YOLOV12-SA-EVC: MÔ HÌNH PHÁT HIỆN "Ổ GÀ" TRÊN MẶT ĐƯỜNG BỘ

Dương Thăng Long[1], Vũ Xuân Hạnh[1], Bùi Anh Tuấn[1], Trần Thanh Tùng[1]
Email: duongthanglong@hou.edu.vn, ORCID: 0000-0003-0609-9534

Ngày tòa soạn nhận được bài báo: 02/12/2025. Ngày phản biện đánh giá: 02/06/2026.
Ngày bài báo được duyệt đăng: 23/06/2026
DOI: 10.59266/houjs.2026.1332

***Tóm tắt**: Hư hỏng mặt đường, đặc biệt là "ổ gà", là mối đe dọa nghiêm trọng đối với an toàn giao thông và gây thiệt hại kinh tế lớn tại nhiều quốc gia. Các phương pháp khảo sát truyền thống tốn kém, chủ quan và thiếu hiệu quả, trong khi các giải pháp tự động hiện có vẫn gặp khó khăn trong các điều kiện thực tế phức tạp như bóng râm, vũng nước, ánh sáng yếu và ổ gà kích thước nhỏ. Nghiên cứu này đề xuất cải tiến kiến trúc YOLOv12 kết hợp với module self-attention (SA) để nắm bắt mối quan hệ ngữ cảnh toàn diện và Enhanced Visual Center Block (EVCBlock) tập trung trích xuất đặc trưng trung tâm ổ gà và ức chế nhiễu ngoại vi. Mô hình được huấn luyện và đánh giá trên bộ dữ liệu 19,267 ảnh tổng hợp từ RDD2020, RDD2022 và Roboflow. Kết quả thực nghiệm cho thấy mô hình đề xuất đạt Precision = 84.23%, Recall = 74.97%, F1-score = 79.33%, mAP@0.5 = 82.36% - vượt trội so với YOLOv12 gốc (+6.28% mAP@0.5) và một số biến thể YOLO nano (YOLOv5n, YOLOv8n, YOLOv11n). So với kiến trúc ResNet50, Vision Transformer và Swin Transformer, mô hình đề xuất đạt độ chính xác tương đương trong khi giảm 66-68% chi phí tính toán và 67-69% tiêu thụ năng lượng, duy trì tốc độ 122 FPS trên GPU NVIDIA RTX 3050.*

***Từ khóa:** phát hiện ổ gà, YOLOv12, self-attention, EVCBlock, thị giác máy tính, hư hỏng mặt đường, học sâu*

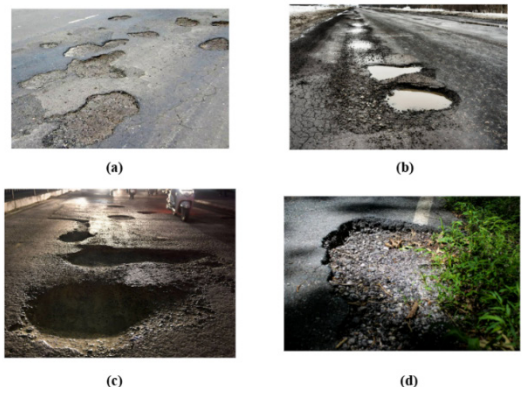
I. Đặt vấn đề

Sự xuống cấp của hạ tầng đang là thách thức nghiêm trọng đối với bất kỳ nước nào, khi nhiều tai nạn giao thông xuất phát từ việc hạ tầng không đáp ứng tiêu chuẩn. Trong số các loại hư hỏng mặt đường, "ổ gà" đang là mối đe dọa trực tiếp và nghiêm trọng đối với người tham gia giao thông (Cuthbert & cộng sự, 2024). Chỉ riêng tại Hoa Kỳ, theo báo cáo của Ellen Edmonds (2022) ổ gà khiến người lái xe thiệt hại khoảng 26.5 tỷ USD mỗi năm. Cũng theo hồ sơ an toàn đường bộ Việt Nam năm 2025, đất nước ghi nhận khoảng 42 ca tử vong/1.000 km, với tổng chi phí kinh tế ước tính lên tới 19 tỷ USD (tương đương 5% GDP). Vì vậy, chất lượng hạ tầng đường bộ, bao gồm tình trạng mặt đường và ổ gà được xác định là một trong những yếu tố then chốt ảnh hưởng đến an toàn giao thông (ATO, 2025).

[1] Trường Đại học Mở Hà Nội, Hà Nội, Việt Nam

Các khảo sát thủ công truyền thống phụ thuộc vào nhân viên thực địa quan sát và đo lường mức độ hư hỏng. Những phương pháp này không chỉ tốn kém, dễ sai sót mà còn mang tính chủ quan cao (Cuthbert & cộng sự, 2024). Các giải pháp tự động sử dụng máy quét laser 3D hoặc xe đo chuyên dụng tuy chính xác nhưng đòi hỏi vốn đầu tư lớn.

Những tiến bộ trong học sâu và thị giác máy tính với cách tiếp cận theo hướng chi phí thấp và thời gian thực. Họ mô hình YOLO (You Only Look Once) nổi lên như một trong những lựa chọn hàng đầu cho bài toán phát hiện đối tượng nhờ kiến trúc bộ phát hiện một giai đoạn, giúp cân bằng hiệu quả giữa tốc độ và độ chính xác (Redmon & cộng sự, 2016). Tuy nhiên, mô hình này vẫn tồn tại hạn chế trong môi trường phức tạp: hiệu suất giảm mạnh với ổ gà nhỏ hoặc hình dạng bất quy tắc và khả năng trích xuất đặc trưng kém trong nền phức tạp (Xuefang & cộng sự, 2025).



Hình 1. Ví dụ ổ gà trong điều kiện thực tế: (a) ổ gà khô ban ngày, (b) ổ gà ngập nước, (c) ổ gà ban đêm, (d) ổ gà có bóng râm và bị che khuất.

Trên cơ sở phân tích các nghiên cứu trước đây, đóng góp của nhóm nghiên cứu gồm: (1) đề xuất tích hợp self-attention toàn cục vào backbone và Enhanced Visual Center Block vào neck của kiến trúc YOLOv12 nhằm nâng cao hiệu xuất nhận diện ổ gà (2) tiền xử lý kết hợp cân bằng histogram thích nghi có giới hạn tương phản (Contrast Limited Adaptive Histogram Equalization - CLAHE) thích ứng giúp giảm khoảng cách giữa các bộ dữ liệu RDD2020, RDD2022 và Roboflow; (3) đánh giá thực nghiệm trên tập kiểm thử với so sánh giữa các biến thể của kiến trúc YOLO và một số kiến trúc khác.

II. Phương pháp nghiên cứu

2.1. Nghiên cứu có liên quan

Trong những năm gần đây, bài toán phát hiện ổ gà đã nhận được sự quan tâm trong cộng đồng nghiên cứu thị giác máy tính. Xu hướng phát triển dựa trên mô hình YOLO, đặc biệt khi kết hợp với các bộ dữ liệu như RDD2020 & RDD2022 được xây dựng bởi Arya và cộng sự (2022; 2021) đều tập trung vào việc cải tiến kiến trúc để nâng cao độ chính xác, tốc độ thời gian thực và khả năng khái quát hóa trong điều kiện phức tạp.

Cuthbert và cộng sự (2024) đề xuất sử dụng mô hình YOLOv5 kết hợp với hệ thống camera gắn trên phương tiện và ước lượng kích thước ổ gà. Mô hình được huấn luyện trên 1,876 ảnh đạt precision 93.0%, recall 91.6% và mAP@0.5 đạt 96.3%.

Yue và cộng sự (2024) phát triển mô hình RDD-YOLO dựa trên nền tảng YOLOv8 nhằm phát hiện đa dạng các loại hư hỏng mặt đường. Mô hình được huấn luyện trên bộ dữ liệu RDD2022. Kết quả đạt mAP@0.5 là 62.5% và F1-score là 69.6%.

Jiayi và Han (2024) đề xuất mô hình YOLOv8-PD, một phiên bản tối ưu hóa của YOLOv8n dành cho thiết bị

tài nguyên hạn chế. Kết quả thực nghiệm cho thấy mô hình cải thiện đáng kể độ chính xác so với YOLOv8 gốc trong khi vẫn duy trì tốc độ xử lý thời gian thực.

Hao và cộng sự (2026) giới thiệu mô hình SDC-YOLOv8, trong đó tích hợp trích xuất đặc trưng đa tỉ lệ, tái cấu trúc đặc trưng thích ứng và cơ chế chú ý theo tọa độ. Mô hình đánh giá trên tập dữ liệu Pothole_665 và đạt mAP@0.5 = 78.0%, F1-score = 75.5% đồng thời đạt tốc độ suy luận 85 FPS.

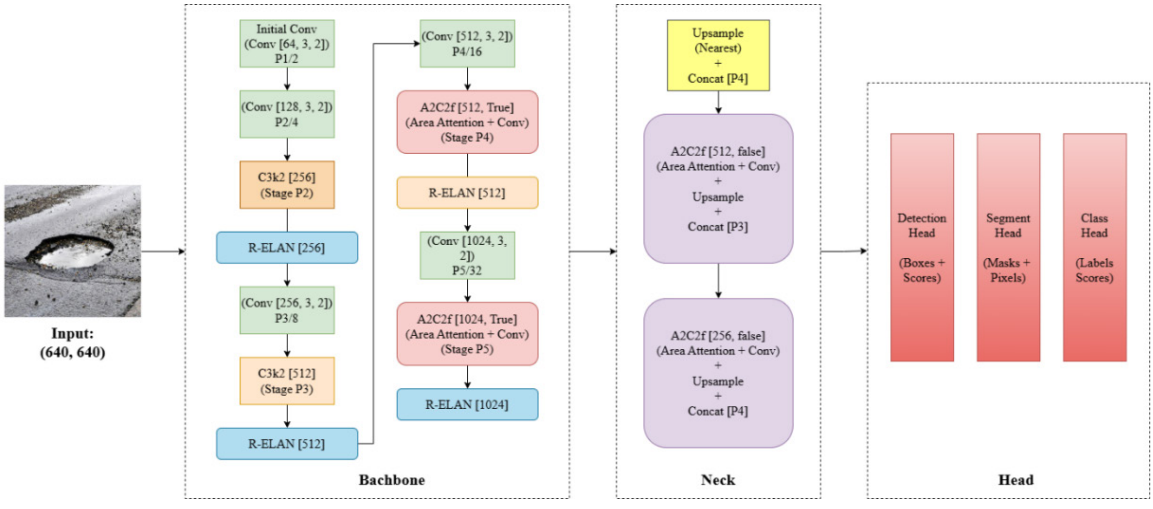
Bảng 1. Tổng hợp công trình nghiên cứu có liên quan

Tác giả	Mô hình	Dataset	Kiến trúc chính	F1-score(%)	mAP@0.5(%)
Cuthbert và cộng sự (2024)	YOLOv5	Tự thu thập (1,876 ảnh)	CSPNet + PANet + LKA	87.0	96.3
Yue và cộng sự (2024)	RDD-YOLO	RDD2022	YOLOv8x + cơ chế cổng chú ý kết hợp đa tỷ lệ	69.6	62.5
Jiayi và Han (2024)	YOLOv8-PD	RoadDamage RDD2022	YOLOv8n + attention nhẹ + PAN-FPN	63.0 68.0	64.2 70.6
Hao và cộng sự (2026)	SDC-YOLOv8	Pothole_665	YOLOv8 + SPPF-LSKA + cơ chế chú ý theo tọa độ	75.5	78.0

Từ các nghiên cứu trên, có thể nhận thấy mô hình YOLO giữ vai trò trung tâm trong các hệ thống phát hiện ổ gà nhờ khả năng xử lý thời gian thực và kiến trúc linh hoạt. Nhưng cũng đặt ra hạn chế hiệu năng suy giảm trên các tập dữ liệu phức tạp, các yếu tố như bóng râm, vũng nước, kích thước ổ gà nhỏ vẫn đang là thách thức lớn với mô hình.

2.2. Kiến trúc YOLOv12

YOLOv12 là một kiến trúc được tùy biến bởi nhóm nghiên cứu của Yunjie Tian và cộng sự (2025). YOLOv12 được thiết kế theo hướng lai, kết hợp giữa lớp tích chập và cơ chế attention, từ đó cải thiện hiệu năng phát hiện đối tượng trong kịch bản phức tạp.



Hình 2. Kiến trúc YOLOv12

Kiến trúc tổng thể của YOLOv12 tương tự các mô hình YOLO hiện đại. Tuy nhiên, điểm khác biệt cốt lõi nằm ở việc tích hợp các khối attention vào bước xử lý đặc trưng. YOLOv12 áp dụng cơ chế area attention (AA), trong đó bản đồ đặc trưng được chia thành các vùng thay vì từng pixel. Khi đó, attention được tính trên các vùng theo hàm softmax:

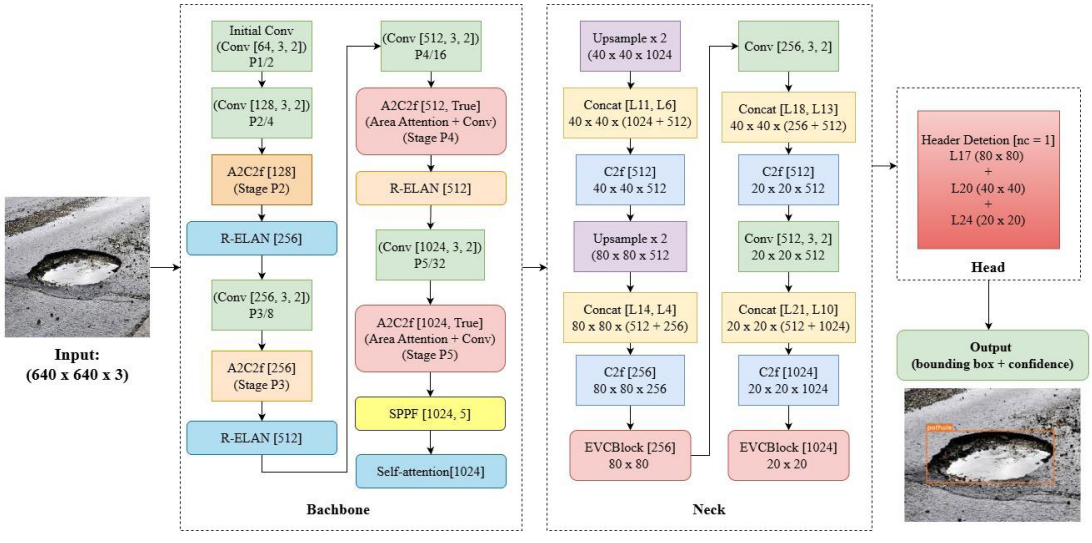
$$Attention_{area} = Softmax\left(\frac{Q_r K_r^T}{\sqrt{d}}\right)V_r\# \quad (1)$$

với Q_r, K_r, V_r là biểu diễn đặc trưng theo vùng. Cách tiếp cận này giúp giảm đáng kể chi phí tính toán, đồng thời vẫn giữ được thông tin ngữ cảnh quan trọng.

Trong nghiên cứu này nhóm sử dụng phiên bản YOLOv12 gốc commit 28a618b từ kho mã nguồn chính thức tại GitHub sunsmarterjie/yolov12 làm mô hình cơ sở. Mô hình được huấn luyện từ đầu, không sử dụng trọng số tiền huấn luyện, với cùng cấu hình tham số và tập dữ liệu như mô hình đề xuất nhằm đảm bảo tính công bằng trong so sánh.

2.3. Mô hình đề xuất

Nhóm nghiên cứu đề xuất mô hình YOLOv12-SA-EVC, được xây dựng trên nền tảng YOLOv12 với hai cải tiến cốt lõi: (i) tích hợp module SA vào cuối backbone nhằm nắm bắt quan hệ toàn cục, (ii) bổ sung EVC vào neck nhằm tập trung trích xuất đặc trưng trung tâm của ổ gà và loại bỏ nhiễu ngoại vi.



Hình 3. Kiến trúc mô hình đề xuất YOLOv12-SA-EVC

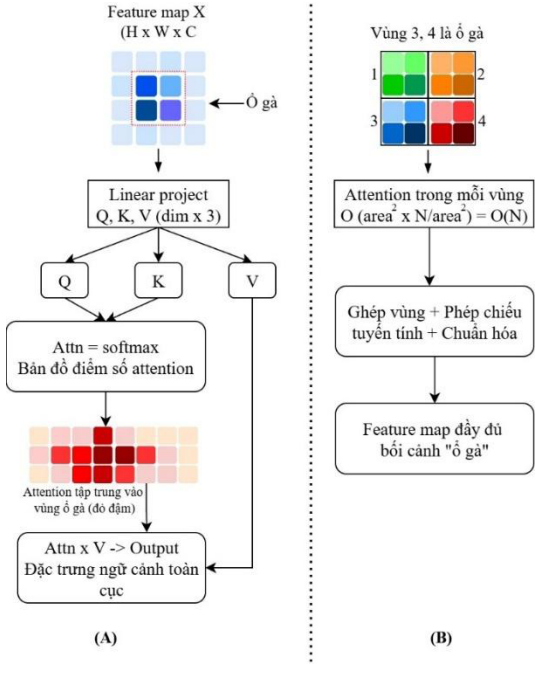
2.3.1. Module self-attention toàn cục

Mặc dù YOLOv12 đã có sẵn cơ chế AA, cơ chế này tính attention từng vùng cục bộ nên khả năng nắm bắt mối quan hệ giữa ổ gà và ngữ cảnh còn hạn chế. Nhằm giải quyết vấn đề đó module SA được đặt ngay sau khối R-ELAN [512], với biểu đồ đặc trưng có kích thước 20×20×1024 giúp đầu ra của backbone giàu thông tin ngữ cảnh hơn.

SA thực hiện attention toàn cục cho phép mọi vị trí (i, j) trong P5 chú ý đến tất cả các vị trí (k, l) còn lại. Cụ thể, đầu vào biểu đồ đặc trưng $X \in R^{H \times W \times C}$ được chiếu thành ba ma trận Q, K, V thông qua các phép biến đổi tuyến tính (Vaswani & cộng sự, 2017):

$$Attention\ (Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V\# \quad (2)$$

Mỗi vị trí (i, j) sinh ra một query vector $q_{i,j}$ từ ma trận Q; vector này so khớp với toàn bộ tập key $k_{(k,l)}$ qua tích vô hướng sau đó chuẩn hóa bằng hàm softmax để tạo phân phối chú ý. Cơ chế này quan trọng cho bài toán vì: (i) ổ gà thường có quan hệ phụ thuộc với các yếu tố nền xung quanh; (ii) ổ gà bị che khuất (do nước hoặc bóng) có thể được nhận diện khi mô hình tổng hợp manh mối từ các vùng lân cận.



Hình 4. Cơ chế self-attention (A) và area attention (B)

AA có độ phức tạp $O(N^2/A^2)$ với A là số vùng, còn SA thuần là $O(N^2)$. Vì SA chỉ được đặt tại P5 chi phí thực tế vẫn ở mức quản lý được. Để đảm bảo tốc độ, module áp dụng FlashAttention (Dao & cộng sự, 2022) tối ưu hóa bộ nhớ của GPU, đồng thời sử dụng Conv 7×7 thay cho mã hóa vị trí truyền thống giúp mô hình nắm thông tin vị trí không gian mà không làm chậm suy luận.

2.3.2. Module EVCBlock

Ổ gà có hình thái đặc trưng với vùng trung tâm lõm tương phản cao so với bề mặt đường, trong khi vùng biên thường bị nhiễu bởi bóng râm, vết nứt hoặc nước đọng. Tuy nhiên, các khối C2f trong YOLOv12 gốc xử lý đặc trưng đồng đều trên toàn feature map, không phân biệt mức độ quan trọng giữa vùng trung tâm và ngoại vi dẫn đến nguy cơ nhiễu nền lấn át tín hiệu đặc trưng đối tượng. Trong nghiên cứu này, nhóm nghiên cứu giải quyết vấn đề bằng cách tùy biến EVCBlock từ ý tưởng của Quan và cộng sự (2023).

EVC bao gồm hai nhánh xử lý song song:

Nhánh Visual Center Attention: Global Avg Pooling nén biểu đồ đặc trưng thành vector kênh 1×1×C, sau đó qua hai lớp kết nối với tỷ lệ nén r và hàm kích hoạt ReLU, tạo ra trọng số $\alpha \in (0,1)$ qua hàm Sigmoid. Sau đó, một Spatial Center Mask được xây dựng dựa trên phân phối Gaussian 2D tập trung vào vùng trung tâm của biểu đồ đặc trưng, mô phỏng xu hướng đặc trưng trung tâm đối tượng nằm ở vùng giữa bounding box (bbox).

$$M_s(i,j) = exp[-((i - H/2)^2 / (2\sigma H^2) + (j - W/2)^2 / (2\sigma W^2))] \quad (3)$$

Công thức (3) dựa trên ý tưởng Spatial Center Mask của Quan và cộng sự (2023) và tùy biến cho phân bố ổ gà tập trung ở vùng giữa - dưới ảnh. Trong đó:

i, j là tọa độ pixel trên biểu đồ đặc trưng;

H, W là chiều cao, chiều rộng của biểu đồ đặc trưng tại P3: 80×80, P4: 40×40, P5: 20×20;

σ_H, σ_W là độ lệch chuẩn theo trục dọc và ngang. Trong thực nghiệm chọn $\sigma_H = \frac{H}{4}, \sigma_W = \frac{W}{4}$ để phủ ≈95% vùng trung tâm.

Sau đó kết quả được kết hợp với trọng số kênh α qua tích Hadamard để tạo bản đồ chú ý tổng hợp $M = \alpha \times M_s$, khi đó M được nhân từng phần tử với đặc trưng đầu ra của nhánh Local Feature Extraction theo công thức (4).

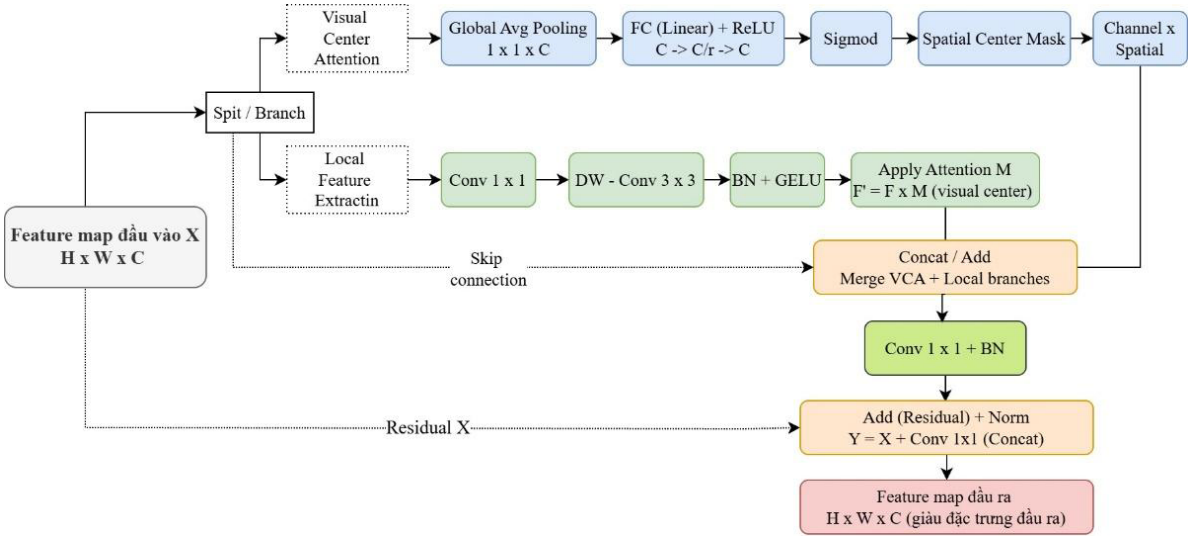
Nhánh Local Feature Extraction: Áp dụng chuỗi Conv 1×1 → DW-Conv 3×3 → BN + GELU để trích xuất đặc trưng cục bộ. Bản đồ chú ý M được nhân từng phần tử vào đặc trưng đầu ra:

$$F' = F_{local} \odot M \# \quad (4)$$

Đầu ra của hai nhánh được Concat, chiếu lại kích thước gốc qua Conv 1×1 + BN, rồi cộng với skip connection từ đầu vào gốc X.

$$Y = X + Conv_{1\times1}(Concat(F_{VCA}, F')) \# \quad (5)$$

Kết nối residual đảm bảo gradient lưu thông tốt trong huấn luyện và không làm mất thông tin đặc trưng ban đầu.



Hình 5. Kiến trúc chi tiết EVCBlock

2.4. Bộ dữ liệu và tiền xử lý dữ liệu

Bộ dữ liệu được thu thập từ nền tảng Roboflow và lọc chọn từ hai tập RDD2020 & RDD2022 với nhãn Pothole (D40). Sau khi loại bỏ các ảnh trùng lặp và không đạt, bộ dữ liệu cuối cùng gồm 19,267 ảnh được chia theo Bảng 2.

Bảng 2. Phân chia bộ dữ liệu

Tập dữ liệu	Số ảnh	Số bbox	Tỷ lệ (%)
Huấn luyện	16,118	30,443	83.7
Kiểm tra	2,094	5,212	10.9
Kiểm thử	1,055	2,649	5.4
Tổng cộng	19,267	38,304	100

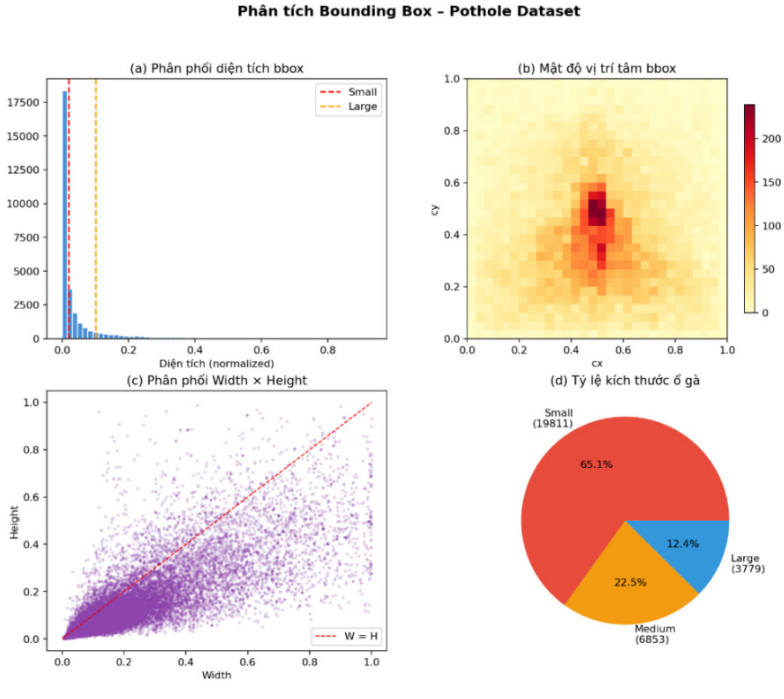
Tập dữ liệu bao quát các điều kiện thực tế phức tạp: ánh sáng ban ngày/đêm, bóng râm, vũng nước, bề mặt ướt và môi trường đô thị giúp mô hình có khả năng khái quát hóa tốt khi triển khai thực tế.

Quy trình lọc dữ liệu gồm: (1) Chuẩn hóa nhãn - nhãn D40 từ RDD2020/ RDD2022 và Roboflow được ánh xạ thống nhất về một lớp "pothole". (2) Loại bỏ ảnh trùng lặp - sử dụng thuật toán Perceptual Hash để tính toán vector đặc trưng thị giác của từng ảnh và so sánh độ tương đồng giữa các cặp ảnh; các ảnh có độ tương đồng ≥ 90% được loại bỏ, chỉ giữ lại một ảnh đại diện. (3) Loại bỏ các ảnh bị mờ, thiếu nhãn hoặc bbox không hợp lệ. Tiếp theo, tất cả ảnh được chuyển về kích thước 640×640 pixel bằng letterbox resize. Đồng thời, kỹ thuật CLAHE (Zuiderveld, 1994) được

áp dụng trên cả ba tập, thực hiện riêng trên kênh L của không gian màu LAB để tăng độ tương phản cục bộ mà không gây lệch màu. Việc áp dụng cho cả 3 tập nhằm loại bỏ khoảng cách miền dữ liệu, đảm bảo tính nhất quán khi suy luận.

Hình 6. So sánh ảnh gốc và sau CLAHE với clip_limit=[2,4]

Phân tích phân phối bbox trên tập huấn luyện cho thấy 65.1% là đối tượng kích thước nhỏ (diện tích bbox < 32x32 px), phân bố tập trung ở vùng giữa, dưới ảnh và tỉ lệ chiều rộng/chiều cao đa dạng từ 0.5 - 2.5. Đặc trưng này cung cấp cơ sở khoa học quan trọng để ưu tiên thiết kế SA và EVC tập trung vào đặc trưng đa tỷ lệ.



Hình 7. Kết quả phân tích tập dữ liệu

2.5. Phương pháp đánh giá

Để đánh giá toàn diện hiệu năng của mô hình, nhóm nghiên cứu sử dụng hai nhóm chỉ số: chính xác phát hiện và chi phí tính toán - năng lượng (Padilla & cộng sự, 2020).

1) Chỉ số chính xác phát hiện

Sử dụng bộ chỉ số tiêu chuẩn cho bài toán phát hiện gồm Precision (P), Recall (R), F1-score (F1) và Average Precision (AP).

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \# \quad (6,7)$$

$$F1 - score = \frac{2 \times P \times R}{P + R} \# \quad (8)$$

$$mAP@0.5 = AP_{IoU=0.5} \# \quad (9)$$

$$mAP@0.5:0.95 = \frac{1}{10} \sum_{t=0.5}^{0.95} AP_{IoU=t} \# \quad (10)$$

Với TP, FP, FN lần lượt là số bbox dự đoán trùng khớp với nhãn thực tế ($IoU \geq 0.5$), số dự đoán dương sai và số ổ gà thực tế bị bỏ sót. AP được tính bằng diện tích dưới đường cong Precision-Recall.

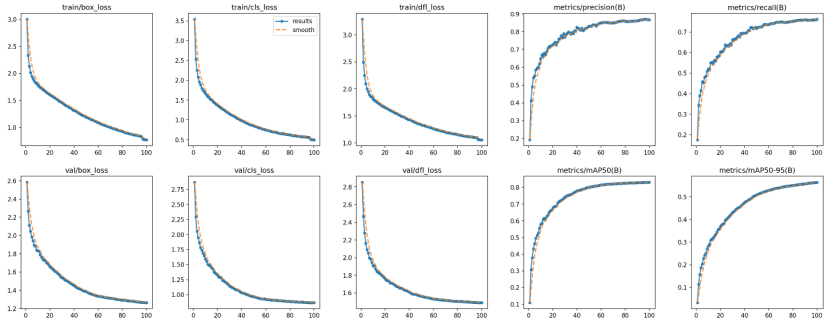
2) Chỉ số chi phí tính toán và năng lượng

Giga Floating-Point Operations (GFLOPs) đo số phép tính dấu phẩy động cần cho một forward pass với đầu vào 640×640, đo bằng thư viện THOP ở precision FP32. Frames Per Second (FPS) là tốc độ suy luận trên GPU NVIDIA RTX 3050, đo qua 100 lần warm-up và 1.000 lần suy luận, FPS = 1.000 / avg_time_ms. Công suất tiêu thụ trung bình (W) được đo qua nvidia-smi trong quá trình suy luận, lấy trung bình các mẫu cách nhau một giây.

III. Kết quả và đánh giá

3.1. Thiết lập thực nghiệm

Toàn bộ quá trình huấn luyện và đánh giá mô hình được triển khai trên cấu hình Intel Core i7-12700 và NVIDIA GeForce RTX 3050 (6 GB VRAM). Quá trình tối ưu sử dụng bộ tối ưu AdamW kết hợp lịch học Cosine, với tốc độ học ban đầu $lr_0 = 0.001$ và tối thiểu $lr_{min} = 1 \times 10^{-5}$, trong đó warmup được áp dụng trong 3 epoch đầu để ổn định quá trình hội tụ. Ngoài ra, weight decay = 5×10^{-4}, batch size 16, số epoch huấn luyện là 100 và toàn bộ mô hình được huấn luyện từ đầu, không sử dụng trọng số tiền huấn luyện. Tăng cường dữ liệu gồm mosaic (0.70), mixup (0.10), flip ngang (0.5), jitter màu (hsv: 0.015/0.75/0.45) và biến đổi hình học nhẹ (degrees = 3°, scale = 0.65); các tham số bổ sung gồm close_mosaic = 4, multi_scale = False, translate = 0,02, shear = 0,0, perspective = 0,0, box = 7,5, cls = 0,5, dfl = 1,5, val = True, plots = False, save_period = -1 và patience = 10, nhằm cân bằng giữa hiệu quả học, độ ổn định và khả năng tổng quát hóa của mô hình.



Hình 8. Đường cong loss và mAP trên tập huấn luyện/kiểm thử qua 100 epoch với batch size 16

Mô hình cho thấy xu hướng hội tụ ổn định, không có dấu hiệu overfitting: loss train và validation giảm đồng đều, trong khi mAP@0.5 trên tập kiểm tra liên tục cải thiện và ổn định từ khoảng epoch 70 trở đi.

3.2. So sánh các biến thể của YOLOv12

Kết quả đánh giá trên tập kiểm thử cho thấy mô hình YOLOv12-SA-EVC đạt Precision = 84.23%, Recall = 74.97%, F1 = 79.33%, mAP@0.5 = 82.36% và mAP@0.5:0.95 = 60.32%. Kết quả tổng hợp tại Bảng 3.

Bảng 3. So sánh hiệu suất biến thể YOLOv12 trên tập kiểm thử

Mô hình	P (%)	R (%)	F1 (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
YOLOv12	78.21	67.65	72.55	76.08	55.12
YOLOv12-SA	81.06	72.82	77.61	80.51	58.07
YOLOv12-EVC	80.82	69.84	73.94	79.82	56.81
YOLOv12-SA-EVC	84.23	74.97	79.33	82.36	60.32

So với YOLOv12 gốc (76,08% mAP@0.5), việc bổ sung riêng module SA giúp tăng 4.43%, trong khi bổ sung riêng EVCBlock giúp tăng 3.74%. Khi kết hợp đồng thời cả hai module, YOLOv12-SA-EVC đạt 82.36% mAP@0.5, tương ứng tăng 6.28% so với mô hình gốc. Xét hiệu ứng bổ sung, việc thêm EVCBlock vào YOLOv12-SA tiếp tục cải thiện 1.85%, trong khi thêm SA vào YOLOv12-EVC mang lại mức tăng 2.54%, cho thấy hai module có tính bổ trợ lẫn nhau thay vì đóng góp theo cách cộng tuyến tính. Từ kết quả nhận diện thực tế, khả năng nhận diện trong điều kiện phức tạp của YOLOv12-SA-EVC vượt trội rõ rệt so với mô hình gốc. Mô hình đề xuất phát hiện tốt hơn các trường hợp ổ gà ngập nước, bị bóng râm che khuất và ổ gà kích thước nhỏ.

Hình 9. So sánh nhận diện thực tế giữa YOLOv12-SA-EVC và YOLOv12 gốc

3.3. Đánh giá kỹ thuật tăng cường

Để định lượng đóng góp của kỹ thuật tăng cường tương phản CLAHE, nhóm thực hiện 3 cấu hình clip_limit trên kiến trúc đề xuất. Kết quả cho thấy với clip_limit = 2 là lựa chọn tối ưu. Khi clip_limit quá cao, việc khuếch đại nhiễu cục bộ ở các vùng phức tạp (bề mặt ướt, bóng râm) gây khó khăn trong việc phán đoán và làm giảm hiệu suất.

Bảng 4. So sánh kỹ thuật tăng cường CLAHE với clip_limit khác nhau

Cấu hình tiền xử lý	P (%)	R (%)	F1(%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
Không CLAHE	82.41	71.83	76.76	78.90	58.21
CLAHE clip_limit = 2	84.23	74.97	79.33	82.36	60.32
CLAHE clip_limit = 4	83.65	73.54	78.37	80.71	52.93

3.4. So sánh với một số kiến trúc nhận diện khác

Để đánh giá tính cạnh tranh của mô hình đề xuất nhóm tiến hành so sánh với ba kiến trúc đại diện: ResNet50 Faster R-CNN, Vision Transformer ViT-B/16, và Swin Transformer, tất cả được triển khai trong cùng điều kiện dữ liệu, cùng kích thước ảnh đầu vào, cùng số epoch và cùng tiêu chí đánh giá. Tất cả detection head đều được khởi tạo và huấn luyện trên tập dữ liệu ổ gà, trong khi các tham số huấn luyện khác như optimizer, learning rate, batch size, warmup, augmentation và số epoch được giữ nguyên với mô hình đề xuất để đảm bảo tính khách quan của phép so sánh.

Bảng 5. So sánh với một số kiến trúc nhận diện khác

Mô hình	P (%)	R (%)	F1 (%)	mAP@0.5 (%)	GFLOPs	Công suất (W)	FPS
ResNet50 Faster R-CNN	78.53	68.45	73.14	76.81	180.4	145	130.3
ViT-B/16	85.47	78.14	81.64	83.36	337.6	245	89.1
Swin Transformer	88.04	80.47	84.09	84.17	354.2	265	84.8
YOLOv12-SA-EVC	84.23	74.97	79.33	82.36	112.8	82	122.4

Có thể thấy Swin Transformer đạt mAP@0.5 cao nhất (84.17%) nhưng đòi hỏi tới 354.2 GFLOPs và công suất 265W - cao gấp khoảng 3× so với YOLOv12-SA-EVC. Tương tự. ViT-B/16 đạt mAP@0.5 = 83.36% nhưng tiêu tốn 337.6 GFLOPs và 245W. Trong khi đó. YOLOv12-SA-EVC chỉ với 112.8 GFLOPs và 82W vẫn đạt mAP@0.5 = 82.36%. chỉ kém Swin 1.81% và vượt qua ResNet50 Faster R-CNN cả về độ chính xác lẫn chi phí (giảm 37.5% GFLOPs. giảm 43.4% công suất). Mô hình đề xuất đạt sự cân bằng tốt giữa độ chính xác và chi phí tính toán đặc biệt phù hợp với các nền tảng nhúng tài nguyên hạn chế.

3.5. So sánh với một số biến thể YOLO

Để đánh giá vị trí của YOLOv12-SA-EVC trong họ YOLO, nhóm nghiên cứu thực hiện so sánh với các phiên bản YOLOv5n. YOLOv8n. YOLOv11n. Kết quả tổng hợp tại Bảng 6.

Bảng 6. So sánh một số biến thể YOLO

Mô hình	P (%)	R (%)	F1 (%)	mAP@0.5 (%)	GFLOPs	Công suất TB (W)	FPS
YOLOv5n	76.80	67.90	72.08	74.50	66.50	40	106.32
YOLOv8n	79.70	70.60	74.87	77.20	78.60	45	113.61
YOLOv11n	80.10	73.40	76.9	78.41	80.50	43	114.23
YOLOv12	78.21	67.65	72.55	76.08	55.12	62	118.02
YOLOv12-SA-EVC	84.23	74.97	79.33	82.36	112.8	82	122.43

Kết quả cho thấy vượt trội hơn một số biến thể nano về độ chính xác: cải thiện +7.86% mAP@0.5 so với YOLOv5n. +5.16% so với YOLOv8n và +4.95% so với YOLOv11n. Đánh đổi là chi phí tính toán cao hơn do tích hợp hai module. Tuy nhiên, mức công suất 82W vẫn triển khai được trên các thiết bị nhúng tầm trung như NVIDIA Jetson Xavier NX. Tùy thuộc vào yêu cầu cần chính xác cao hay tiết kiệm tài nguyên người dùng có thể lựa chọn giữa mô hình phù hợp.

IV. Kết luận

Nghiên cứu này đề xuất mô hình YOLOv12-SA-EVC - phát hiện ổ gà tự động theo thời gian thực với kết quả thực nghiệm trên bộ dữ liệu 19.267 ảnh tổng hợp từ RDD2020, RDD2022 và Roboflow cho thấy mô hình đề xuất đạt mAP@0.5 = 82.36%, cải thiện 6.28% so với YOLOv12 gốc và cao hơn so với một số biến thể YOLO Nano (YOLOv5n, YOLOv8n, YOLOv11n). So với kiến trúc ResNet50, ViT-B/16 và Swin Transformer, mô hình đề xuất đạt độ chính xác tương đương trong khi giảm 66-68% chi phí tính toán và 67-69% mức tiêu thụ năng lượng, đồng thời duy trì tốc độ suy luận 122 FPS trên GPU NVIDIA RTX 3050, mô hình có thể triển khai trực tiếp trên camera hành trình, thiết bị giám sát đường bộ hoặc xe tuần tra, cung cấp cảnh báo tức thời và dữ liệu định vị ổ gà để lập kế hoạch bảo trì chủ động - từ đó góp phần giảm thiểu nguy cơ tai nạn giao thông do hư hỏng mặt đường gây ra. Hướng phát triển tiếp theo sẽ tập trung vào: (i) thu thập thêm dữ liệu địa phương để nâng cao khả năng khái quát hóa cho điều kiện đường bộ Việt Nam; (ii) tối ưu hóa mô hình cho triển khai trên thiết bị nhúng.

Tài liệu tham khảo

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., Seto, T., & Sekimoto, Y. (2022). Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022). *2022 IEEE International Conference on Big Data (Big Data)*, 6378--6386.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., Seto, T., Mraz, A., & Sekimoto, Y. (2021). RDD2020: An Image Dataset for Smartphone-based Road Damage Detection and Classification. *Mendeley Data,*(10.17632/5ty2wb6gvg.1).

ATO.(2025).*VietNamRoadSafetyProfile2025*. https://asiantransportobservatory.org/analytical-outputs/roadsafetyprofiles/viet-nam-road-safety-profile-2025/

Cuthbert, R., Judith, M., Gurcan, C., Saidi, S., Frank, N., & Quincy, A. (2024). Augmenting roadway safety with machine learning and deep learning: Pothole detection and dimension estimation using in-vehicle technologies. *Machine Learning with Applications*, *16*, 100547. https://doi.org/10.1016/j.mlwa.2024.100547

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *arXiv*.

Edmonds, E. (2022). *AAA: Potholes Pack a Punch as Drivers Pay $26.5 Billion in Related Vehicle Repairs*. https://newsroom.aaa.com/2022/03/aaa-potholes-pack-a-punch-as-drivers-pay-26-5-billion-in-related-vehicle-repairs/

Hao, Y., Yulong, S., Yue, L., Enhao, T., & Danyang, C. (2026). SDC-YOLOv8: An Improved Algorithm for Road Defect Detection Through Attention-Enhanced Feature Learning and Adaptive Feature Reconstruction. *Sensors*, *26*(2), 609.

Jiayi, Z., & Han, Z. (2024). YOLOv8-PD: an improved road damage detection algorithm based on YOLOv8n model. *Scientific Reports*, *14*(1), 12052. https://doi.org/10.1038/s41598-024-62933-z

Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237-242.

Quan, Y., Zhang, D., Zhang, L., & Tang, J. (2023). Centralized Feature Pyramid for Object Detection. *IEEE Transactions on Image Processing*, *32*, 4341-4354. https://doi.org/10.1109/tip.2023.3297408

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, 27-30 June 2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). *Attention is All you Need* https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Xuefang, L., Tianyu, C., Caixia, S., Chaoran, Y., & Tao, P. (2025). Application of YOLO Algorithm for Intelligent Transportation Systems: A Survey and New Perspectives. *International Journal of Distributed Sensor Networks*, *2025*(1), 2859040. https://doi.org/10.1155/dsn/2859040

Yue, L., Chang, Y., Yutian, L., Jiale, Z., & Yiting, Y. (2024). RDD-YOLO: Road Damage Detection Algorithm Based on Improved You Only Look Once Version 8. *Applied Sciences*, *14*(8), 3360.

Zuiderveld, K. J. (1994). Contrast Limited Adaptive Histogram Equalization. In P. S. Heckbert (Ed.), *Graphics Gems IV* (pp. 474-485). Academic Press.

YOLOV12-SA-EVC: A ROAD POTHOLE DETECTION MODEL FOR ROAD SURFACE INSPECTION

Duong Thang Long[1], Vu Xuan Hanh[1], Bui Anh Tuan[1], Tran Thanh Tung[1]

Abstract: *Road surface deterioration, particularly potholes, poses a significant threat to traffic safety and causes substantial economic losses in many countries. Conventional road inspection methods are costly, subjective, and inefficient, whereas existing automated approaches still struggle under challenging real-world conditions, including shadows, water puddles, low-light environments, and small-sized potholes. This study proposes an enhanced YOLOv12 architecture by integrating the self-attention (SA) module to capture comprehensive contextual dependencies and the Enhanced Visual Center Block (EVCBlock) to emphasize discriminative features at the pothole center while suppressing peripheral noise. The proposed model was trained and evaluated on a combined dataset of 19,267 images collected from RDD2020, RDD2022, and Roboflow. Experimental results demonstrate that the proposed model achieves a Precision of 84.23%, Recall of 74.97%, F1-score of 79.33%, and mAP@0.5 of 82.36%, outperforming the baseline YOLOv12 by 6.28 percentage points in mAP@0.5 and surpassing several lightweight YOLO variants, including YOLOv5n, YOLOv8n, and YOLOv11n. Compared with ResNet50, Vision Transformer (ViT), and Swin Transformer, the proposed model achieves comparable detection accuracy while reducing computational cost by 66-68% and energy consumption by 67-69%, maintaining a real-time inference speed of 122 FPS on an NVIDIA RTX 3050 GPU.*

Keywords: *pothole detection, YOLOv12, self-attention, EVCBlock, computer vision, road surface damage, deep learning*

---

[1] Hanoi Open University, Hanoi, Vietnam