' section shows the 'Results' for 'Model version: Unoptimized (float32)'. It highlights an 'ACCURACY' of 71.70% and provides a detailed table of 'Metrics for Object detection' including mAP, mAP@[IoU=50], mAP@[IoU=75], mAP@[area=small], mAP@[area=medium], mAP@[area=large], Recall@[max_detections=1], Recall@[max_detections=10], Recall@[max_detections=100], Recall@[area=small], and Recall@[area=medium].

Metric	Value
mAP	0.57
mAP@[IoU=50]	0.77
mAP@[IoU=75]	0.71
mAP@[area=small]	-1.00
mAP@[area=medium]	-1.00
mAP@[area=large]	0.57
Recall@[max_detections=1]	0.66
Recall@[max_detections=10]	0.66
Recall@[max_detections=100]	0.66
Recall@[area=small]	-1.00
Recall@[area=medium]	-1.00

Đánh giá

Kết quả huấn luyện và kiểm tra cho thấy mô hình đã đạt mức hiệu suất khá tốt trên tập dữ liệu giới hạn. Tuy nhiên, với điểm Precision thấp, mô hình cần được cải thiện thêm để giảm tỷ lệ dự đoán sai.

Việc tinh chỉnh thêm các siêu tham số như tốc độ học, số chu kỳ huấn luyện, hoặc thử nghiệm các kiến trúc mô hình khác có thể giúp tăng hiệu quả. Tuy nhiên, đây là những siêu tham số mà em có thể tìm được để đem lại kết quả tối ưu khi sử dụng mô hình MobileNetV2 SSD FPN-Lite. Mặt khác, mô hình này có kích thước còn khá lớn nên chưa thể sử dụng để deploy trên thiết bị ESP32. Vì vậy, việc cân nhắc tìm hiểu và đánh giá thêm để sử dụng mô hình khác là cần thiết.

b) FOMO (Faster Objects, More Objects) MobileNetV2:

FOMO MobileNetV2 được xây dựng trên nền tảng của MobileNetV2, một kiến trúc hiệu quả trong việc giảm kích thước mô hình và tính toán mà vẫn duy trì hiệu suất cao. MobileNetV2 sử dụng các lớp depthwise separable convolutions để giảm số lượng tham số và tính toán, giúp mô hình nhẹ và nhanh chóng, phù hợp với các thiết bị có tài nguyên hạn chế như điện thoại di động, IoT, hoặc các thiết bị nhúng như ESP32.

Các lượt thử nghiệm đem được kết quả như sau:

Model	Learning Rate (LR)	Training Cycles	Validation Set Size	Batch Size	F1 Score	Accuracy
FOMO MobileNetV2 0.1	0.015	90	20%	32	54.3%	41.51%
FOMO MobileNetV2 0.35 (LR=0.01)	0.01	90	20%	32	82.6%	60.32%
FOMO MobileNetV2 0.35 (LR=0.015)	0.015	90	20%	32	76.3%	72.22%
FOMO MobileNetV2 0.35 (LR=0.02)	0.02	90	20%	32	71.4%	52.38%

Tổng quan:

FOMO MobileNetV2 0.35 là lựa chọn vượt trội hơn so với FOMO MobileNetV2 0.1 khi cần tối ưu hóa độ chính xác và hiệu suất trong bài toán nhận diện ký hiệu tay. Mô hình 0.35 không chỉ cải thiện độ chính xác mà còn giúp giảm thiểu các lỗi phân loại, mang lại sự ổn định và hiệu quả cao hơn trong môi trường thực tế.

Về chi tiết:

FOMO MobileNetV2 0.1 (LR=0.015): Độ chính xác thấp nhất (41.51%), cho thấy khả năng nhận diện kém và ít hiệu quả đối với dữ liệu.

FOMO MobileNetV2 0.35 (LR=0.01): Độ chính xác là 60.32%, trong khi F1 Score khá cao (82.6%), cho thấy mô hình này có khả năng nhận diện chính xác hơn mặc dù có độ chính xác thấp hơn so với mô hình LR=0.015.

FOMO MobileNetV2 0.35 (LR=0.015): Độ chính xác tốt nhất (72.22%) và F1 Score khá cao (76.3%), là mô hình hoạt động ổn định nhất trong các thử nghiệm.

FOMO MobileNetV2 0.35 (LR=0.02): Độ chính xác thấp hơn (52.38%) và F1 Score giảm (71.4%) so với mô hình sử dụng learning rate 0.015.

Kết Luận:

Mô hình FOMO MobileNetV2 0.35 với Learning Rate 0.015 cho kết quả tốt nhất về độ chính xác (72.22%) và F1 Score (76.3%), là lựa chọn tối ưu.

2.3 Hiệu suất của mô hình:

Kết quả đạt được với mô hình được chọn: FOMO MobileNetV2 0.35 (LR = 0.015).

Hiệu suất mô hình:

- F1 Score (validation set): 76.3%
- Accuracy: 72.22%
- Precision (non-background): 80%
- Recall (non-background): 89%
- F1 Score (non-background): 84%

Confusion Matrix (validation set):

	BACKGRO	EIGHT	FIVE	FOUR	NINE	ONE	SEVEN	SIX	THREE	TWO	ZERO
BACKGROUND	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0%	0.0%	0.0%	0.0%
EIGHT	54.5%	45.5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
FIVE	45.5%	0%	54.5%	0%	0%	0%	0%	0%	0%	0%	0%
FOUR	7.1%	0%	0%	92.9%	0%	0%	0%	0%	0%	0%	0%
NINE	50%	0%	0%	0%	50%	0%	0%	0%	0%	0%	0%
ONE	50%	0%	0%	0%	0%	50%	0%	0%	0%	0%	0%
SEVEN	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
SIX	9.1%	0%	0%	0%	0%	0%	0%	90.9%	0%	0%	0%
THREE	33.3%	0%	0%	0%	0%	0%	0%	0%	66.7%	0%	0%
TWO	7.1%	0%	0%	0%	0%	0%	0%	0%	0%	92.9%	0%
ZERO	9.1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	90.9%
F1 SCORE	1.00	0.56	0.63	0.72	0.60	0.55	0.95	0.95	0.75	0.90	0.77

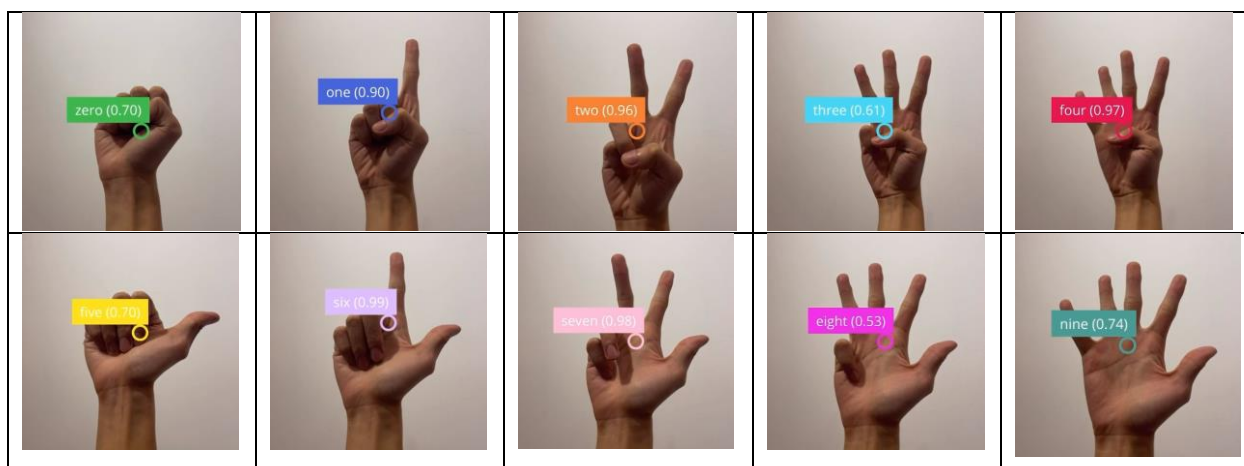
Mô hình hoạt động khá tốt với các ký hiệu dễ nhận diện. Một số điểm cần chú ý:

Các ký hiệu "four", "seven", "six", "two", "three" và "zero": Độ chính xác cao.

Các ký hiệu "eight", "five", "nine", "one": Có độ chính xác thấp hơn, với các sai sót chủ yếu liên quan đến nhầm lẫn giữa các lớp này.

Trong đó, khả năng "eight" bị nhận diện sai sót là cao nhất, eight có đặc điểm gần giống với các số gần nó. Đồng thời, bộ data của "eight" chưa đa dạng dẫn đến kết quả dự đoán chưa khả quan.

Kết quả khi kiểm tra trên thiết bị điện thoại thông minh với mô hình FOMO MobileNetV2 0.35 (LR = 0.015).



Nhận xét:

Mô hình đạt F1 Score khá cao (76.3%) và độ chính xác 72.22%, cho thấy khả năng phân loại tốt trong môi trường kiểm tra. Precision (80%) và Recall (89%) cho các lớp không phải background cũng khá ấn tượng, chứng tỏ mô hình có thể phát hiện hầu hết các đối tượng nhưng vẫn cần cải thiện độ chính xác trong một số trường hợp.

Các số đa phần đều được nhận diện tốt, trường hợp số 8 với nhãn "eight" vì có hình dáng khó nhận biết nên cần mất thời gian lâu để có thể nhận diện, đồng thời tỉ lệ nhận diện được là chưa cao. Bộ dataset cần được bổ sung thêm các mẫu nhãn "eight" có chất lượng tốt để nâng cao hiệu suất của mô hình.

3. Thực nghiệm trên thiết bị và kết quả:

3.1 Thực hiện deploy trên thiết bị ESP32:


Thiết bị ESP32-CAM Ai-Thinker:




Các bước thực hiện:

Bước 1: Tải mô hình từ Edge Impulse:

- Tải mô hình đã huấn luyện từ Edge Impulse dưới dạng tệp .zip trong phần Deployment của project.

 **SELECTED DEPLOYMENT**
Arduino library
An Arduino library with examples that runs on most Arm-based Arduino development boards.

MODEL OPTIMIZATIONS
Model optimizations can increase on-device performance but may reduce accuracy.

 **EON™ Compiler**
Same accuracy, 17% less RAM, 29% less ROM.

Quantized (int8)
Selected ✓

	IMAGE	OBJECT DETECTION	TOTAL
LATENCY	15 ms.	1,013 ms.	1,028 ms.
RAM	4.0K	239.6K	239.6K
FLASH	-	79.6K	-
ACCURACY			-

Unoptimized (float32)
Select

	IMAGE	OBJECT DETECTION	TOTAL
LATENCY	15 ms.	2,563 ms.	2,578 ms.
RAM	4.0K	887.1K	887.1K
FLASH	-	104.6K	-
ACCURACY			72.22%

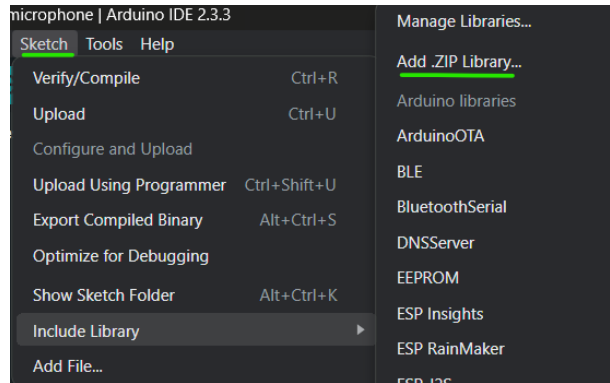
To compare model accuracy, run model testing for all available optimizations.
[Run model testing](#)

Estimate for Espressif ESP-EYE (ESP32 240MHz) - [Change target](#)

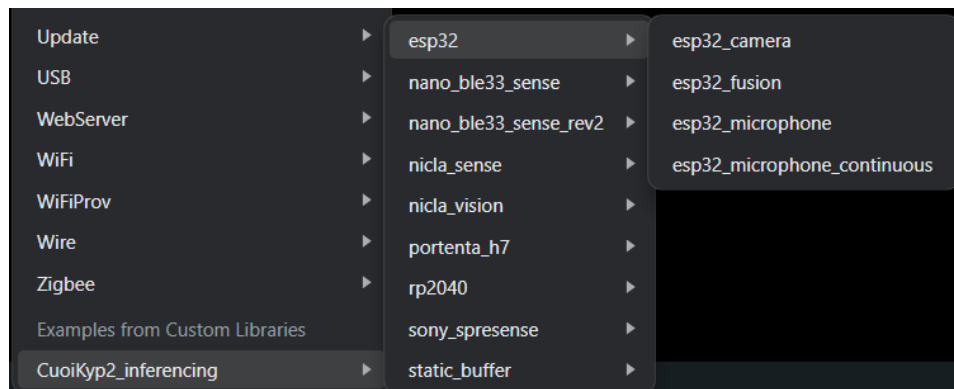
Build

Bước 2: Sử dụng Arduino IDE và include thư viện .zip:

- Trong Arduino IDE, vào Sketch > Include Library > Add .ZIP Library.
- Chọn tệp .zip đã tải từ Edge Impulse chứa thư viện và mô hình đã huấn luyện.



Bước 3: Cài đặt và cấu hình mô hình:

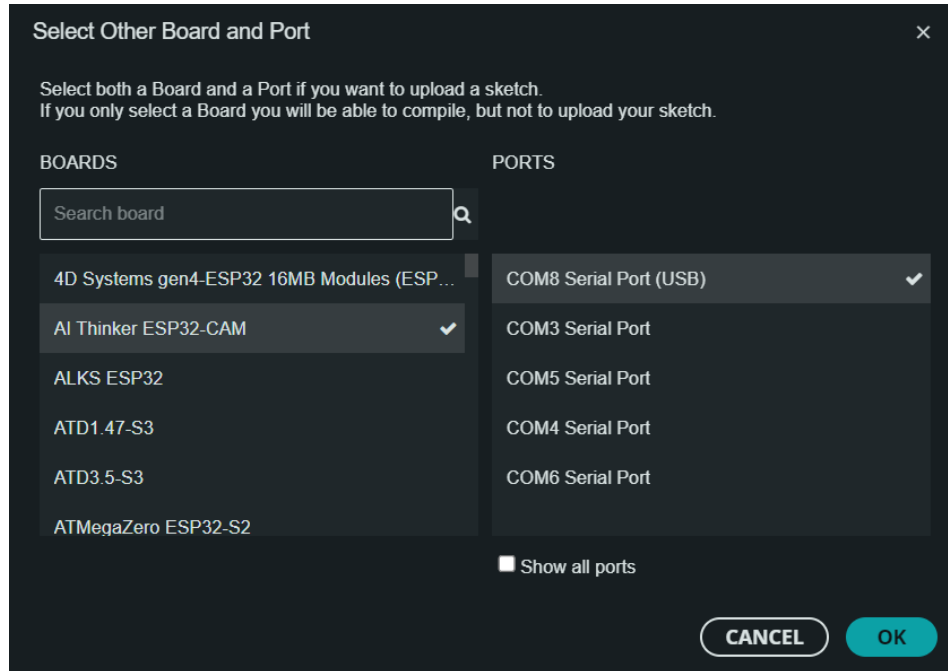


Chọn #define CAMERA_MODEL_AI_THINKER

```
35 // #define CAMERA_MODEL_ESP_EYE // Has PSRAM
36 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

Bước 4: Upload mã lên ESP32:

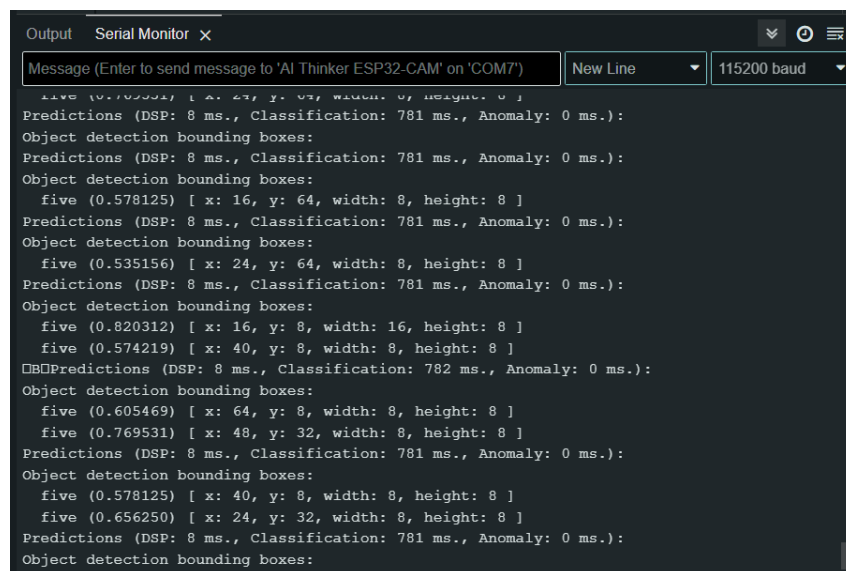
- Chọn Board là AI Thinker ESP32-CAM và Port phù hợp.



- Trong Arduino IDE, chọn Upload để tải chương trình lên ESP32 đến khi nạp thành công.
- Cuối cùng là rút dây và cấp lại nguồn cho thiết bị.

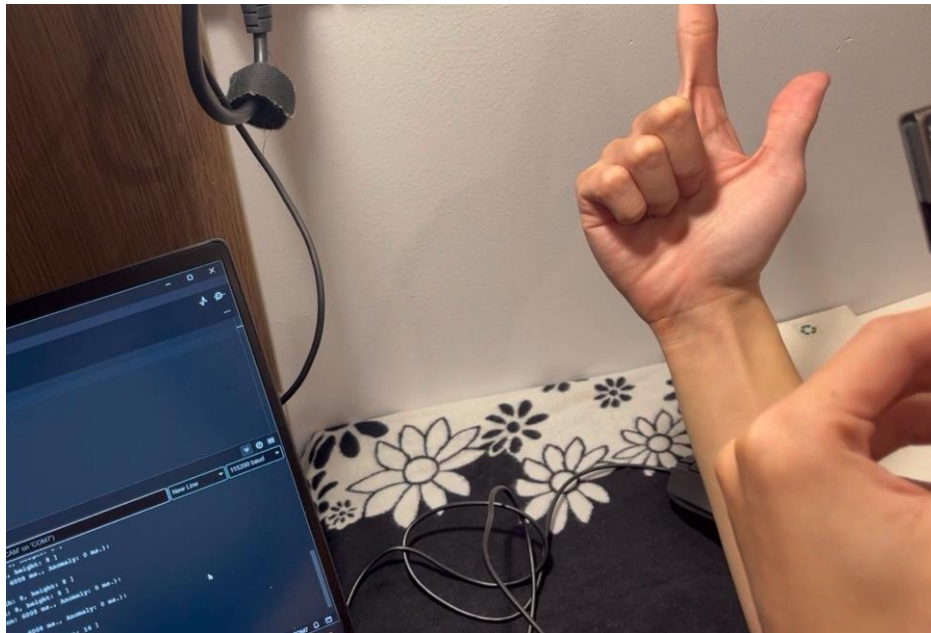
Bước 5: Kiểm tra Kết quả:

Sau khi chương trình được tải lên và cấp lại nguồn cho thiết bị, mở Serial Monitor trong Arduino IDE để kiểm tra kết quả. Chuyển baud rate setting thành 115200 để thấy kết quả. Kết quả tương tự như sau là thành công:

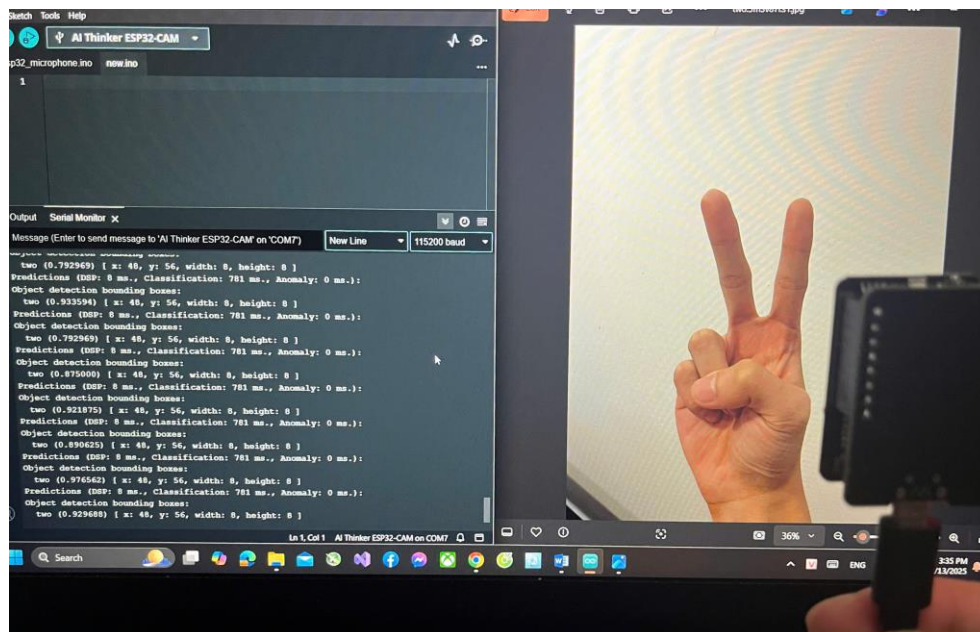


3.2 Kết quả thu được:

Quá trình kiểm tra có thể sử dụng ký hiệu bàn tay cùng với nền trống để tránh gây nhiễu:



Hoặc cũng có thể dùng thiết bị phân để loại ảnh các nhân trên màn hình máy tính và đánh giá độ chính xác:



Kết quả nhận diện các nhãn:

Nhãn	Kết quả
Zero	<pre> Object detection bounding boxes: zero (0.562500) [x: 40, y: 40, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: zero (0.617188) [x: 48, y: 40, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: zero (0.777344) [x: 40, y: 40, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: zero (0.695312) [x: 40, y: 40, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.507812) [x: 40, y: 40, width: 8, height: 8] zero (0.687500) [x: 40, y: 48, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: zero (0.726562) [x: 40, y: 56, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: zero (0.566406) [x: 40, y: 56, width: 8, height: 8] </pre>
One	<pre> Object detection bounding boxes: one (0.937500) [x: 56, y: 32, width: 16, height: 24] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.945312) [x: 56, y: 32, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.992188) [x: 40, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.894531) [x: 40, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.976562) [x: 48, y: 24, width: 8, height: 24] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.972656) [x: 48, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.980469) [x: 48, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.984375) [x: 48, y: 32, width: 16, height: 16] </pre>

Two	<pre> Object detection bounding boxes: two (0.945312) [x: 32, y: 40, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: two (0.972656) [x: 32, y: 40, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: two (0.968750) [x: 40, y: 40, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: two (0.964844) [x: 24, y: 48, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.707031) [x: 32, y: 48, width: 16, height: 8] two (0.917969) [x: 40, y: 56, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.832031) [x: 32, y: 40, width: 8, height: 16] two (0.945312) [x: 24, y: 56, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): </pre>
Three	<div> Output Serial Monitor X <div> Not connected. Select a board and a port to connect automatically. New Line 115200 baud </div> </div> <pre> Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.796875) [x: 32, y: 48, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.832031) [x: 40, y: 48, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.960938) [x: 32, y: 40, width: 8, height: 8] three (0.769531) [x: 32, y: 48, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.984375) [x: 32, y: 40, width: 8, height: 8] three (0.796875) [x: 32, y: 40, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.738281) [x: 32, y: 40, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: three (0.835938) [x: 32, y: 40, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.531250) [x: 32, y: 40, width: 8, height: 8] three (0.796875) [x: 40, y: 40, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: </pre>
Four	<pre> Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.992188) [x: 32, y: 32, width: 16, height: 24] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.941406) [x: 32, y: 40, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.976562) [x: 16, y: 16, width: 8, height: 16] three (0.742188) [x: 24, y: 24, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.992188) [x: 32, y: 32, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.996094) [x: 24, y: 24, width: 16, height: 24] three (0.597656) [x: 40, y: 24, width: 8, height: 8] two (0.511719) [x: 40, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.996094) [x: 32, y: 32, width: 16, height: 24] three (0.542969) [x: 40, y: 32, width: 8, height: 8] </pre>

Five	<div> <div>Output Serial Monitor</div> <div>Not connected. Select a board and a port to connect automatically.</div> <div>New Line</div> <div>115200 baud</div> </div> <pre> five (0.761719) [x: 24, y: 8, width: 8, height: 8] five (0.652344) [x: 40, y: 8, width: 8, height: 8] five (0.613281) [x: 56, y: 8, width: 16, height: 8] five (0.593750) [x: 32, y: 56, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.957031) [x: 32, y: 24, width: 32, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.976562) [x: 16, y: 24, width: 32, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.976562) [x: 40, y: 16, width: 16, height: 24] four (0.613281) [x: 56, y: 56, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.937500) [x: 48, y: 16, width: 16, height: 8] five (0.605469) [x: 40, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.980469) [x: 40, y: 16, width: 24, height: 24] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: five (0.785156) [x: 24, y: 8, width: 8, height: 8] five (0.968750) [x: 32, y: 24, width: 24, height: 8] five (0.570312) [x: 24, y: 40, width: 8, height: 8] </pre>
Six	<pre> Object detection bounding boxes: six (0.960938) [x: 32, y: 40, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: six (0.976562) [x: 40, y: 32, width: 16, height: 16] five (0.867188) [x: 40, y: 56, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: six (0.992188) [x: 40, y: 32, width: 16, height: 16] five (0.605469) [x: 40, y: 64, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: one (0.589844) [x: 40, y: 32, width: 8, height: 8] six (0.980469) [x: 48, y: 32, width: 16, height: 16] five (0.625000) [x: 40, y: 64, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: six (0.980469) [x: 32, y: 32, width: 16, height: 16] five (0.820312) [x: 32, y: 56, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: six (0.984375) [x: 40, y: 32, width: 8, height: 24] seven (0.789062) [x: 48, y: 48, width: 8, height: 8] five (0.785156) [x: 32, y: 64, width: 8, height: 8] </pre>
Seven	<pre> Object detection bounding boxes: seven (0.992188) [x: 48, y: 48, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: seven (0.972656) [x: 32, y: 24, width: 24, height: 16] six (0.644531) [x: 48, y: 24, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: seven (0.968750) [x: 32, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: seven (0.968750) [x: 48, y: 24, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: seven (0.957031) [x: 40, y: 24, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: seven (0.890625) [x: 40, y: 24, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): </pre>

Eight	<p>Object detection bounding boxes: eight (0.648438) [x: 48, y: 32, width: 16, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: eight (0.628906) [x: 40, y: 32, width: 8, height: 8] seven (0.531250) [x: 40, y: 40, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: eight (0.542969) [x: 40, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: eight (0.660156) [x: 40, y: 32, width: 16, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.785156) [x: 32, y: 24, width: 8, height: 8] eight (0.675781) [x: 40, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: eight (0.695312) [x: 40, y: 24, width: 8, height: 16] seven (0.707031) [x: 48, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: four (0.894531) [x: 40, y: 24, width: 8, height: 8] eight (0.640625) [x: 48, y: 32, width: 8, height: 8]</p>
Nine	<p>Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.664062) [x: 16, y: 24, width: 16, height: 8] eight (0.585938) [x: 32, y: 24, width: 8, height: 8] two (0.554688) [x: 40, y: 48, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.789062) [x: 32, y: 24, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6009 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.765625) [x: 40, y: 24, width: 8, height: 16] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.585938) [x: 48, y: 24, width: 8, height: 8] seven (0.609375) [x: 56, y: 32, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.570312) [x: 32, y: 24, width: 8, height: 8] eight (0.664062) [x: 40, y: 24, width: 8, height: 8] Predictions (DSP: 8 ms., Classification: 6008 ms., Anomaly: 0 ms.): Object detection bounding boxes: nine (0.699219) [x: 32, y: 24, width: 16, height: 8]</p>

3.3 Đánh giá kết quả:

Mô hình nhận dạng số từ ký hiệu tay hoạt động khá tốt, cho thấy hiệu quả trong việc xử lý và nhận dạng các ký hiệu số. Việc triển khai thành công trên thiết bị ESP32 chứng minh khả năng tối ưu hóa mô hình cho các ứng dụng nhúng. Mô hình có thời gian xử lý và hiển thị kết quả ổn định, phù hợp để áp dụng vào các tình huống thực tế. Qua đó, ta có thể kết luận mô hình đã đạt được sự tin cậy nhất định và có thể sử dụng để phân loại ký hiệu tay.

Tuy nhiên, một số hạn chế vẫn còn tồn tại. Cụ thể, mô hình thường xảy ra nhầm lẫn giữa số 0 và số 5, có thể do sự tương đồng về hình dạng của hai ký hiệu này hoặc do đặc trưng của chúng không đủ phân biệt trong tập dữ liệu huấn luyện. Ngoài ra, số 8 khó nhận diện và dễ bị nhầm lẫn với các số khác, có khả năng do dữ liệu huấn luyện cho số 8 chưa đủ đa dạng và không nổi bật trong không gian đặc trưng của mô hình. Một số nhầm lẫn khác có thể đến từ điều kiện ánh sáng, góc độ, hoặc chất lượng hình ảnh khi sử dụng thiết bị nhúng.

Để cải thiện hiệu năng, việc tăng cường dữ liệu huấn luyện là cần thiết. Cần thu thập thêm dữ liệu cho các số khó nhận diện như 0, 5, 8 từ nhiều người dùng khác nhau với điều kiện ánh sáng và góc độ đa dạng.

Ngoài ra, có thể cải thiện mô hình bằng cách tinh chỉnh các siêu tham số như learning rate, batch size, hoặc optimizer cũng là một hướng tiếp cận hiệu quả. Có thể thử nghiệm các mô hình phức tạp hơn bằng cách thêm các lớp convolutional hoặc sử dụng các mô hình đã được pre-trained để fine-tune trên tập dữ liệu. Hiệu quả nhất có thể kể đến là việc cải tiến mô hình FOMO MobileNetV2 bằng cách tinh chỉnh ở các lớp để đem lại hiệu suất tốt nhất.

4. Tài liệu tham khảo:

[1] FOMO: Object detection for constrained devices

<https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>

[2] How to do Object Detection using ESP32-CAM and Edge Impulse YOLO Model

<https://www.youtube.com/watch?v=bZIKVaD3dRk>

[3] Object Detection with ESP32-Cam using Custom Model Edge Impulse

<https://www.youtube.com/watch?v=KEcLCgKp1Ls&t=983s>