https://doi.org/10.31436/iiumej.v27i3.4693

Similar Accuracy, Different Explanations in YOLO-Based Palm Fruit Ripeness Detection

QUEENTA PARADISSA RAMADHANI[1], FAVIAN DEWANTA[1], TEDDY SURYA GUNAWAN[2*], MIRA KARTIWI[3], AND BAMBANG SETIA NUGROHO[1]

[1] *School of Electrical Engineering, Telkom University, Bandung, Indonesia.*

[2] *Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia*

[3] *Department of Information Systems, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia*

**Corresponding author: tsgunawan@iium.edu.my*

(Received: 31 July 2026; Accepted: 30 August 2026; Published online: 11 September 2026)

ABSTRACT: Modern palm-fruit detectors can achieve similarly high benchmark accuracy, yet class activation mapping (CAM) methods are often applied without establishing whether their explanatory behavior transfers across detector configurations. This study is a controlled exploratory within-dataset comparison: model scale and target layer were selected on the same 416-image test partition used for final CAM evaluation, so the reported values are not independent estimates of explanation generalization. We evaluated 15 YOLOv8, YOLO11, and YOLO26 variants under a matched training protocol and assessed representative medium models across three random seeds. Backbone representations were then audited on the 416 test images using class-agnostic EigenCAM, backbone-adapted Grad-CAM++ (bGrad-CAM++), and backbone-adapted Score-CAM (bScore-CAM). Explanation behavior was quantified with CAM-box intersection over union (IoU) and Pointing Game for spatial localization and with Mean Absolute Confidence Drop (MACD) and Increase in Confidence as image-level confidence-retention diagnostics. Repeated-run mean mAP50–95 was narrowly separated at 0.903 ± 0.002, 0.899 ± 0.002, and 0.901 ± 0.003 for YOLOv8m, YOLO11m, and YOLO26m, respectively. In contrast, Pointing Game varied by 68.03 percentage points and MACD by 48.62 percentage points across model–explainer configurations. EigenCAM showed the strongest observed profile on YOLOv8m, whereas bScore-CAM achieved the highest localization on YOLO11m; no explainer dominated across all evaluated backbones. The ranking reversal was also present within the difficult underripe, ripe, and overripe subset, and the recorded layer search showed that layer 8 was the strongest compatible backbone target across the layer 6–10 window in all three detectors. Similar detection accuracy therefore did not imply transferable explanation behavior. CAM explanations should be validated jointly with the explanatory target, layer, and detector rather than treated as interchangeable post-hoc visualizations.

ABSTRAK: Pengesan buah sawit moden boleh mencapai ketepatan penanda aras tinggi dan hampir setara, namun kaedah pemetaan pengaktifan kelas (CAM) sering diguna tanpa memastikan sama ada tingkah laku penjelasannya boleh dipindah merentas konfigurasi pengesan. Kajian ini merupakan perbandingan penerokaan terkawal dalam set data: skala model dan lapisan sasaran dipilih menggunakan pecahan ujian 416 imej sama seperti penilaian CAM akhir, di mana dapatan yang dilaporkan bukan anggaran bebas bagi generalisasi penjelasan. Sebanyak 15 varian YOLOv8, YOLO11 dan YOLO26 dinilai menggunakan protokol latihan sepadan, manakala model sederhana yang mewakili setiap keluarga dinilai menggunakan tiga benih rawak. Perwakilan tulang belakang kemudiannya diaudit pada 416 imej ujian menggunakan EigenCAM yang bebas kelas, Grad-CAM++

teradaptasi tulang belakang (bGrad-CAM++), dan Score-CAM teradaptasi tulang belakang (bScore-CAM). Tingkah laku penjelasan dinilai menggunakan CAM-box IoU dan Pengarahan Perburuan bagi menempat spatial, serta Purata Mutlak Keyakinan Menurun (MACD) dan Keyakinan Bertambah sebagai diagnostik pengekalan keyakinan pada aras imej. Purata mAP50–95 bagi larian berulang masing-masing adalah 0.903 ± 0.002, 0.899 ± 0.002 dan 0.901 ± 0.003 untuk YOLOv8m, YOLO11m dan YOLO26m. Sebaliknya, Pengarahan Pemburuan berbeza sebanyak 68.03 mata peratusan dan MACD sebanyak 48.62 mata merentas konfigurasi model–penerang. EigenCAM menunjukkan profil pemerhatian terkuat pada YOLOv8m, manakala bScore-CAM mencapai penyetempatan tertinggi pada YOLO11m; tiada penerang yang mendominasi semua tulang belakang yang dinilai. Pembalikan kedudukan ini turut hadir dalam subset sukar seperti kurang masak, masak dan terlebih masak, manakala carian lapisan yang direkod menunjukkan bahawa lapisan 8 ialah sasaran tulang belakang serasi terkuat dalam tetingkap lapisan 6–10 bagi ketiga-tiga pengesan. Dapatan ini menunjukkan bahawa ketepatan pengesanan yang hampir setara tidak semestinya menghasilkan tingkah laku penjelasan yang boleh dipindahkan. Oleh itu, penjelasan CAM perlu disahkan secara bersama dengan sasaran penjelasan, lapisan dan pengesan.

KEYWORDS: *Explainable Artificial Intelligence; Class Activation Mapping; Object Detection; YOLO; Oil Palm Fruit.*

1. INTRODUCTION

Oil palm fresh fruit bunch (FFB) ripeness is an operational quality determinant, not merely a visual classification problem. Harvesting bunches outside the appropriate maturity stage can reduce recoverable oil yield or compromise crude palm oil quality, while inconsistent grading can propagate economic losses from harvesting through mill processing. This problem is substantial. Malaysia's oil palm sector generated RM109.39 billion in export revenue in 2024, yet the national oil extraction rate declined from 19.86% in 2023 to 19.67% in 2024 despite a 5.8% increase in FFB yield [1]. This decline was partly attributed to lower-quality FFB processed following prolonged rainfall and flooding [2], illustrating that increased fruit yield does not necessarily translate into greater extraction efficiency. Reliable and auditable ripeness assessment therefore remains important for both operational performance and economic value.

Computer vision provides a scalable approach to improving grading consistency, and one-stage YOLO detectors are particularly attractive because they integrate localization, classification, and real-time inference. Previous work identified YOLOv8m as offering a favorable accuracy–computational cost trade-off for six-class FFB detection [3], while subsequent studies have continued to compare detector generations primarily using precision, recall, inference speed, and mean average precision [4]. These metrics quantify predictive performance but provide limited evidence about the visual representations underlying a model's decisions. Two detectors can achieve nearly identical benchmark accuracy while relying on markedly different spatial features, and a visually plausible heatmap does not necessarily indicate that an explanation is informative or transferable across models.

Class activation mapping (CAM) methods, including EigenCAM, Grad-CAM++, and Score-CAM, are widely used to visualize salient image regions. However, these methods were developed largely for classifier-style networks with tensor outputs and scalar class scores. Modern object detectors present a more complex explanatory setting: they produce multiple bounding boxes, class labels, and confidence scores; their necks combine skip-connected feature maps through multi-path operations; and recent architectures increasingly modify the detection and post-processing pathway. Consequently, applying a classifier-oriented CAM

https://doi.org/10.31436/iiumej.v27i3.4693

method to an object detector requires explicit choices about the explanatory target and target layer. Without these choices, a plausible saliency map may be incorrectly interpreted as attribution of a specific final box–class prediction.

The resulting research gap is therefore not another comparison of YOLO generations, but whether explanation behavior transfers across detector configurations with closely matched predictive performance. Differences in detector configuration can coincide with differences in learned representations, while CAM methods themselves interrogate those representations through distinct activation-, gradient-, or perturbation-based mechanisms. Whether their relative behavior remains consistent across closely performing YOLO detectors has received limited quantitative examination in FFB ripeness detection.

Accordingly, this study asks whether YOLO detectors with practically similar ripeness-detection accuracy exhibit similarly consistent backbone-level CAM explanations across detector–explainer configurations. The study makes three contributions. First, it provides a controlled exploratory comparison showing that explainer rankings can change across closely performing YOLOv8m, YOLO11m, and YOLO26m configurations. Second, it establishes a matched benchmark of 15 YOLO variants, supported by three-seed evaluations of the representative medium models, to place explanation differences alongside small variations in predictive accuracy. Third, it presents a backbone-level compatibility procedure for EigenCAM, backbone-adapted Grad-CAM++ (bGrad-CAM++), and backbone-adapted Score-CAM (bScore-CAM) that preserves trained weights while explicitly distinguishing feature-representation auditing from attribution of a final box–class prediction.

Two scope conditions apply throughout. First, all explanation results are strictly within-dataset because model-scale and target-layer selection reused the 416-image test partition used for final CAM evaluation. The reported metrics therefore describe relative behavior on this dataset rather than out-of-sample explanation generalization. Second, the explanatory target is the pre-neck backbone representation rather than a final box–class decision. The maps audit spatial selectivity in the selected representation and do not provide causal attributions for individual detections.

2. Related Work

Research on automated fresh fruit bunch (FFB) grading has advanced rapidly in predictive performance, particularly through recent YOLO-based detectors. However, the methodological basis for interpreting and auditing those detections remains less mature. This gap matters because high benchmark accuracy does not ensure that the underlying visual evidence is spatially meaningful, stable across detector configurations, or suitable for post-hoc explanation.

2.1. FFB Ripeness Detection and YOLO Benchmarking

FFB ripeness assessment has progressed from manual color inspection and specialized optical sensing toward RGB-based deep learning. Camera-based systems are non-destructive, comparatively inexpensive, and readily integrated into harvesting and mill workflows [5, 6]. Nevertheless, reliable ripeness discrimination remains difficult because biologically adjacent stages, particularly underripe, ripe, and overripe, exhibit gradual rather than discrete changes in color, texture, and fruitlet detachment. Field imagery further introduces substantial variability in illumination, viewing angle, distance, occlusion, and plantation background [7]. Consequently, near-ceiling mAP50 on curated datasets does not imply that ripeness boundaries are fully resolved or that comparable performance will generalize to field conditions.

YOLO-based approaches have established strong FFB detection baselines. Comparisons across YOLOv8 scales have shown favorable accuracy–cost trade-offs for medium-sized models [3], while cross-generation studies have evaluated YOLOv8, YOLOv9, and optimized YOLOv7 variants primarily in terms of precision, recall, mean average precision, and inference speed [4, 8]. YOLOv8 has also achieved near-saturated performance on curated ripeness datasets [9]. However, reported results are difficult to compare directly because studies vary in class definitions, dataset composition, augmentation, training configuration, hardware, and model-selection criteria. More broadly, controlled agricultural benchmarks indicate that the relative ranking of recent YOLO generations is sensitive to both dataset and evaluation protocol [10]. Cross-generation comparisons are therefore most informative when data, training, model selection, and evaluation conditions are held constant.

2.2. Evolving Detector Architectures

Successive YOLO generations differ not only in computational cost but also in the internal representations exposed to explanation methods. YOLOv8 uses an anchor-free design with C2f-based feature extraction and decoupled multiscale prediction [11]. YOLO11 replaces key C2f stages with C3k2 blocks and incorporates C2PSA attention while retaining the conventional backbone–neck–head organization [10]. YOLO26 extends the design toward deployment-oriented end-to-end detection through an NMS-free inference pathway and lighter regression components, reflecting a broader movement toward reducing dependence on post-processing [12].

These changes do not guarantee monotonic gains in predictive accuracy. Controlled architectural analyses instead indicate shifting accuracy–latency trade-offs across model generations [13]. This distinction matters for explainability because CAM methods operate on intermediate activations, gradients, or perturbation responses rather than mAP itself. Two detectors can therefore produce similar final predictions while exposing materially different internal representations to the same explanatory procedure. The present study evaluates that empirical possibility without attributing the observed differences to any single architectural block.

2.3. CAM Methods and the Detector Barrier

CAM methods interrogate learned representations through different mechanisms. Grad-CAM and Grad-CAM++ derive saliency from gradient information, with Grad-CAM++ using higher-order weighting to improve localization in complex scenes [14, 15]. EigenCAM is class-agnostic and derives saliency from the dominant principal component of the activation tensor [16], whereas Score-CAM estimates channel importance through masked forward responses rather than gradients [17]. Because these methods probe different properties of the learned representation, their relative performance may not transfer unchanged across detector configurations.

A further difficulty is that conventional CAM methods were developed primarily for classifier-style networks with tensor outputs and scalar class scores. Object detectors instead produce multiple boxes, classes, and confidence values, making the explanatory target inherently more complex. Detector-specific methods such as D-RISE address this issue by combining category confidence with bounding-box similarity to explain complete detection outputs [18]. Quantitative studies of detector explanations also distinguish localization, faithfulness, and computational cost, emphasizing that visually plausible heatmaps may correspond only weakly to actual detector behavior [19, 20]. In agricultural imaging, CAM-

family explanations are increasingly common, but quantitative validation remains comparatively limited [21].

2.4. Research Positioning

Detector-output attribution and backbone-level feature auditing address different explanatory questions. The former asks why a particular box–class prediction was produced; the latter asks whether the representation entering subsequent neck fusion and detection stages is spatially selective for the object of interest. This distinction matters because predictive accuracy can conceal substantial differences in internal feature organization.

Table 1. Comparison of Prior Research and the Positioning of the Present Study

Research focus	Typical evidence	Explanatory target	Key limitation relative to this study
FFB detector benchmarking	mAP, precision, recall, latency	Final detector output	Does not quantify explanation behavior
Agricultural CAM visualization	Qualitative or limited quantitative saliency	Intermediate feature maps	Often evaluates one detector–explainer pairing
Detector-specific XAI, e.g., D-RISE	Box- and class-conditioned attribution	Complete detection decision	Addresses final prediction attribution rather than backbone representation
This study	*Matched detection benchmark, CAM-box IoU, Pointing Game, confidence-retention diagnostics*	*Pre-neck backbone representation*	*Tests whether explanation behavior remains consistent across closely performing YOLO configurations*

Accordingly, this study treats backbone saliency as an explanatory target in its own right rather than as a surrogate for complete detector attribution. EigenCAM is applied in its original class-agnostic form, whereas Grad-CAM++ and Score-CAM are adapted to explicit backbone targets as bGrad-CAM++ and bScore-CAM. The central question is whether the relative localization and confidence-retention behavior of these feature-auditing methods remains stable across YOLOv8m, YOLO11m, and YOLO26m when predictive accuracy is practically close.

3. METHODOLOGY

This section describes the matched detector benchmark, backbone-level CAM compatibility procedure, saliency evaluation, and statistical analysis used to determine whether closely performing YOLO detectors exhibit consistent explanation behavior. The experimental design separates predictive performance from backbone-level feature auditing so that differences in CAM behavior are not conflated with differences in detector accuracy.

3.1. Problem Formulation and Experimental Overview

Let an input image be denoted by $x \in [0,1]^{3\times H_0\times W_0}$, where $H_0 = W_0 = 640$. A detector D_θ, parameterized by θ, predicts a set of J detections,

$$D_\theta(x) = \left[\left(\hat{b}_j, \hat{c}_j, \hat{p}_j\right)\right]_{j=1}^{J} \quad (1)$$

where $\hat{b}_j$ is a predicted bounding box, $\hat{c}_j \in \mathcal{C}$ is its predicted class, and $\hat{p}_j \in [0,1]$ is its confidence score. The class set is $\mathcal{C} =$ [unripe, underripe, ripe, overripe, empty bunch, abnormal]. The detection objective is to localize each FFB and assign it to the appropriate Malaysian Palm Oil Board (MPOB) ripeness class. The explanatory objective is deliberately

https://doi.org/10.31436/iiumej.v27i3.4693

narrower. Given a trained detector D_θ, an explainer E_m, and a target backbone layer L, the objective is to generate a normalized spatial saliency map

$$S_{m,L}(x) \in [0,1]^{H_0 \times W_0}, \tag{2}$$

and evaluate its spatial correspondence with annotated FFB regions and its ability to preserve image-level detector confidence after saliency-based masking.

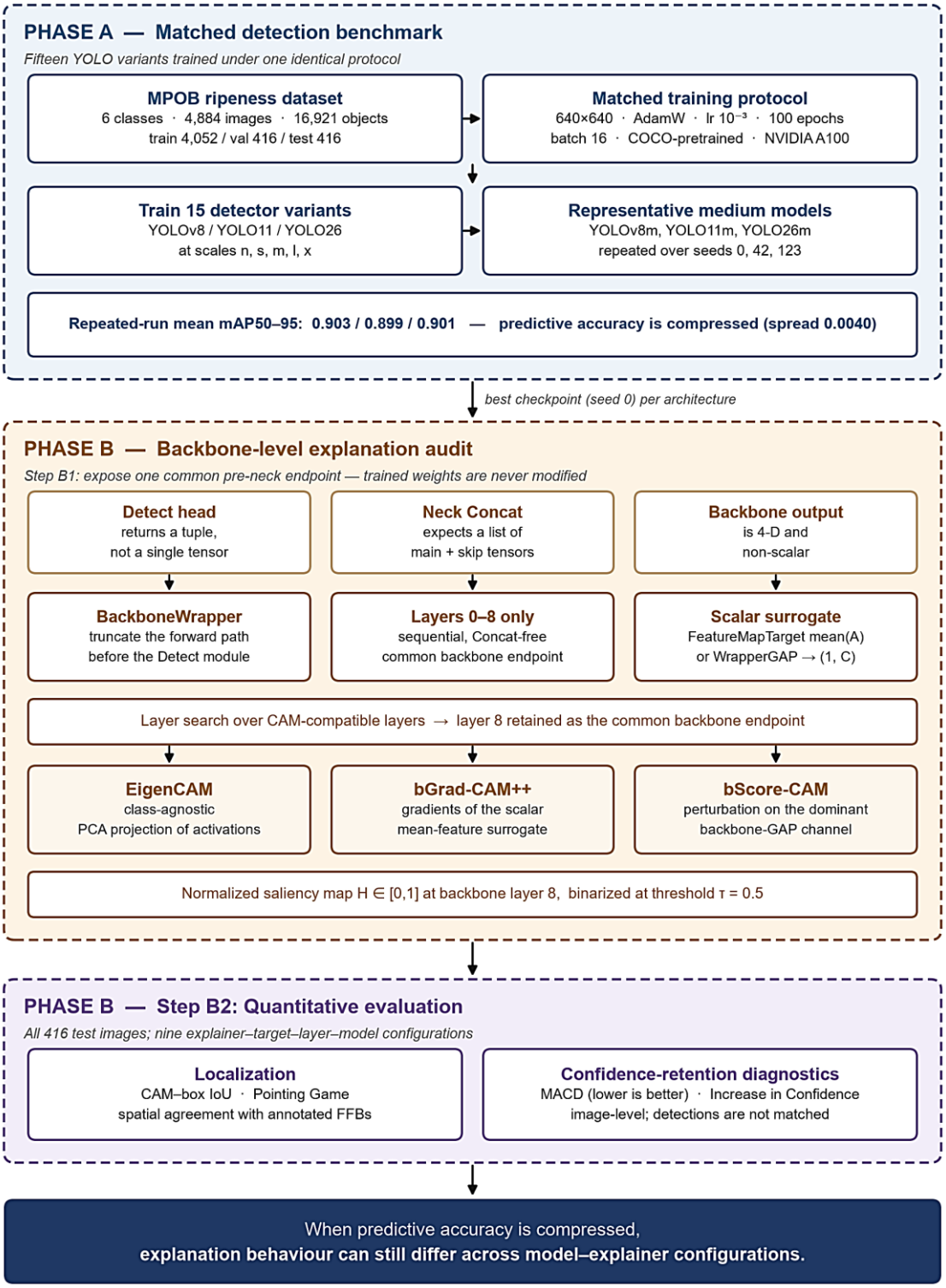

Figure 1. Two-phase experimental framework.

https://doi.org/10.31436/iiumej.v27i3.4693

The explanatory target is defined at the *backbone representation*, rather than at the final detector output. Accordingly, $S_{m,L}(x)$ characterizes whether the selected pre-neck feature representation is spatially concentrated on annotated FFB regions before multiscale feature fusion and final box–class prediction. The resulting saliency maps therefore support backbone-level feature auditing and should not be interpreted as causal attribution of a specific final detection. This objective complements box- and class-conditioned detector attribution rather than replacing it.

The experimental pipeline comprises two phases, as illustrated in Figure 1. Phase A benchmarks 15 detector configurations formed by three YOLO families, including YOLOv8, YOLO11, and YOLO26, at five model scales: nano, small, medium, large, and extra-large. We then evaluate the medium variant from each family across three random seeds to characterize variability in predictive performance. Phase B exposes a common compatible pre-neck backbone endpoint from the three medium detectors to EigenCAM, backbone-adapted Grad-CAM++ (bGrad-CAM++), and backbone-adapted Score-CAM (bScore-CAM), after which the resulting saliency maps are evaluated on the test set.

Because YOLOv8m, YOLO11m, and YOLO26m achieve practically close mAP50–95 under the matched training protocol, Phase B addresses the central methodological question of the study: whether similarly accurate detectors also exhibit similarly consistent explanation behavior across model–explainer configurations.

3.2. Dataset and Preprocessing

The experiments used a publicly available oil palm fresh fruit bunch (FFB) dataset developed for six-class ripeness grading [22]. The original dataset comprised 4,160 images with predefined training, validation, and test partitions in an approximately 80:10:10 ratio. The six classes were abnormal, empty bunch, overripe, ripe, underripe, and unripe.

Because the empty-bunch class was underrepresented in the training partition, with 266 dominant-class images, offline augmentation was applied exclusively to training images dominated by this class. The augmentation operations comprised horizontal and vertical flipping, 90° rotation, hue–saturation–value adjustment, brightness and contrast modification, blur, additive noise, synthetic shadow, and contrast-limited adaptive histogram equalization. This procedure increased the empty-bunch subset to 990 dominant-class images and yielded a final dataset of 4,884 unique images containing 16,921 annotated object instances. Neither the validation nor the test partition was modified by offline augmentation.

Table 2. Dataset distribution after training-only offline augmentation

Class	Dominant-class images	Images containing class	Object instances
Abnormal	869	1,237	2,599
Empty bunch	990	1,197	2,574
Overripe	658	1,366	2,845
Ripe	606	2,080	3,173
Underripe	819	1,332	2,618
Unripe	942	1,274	3,112
Total	**4,884**	**8,486**	**16,921**

Table 2 reports the resulting class distribution using three complementary counts. Dominant-class images assign each image to the class with the most annotated objects. Images containing class count an image once for each class represented in that image; consequently, these counts do not sum to the total number of unique images. Object instances represent

individual bounding-box annotations. This distinction matters because a single image may contain multiple FFBs from different ripeness classes.

During detector training, images were resized to 640×640 pixels. The Ultralytics training pipeline additionally applied online mosaic, mixup, HSV transformation, scaling, translation, and RandAugment, with mosaic augmentation disabled during the final ten epochs. For CAM analysis, each test image was resized to 640×640, converted to RGB, normalized to [0,1], and represented as a 1×3×640×640 tensor. The unchanged test partition contained 416 images and 1,493 annotated FFB instances.

3.3. Detection Models and Training Protocol

The detector benchmark comprised 15 configurations obtained by combining three YOLO families, including YOLOv8, YOLO11, and YOLO26, with five model scales: nano, small, medium, large, and extra-large. All configurations were trained under the same externally controlled protocol to minimize confounding from model-specific hyperparameter tuning. Each model was initialized from COCO-pretrained weights and trained for 100 epochs using AdamW with an initial learning rate of 1×10^{-3}, a batch size of 16, and 640×640 inputs. Experiments were conducted on an NVIDIA A100-SXM4-40GB GPU using Ultralytics 8.4.41 and PyTorch 2.11.0. Table 3 summarizes the training configuration.

Table 3. Externally controlled training configuration applied to all 15 detector variants.

Parameter	Value
Optimizer	AdamW
Initial learning rate	0.001
Epochs	100
Batch size	16
Input resolution	640×640
Initialization	COCO-pretrained weights
Initial benchmark seed	0
Repeated-training seeds	0, 42, and 123
Hardware	NVIDIA A100-SXM4-40GB

The initial benchmark used seed 0. Within each YOLO family, we ranked the five scales by test-set mAP50–95, and carried forward the highest-scoring scale to the explanation analysis. We selected the medium variant for all three families. To assess whether the small performance differences among these representative models were stable across training runs, we additionally trained YOLOv8m, YOLO11m, and YOLO26m using seeds 42 and 123. The subsequent CAM analysis used the primary seed-0 checkpoint, while the additional checkpoints were used only to characterize variation in detection performance.

Because model-scale selection used test-set mAP50–95, the test partition was not fully independent of model selection. The same partition was then used for target-layer selection and CAM evaluation. These choices may introduce selection optimism; therefore, the explanation results are interpreted as a controlled exploratory within-dataset comparison rather than an independent estimate of explanation generalization. An independent study should select model scale and explanatory layer using validation data before evaluating CAM behavior on a separate test set.

3.4. Backbone-Level Compatibility Procedure

Conventional CAM implementations could not be applied directly to the complete YOLO computation graph because of differences in detector outputs, internal feature routing, and

https://doi.org/10.31436/iiumej.v27i3.4693

explanatory target requirements. Rather than modifying the trained weights, we used a backbone-level compatibility procedure to expose a common intermediate representation to each CAM method. The procedure addresses the three implementation barriers summarized in Table 4 while explicitly restricting interpretation to pre-neck backbone feature auditing.

Let B_L denote the YOLO backbone truncated after layer L. For the common compatible endpoint $L = 8$,

$$A = B_8(x) \in \mathbb{R}^{1\times K\times h\times w}, \tag{3}$$

where K is the number of channels and $h \times w$ is the spatial resolution of the activation tensor. In YOLOv8m, YOLO11m, and YOLO26m, layers 0–8 form a sequential backbone path before the multi-input concatenation operations used in the neck. Layer 8 therefore provides a common evaluable representation while avoiding both structured detection-head outputs and neck operations that a simple sequential wrapper cannot reproduce.

Table 4. CAM–YOLO incompatibilities and corresponding backbone-level compatibility procedures.

Incompatibility	Observed error	Cause	Procedure response
Detection-head output	tuple object has no attribute CPU	The complete detector returns structured detection outputs rather than a single tensor compatible with the CAM interface.	Terminate the explanatory forward path before the detection head.
Neck concatenation	Invalid cat() argument combination	Neck concatenation requires multiple tensors from main and skip paths, whereas a sequential wrapper propagates a single tensor.	Restrict the common explanatory path to the sequential backbone through layer 8.
Non-scalar backbone output	Scalar-gradient or tensor-to-scalar error	The truncated backbone produces a four-dimensional activation tensor rather than the scalar or vector target required by the adapted CAM methods.	Use the scalar mean target in Eq. (4) for bGrad-CAM++ and global average pooling in Eq. (5) for bScore-CAM; EigenCAM requires no target adaptation.

The three CAM methods require different interfaces to this backbone representation. EigenCAM operates directly on A because it is class-agnostic and requires neither a scalar class score nor a gradient-defined target. By contrast, conventional Grad-CAM++ and Score-CAM are formulated around class-discriminative targets and therefore require explicit adaptation when the explanatory target is a backbone representation rather than a detector class output. To distinguish these implementations from their original formulations, we denote them as backbone-adapted Grad-CAM++ (bGrad-CAM++) and backbone-adapted Score-CAM (bScore-CAM).

For bGrad-CAM++, the four-dimensional backbone activation is reduced to the scalar surrogate

$$q_G(A) = \frac{1}{Khw}\sum_{k=1}^{K}\sum_{i=1}^{h}\sum_{j=1}^{w} A_{ij}^{k}. \tag{4}$$

which provides the scalar target required for gradient computation. The resulting gradients therefore characterize the sensitivity of the aggregated backbone representation rather than the sensitivity of a particular FFB class score.

For bScore-CAM, the truncated backbone is followed by global average pooling,

https://doi.org/10.31436/iiumej.v27i3.4693

$$z(x) = \mathrm{GAP}\big(B_8(x)\big) \in \mathbb{R}^K, \quad z_k = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w} A_{ij}^{k} \tag{5}$$

producing a channel-response vector that supports perturbation-based weighting. As described in Section 3.5, the dominant pooled backbone channel is then used as the feature-level target for each image. This target is a learned backbone feature response, not one of the six FFB ripeness classes.

These adaptations preserve the trained detector parameters and the core weighting principles of the three CAM methods while placing them on a common explanatory interface. EigenCAM retains its original class-agnostic formulation, whereas bGrad-CAM++ and bScore-CAM replace their conventional class-oriented targets with explicitly defined backbone targets.

The resulting saliency maps therefore characterize the spatial selectivity of the selected *pre-neck backbone representation*. They should not be interpreted as causal explanations of a specific final bounding box or ripeness-class prediction. Saliency generation is defined mathematically in Section 3.5, while target-layer selection and quantitative evaluation are described in Sections 3.6 and 3.7, respectively.

3.5. Backbone-Level CAM Formulations

Let $A = [A_k]_{k=1}^{K}, A_k \in \mathbb{R}^{h\times w}$, denote the activation tensor at the selected backbone layer. Let $\mathcal{U}(\cdot)$ denote bilinear upsampling to the input resolution and $\mathcal{N}(\cdot)$ denote min-max normalization to $[0,1]$.

3.5.1. EigenCAM

EigenCAM derives a class-agnostic saliency map from the dominant variation in the backbone activation tensor. The tensor is reshaped as $Z \in \mathbb{R}^{n\times K}, n = hw$, where each row represents a spatial location and each column an activation channel. After channel-wise centering,

$$\tilde{Z} = Z - \mathbf{1}\mu^T, \mu = \frac{1}{n}Z^T\mathbf{1}. \tag{6}$$

The channel covariance matrix and its dominant eigenvector are

$$C = \frac{1}{n-1}\tilde{Z}^T\tilde{Z}, Cv_1 = \lambda_1 v_1, \tag{7}$$

where v_1 corresponds to the largest eigenvalue. The raw EigenCAM map is

$$R^E = \mathrm{reshape}\big(\tilde{Z}v_1\big) \tag{8}$$

and the final normalized heatmap is

$$S^E = \mathcal{N}\big[\mathcal{U}\big(\mathrm{ReLU}(R^E)\big)\big]. \tag{9}$$

Because EigenCAM does not require a class score, S^E represents dominant spatial variation in the selected backbone representation rather than evidence for a specific ripeness class.

3.5.2. Backbone-Adapted Grad-CAM++ (bGrad-CAM++)

bGrad-CAM++ retains the Grad-CAM++ weighting mechanism but replaces the conventional class score with the backbone-derived scalar target $q_G(A)$ defined in Eq. (4). The gradient at each spatial location is

https://doi.org/10.31436/iiumej.v27i3.4693

$$g_{i,j}^{k} = \frac{\partial q_G}{\partial A_{i,j}^{k}}. \tag{10}$$

The implementation-equivalent pixel coefficient is

$$\alpha_{i,j}^{k} = \frac{\left(g_{i,j}^{k}\right)^2}{2\left(g_{i,j}^{k}\right)^2 + \left(\sum_{a=1}^{h}\sum_{b=1}^{w} A_{ab}^{k}\right)\left(g_{i,j}^{k}\right)^3 + \varepsilon}, \tag{11}$$

where $\varepsilon > 0$ prevents division by zero. The channel weight is

$$w_k^G = \sum_{i=1}^{h}\sum_{j=1}^{w} \alpha_{i,j}^{k}\, ReLU\left(g_{ij}^{k}\right), \tag{12}$$

yielding

$$S^G = \mathcal{N}\left[\mathcal{U}\left(\mathrm{ReLU}\left(\sum_{k=1}^{K} w_k^G A_k\right)\right)\right]. \tag{13}$$

Because $q_G(A)$ aggregates the backbone representation rather than a detector class score, S^G is interpreted as gradient-weighted backbone saliency and is not class-discriminative with respect to the six FFB labels.

3.5.3. Backbone-Adapted Score-CAM (bScore-CAM)

For bScore-CAM, each activation channel is first upsampled and normalized to form a spatial mask,

$$M_k = \mathcal{N}[\mathcal{U}(A_k)]. \tag{14}$$

The corresponding masked input is

$$x_k = x \odot M_k, \tag{15}$$

where M_k is broadcast across the RGB channels. Using the pooled backbone response $z(x)$ from Eq. (5), the dominant feature channel is selected as,

$$k^* = \arg\max_k z_k(x). \tag{16}$$

Each masked input x_k is then forwarded through the same backbone-GAP wrapper, and its response at the fixed target channel k^* determines the Score-CAM weight w_k^S. The resulting heatmap is

$$S^S = \mathcal{N}\left[\mathcal{U}\left(\mathrm{ReLU}\left(\sum_{k=1}^{K} w_k^S A_k\right)\right)\right]. \tag{17}$$

Importantly, k^* denotes a backbone feature channel rather than an FFB ripeness class. Thus, bScore-CAM preserves the perturbation-based principle of Score-CAM while explaining the dominant pooled backbone response rather than a specific final box–class prediction.

3.6. Target-Layer Selection

CAM behavior depends on the depth of the selected representation. Shallower layers generally preserve finer spatial detail, whereas deeper layers encode increasingly abstract semantic information [23]. We therefore conducted a target-layer search separately for each representative medium detector. For candidate layer L, localization quality was summarized as

$$Q_L = \frac{1}{2}\left[\frac{1}{N}\sum_{i=1}^{N} IoU_{i,L} + \frac{1}{N}\sum_{i=1}^{N} PG_{i,L}\right], \tag{18}$$

where CAM-box IoU and Pointing Game (PG) are both expressed on the [0,1] scale. We excluded layers associated with the final Detect module.

https://doi.org/10.31436/iiumej.v27i3.4693

The highest observed localization scores occurred in the neck: layer 18 for YOLOv8m and layer 19 for YOLO11m and YOLO26m. These layers require multi-input, list-based concatenation and therefore cannot be exposed through the common sequential wrapper described in Section 3.4. Selecting them would have required a different explanatory pathway for each detector and would have confounded the cross-model comparison with implementation differences, which is precisely the confound this design is intended to avoid; Table 4 documents the specific failures encountered. Layer 8 was consequently selected as the strongest compatible backbone layer and the common evaluable endpoint across all three detectors. This choice should not be interpreted as evidence that layer 8 is universally optimal for CAM analysis. Rather, it reflects a trade-off between observed localization performance and implementation compatibility. To show that this is not a knife-edge choice within its neighborhood, we also report the layer search over the full compatible window from layer 6 to layer 10 in Section 4.3. The target layer is therefore part of the explanatory design and must be reported explicitly.

The layer search used all 416 test images, which were subsequently reused for final CAM evaluation. This reuse may favor the selected layer and introduces selection optimism. Accordingly, the target-layer analysis and CAM results are interpreted as exploratory within-dataset evidence rather than independent estimates of explanation generalization.

3.7. Evaluation Metrics

3.7.1. Detection Performance

We assessed detection performance using precision, recall, F1-score, mAP50, and mAP50–95. The primary predictive metric was mAP50–95, which averages class-level average precision over the IoU thresholds $\mathcal{T} = [0.50,\ 0.55, \ldots,\ 0.95]$. We reported computational efficiency using end-to-end inference time, frames per second, and serialized model size. Because latency depends on both hardware and software configuration, these values characterize execution on the stated NVIDIA A100 platform and should not be interpreted as hardware-independent measures.

Latency benchmarking used the native PyTorch 2.11.0 Ultralytics 8.4.41 validation runtime on the NVIDIA A100-SXM4-40GB GPU recorded in the Colab notebooks, with batch size one and 640×640 inputs. No TensorRT engine, ONNX Runtime backend, explicit FP16 mode, INT8 quantization, pruning, or batched-throughput optimization was used; the reported values are therefore native PyTorch full-precision timings rather than optimized deployment measurements. Total time is the sum of the recorded per-image preprocessing, inference, and post-processing times, and FPS is its reciprocal. Because all 15 variants were evaluated under the same runtime conditions, the values support relative accuracy–efficiency comparison on this platform but should not be transferred directly to TensorRT, quantized, or different-hardware deployments.

3.7.2. Saliency Localization

Let S_i denote the normalized saliency map for the test image i. Using the common threshold $\tau = 0.5$, the binary saliency mask is

$$P_i(u, v) = \mathbf{1}[S_i(u, v) \geq \tau]. \tag{19}$$

For an image containing R_i annotated FFBs, each bounding box is rasterized into a binary mask G_{ir}. The image-level ground-truth mask is their union,

$$G_i = \bigvee_{r=1}^{R_i} G_{ir}. \tag{20}$$

https://doi.org/10.31436/iiumej.v27i3.4693

CAM-box IoU is then

$$\mathrm{IoU}_i^{\mathrm{CAM}} = \frac{|P_i \cap G_i|}{|P_i \cup G_i|}. \tag{21}$$

The reported CAM-box IoU is the mean image-level IoU across the test set. Because all annotated boxes in an image are combined, this metric measures spatial overlap with the overall annotated FFB region rather than instance-specific attribution.

Pointing Game evaluates whether the most salient spatial location falls within any annotated FFB. Let

$$(u_i^*, v_i^*) = \arg\max_{u,v} S_i(u, v), \tag{22}$$

and define

$$PG_i = \mathbf{1}\left[(u_i^*, v_i^*) \in \bigcup_{r=1}^{R_i} b_{ir}\right]. \tag{23}$$

The dataset-level Pointing Game score is

$$PG = \frac{100}{N} \sum_{i=1}^{N} PG_i. \tag{24}$$

Unlike CAM-box IoU, Pointing Game is independent of τ because it depends only on the location of maximum saliency.

3.7.3. Image-Level Confidence Retention

Confidence retention was evaluated using the complete YOLO detector rather than the truncated backbone. For the original image,

$$p_i = \begin{cases} \max_j \hat{p}_{ij}, & J_i > 0, \\ 0, & J_i = 0, \end{cases} \tag{25}$$

where J_i is the number of detections. The keep-bright masked image is

$$\tilde{x}_i = x_i \odot P_i, \tag{26}$$

with P_i broadcast across the three RGB channels. The corresponding masked-image confidence is

$$\tilde{p}_i = \begin{cases} \max_j \tilde{p}_{ij}, & \tilde{J}_i > 0, \\ 0, & \tilde{J}_i = 0, \end{cases} \tag{27}$$

Mean Absolute Confidence Drop (MACD) is defined as

$$\mathrm{MACD} = \frac{100}{N} \sum_{i=1}^{N} \max(0, p_i - \tilde{p}_i). \tag{28}$$

MACD is a study-specific descriptive measure of the absolute reduction in maximum image-level confidence after masking. Unlike conventional normalized Average Drop, it is not divided by the original confidence. Lower MACD therefore indicates greater confidence retention.

Increase in Confidence (IC) is

$$\mathrm{IC} = \frac{100}{N} \sum_{i=1}^{N} 1[\tilde{p}_i > p_i]. \tag{29}$$

Because detections before and after masking are not matched by object, box, or class, MACD and IC are interpreted strictly as image-level confidence-retention diagnostics, not as object-specific or class-specific faithfulness measures.

All CAM methods used the same fixed threshold $\tau = 0.5$, with no model- or explainer-specific tuning. This value is the implementation default and was fixed before any model–explainer comparison: the identical threshold constant was already in place during the target-layer search of Section 3.6, which preceded the CAM evaluation, and it was not adjusted afterward for any configuration. It is therefore prespecified for the comparison reported here, although it was not registered in advance of the study as a whole. Because CAM-box IoU nevertheless depends on this threshold, every threshold-dependent conclusion is cross-checked against Pointing Game, which is threshold-free by construction, and threshold-dependent metrics are interpreted comparatively across configurations rather than as absolute measures of explanation validity.

3.8. Class-Stratified Analysis

The primary EigenCAM–bScore-CAM comparison was stratified post hoc using the per-image CAM records already produced in the original Colab evaluation. Each image was assigned to the first annotated class recorded by the evaluation code. Overripe, ripe, and underripe images were also pooled as a difficult subset, while abnormal, empty-bunch, and unripe images formed a visually distinctive subset. This aggregation introduces no new model, checkpoint, image, or CAM run; it describes the existing 416-image results at a finer class resolution.

3.9. Statistical Analysis

Uncertainty in CAM localization was quantified using percentile bootstrap 95% confidence intervals based on 10,000 resamples of the 416 test images with a fixed random seed of 0. For the primary localization comparison, paired Wilcoxon signed-rank tests evaluated image-level CAM-box IoU differences between EigenCAM and bScore-CAM within each detector. Holm adjustment across the three detector-specific tests controlled the family-wise error rate; all tests were two-sided. Because model-scale and target-layer selection reused the test partition, these analyses are interpreted as exploratory estimates of effect magnitude and consistency rather than confirmatory estimates of out-of-sample explanation generalization. The class-stratified results are reported descriptively.

4. RESULTS

The results are organized to test the study's central proposition in two stages. First, the 15 YOLO variants are compared to determine whether conventional detection accuracy and computational efficiency provide a clear basis for model selection. Second, repeated runs and class-level performance are examined to establish whether the three representative medium models are sufficiently close in predictive performance to justify a separate analysis of explanation behavior.

4.1. Detection Accuracy and Efficiency

All 15 detector variants achieved mAP50 values above 0.993, while mAP50–95 ranged from 0.8860 to 0.9048 (Table 5). Within each YOLO family, the medium-scale model achieved the highest mAP50–95: 0.9048 for YOLOv8m, 0.8998 for YOLO11m, and 0.9043 for YOLO26m. Increasing model scale beyond medium did not yield consistent accuracy gains. Instead, the large and extra-large variants generally incurred higher model size and latency while remaining within the same narrow mAP50–95 range. The benchmark therefore indicates a compressed accuracy regime rather than a clear accuracy-driven ranking.

https://doi.org/10.31436/iiumej.v27i3.4693

Table 5. Detection accuracy and end-to-end inference performance of the 15 YOLO variants on the NVidia A100.

Model	Size (MB)	mAP50–95	Total (ms)	FPS
YOLOv8n	5.97	0.8901	1.61	620.25
YOLOv8s	21.49	0.8997	2.60	384.13
YOLOv8m	49.63	**0.9048**	5.05	197.97
YOLOv8l	83.60	0.8970	7.79	128.40
YOLOv8x	130.40	0.9027	12.12	82.48
YOLO11n	5.23	0.8921	1.79	559.51
YOLO11s	18.30	0.8984	3.10	322.28
YOLO11m	38.65	**0.8998**	5.20	192.44
YOLO11l	48.84	0.8982	6.24	160.35
YOLO11x	109.10	0.8965	11.45	87.36
YOLO26n	5.15	0.8860	1.29	773.49
YOLO26s	19.38	0.8998	2.73	365.70
YOLO26m	42.00	**0.9043**	4.76	210.19
YOLO26l	50.54	0.8969	5.77	173.39
YOLO26x	112.86	0.8995	10.93	91.49

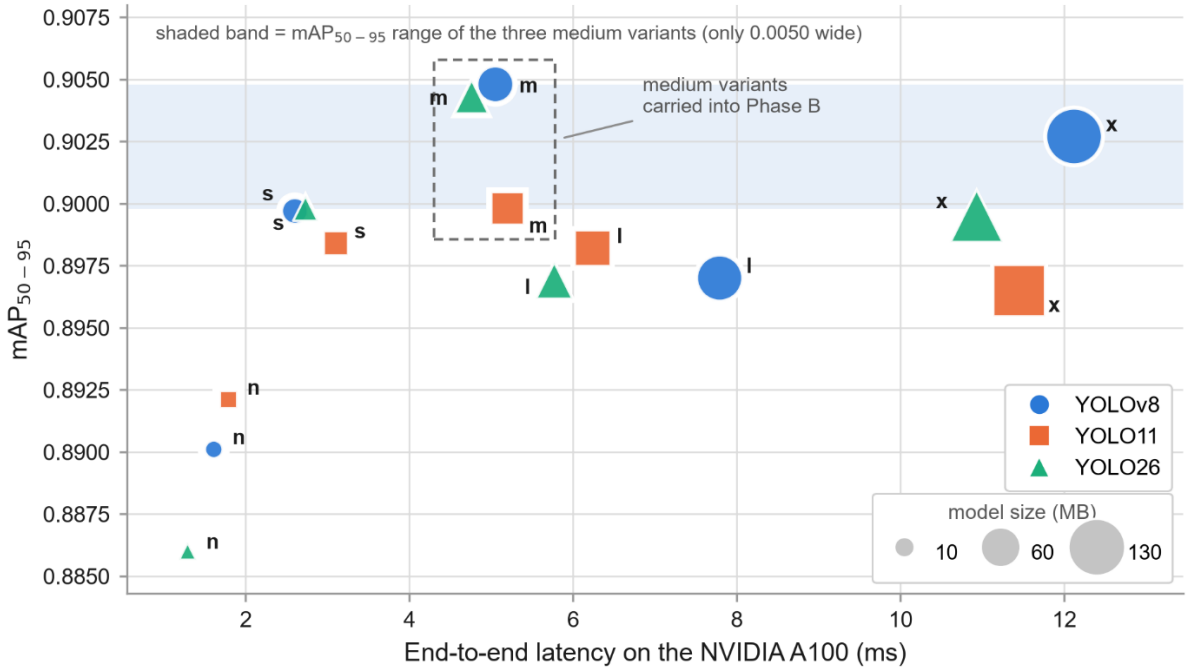

Figure 2. Accuracy–efficiency trade-off of the 15 YOLO variants on the NVidia A100. Marker area represents serialized model size; *n*, *s*, *m*, *l*, and *x* denote model scale. The three medium variants occupy a narrow high-accuracy region, while latency and model size more clearly differentiate their deployment profiles.

Figure 2 further illustrates this compressed accuracy–efficiency region. The three medium models achieved single-run mAP50–95 values of 0.9048, 0.8998, and 0.9043 for YOLOv8m, YOLO11m, and YOLO26m, respectively, corresponding to a maximum spread of only 0.0050. However, their deployment characteristics were more clearly separated. YOLO26m required 4.76 ms per image, compared with 5.05 ms for YOLOv8m and 5.20 ms for YOLO11m, corresponding to 210.19, 197.97, and 192.44 FPS, respectively. Thus, YOLO26m offered the lowest measured latency among the medium variants, although the absolute difference remained modest and is specific to the reported hardware and software environment.

Two observations follow from these results. First, near-ceiling mAP50 should not be interpreted as evidence that the detection task is saturated. The lower mAP50–95 values, centered near 0.90, indicate that stricter localization criteria continue to expose performance differences that are largely hidden at IoU = 0.50. Second, increasing detector size does not produce monotonic accuracy improvements. The medium variants provide the strongest overall accuracy–efficiency balance, whereas larger variants incur substantial computational cost without consistent gains in localization performance.

Accordingly, the benchmark does not identify a dominant detector solely on predictive accuracy. Instead, it establishes a setting in which several models achieve closely matched accuracy while differing in computational cost. This compressed performance regime provides the basis for examining whether explanation behavior remains similarly close.

4.2. Repeated Runs and Class-Level Behavior

The single-run ranking of the three medium detectors was examined across three random seeds to determine whether their small accuracy differences were stable or comparable to ordinary training variation.

Table 6. Repeated-training mAP50–95 of the three medium YOLO models.

Seed	YOLOv8m	YOLO11m	YOLO26m
0	0.905	0.900	0.904
42	0.904	0.900	0.901
123	0.901	0.897	0.899
Mean ± SD	**0.903 ± 0.002**	**0.899 ± 0.002**	**0.901 ± 0.003**

As shown in Table 6, across the three runs, mean mAP50–95 was 0.903±0.002 for YOLOv8m, 0.899±0.002 for YOLO11m, and 0.901±0.003 for YOLO26m. The maximum separation between the model means was only 0.004, which is of the same order as the observed run-to-run standard deviations. The small differences in mean performance therefore provide limited evidence for a practically meaningful ranking among the three medium detectors. Importantly, these results support practical closeness, not statistical equivalence: three seeds are sufficient to show that a single favorable run should not determine model selection, but they are insufficient for a formal equivalence claim or robust between-model significance analysis.

Table 7. Class-level mAP50–95 of the three representative medium detectors on the test partition.

Class	YOLOv8m	YOLO11m	YOLO26m
Abnormal	0.854	0.852	0.851
Empty bunch	0.910	0.907	0.902
Overripe	0.893	0.885	0.876
Ripe	0.915	0.908	0.919
Underripe	0.914	0.907	0.909
Unripe	0.931	0.929	0.935
All classes	0.9048	0.8998	0.9043

Note: Values are seed-0 checkpoints evaluated on the unmodified test partition of 416 images and 1,493 annotated instances. The all-classes row reports the aggregate mAP50–95 of Table 5.

Class-level performance further clarifies where detection remains difficult. As Table 7 shows, abnormal bunches consistently produced the lowest mAP50–95, at 0.854, 0.852, and 0.851 for YOLOv8m, YOLO11m, and YOLO26m, whereas unripe bunches achieved 0.931,

https://doi.org/10.31436/iiumej.v27i3.4693

0.929, and 0.935. The most operationally consequential ambiguity occurred among underripe, ripe, and overripe FFBs, where visual characteristics evolve gradually rather than at sharply defined boundaries [6]. Overripe was the weakest of the three stages for every detector, at 0.893, 0.885, and 0.876. Aggregate mAP therefore conceals meaningful class asymmetry: visually distinctive classes are detected more reliably, whereas the biologically adjacent ripeness stages remain comparatively difficult. The class ordering is nearly identical across the three detectors, further indicating that they have converged on similar predictive behavior and motivating the class-stratified explanation analysis in Section 4.5.

These findings change the role of predictive accuracy in the subsequent analysis. Detection performance alone does not establish that YOLOv8m, YOLO11m, and YOLO26m are interchangeable, but the observed differences are too small to provide a compelling basis for selection on this dataset. Once predictive accuracy becomes compressed, *computational efficiency and explanation behavior become complementary evaluation dimensions*, motivating the backbone-level CAM analysis that follows.

4.3. Compatible Target-Layer Selection

Because CAM behavior depends on the representation being interrogated, target-layer selection was examined before comparing explainers. Table 8 summarizes the exploratory layer search using the combined localization score defined in Eq. (18).

Table 8. Exploratory target-layer search.

Metric	YOLOv8m	YOLO11m	YOLO26m
Overall best layer	18 (C2f)	19 (C3k2)	19 (C3k2)
Overall combined score	53.40%	54.78%	51.27%
Compatible backbone layer	8 (C2f)	8 (C3k2)	8 (C3k2)
Backbone combined score	51.23%	39.08%	46.70%

Note. Combined score is the mean of CAM-box IoU and Pointing Game, both expressed on the same scale. The formal cross-model comparison uses the common compatible backbone layer 8.

The highest observed localization scores occurred in the neck: layer 18 for YOLOv8m and layer 19 for YOLO11m and YOLO26m. However, these representations depend on multi-input list-based concatenation and could not be exposed through the common sequential CAM wrapper. The formal comparison therefore used layer 8, the strongest compatible backbone layer shared across all three detectors.

This constraint affected the explanatory target differently across models. For YOLOv8m, the combined score decreased only slightly from 53.40% at the overall-best layer to 51.23% at layer 8. The reduction was more pronounced for YOLO11m, from 54.78% to 39.08%, and more moderate for YOLO26m, from 51.27% to 46.70%. Thus, implementation compatibility is not methodologically neutral: the strongest common evaluable layer does not necessarily coincide with the representation showing the strongest observed localization.

Because the choice of layer 8 governs every subsequent comparison, its sensitivity was examined across the neighboring layers rather than asserted from a single value. Table 9 reports the full layer 6 to layer 10 window for all three detectors using the same combined localization score.

Layer 8 attains the highest combined score in all three detectors within this window, exceeding the next-best candidate by 26.10 percentage points in YOLOv8m, 4.96 points in YOLO11m, and 12.82 points in YOLO26m. The selection is therefore not a knife-edge choice among neighboring representations, although the margin is far smaller for YOLO11m than for

https://doi.org/10.31436/iiumej.v27i3.4693

the other two detectors, and a study adopting a different compatible endpoint in that model would obtain a correspondingly different explanatory target.

Table 9. Target-layer sensitivity across the compatible window, layers 6 to 10. Each cell reports Pointing Game (%) / CAM-box IoU (%) / combined score (%).

Layer	Block type (v8m / 11m / 26m)	YOLOv8m	YOLO11m	YOLO26m
6	C2f / C3k2 / C3k2	36.54 / 5.02 / 20.78	60.10 / 8.02 / 34.06	58.89 / 8.87 / 33.88
7	Conv / Conv / Conv	35.10 / 15.16 / 25.13	54.87 / 13.37 / 34.12	7.69 / 11.14 / 9.42
8	C2f / C3k2 / C3k2	85.82 / 16.64 / 51.23	63.94 / 14.22 / 39.08	79.33 / 14.06 / 46.70
9	SPPF / SPPF / SPPF	0.96 / 6.26 / 3.61	43.79 / 12.57 / 28.18	20.91 / 13.72 / 17.32
10	Upsample / C2PSA / C2PSA	1.20 / 6.47 / 3.83	38.46 / 15.18 / 26.82	1.92 / 3.12 / 2.52

Note: Scores are computed from the unweighted spatial mean of each layer's activation tensor over all 416 test images, so they characterize the representation rather than any particular CAM method. Layer 10 lies outside the backbone in YOLOv8m, where the SPPF stage at layer 9 terminates the sequential path.

The window also shows that localization degrades sharply and unevenly once the sequential backbone terminates. Pointing Game at the post-backbone stages falls to 0.96% and 1.20% in YOLOv8m and to 20.91% and 1.92% in YOLO26m, while YOLO11m retains 43.79% and 38.46% at the same positions. Layer selection is thus a model-specific decision even within a narrow window, which reinforces the study's central point that the explanatory configuration, rather than the explainer alone, is the unit that must be reported.

Layer 8 should therefore be interpreted as a common compatible backbone endpoint, not as an optimal CAM layer. Because both layer selection and final CAM evaluation reused the same 416-image test set, these results remain exploratory and may reflect selection optimism; the sensitivity window of Table 9 shows only that the choice is stable among its neighbors, not that it is independent of the evaluation data.

4.4. Model–Explainer Explanation Behavior

Table 10 presents the central empirical result. Predictive accuracy among the three medium detectors was tightly compressed, whereas explanation behavior varied substantially across model–explainer configurations. No explainer dominated across all three backbones.

For YOLOv8m, EigenCAM produced the highest Pointing Game score (71.39%) and lowest MACD (32.68%), while CAM-box IoU differed only marginally among the three methods (14.85–15.70%). The EigenCAM–bScore-CAM IoU contrast was not significant after Holm adjustment ($p=0.650$). EigenCAM therefore showed the strongest observed peak-localization and confidence-retention profile for YOLOv8m, but did not statistically establish overall superiority.

The ranking changed markedly for YOLO11m. EigenCAM yielded 7.00% CAM-box IoU, 35.10% Pointing Game, and 81.30% MACD, whereas bScore-CAM achieved 22.83%, 75.72%, and 41.89%, respectively. bGrad-CAM++ produced similarly strong localization, with 21.38% IoU and 67.07% in Pointing Game. The EigenCAM–bScore-CAM IoU difference was 15.82 percentage points and remained significant after Holm adjustment ($p<0.001$). This rank reversal is the primary empirical finding: an explainer that performed strongly on one detector did not retain the same relative behavior on another detector with similar predictive accuracy.

https://doi.org/10.31436/iiumej.v27i3.4693

Table 10. Backbone-CAM evaluation on all 416 test images at layer 8.

Model	Method	IoU (%)	PG (%)	MACD (%) ↓	IC (%)
YOLOv8m	EigenCAM	15.70	71.39	32.68	0.24
YOLOv8m	bGrad-CAM++	15.27	35.10	40.14	0.72
YOLOv8m	bScore-CAM	14.85	66.35	48.62	0.00
YOLO11m	EigenCAM	7.00	35.10	81.30	0.00
YOLO11m	bGrad-CAM++	21.38	67.07	43.66	0.00
YOLO11m	bScore-CAM	22.83	75.72	41.89	0.00
YOLO26m	EigenCAM	10.68	19.71	66.51	0.24
YOLO26m	bGrad-CAM++	11.16	7.69	74.65	0.00
YOLO26m	bScore-CAM	10.50	9.86	78.03	0.00

Note: Higher IoU and PG are better; lower MACD is better. MACD is the study-specific, non-normalized reduction in maximum image-level detector confidence. bGrad-CAM++ and bScore-CAM are the backbone-adapted formulations defined in Section 3.4. Shading, if retained, should indicate the strongest observed value within each architecture and should not imply statistical superiority.

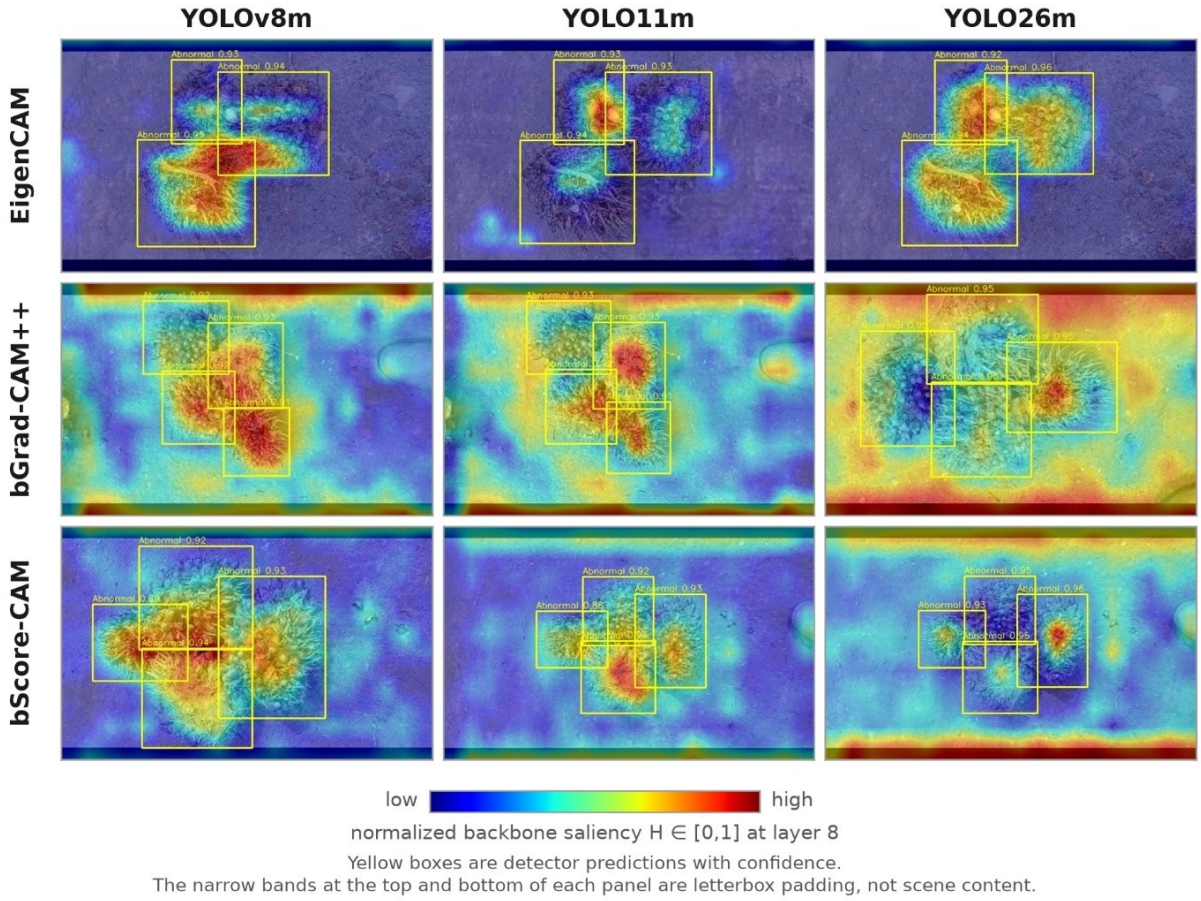

Figure 3. Representative layer-8 backbone heatmaps for an abnormal-bunch scene. EigenCAM is applied in its original class-agnostic form, whereas bGrad-CAM++ and bScore-CAM use the backbone-derived targets defined in Section 3.4. The examples illustrate the quantitative trends but do not establish statistical superiority. All overlays represent backbone feature saliency rather than attribution to a specific final box–class prediction.

For YOLO26m, all evaluated layer-8 configurations showed comparatively weak localization. Pointing Game ranged from 7.69% to 19.71%, while MACD ranged from 66.51% to 78.03%. EigenCAM and bScore-CAM produced nearly identical IoU values (10.68% and

10.50%; Holm-adjusted p=0.650), with overlapping bootstrap 95% confidence intervals of 9.60–11.78% and 10.01–11.00%, respectively. This result applies only to the selected backbone target and layer and should not be generalized to YOLO26 as a whole.

Across all nine configurations, Pointing Game spanned 68.03 percentage points, and MACD spanned 48.62 points, whereas repeated-run mean mAP50–95 differed by only 0.0040 across the three medium detectors. Thus, models that are nearly indistinguishable by predictive accuracy need not be interchangeable when explanation behavior is included as an evaluation criterion. Figure 3 shows representative heatmaps illustrating these contrasting saliency patterns.

The absolute CAM-box IoU values remained modest, with a maximum of 22.83%. This is consistent with the study design: the maps originate from a deep, lower-resolution backbone representation, are not conditioned on individual detections, and are evaluated against the union of annotated boxes in multi-object images. They are therefore best interpreted as relative measures of backbone spatial selectivity rather than precise pixel-level attribution. Section 5.4 assesses whether this level of overlap is practically sufficient and why direct comparison with COCO-style detection benchmarks is not meaningful. A class-stratified analysis next examines whether the aggregate pattern is concentrated in visually distinctive classes or also occurs in the biologically adjacent underripe, ripe, and overripe stages.

Increase in Confidence was only 0–0.72% across all configurations. Because detections were not matched before and after masking, IC and MACD remain coarse image-level confidence-retention diagnostics rather than object-specific faithfulness measures.

4.5. Class-Stratified Explanation Behavior

Aggregate metrics can conceal class-specific behavior, and the operationally consequential decisions in FFB grading occur among the biologically adjacent underripe, ripe, and overripe stages. Table 11 therefore stratifies the primary contrast, EigenCAM against bScore-CAM, by the dominant annotated class of each test image, and pools the three adjacent stages into a difficult subset and the visually distinctive classes into a comparison subset.

Table 11. Class-stratified localization for the primary EigenCAM–bScore-CAM contrast at layer 8. Each cell reports CAM-box IoU (%) / Pointing Game (%).

Class (n)	YOLOv8m EigenCAM	YOLOv8m bScore-CAM	YOLO11m EigenCAM	YOLO11m bScore-CAM	YOLO26m EigenCAM	YOLO26m bScore-CAM
Abnormal (56)	12.71 / 62.50	19.91 / 92.86	5.39 / 35.71	22.95 / 80.36	11.43 / 12.50	8.44 / 10.71
Empty bunch (27)	12.21 / 70.37	14.60 / 66.67	4.45 / 25.93	25.30 / 70.37	9.21 / 22.22	8.86 / 11.11
Overripe (72)	14.51 / 63.89	15.64 / 52.78	4.80 / 38.89	24.05 / 93.06	11.88 / 29.17	12.99 / 8.33
Ripe (103)	14.46 / 72.82	16.04 / 66.02	5.03 / 24.27	22.11 / 84.47	12.36 / 19.42	11.05 / 16.50
Underripe (77)	15.11 / 70.13	14.27 / 80.52	5.46 / 31.17	19.27 / 63.64	12.18 / 16.88	10.23 / 2.60
Unripe (81)	22.12 / 83.95	9.78 / 46.91	14.89 / 51.85	25.12 / 59.26	6.04 / 18.52	9.80 / 8.64
Difficult subset (252)	14.67 / 69.44	15.38 / 66.67	5.10 / 30.56	21.80 / 80.56	12.17 / 21.43	11.35 / 9.92
Distinctive subset (164)	17.27 / 74.39	14.03 / 65.85	9.93 / 42.07	24.41 / 68.29	8.40 / 17.07	9.18 / 9.76

Note: *n* is the number of test images whose first annotated instance carries the listed class. The difficult subset pools overripe, ripe, and underripe; the distinctive subset pools abnormal, empty bunch, and unripe

The descriptive rank reversal is present within the difficult subset and is not produced by the visually distinctive classes. On YOLO11m, bScore-CAM exceeds EigenCAM by 16.70 percentage points of IoU on these 252 images, at 21.80% versus 5.10%, and by 50.00 points of Pointing Game, at 80.56% versus 30.56%. On YOLOv8m, the corresponding IoU values are close, at 15.38% for bScore-CAM and 14.67% for EigenCAM; on YOLO26m, they are 11.35% and 12.17%, respectively. The class-stratified evidence therefore supports the large YOLO11m contrast but does not justify superiority claims where the methods remain closely matched.

Two class-specific effects nevertheless qualify any single-number summary. First, the aggregate EigenCAM advantage on YOLOv8m is concentrated in the unripe class, where it reaches 22.12% IoU and 83.95% Pointing Game against 9.78% and 46.91% for bScore-CAM; restricted to the difficult subset, bScore-CAM is marginally ahead on IoU. Second, on YOLO26m, every configuration is weak, but the failure is not uniform: bScore-CAM Pointing Game falls to 2.60% on underripe images while reaching 16.50% on ripe images. Explanation quality therefore depends on the class, the model, the explainer, and the configuration; aggregate metrics may be weakest on the classes that matter most for grading.

5. DISCUSSION

The results show that the principal distinction among the evaluated detectors is not simply predictive accuracy, but how much explanation behavior depends on the specific model, explanatory target, selected layer, and CAM procedure. The three medium detectors achieved practically close detection performance, yet their backbone-level saliency profiles differed substantially. This contrast is central to interpreting the study: explanation behavior cannot be inferred from mAP alone and should not be treated as a model-independent property of the explainer.

5.1. Detection Accuracy Has Limited Discriminative Power

The three medium detectors occupy a narrow accuracy range. Their repeated-run mean mAP50–95 values differ by no more than 0.004, with standard deviations of 0.002–0.003. These results do not establish statistical equivalence, but they provide little basis for claiming meaningful superiority from the observed ordering alone. For example, selecting YOLOv8m over YOLO26m because its mean mAP50–95 is higher by 0.002 would assign disproportionate importance to a difference that is comparable to ordinary run-to-run variation.

This compressed accuracy regime also limits the scientific value of ranking newer YOLO generations exclusively by mAP. Once predictive performance converges, other criteria matter more. YOLO26m illustrates this point: it retained nearly the same detection accuracy as YOLOv8m while achieving lower measured end-to-end latency on the NVidia A100. Computational efficiency therefore constitutes a legitimate additional selection criterion when deployment resources matter.

The latency and CAM findings, however, must remain analytically separate. The CAM procedure terminates at the backbone and does not interrogate the NMS-free detection pathway. Consequently, the weaker layer-8 saliency observed for YOLO26m cannot be attributed to its NMS-free design or other downstream head characteristics. Such an explanation would be causal speculation beyond the experimental evidence. The reference-baseline diagnostics of Section 4.7 support the same restriction: the weaker layer-8 behavior of YOLO26m is visible relative to its own unweighted channel mean, which is a property of how the explainers interact with that representation rather than of the detection head.

https://doi.org/10.31436/iiumej.v27i3.4693

5.2. Explanation Behavior Is Conditional on the Model–Explainer Configuration

The central finding is not that one CAM method is universally superior, but that explainer performance changes across model–target–layer configurations. EigenCAM captures dominant activation variation, bGrad-CAM++ weights features using gradients of a backbone-derived scalar target, and bScore-CAM uses perturbation responses associated with a pooled backbone feature. Because these mechanisms interrogate different aspects of the learned representation, their relative rankings need not remain stable across detectors even when predictive accuracy is nearly identical.

The clearest evidence is the rank reversal between YOLOv8m and YOLO11m. EigenCAM produced the strongest observed peak-localization and confidence-retention profile on YOLOv8m, whereas bScore-CAM produced substantially stronger localization on YOLO11m. This reversal is more informative than the absolute ranking of any single method because it directly challenges the assumption that a CAM method validated on one detector can be transferred unchanged to another.

The present evidence establishes an empirical association but does not identify its mechanism. EigenCAM, bGrad-CAM++, and bScore-CAM interrogate the representation through principal-component, gradient-weighted, and perturbation-weighted operations, respectively, so differences in activation covariance, gradient stability, or channel-response distributions plausibly explain the reversal. These quantities were not recorded in the Colab runs and cannot be reconstructed reliably from aggregate CAM metrics. The observed differences therefore cannot be attributed to C2f, C3k2, C2PSA, or any other architectural component in isolation. Instrumented analysis of the existing checkpoints is reserved for future work.

The appropriate unit of evaluation is therefore the complete *explainer–target–layer–model configuration*, not the explainer alone. This distinction matters for post-hoc XAI, where visually plausible saliency maps can encourage overgeneralization if their validity is assumed to transfer across models.

5.3. Implications for Model Selection and Deployment

The results suggest three complementary model-selection dimensions: *predictive accuracy*, *computational efficiency*, and *explanation auditability*. In this study, accuracy alone provided little separation among the three medium detectors, while latency and explanation behavior produced materially different comparative profiles.

For the evaluated layer-8 backbone target, YOLOv8m–EigenCAM yielded the highest Pointing Game score and lowest MACD, whereas YOLO11m–bScore-CAM achieved the strongest localization. YOLO26m provided the lowest latency among the medium variants but weaker localization across the evaluated layer-8 CAM configurations. These rankings are descriptive and configuration-specific; they should not be interpreted as universal recommendations for either a detector or an explainer.

This distinction matters for FFB grading. The most difficult classes are the biologically adjacent underripe, ripe, and overripe stages, where visual differences are gradual and grading decisions are consequential, and Section 4.5 shows that the explainer ranking reverses within exactly this subset rather than being averaged into existence across easier classes. It also shows that failure is class-specific: on YOLO26m, bScore-CAM localizes correctly on 16.50% of ripe images but only 2.60% of underripe images. In such settings, a high aggregate mAP may be insufficient if the accompanying explanation consistently emphasizes regions that are difficult to reconcile with annotated FFB locations, and an aggregate explanation score may itself be

https://doi.org/10.31436/iiumej.v27i3.4693

insufficient if it conceals a class on which the explanation fails. Explanation auditing therefore provides an additional diagnostic dimension rather than a substitute for predictive accuracy, and it should be reported per class.

More broadly, CAM should not be treated as an interchangeable visualization layer attached after model training. The explanatory target and layer should be specified explicitly, and the selected explainer–target–layer–model configuration should be validated before its saliency output is used for interpretation or decision support.

5.4. Is the Observed Localization Practically Sufficient?

The largest CAM-box IoU observed in this study, 22.83%, is modest in absolute terms and not sufficient on its own to establish deployment-ready explanation quality. A direct numerical comparison with COCO detection IoU or ImageNet localization accuracy would be misleading: those benchmarks evaluate supervised predicted regions or class-conditioned localization, whereas the present metric compares a thresholded, class-agnostic saliency mask from one intermediate tensor with the union of all annotated boxes. Threshold choice, feature-map resolution, and multi-object box union therefore affect the value in ways that do not apply to standard detector mAP [24, 25]. The result is best interpreted comparatively within the present protocol.

Within that protocol, bScore-CAM on YOLO11m combined the highest observed CAM-box IoU (22.83%) with a Pointing Game score of 75.72%, indicating that its peak saliency often fell on an annotated bunch even though the thresholded map did not closely reproduce box extent. This supports its use as a backbone-auditing signal, not as a precise segmentation or as evidence that the explanation is sufficient for autonomous deployment.

Pointing Game and CAM-box IoU should therefore be read together: the former tests whether peak saliency reaches an annotated FFB and the latter tests coarse spatial extent at the fixed threshold. Neither metric validates a specific class decision, and this study did not establish an acceptance threshold for operational deployment.

Instance-level IoU was not added because the evaluated saliency map is class-agnostic and is not conditioned on an individual detection. In images containing multiple bunches, assigning one shared saliency map to a particular instance would require an arbitrary matching rule and would not measure the stated backbone target. A future box- and class-conditioned detector explanation, such as D-RISE [18], would make instance-level overlap well defined.

5.5. Limitations and Validity Threats

Four limitations bound the interpretation of these findings. First, model-scale and target-layer selection reused the test partition. The layer 6–10 sensitivity analysis shows that layer 8 is the strongest compatible candidate within the recorded window, but it does not remove selection dependence. The study therefore remains a controlled exploratory within-dataset analysis rather than a confirmatory estimate of explanation generalization. Second, the explanatory scope is restricted to pre-neck backbone saliency and does not attribute individual box–class decisions; CAM analysis was also limited to the primary seed-0 checkpoints.

Third, localization uses the union of annotated boxes rather than instance-level assignment, together with a single saliency threshold. The threshold was defined in the shared Colab evaluation functions before the model–explainer runs and was not tuned by configuration. However, it was not preregistered, and no threshold sweep was performed. Threshold-dependent IoU is therefore reported alongside the threshold-free Pointing Game. MACD and IC operate on unmatched image-level maximum confidence rather than object- or class-specific

detector responses. Fourth, external validity is limited to one public FFB dataset without independent field validation, repeated CAM evaluation across seeds, explanation-runtime measurement, or operator-based assessment; latency reflects an unoptimized native PyTorch run on the recorded NVIDIA A100 environment rather than TensorRT or a quantized edge deployment.

These constraints limit the generalizability of the absolute CAM values and rankings. Future work should prespecify model and layer selection on validation data, evaluate a genuinely independent test set, sweep the saliency threshold, repeat CAM analysis across seeds, record activation, gradient, and perturbation statistics, and evaluate box- and class-conditioned detector attribution with instance-level overlap on field imagery.

6. CONCLUSION

This study shows that similar detection accuracy does not imply similar explanation behavior in YOLO-based palm fruit ripeness detection. Under a matched protocol, YOLOv8m, YOLO11m, and YOLO26m achieved repeated-run mean mAP50–95 values of 0.903, 0.899, and 0.901, respectively, with a maximum separation of 0.004. In contrast, Pointing Game and MACD varied by 68.03 and 48.62 percentage points across model–explainer configurations, and the explainer ranking changed across detectors. EigenCAM produced the strongest observed peak-localization and confidence-retention profile for YOLOv8m, whereas bScore-CAM achieved the strongest localization for YOLO11m; no method dominated all evaluated backbones. The descriptive reversal remained evident within the difficult underripe, ripe, and overripe subset, while the recorded layer 6–10 search confirmed layer 8 as the strongest common compatible backbone endpoint. These findings support evaluating the complete explainer–target–layer–model configuration rather than treating a CAM method as an interchangeable visualization step. Because the study audits pre-neck representations and reuses the test partition for selection, its results are exploratory, within-dataset evidence rather than attribution of individual box–class decisions or independent estimates of generalization. Future work should address those boundaries with validation-based selection, independent field data, instrumented representation analysis, and box- and class-conditioned explanations.

ACKNOWLEDGEMENT

This work was supported by the International Islamic University Malaysia through the Recognition in Scholarly Excellence 2025 (RISE) Grant (RISE25-007-0007). The first and second authors also acknowledge support from the Ministry of Higher Education, Science, and Technology of Indonesia (Kemendiktisaintek) under grant numbers 1609/LL4/PG/2026 and 069/LIT07/PPM-LIT/2026. Generative AI tools were used solely to assist with language refinement and proofreading. All AI-assisted content was reviewed, verified, and revised by the authors, who retain full responsibility for the accuracy, originality, and integrity of the work.

REFERENCES

[1] T. S. Gunawan, M. Kartiwi, and A. Jamali, "Drone and deep learning-based instrumentation for palm fruit detection and yield measurement: A Southeast Asian roadmap," *IEEE Instrumentation & Measurement Magazine,* pp. 1-17, 2026, doi: 10.1109/MIM.2026.11224873.

[2] G. K. A. Parveez *et al.*, "Oil palm economic performance in Malaysia and R&D progress in 2024," *Journal of Oil Palm Research,* vol. 37, no. 2, pp. 187-208, 2025, doi: https://doi.org/10.21894/jopr.2025.0031.

[3] T. S. Gunawan, M. Kartiwi, H. Mansor, and N. M. Yusoff, "Palm fruit ripeness detection and classification using various yolov8 models," in *2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2023: IEEE, pp. 193-198.

[4] S. J. Purba, W. F. Tandion, and E. Irwansyah, "Comparison of the latest version of deep learning YOLO model in automatically detecting the ripeness level of oil palm fruit in PTPN IV, North Sumatra, Indonesia," in *2024 International Conference on Information Technology and Computing (ICITCOM)*, 2024: IEEE, pp. 295-300.

[5] J. Y. Goh, Y. Md Yunos, and M. S. Mohamed Ali, "Fresh fruit bunch ripeness classification methods: A review," *Food and Bioprocess Technology,* vol. 18, no. 1, pp. 183-206, 2025.

[6] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. Wan Hasan, "Oil palm fresh fruit bunch ripeness detection methods: A systematic review," *Agriculture,* vol. 13, no. 1, p. 156, 2023.

[7] J. Y. Goh, M. S. Mohamed Ali, Y. Md Yunos, U. U. Sheikh, and M. S. Khan, "Outdoor RGB and Point Cloud Depth Dataset for Palm Oil Fresh Fruit Bunch Ripeness Classification and Localization," *Scientific Data,* vol. 12, no. 1, p. 687, 2025.

[8] P. Chotikawanid, P. Saeleung, Y. Pianroj, S. Jumrat, T. Punvichai, and J. Muangprathub, "Optimizing an Object Detection Algorithm for Detecting Oil Palm Fruit Bunches and Their Ripeness," *Applied Computational Intelligence and Soft Computing,* vol. 2025, no. 1, p. 6263757, 2025.

[9] J. Josdaan, V. C. Tamsil, J. Harefa, and K. Jingga, "Revolutionizing palm oil ripeness classification: Utilizing YOLOv8 for ultra-precise ripeness detection," *Procedia Computer Science,* vol. 245, pp. 700-709, 2024.

[10] L. T. Ramos and A. D. Sappa, "A comprehensive analysis of YOLO architectures for tomato leaf disease identification," *Scientific Reports,* vol. 15, no. 1, p. 26890, 2025.

[11] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction,* vol. 5, no. 4, pp. 1680-1716, 2023.

[12] Y. Zhao *et al.*, "DETRs beat YOLOs on real-time object detection," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024: IEEE, pp. 16965-16974.

[13] M. Tornero-Soria, A.-J. Sánchez-Salmerón, and E. Vendrell Vidal, "Toward a Deeper Understanding of YOLO26: Block-Level Architectural Analysis and Ablation Studies," *Applied Sciences,* vol. 16, no. 13, p. 6758, 2026.

[14] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*, 2018: IEEE, pp. 839-847.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618-626.

[16] M. Bany Muhammad and M. Yeasin, "Eigen-CAM: Visual explanations for deep convolutional neural networks," *SN Computer Science,* vol. 2, no. 1, p. 47, 2021.

[17] H. Wang *et al.*, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 24-25.

[18] V. Petsiuk *et al.*, "Black-box explanation of object detectors via saliency maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11443-11452.

[19] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in Neural Information Processing Systems,* vol. 31, 2018.

[20] A. Andres, A. Martinez-Seras, I. Laña, and J. Del Ser, "On the black-box explainability of object detection models for safe and trustworthy industrial applications," *Results in Engineering,* vol. 24, p. 103498, 2024.

[21] D. G. Pai, M. Balachandra, and R. Kamath, "Explainable AI in agriculture: Review of applications, methodologies, and future directions," *Engineering Research Express,* vol. 7, no. 3, p. 032202, 2025.

[22] Suharjito *et al.*, "Annotated datasets of oil palm fruit bunch piles for ripeness grading using deep learning," *Scientific data,* vol. 10, no. 1, p. 72, 2023.

[23] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "LayerCAM: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing,* vol. 30, pp. 5875-5888, 2021.

[24] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, "Evaluating weakly supervised object localization methods right," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020: IEEE, pp. 3133-3142.

[25] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, 2014: Springer, pp. 740-755.