Computers in Biology and Medicine 196 (2025) 110697

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Hyperparameter optimization of YOLO models for invasive coronary angiography lesion detection and assessment

Mario Pascual-González [b], Ariadna Jiménez-Partinen [a,b], Esteban J. Palomo [a,b,*], Ezequiel López-Rubio [a,b], Almudena Ortega-Gómez [b,c,d,e]

[a] ITIS Software, University of Málaga, Málaga, 29071, Spain
[b] Instituto de Investigación Biomédica de Málaga y Plataforma en Nanomedicina-IBIMA Plataforma BIONAND, Málaga TechPark, Campanillas, 29590, Spain
[c] Department of Endocrinology and Nutrition, Virgen de la Victoria University Hospital, Málaga, 29010, Spain
[d] Centro de Investigación Biomédica en Red de la Fisiopatología de la Obesidad y Nutrición (CIBEROBN), Instituto de Salud Carlos III (ISCIII), Madrid, 28029, Spain
[e] Cardiology and Cardiovascular Surgery Department, Virgen de la Victoria University Hospital, Málaga, 29010, Spain

ARTICLE INFO

Dataset link: https://doi.org/10.17632/p9bpx9ctcv.2, https://doi.org/10.5281/zenodo.7981244

Keywords:
Invasive coronary angiography
Detection
Deep learning
YOLOv8
Hyperparameter optimization

ABSTRACT

Coronary artery disease (CAD) remains the leading cause of mortality, creating an urgent need for reproducible, image-based decision support. Although YOLOv8-based detectors underpin much of today's state-of-the-art stenosis detection, their accuracy is sensitive to dozens of interacting hyperparameters. We therefore perform a systematic study of hyperparameter optimizers for YOLO-style models, pairing YOLOv8 and its Double Coordinate Attention (DCA) variant with three model-based engines: Covariance-Matrix-Adaptation Evolution Strategy (CMA-ES), Tree-structured Parzen Estimator (TPE), and a Gaussian-process sampler; and contrasting them with Random Search and the mutation-only routine that serves as the default optimizer in the `ultralytics` package.

Optimization targets binary detection and was benchmarked using the CADICA (full sequences) and ARCADE (single key-frames) datasets, maximizing the F1-Score under stratified three-fold cross-validation within a fixed compute budget. Model-based methods consistently lift the F1-speed Pareto frontier: CMA-ES attains 0.355 ± 0.079 on *v8l*, while Bayesian strategies top the medium and small backbones with 0.346 ± 0.048 (*v8m*) and 0.304 ± 0.054 (*v8s*). All surpass the default optimizer and yield more lesion-centric EigenCAM saliency, confirming the value of adaptive probabilistic search for tuning high-dimensional YOLO-based CAD pipelines. The complete code-base is open-source and released at https://github.com/MarioPasc/Coronary_Angiography_Detection.

1. Introduction

Coronary artery disease (CAD), characterized by the obstruction of myocardial arteries, is the leading cause of death worldwide [1–3]. These coronary artery obstructions, or stenosis, can be categorized by their percentage of narrowing; common categories are mild ($< 50\%$), moderate ($[50\%, 70\%]$), and severe $> 70\%$, also named obstructive lesion, where a lesion 100% imply a total occlusion of the artery. Although non-invasive methods exist, invasive coronary angiography (ICA), also known as X-ray coronary angiogram (XCA) or X-ray angiography (XRA), remains the gold standard method for anatomical imaging diagnosis and treatment when CAD is suspected [4]. The ICA acquisition protocol involves the insertion of a radiocontrast agent through a catheter by a percutaneous incision and X-ray imaging to visualize coronary arteries. To evaluate the complete state of myocardial vessels, the left coronary artery (LCA) and the right coronary artery (RCA) have to be inspected by different projections and angles, such as the right and left anterior obliques, with cranial and caudal angulation [5]. The identification of abnormalities and severity assessment are commonly determined visually by clinical experts. Although this protocol is standard clinical practice, it may still suffer from inter-observer and intra-observer variability due to subjectivity.

Computer-aided diagnosis based on deep learning (DL) techniques, as decision-making support, could lead to more efficient evaluation with more reliable assessments and reduced workload.

DL applications for interpreting ICA images have been limited because of the shortage of open-access datasets, which is the main drawback. ICA image state-of-the-art (SOTA) is focused on applying convolutional neural network (CNN) architectures across diverse computer

* Corresponding author at: ITIS Software, University of Málaga, Málaga, 29071, Spain.
E-mail address: ejpalomo@uma.es (E.J. Palomo).

https://doi.org/10.1016/j.compbiomed.2025.110697
Received 13 February 2025; Received in revised form 26 June 2025; Accepted 30 June 2025
Available online 24 July 2025

vision tasks [6,7]. Deep learning-based solutions for ICAs are challenging because of the poor signal-to-noise ratio, complex and overlapped vessel structures, non-uniform illumination, and inhomogeneous lumen intensity. The CNN applications have mainly focused on artery extraction through segmentation-based methods [8–11].

Focusing only on the stenosis detection task, only a few methods address the development of automatic lesion detection. For instance, a three-stage system was proposed by Wu et al. [12], where firstly, the U-Net network selected from 148 ICA videos the contrast-filled frames. In the second step, a deconvolutional single-shot multi-box model detected promising lesion bounding boxes from those key-frames. Finally, a customized "sequence-false positive suppression" module filtered the predicted bounding boxes corresponding to false positives, considering the comparing frames of the same sequence. The proposed method achieves an F_1-score of 83.2%, sensitivity of 87.2%, and precision of 79.5%.

Rodrigues et al. [13] used a two-stage framework to locate stenosis. First, a ResNet-50 architecture was used to classify between RCA and LCA images. Next, the RetinaNet model was trained to determine the bounding boxes of the regions of interest in frames where stenosis is visible. The first step attained an F_1-score of 96% and the second a recall of 68% and 73% for LCA and RCA, respectively.

Han et al. [14] proposed a stenosis detection framework based on a Transformer-based module to aggregate proposal-level spatiotemporal features. Experiments on 233 ICA sequences achieved a high F_1-score of 90.88%, sensitivity of 89.56%, and Precision of 92.27%.

Wang et al. [15] develop an end-to-end YOLOv5-based system to detect lesions in RCA images classifying them into $[25\% - 70\%]$ or $[70\% - 100\%]$. The study was carried out using 201 RCA sequences, and the outcomes attained a recall rate of 85.4%, an F_1-score of 87.1%, and a mAP value of 87.5%.

Despite the good performance achieved, the proposed detection solutions in the state-of-the-art for ICA images have limitations that imply unfair comparison and difficult reproducibility, such as private datasets, unavailable code, or lack of image information, as patient-level partition implemented, lesion degree considered, or multi-lesion frames inclusion. Additionally, there are no universal default parameters that ensure optimal results for every research task. Default settings may not suit all image detection problems, and manually selecting hyperparameters can be difficult and suboptimal, but many of the DL approaches underreport a comprehensive hyperparameter exploration [16]. This is where hyperparameter optimization (HPO) algorithms become valuable, as they systematically explore the hyperparameter space and typically yield better results than manual selection. Most popular HPO algorithms are Tree-structure Parzen estimator (TPE), harmony search, differential evolution (DE), genetic algorithm (GA), grid search (GS), random search (RS), or Bayesian optimization (BO) [17].

The work of Wojciukis et al. [18] focused on HPO in transfer learning to understand how different methods influence the performance. They evaluated key hyperparameters and their optimal ranges for training CNNs, comparing four optimization methods: GS, RS, BO, and the Asynchronous Successive Halving Algorithm (ASHA). The experiments conducted demonstrated that using ASHA or BO techniques can effectively boost the performance accuracy of CNN models, although it depends on the specific task and dataset. For instance, Ramos et al. [19] proposed the YOLOv8 architecture for smoke and fire detection, using a two-stage HPO process. First, the One Factor At a Time (OFAT) methodology was applied to focus on key parameters such as learning rate and batch size. Insights from this led to an RS, resulting in a model that significantly outperformed the initial fine-tuned version, with improvements of 1.39% in precision, 1.48% in recall, and 1.44% in F1-score. Wahyudi et al. [20] studied the improvement of the Yolov7 model performance on MRI brain tumor detection, exploring three HPO methods: RS, GA, and BO. Their findings show that BO was the most efficient, improving YOLOv7's performance by 8%, being 1.5 times faster than RS and 4 times faster than GA.

This work aims to report a reproducible and available HPOframework for YOLOv8 architecture to locate $> 50\%$ lesions using the open-access CADICA [21] and ARCADE [22] datasets to approach a benchmark as close to clinical settings as possible. The rest of the paper is organized as follows: Section 2 describes in detail both the source datasets and the methodology addressed to tackle our problem. Several experimental setups and results are reported in Section 3. Section 4 is devoted to discussing and highlighting findings. Finally, Section 5 outlines the conclusions of the paper.

Table 1
Distribution of the CADICA dataset: number of frames and bounding boxes from *lesion* frames.

			Total
Images	Non-lesion	2,130	6,126
	Lesion	3,996	
Bounding boxes	$< 20\%$	1944	6,161
	$[20\%, 50\%)$	1128	
	$[50\%, 70\%)$	999	
	$[70\%, 90\%)$	893	
	$[90\%, 98\%]$	930	
	99%	63	
	100%	204	

2. Methodology

2.1. Source data

Two open-access annotated datasets – CADICA [21] and ARCADE [22] – were chosen to reliably study the impact of hyperparameter optimization approaches for narrowing detection with DL-driven methods. Both datasets are composed of enough ICA frames from different projections of LCA and RCA, but they also differ. ARCADE reports a single key-frame per patient, while the CADICA dataset is composed of complete videos with a performed frame selection and zero or more bounding boxes per frame. However, these differences allow for a wider study scope for the establishment of an approach closer to clinical settings as a baseline framework and a starting point to improve stenosis detection in ICA images.

2.1.1. Dataset I: CADICA

The CADICA dataset [21] is composed of invasive coronary angiography frames from 42 patients. Each patient case contains videos selected for CAD protocol inspection. Different projections for LCA and RCA, depending on the diagnostic case and its difficulties, are included. The dataset incorporates a frame selection where the radiocontrast was perfused correctly or the lesion was discernible. ICA images are in PNG format with a size of 512×512 pixels.

The frame annotation consists of locating and delimiting the region of interest with bounding boxes that are categorized according to their clinical diagnosis. The categories correspond to the lesion degree range, which depends on the narrowing of the vessel: $< 20\%$, $[20\%, 50\%)$, $[50\%, 70\%)$, $[70\%, 90\%)$, $[90\%, 98\%]$, 99%, and 100%.

To carry out the study using a supervised binary approach, frames without lesions were chosen as *non-lesion* and with bounding boxes, independently of lesion degree, as *lesion* samples. Therefore, the model must detect all such boxes regardless of their exact sub-range. Three samples are illustrated in Fig. 1. A total of 6126 frames and 6161 bounding boxes were used, whose distribution is detailed in Table 1. Notice that the number of labels is higher than the number of *lesion* images since a frame may have zero or more labels. Considering LCA and RCA projections and a wide range of lesions as positive instances became more complex the detection task.

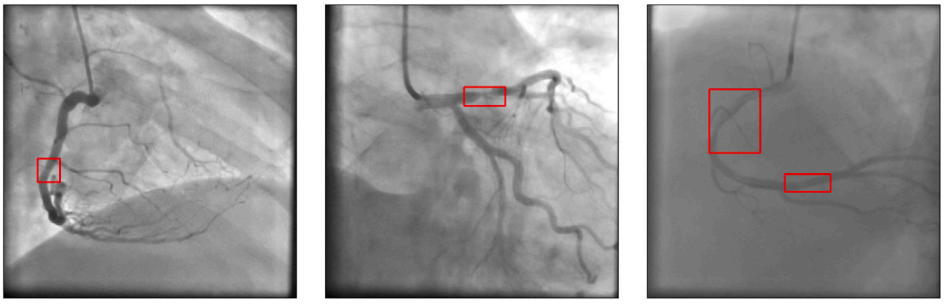

Fig. 1. CADICA dataset samples with bounding boxes.

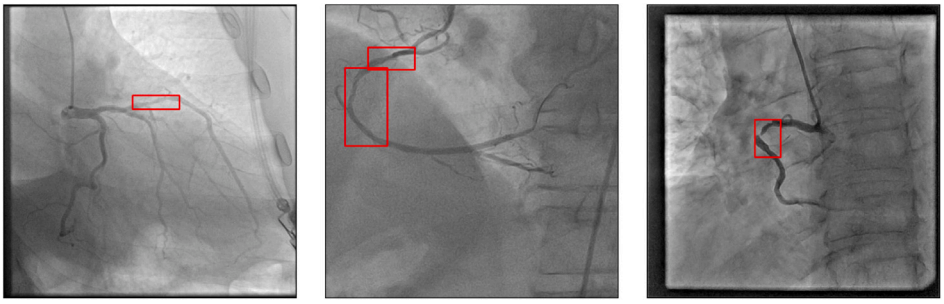

Fig. 2. ARCADE dataset samples with bounding boxes.

Table 2
Distribution of the ARCADE dataset: number of frames and bounding boxes from *lesion* frames.

		Total
Images	Lesion	1,500
Bounding boxes	[50%, 99%]	2,417

2.1.2. Dataset II: ARCADE

Automatic Region-based Coronary Artery Disease Diagnostics using X-ray angiography images (ARCADE) [22], where 3,0000 invasive coronary angiographies are annotated in order to tackle two different tasks: arterial tree segmentation according to syntax score, and stenosis detection. In the present work, only the 1500 stenosis task frames were considered. These ICAs are characterized by being in PNG format with a size of 512 × 512 pixels, coming from different projections, and present > 50% lesion. However, this dataset lacks *non-lesion* frames. Table 2 showcases dataset size and lesion severity range covered in the ARCADE dataset. Three samples are shown in Fig. 2.

2.2. Convolutional neural networks

Convolutional Neural Networks (CNNs) serve as the primary tool for detecting lesions in ICA images. CNNs are a type of Deep Learning architecture that includes at least one convolutional layer for feature extraction from images. As network depth increases, these features become more complex, enabling the model to capture intricate patterns in the data [23]. CNNs are particularly effective when combined with transfer learning, where a network initially trained on a large dataset is fine-tuned on a different dataset using the pre-trained weights obtained to address a specific problem [24]. The architecture chosen to conduct the study belongs to the YOLO (You Only Look Once) family as it is a SOTA benchmark framework for DL-driven tasks with ICA images [15,25,26], although there is a gap in the application of hyperparameter optimization algorithms.

2.2.1. You Only Look Once (YOLO)

YOLO (You Only Look Once) is a family of Deep Learning-based methods designed for real-time object detection, striking a balance between accuracy and speed [27]. Unlike Deformable Parts Models (DPM) [28], which rely on sliding windows and part-based filters, or Region-Based CNN (R-CNN) [29], which separates region proposal, feature extraction, and classification into distinct stages, YOLO performs detection in a single pass. The network divides the image into an $S \times S$ grid, where each cell predicts bounding-box parameters, confidence scores (indicating the likelihood of an object), and class probabilities. The cells do not act independently; global features extracted from the entire image inform their predictions. These outputs are merged to yield final object detections. As a post-processing step, Non-Maximum Suppression (NMS) discards overlapping boxes and retains only the most confident detections to avoid redundancy.

In 2023, Ultralytics – creators of the first Python-based YOLO model, YOLOv5 [30] – introduced YOLOv8, a state-of-the-art architecture for object detection [31]. Pre-trained on the COCO dataset [32], YOLOv8 demonstrated notable advancements in mean Average Precision (mAP) and computational efficiency compared to its predecessors. In this study, three out of five backbone sizes for YOLOv8 were explored, the *s*, *m*, and *l* sizes. Key innovations include an anchor-free design for streamlined detection, the C2f module for improved feature aggregation and gradient flow, and Spatial Pyramid Pooling Fast (SPPF) for robust multi-scale detection [33]. YOLOv8 uses three primary loss terms to guide its predictions, where each loss component targets a distinct aspect of detection performance:

- **Box Loss** $\mathcal{L}_{\text{box}}$: this term enforces accurate localization of objects. Includes a Complete IoU (CIoU) to measure how closely the predicted box $\hat{b}_{x,y}$ aligns with the ground-truth box $b_{x,y}$:

$$\mathcal{L}_{\text{box}} = \sum_{x,y} \mathbf{1}_{c^*_{x,y}} \left(1 - q_{x,y} + \|\hat{b}_{x,y} - b_{x,y}\|_2^2 / \rho^2 + \alpha_{x,y}\, v_{x,y}\right) \quad (1)$$

  where $q_{x,y}$ is an IoU-based term, CIoU, ρ is the diagonal length of the enclosing box covering predictions and ground truth, and $\alpha_{x,y}\, v_{x,y}$ optionally penalizes angular deviations [34].
- **Classification Loss** $\mathcal{L}_{\text{cls}}$: this penalty ensures each cell correctly classifies whether it contains an object (and, if so, which class). YOLOv8 uses binary cross-entropy, allowing multi-label outputs when multiple classes may appear in the same region:

$$\mathcal{L}_{\text{cls}} = \sum_{x,y} \sum_{c \in \text{classes}} \left[y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c) \right] \quad (2)$$

- **Distribution Focal Loss (DFL)** $\mathcal{L}_{\text{dfl}}$: instead of merely penalizing misclassifications of objectness, DFL transforms bounding-box localization into a discrete distribution problem. Each coordinate dimension is modeled by multiple "bins", and the network must learn the probability distribution over these bins. By assigning higher weights to uncertain or challenging samples, DFL improves localization and stability:

$$\mathcal{L}_{\text{dfl}} = \sum_{x,y} \mathbf{1}_{c^*_{x,y}} \left[-\cdots \log(\hat{q}_{x,y}) + \cdots \right] \tag{3}$$

where $\hat{q}_{x,y}$ is the predicted probability distribution over possible coordinate locations [35].

2.2.2. Double coordinate attention (DCA-YOLO)

In order to conduct our study within a comprehensive scope for Yolov8-based methods, the proposed framework by Duan et al. [36] is used. Duan's proposal, called DCA-YOLOv8, is focused on maximizing ICA feature extraction by incorporating a double coordinate attention (DCA) feature extraction module before the C2f module, which is part of the feature extraction module and neck feature fusion module of the YOLOv8 backbone. The DCA module addition enhances the extraction of stenosis features by placing greater emphasis on the positional information of stenosis in blood vessels within the images. Additionally, the final module was updated, replacing the baseline loss function CIoU with the AICI function loss. This change was specifically designed to consider the characteristics of coronary artery stenosis, making it more suitable for detecting and performing bounding box regression on small targets in the box localization loss function.

2.3. Hyperparameter search

Hyperparameters are the external parameters of a neural network, whose values are fixed *before* training begins and remain unchanged during gradient-based learning. Formally, let $\mathcal{H} \subset \mathbb{R}^d$ denote the d-dimensional hyperparameter space and let $g(\theta, h\,; \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}) \in \mathbb{R}$ be a scalar performance metric (here the F_1-score) obtained after training the model parameters θ with a configuration $h \in \mathcal{H}$. The hyperparameter optimization problem (HPO) can therefore be written as the black-box optimization task

$$h^* = \arg\max_{h \in \mathcal{H}} f(h), \qquad f(h) = g\big(\theta^\star(h), h\big), \tag{4}$$

where $\theta^\star(h) = \arg\min_\theta \mathcal{L}_{\text{train}}(\theta, h)$ is obtained by empirical risk minimization on $\mathcal{D}_{\text{train}}$. Because a single evaluation of $f(h)$ entails a costly end-to-end YOLOv8 training run, the HPO strategy must balance exploration of $\mathcal{H}$ and exploitation of already promising regions.

Throughout this work we consider three algorithmic families for solving (4): (i) *Random Search* (Section 2.3.1), a baseline that probes the space uniformly at random; (ii) *Bayesian Optimization* (Section 2.3.2), which leverages probabilistic surrogate models to guide sampling; and (iii) *Evolutionary Strategies* (Section 2.3.3), population-based heuristics inspired by natural evolution. Each offers a different trade-off between computational overhead and sample-efficiency, and their comparative behavior on stenosis detection constitutes the experimental core of this study.

2.3.1. Random search

Random Search (RS) samples hyperparameter vectors independently from a user-defined prior $p(h)$, usually factorized and uniform inside practical bounds [37]. For a fixed computational budget of T trials, the algorithm generates $\{h_t\}_{t=1}^{T} \sim p(h)$ and returns $h^* = \arg\max_t f(h_t)$. Reproducibility is ensured by initializing the pseudo-random number generator with a fixed seed s (i.e. rng $\leftarrow$ `Seed`(s)), so that $\{h_t\}$ and their associated performances can be exactly revisited.

2.3.2. Bayesian optimization

Bayesian Hyperparameter Optimization (BHO) is an iterative approach for optimizing black-box objective functions $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the hyperparameter space and $f(x)$ measures the performance of a given configuration x under the assumption that direct evaluations of $f(x)$ are computationally costly [38]. The key elements of Bayesian optimization are: (i) a surrogate model, used as a computationally efficient proxy for $f(x)$, predicting the performance of untested configurations based on previous evaluations, and (ii) an acquisition function, which uses the posterior distribution of $f(x)$ (updated after each observation) to determine the most promising hyperparameter candidates to sample next. Bayesian optimization aims to solve

$$\max_{x \in \mathcal{X}} f(x) \tag{5}$$

Unlike grid or random search, where each hyperparameter set is sampled blindly from the search space, Bayesian optimization updates a probabilistic model of the objective function at every iteration based on past observations, thereby informing the selection of future query points.

In this study, the validation F_1-score is the objective, and the *Expected Improvement* (EI) acquisition is employed to optimize its value,

$$\alpha_{\text{EI}}(x) = \mathbb{E}\big[\max\big(f(x) - f_{\text{best}}, 0\big)\big] = \Delta(x)\,\Phi\Big(\frac{\Delta(x)}{s(x)}\Big) + s(x)\,\phi\Big(\frac{\Delta(x)}{s(x)}\Big), \tag{6}$$

where $\Delta(x) = \mu(x) - f_{\text{best}}$, $\mu(x)$ and $s(x)$ are the posterior mean and standard deviation of the surrogate, and $\Phi(\cdot)$ and $\phi(\cdot)$ denote the standard-normal cdf and pdf, respectively. The next configuration is therefore obtained via

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha_{\text{EI}}(x). \tag{7}$$

The objective function $f(x^*)$ is then evaluated at the chosen configuration x^*, and the result is used to update the surrogate model. This process continues until the stopping criterion, determined by the computational budget and empirical signs of saturation, is reached. Employing the Optuna framework for BHO [39], two specific models were implemented to fine-tune the YOLOv8 hyperparameters:

Gaussian process (GPSAMPLER). In this approach, a Gaussian Process (GP) is used to model the objective function $f(x)$, which in our study corresponds to the F_1-score. Formally, a GP imposes a prior over functions:

$$f(x) \sim \mathcal{GP}\big(\mu(x),\, k\big(x, x'\big)\big), \tag{8}$$

where $\mu(x)$ is the mean function and $k(x, x')$ is the covariance (kernel) function. Here, we employ the Matérn kernel with $\nu = 2.5$ to balance smoothness and flexibility, and we also use Automatic Relevance Determination (ARD) to assign a distinct length scale ρ_i to each hyperparameter dimension. This strategy helps capture the individual importance of each hyperparameter, preventing one dimension from dominating the search.

At each iteration, the GP posterior distribution is updated based on the evaluated hyperparameter configurations $(x_i, f(x_i))$. The algorithm then seeks the next point x_{next} to sample by maximizing an acquisition function $a(x)$. In this study, we use the logarithm of the Expected Improvement (logEI). Mathematically, if the current best value of the F_1-score is f_{best}, the improvement $I(x)$ at a new point x can be defined as $I(x) = \max\big(f(x) - f_{\text{best}}, 0\big)$. We consider the logarithmic form to stabilize the optimization process; therefore, the expected improvement is then

$$\log(\text{EI}(x)) = \log\Big(\mathbb{E}[I(x)]\Big) \tag{9}$$

The optimization of this objective function is done via Quasi-Monte Carlo (QMC) sampling.

Tree-structured Parzen Estimator (TPE). TPE follows a different model-based philosophy for HPO by approximating two conditional likelihood terms, $p(x \mid y < y^*)$ and $p(x \mid y \geq y^*)$, where y refers to the observed F_1-score and y^* is a threshold (quantile of the best scores). Rather than directly modeling $p(y \mid x)$, TPE transforms the problem into modeling the densities $p(x \mid y)$ and $p(y)$. Specifically, TPE divides the evaluated hyperparameter configurations into two sets: one containing "good" F_1-scores ($y \geq y^*$), and one containing the rest ($y < y^*$). Two Gaussian Mixture Models (GMMs) – $\ell(x)$ for the good set and $g(x)$ for the other – approximate these densities. The method then proposes new samples by maximizing the ratio

$$\frac{\ell(x)}{g(x)} \tag{10}$$

which focuses on searching around hyperparameters likely to deliver better results. In our study, we configure TPE with a Gaussian prior to stabilize its Parzen estimators, apply heuristic variance clipping for small variances, and employ a multivariate approach to capture interdependencies among hyperparameters. The optimization begins with a preliminary random search phase, after which TPE constructs and refines the GMMs as data accumulates, thereby guiding the search toward more promising regions of the hyperparameter space.

2.3.3. Evolutionary Strategies (ES)

Evolutionary Strategies form a class of derivative-free optimizers that maintain *populations* of candidate solutions and iteratively refine them via stochastic variation and fitness-based selection [40]. At generation t a population $\mathcal{P}_t = \{h_k^{(t)}\}_{k=1}^{\lambda}$ of λ individuals is available. Each individual is evaluated to obtain the fitness vector $\mathbf{f}_t = \left(f(h_1^{(t)}), \ldots, f(h_\lambda^{(t)})\right)$. The $\mu \leq \lambda$ fittest parents are selected, and offspring are produced by Gaussian mutations,

$$h_k^{(t+1)} = m_t + \sigma_t\, \varepsilon_k^{(t)}, \qquad \varepsilon_k^{(t)} \sim \mathcal{N}(0, C_t),\ k = 1, \ldots, \lambda, \tag{11}$$

where m_t is the current search center (e.g., mean of the parent set), $\sigma_t > 0$ the global step-size and $C_t \in \mathbb{R}^{d\times d}$ a positive-definite covariance matrix modeling search directions. The update of (m_t, σ_t, C_t) distinguishes different ES variants and governs the exploration–exploitation balance.

In a (μ, λ)-ES the next generation is *entirely* composed of offspring, while a $(\mu + \lambda)$-ES allows parents to survive if they outperform their descendants, providing additional stability. Adaptive step-size control can be achieved through cumulative path length or success rules, whereas covariance adaptation learns the principal axes of the objective landscape.

Covariance matrix adaptation (CMAES). CMA-ES is a state-of-the-art member of the (μ, λ)-ES family that augments the basic Gaussian mutation model in (11) with two self-adaptive mechanisms: (i) cumulative path–length control for the global step size σ_t and (ii) rank-$\{1, \mu\}$ updates of the full covariance matrix C_t [41]. At generation t the search distribution is the multivariate normal

$$h \sim \mathcal{N}\left(m_t,\ \sigma_t^2 C_t\right), \qquad C_t \in \mathbb{R}^{d\times d} \tag{12}$$

from which λ offspring are sampled and evaluated. The weighted mean of the best μ individuals, $m_{t+1} = \sum_{k=1}^{\mu} w_k\, h_{k:\lambda}^{(t)}$ with positive weights $\sum_k w_k = 1$, becomes the new search center. A "conjugate evolution path" p_σ accumulates successive steps in a whitened coordinate system and drives the step-size update

$$\sigma_{t+1} = \sigma_t \exp\!\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{\mathbb{E}\|\mathcal{N}(0,I)\|} - 1\right)\right), \tag{13}$$

thereby realizing the Cumulative Step-Size Adaptation (CSA). In parallel, a covariance evolution path p_c is maintained and combined with the rank-μ outer products of normalized steps $y_k = (h_{k:\lambda}^{(t)} - m_t)/\sigma_t$ to yield

$$C_{t+1} = (1 - c_1 - c_\mu) C_t + c_1\, p_c p_c^{\top} + c_\mu \sum_{k=1}^{\mu} w_k\, y_k y_k^{\top}, \tag{14}$$

allowing the algorithm to learn the principal axes and scale of the objective landscape without gradient information.

The resulting search distribution progressively rotates and contracts toward valleys of high fitness. Convergence to sub-optimal basins is mitigated by restart heuristics – most notably *bipop-CMA-ES* – which alternates between increasingly large and small population sizes to broaden the exploration radius and escape local optima.

In this study, CMA-ES is instantiated through the Optuna framework [39]. Our configuration follows classical design rules: the initial population size $\lambda = 4 + \lfloor 3 \ln d \rfloor$; a single-generation bootstrap (`n_startup_trials` = λ); and a `bipop` restart strategy with doubling factor `inc_popsize` = 2. The algorithm therefore adapts (m_t, σ_t, C_t) across generations exactly as formalized above, using the validation F_1-score as fitness and terminating once the computational budget is exhausted.

2.3.3.1. Ultralytics optimization framework (Ultralytics-ES).

In its tuning module, Ultralytics offers a mutation-based algorithm, i.e., evolution strategy (ES), for hyperparameter optimization. This method stochastically modifies hyperparameter values from previously evaluated configurations and measures their performance at each iteration. A parent configuration is selected from a small set of top-scoring solutions (based on weighted fitness), giving preference to higher-performing hyperparameter sets. In each iteration, only one new child is generated via mutation, making the process closely resemble a $(1+\lambda)$-ES evolutionary approach, where λ is the number of potential parents considered for weighted selection. Specifically, each hyperparameter x_i undergoes mutation according to

$$x_i' = x_i \cdot \Big(1 + \sigma \cdot \mathrm{randn} \cdot \mathrm{rand}\Big), \tag{15}$$

where x_i is the parent value, σ the mutation scale, `randn` a Gaussian noise term, and `rand` an additional randomness factor. An internal probability check controls whether each dimension actually mutates (i.e., some hyperparameters may remain unchanged), ensuring dimension-wise variation. This procedure is repeated until at least one hyperparameter mutates, which the code enforces by checking if all factors remain at 1.0. The final mutated values are clipped within user-specified bounds to maintain feasibility.

Although Ultralytics refers to this approach as a genetic algorithm, it lacks crossover operations, or does not include a temperature schedule or acceptance of worse solutions (as in $(1 + \lambda)$-ES), and relies solely on mutation-driven exploration. This design shares fundamental principles with simple evolution strategies: in each generation, the best-performing solutions are selectively chosen, mutated, and re-evaluated. Fitness is determined by computing the F_1-score of the trained model on the validation set, and the search process continues for a fixed number of iterations. The algorithm finally returns the best configuration $\arg\max_{x\in\mathcal{X}} f(x)$, where $f(x)$ is the fitness function over the search space $\mathcal{X}$. The Ultralytics-ES strategy is the standard tuning strategy for the YOLO family, therefore, we consider it as the main competitor of this study.

2.4. Performance metrics

Performance in CNN-based object detection is commonly assessed using Precision, Recall, and the F_1-score. *Precision* measures the fraction of predicted positives that are truly positive, quantifying how many detected objects are correct:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{16}$$

where TP represents true positives and FP indicates false positives. *Recall* captures the model's ability to retrieve all relevant objects:

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{17}$$

with FN being false negatives (i.e., missed detections). The *F_1-score* is then defined as the harmonic mean of Precision and Recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{18}$$

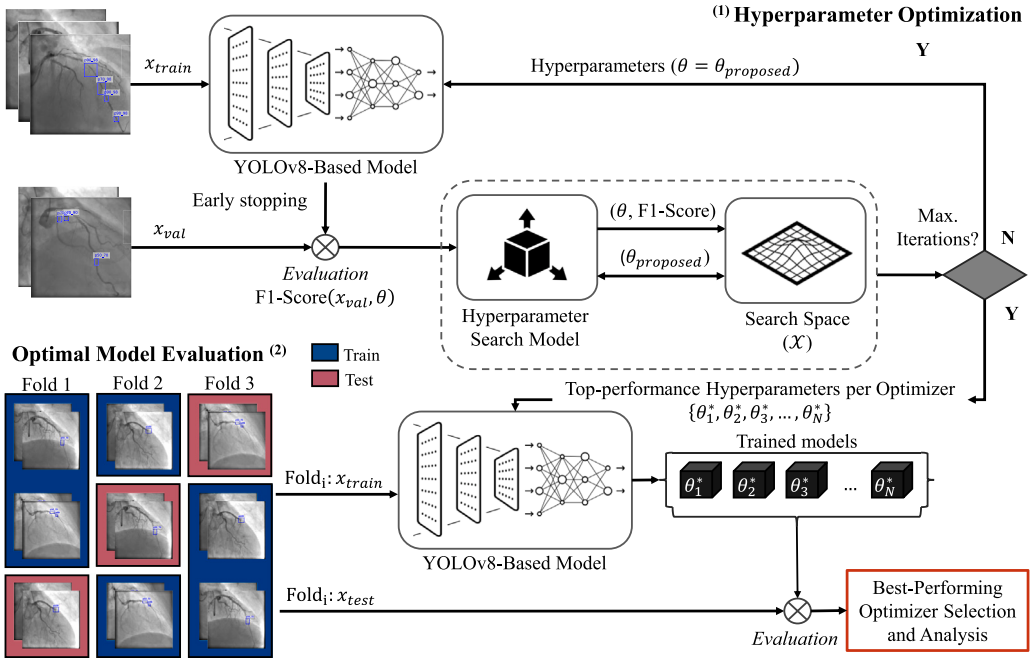

Fig. 3. Hyperparameter Optimization for YOLOv8-based models. The invasive coronariography angiography dataset is partitioned into training and validation sets. The model is configured to detect lesions (*lesion* vs. *non-lesion*). Each of the five optimizers (Covariance Matrix Adaptation, Tree-structured Parzen Estimator, Gaussian process-based algorithm, Random search, and Ultralytics-ES) proposes hyperparameter configurations aimed at maximizing the F_1-score until a maximum number of iterations is reached, with each trial being pruned using early stopping. The optimal configuration from each optimizer is then evaluated using 3-fold stratified cross-validation, yielding the overall performance for the studied dataset.

where all three metrics range from 0 to 1, and the higher the better. In this study, a detection is considered positive if its *Intersection over Union* (IoU) with the ground-truth bounding box exceeds 0.5, a common threshold in the literature [12,14]. The IoU is computed as

$$\text{IoU} = \frac{\text{area}(B_{\text{pred}} \cap B_{\text{GT}})}{\text{area}(B_{\text{pred}} \cup B_{\text{GT}})},$$

where B_{pred} and B_{GT} are the predicted and ground-truth bounding boxes, respectively.

Optimizers will maximize the F_1-score metric since it is a common performance metric for this purpose [14]

3. Results

The proposed HPO framework has been evaluated by several experiments, whose results have been analyzed. The following subsections are devoted to describing them in detail.

3.1. Experimental setup

Fig. 3 illustrates the HPO workflow proposed in this study. The task is a binary detection problem, where each frame is labeled either as *lesion* or *non-lesion*. The model aims to locate and enclose the lesion region in a bounding box.

Following a patient-level split of the dataset into training (67%) and validation (33%) sets, each model underwent 100 trials to maximize the F_1-score on the validation subset. In order to minimize the risk of data leakage, frames were grouped based on patient ID. The explored hyperparameters included batch size, optimizer choice (from the Adam family), initial and final learning rates, momentum, weight decay, warm-up epochs, warm-up momentum, and coefficients for the three YOLOv8 loss terms. Each sub-loss is scaled by a coefficient λ and normalized by the total number of positive cells, N_{pos}. Formally, the generalized YOLOv8 loss can be expressed as:

$$\mathcal{L}(\theta) = \frac{\lambda_{\text{box}}}{N_{\text{pos}}} \mathcal{L}_{\text{box}}(\theta) + \frac{\lambda_{\text{cls}}}{N_{\text{pos}}} \mathcal{L}_{\text{cls}}(\theta) + \frac{\lambda_{\text{dfl}}}{N_{\text{pos}}} \mathcal{L}_{\text{dfl}}(\theta) + \phi \frac{\|\theta\|^2}{2}, \quad (19)$$

where λ_{box}, λ_{cls}, and λ_{dfl} are hyperparameters that modulate the relative weight of box (`box`), classification (`cls`), and distribution focal losses (`dfl`), respectively. The term $\phi \frac{\|\theta\|^2}{2}$ represents the L2 regularization (weight decay).

Each trial ran for 1000 epochs with a patience of 10 epochs to mitigate overfitting, which is likely under data scarcity. Moreover, all data augmentation methods, including those provided by Ultralytics-ES, were disabled. A detection is considered positive if its Intersection over Union (IoU) with the ground-truth box exceeds 0.5, a threshold commonly used in the literature [12,14]. The representative F_1-score of each trial was the highest performance obtained across all epochs. The best-performing hyperparameter configuration underwent a stratified k-fold cross-validation to obtain a final honest evaluation for the model.

Table 3 summarizes the final hyperparameter ranges of our search procedure. In this context, `loguniform` samples values from the logarithmic domain, `uniform` samples them linearly, `int` draws integer values, and parameters enclosed in braces (e.g., $\{2, 4, 8, 16, 32\}$) reflect categorical sampling from discrete sets. Although all listed parameters were optimized under the Bayesian approaches described in Section 2.3.2, the batch size and optimizer choice were excluded from the Ultralytics-ES optimizer, since this algorithm cannot handle categorical hyperparameter, (Section 2.3.3.1) and fixed to 16 and `Adam`, respectively. Initially, broader ranges were used, then narrowed once intermediate experiments demonstrated more efficient exploration within smaller parameter spaces. Additionally, while the Bayesian methods treated `warmup_epochs` as an integer, the Ultralytics-ES algorithm handled all hyperparameters as floats, including `warmup_epochs`.

All experiments were executed on the Picasso super-computing cluster. Each job was assigned three NVIDIA A100 accelerators, amounting

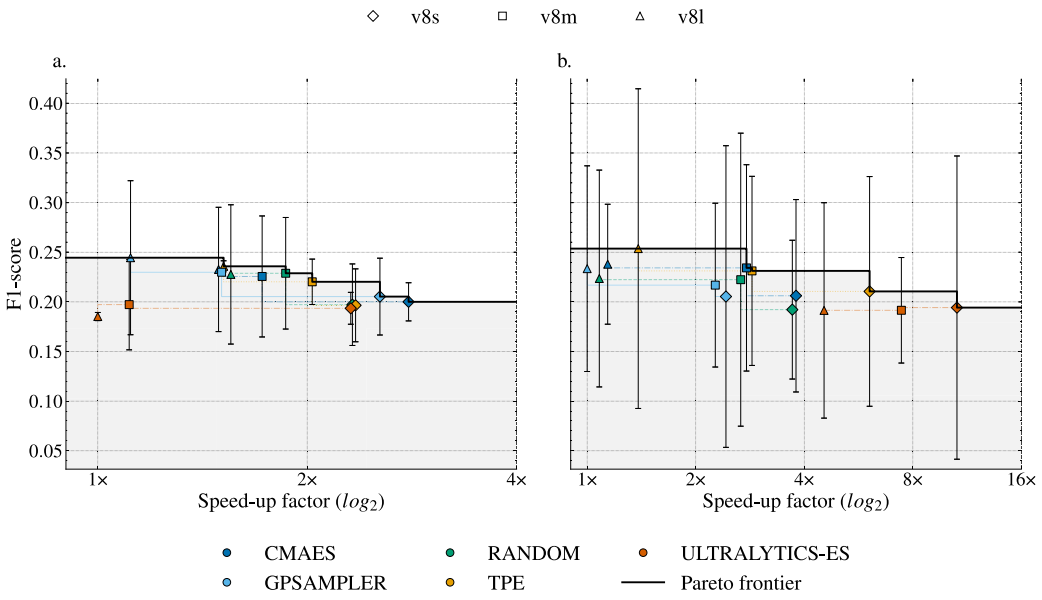

Fig. 4. Trade-off between optimization efficiency and detection accuracy for YOLOv8 (panel a) and DCA-YOLOv8 (panel b) for the CADICA dataset. Each marker corresponds to the best trial returned by one hyperparameter–search strategy (color) on one model size (shape). The shape is the maximum F1 observed during the hyperparameter search. Vertical bars show the 95% confidence interval (CI) obtained by re-training the model with the same hyperparameters under a 3-fold cross-validation protocol.

Table 3
Hyperparameter Tuning Setup.

Hyperparameter	Default	Explored
`optimizer`	`auto`	{Adam, AdamW, NAdam, RAdam}
`batch`	16	{2, 4, 8, 16, 32}
`lr0`	0.01	`loguniform`([1e−5, 5e−3])
`lrf`	0.01	`loguniform`([1e−5, 5e−3])
`momentum`	0.937	`uniform`([0.65, 0.8])
`weight_decay`	0.0005	`loguniform`([1e−5, 5e−2])
`warmup_epochs`	3.0	`int`([5, 10])
`warmup_momentum`	0.8	`uniform`([0.8, 0.99])
`box`	7.5	`uniform`([6.0, 8.5])
`cls`	0.5	`uniform`([0.65, 0.9])
`dfl`	1.5	`uniform`([0.2, 3.5])

to ≈ 13 k Tensor Cores, ≈ 110 k FP32 CUDA cores and a combined 1.28 TB of HBM2e memory. Three hyperparameter trials were launched concurrently for every optimizer.

3.2. Dataset I

This section details the qualitative and quantitative results obtained for both YOLOv8, and the SOTA model, DCA-YOLOv8, after applying the proposed hyperparameter search, for the CADICA dataset.

3.2.1. Performance

Fig. 4 plots the best validation F1–Score achieved by each optimizer against its corresponding training speed-up, where the abscissa is shown on a $\log_2$ scale; therefore, one tick represents a twofold reduction in wall-clock time. Panel **a** reports results for the canonical YOLOv8 detector, while Panel **b** repeats the analysis for DCA-YOLOv8.

Across both panels, model-based algorithms jointly raise the Pareto frontier relative to the mutation-only Ultralytics-ES baseline, delivering higher F1 scores at equal or greater speed-ups. The magnitude of this advantage depends on network capacity: lightweight *v8s* models achieve the largest throughput gains and exhibit the narrowest 95% confidence intervals, whereas the medium and large backbones (*v8m* and *v8l*) trade computational efficiency for peak accuracy and show greater variability across patient splits. Replacing YOLOv8 with DCA-YOLOv8 shifts the entire frontier upward, allowing F1 scores approaching 0.25 to be attained at four-fold speed-ups.

3.2.2. Hyperparameter convergence

Fig. 5 provides a compact view of the search landscape explored by the five optimizers. Each row corresponds to an explored model (base YOLOv8 on top, DCA-YOLOv8 below) and combines two complementary summaries: panel **a** inspects single–parameter marginals, while panel **b** visualizes the joint configuration of the elite 10% of trials.

Panel **a** stacks log-scaled violin–box plots for the four most influential scalars (`lr0`, `lrf`, `momentum` and `weight_decay`) plus two loss weights (`box`, `dfl`). For every row, we apply a Kruskal–Wallis H-test that compares the five optimizer-specific samples under the null hypothesis H_0 : "all strategies draw from the same parent distribution", with significance levels being printed directly above each strip ($p < 0.05 \rightarrow$ *, $p < 0.01 \rightarrow$ **, $p < 0.001 \rightarrow$ ***). Asterisks therefore indicate that at least one optimizer allocates noticeably different values (e.g., CMA-ES and GPSAMPLER converging to a tight range of `lr0`), whereas the absence of a star denotes statistical indistinguishability and hence broad agreement among the five methods.

Panel **b** embeds the top-decile trials into two UMAP coordinates after z-scaling and one-hot encoding. Points are colored by optimizer and shaped by backbone size. Around each color cloud, we draw a 95% confidence ellipse obtained by diagonalizing the group covariance matrix and scaling its principal axes. Smaller ellipses imply a tighter consensus within that optimizer; overlap between ellipses means that two methods ultimately settle in the same basin, whereas non-overlapping regions reveal divergent solutions that could still achieve competitive F1.

Comparing the two model families, base YOLOv8 exhibits broader, multi-modal searches: Ultralytics-ES's UMAP representation clusters away from other methods, while CMA-ES, GPSAMPLER, and TPE already focus on narrower corridors. After domain calibration (DCA-YOLOv8) best trials cluster more densely, learning-rate distributions shift upward, and weight decay becomes the sole parameter with negligible between-method differences, signaling consensus on its optimal scale. Overall, both representations show that all model-based samplers gravitate toward a shared optimum once the network is trained on the CADICA domain, whereas the Ultralytics-ES strategy seems to shift toward a different hyperparameter domain.

3.2.3. Attention analysis (GradCAM)

Gradient-weighted Class Activation Mapping (Grad-CAM) highlights the spatial support of a prediction by weighting the last-layer activations A^k with the average gradient of a target score y with respect to

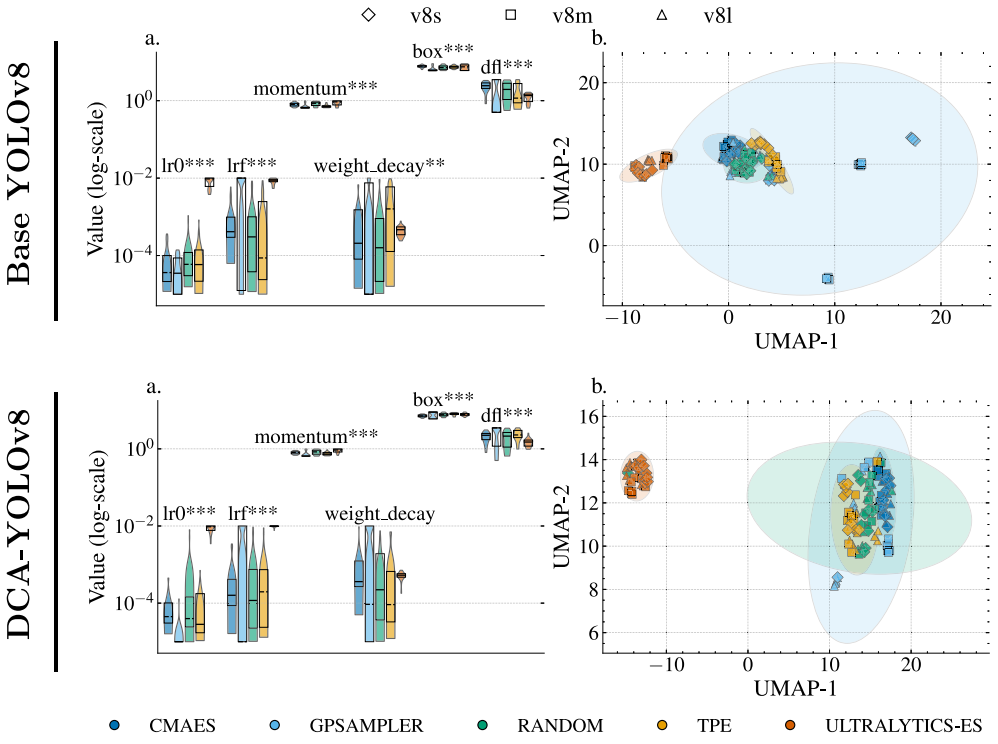

Fig. 5. Hyperparameter decision landscapes for the base YOLOv8 (top) and DCA-YOLOv8 (bottom) models, for the CADICA dataset. Each sub-figure combines (a) violin–box plots of the three most influential hyperparameters with Kruskal–Wallis statistical significance ($p < 0.05 \rightarrow$ *, $p < 0.01 \rightarrow$ **, $p < 0.001 \rightarrow$ ***) and (b) a UMAP embedding of the elite 10% trials, colored by optimization strategy and shaped by network size. Shaded ellipses represent the 95% confidence region of each optimizer in the embedded space.

each channel. A single backward pass is therefore required, which can be cumbersome for detectors that produce many instance-level scores. We consequently adopt *EigenCAM*, a gradient-free variant that projects the activation tensor onto its dominant principal component:

$$F \in \mathbb{R}^{C\times HW},\ (FF^{\top})v_1 = \lambda_1 v_1, \quad S_{\text{Eigen}} = \text{ReLU}\big(v_1^{\top}F\big) \text{ reshaped to } H \times W. \tag{20}$$

Because it captures the direction of maximal variance, EigenCAM yields smoother, class-agnostic heat-maps without additional hyperparameters, making it well suited to multi-head architectures such as YOLOv8.

Fig. 6 showcases the EigenCAM activation maps using the best-performing models. For the large and medium backbones (*v8l, v8m*), CMA-ES produces the most coherent EigenCAM maps, where high-value heat concentrates tightly on the arteries, with minor peripheral spill-over. GPSAMPLER and TPE trace almost the same arterial corridor, suggesting that all three model-based samplers converge on a shared, clinically relevant area, whereas Random Search alternates between frames of good alignment and others with diffuse, vessel-agnostic highlights, reflecting its larger performance variance. Ultralytics-ES rarely locks onto the lesion; its maps wander across background tissue or unrelated branches, indicating that the baseline optimizer fails to identify a stable, diagnostic focus for the same number of trials.

The lightweight *v8s* backbone shifts the trend, TPE now yields the crispest localization despite the reduced capacity, while CMA-ES and GPSAMPLER display slightly broader, lower-confidence lobes. Across every backbone, replacing YOLOv8 with the domain-calibrated architecture DCA-YOLOv8 sharpens the heat-maps, becoming more condensed along the coronary lumen and markedly less dispersed for the best-performing optimizers, underscoring that both principled hyperparameter search and domain adaptation synergistically steer the network's attention toward vessels most predictive of stenosis.

3.3. Dataset II

This section details the qualitative and quantitative results obtained for both YOLOv8, and the SOTA model, DCA-YOLOv8, after applying the proposed hyperparameter search, for the ARCADE dataset.

3.3.1. Performance

Fig. 7 replicates the speed–accuracy analysis for the ARCADE benchmark. Model-based samplers, again, trace the upper edge of the Pareto frontier, whereas the mutation-only Ultralytics-ES stays well below it, unable to stabilize on a competitive hyperparameter set. Backbone-wise, in Panel **a** (base YOLOv8) no single method dominates across all capacities, but CMA-ES, GPSAMPLER and TPE collectively span the frontier, with TPE edging ahead for the lightweight *v8s*. Panel **b** (DCA-YOLOv8) sharpens these distinctions: CMA-ES secures the top F1 for the large backbone (*v8l*), maintains parity with GPSAMPLER on *v8m*, and concedes the small model to TPE, which pairs the best accuracy with the shortest training time. Random Search remains competitive only when speed is paramount, while Ultralytics-ES never improves with backbone size, confirming its inadequate exploration strategy on this dataset.

3.3.2. Hyperparameter convergence

Fig. 8 repeats the marginal–versus–joint analysis of Section 3.2.2 for the ARCADE dataset. In panel **a**, Kruskal–Wallis statistics confirm that the learning-rate schedule (`lr0`, `lrf`), momentum and the two loss weights (`box`, `dfl`) vary significantly across optimizers (*** $p < 0.001$ in most cases), whereas `weight_decay` exhibits only a weak shift (* $p < 0.05$) or none at all—suggesting broad agreement on its optimal scale. In Panel **b**, Ultralytics-ES remains isolated. In the domain-calibrated DCA-YOLOv8 rows, Random Search drifts toward the model-based cluster, TPE shifts closer to CMA-ES, and GPSAMPLER's confidence region inflates, signaling more exploratory behavior.

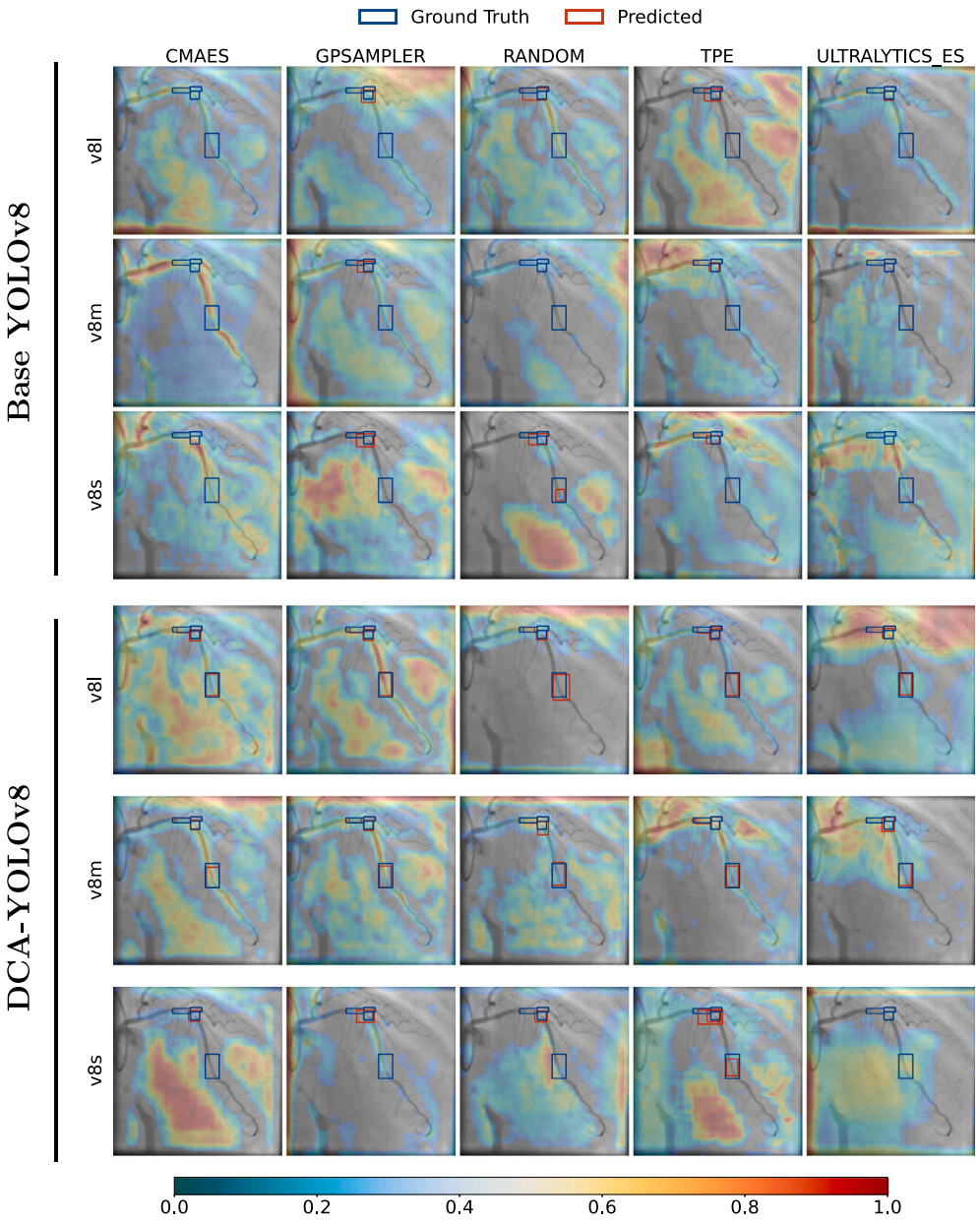

Fig. 6. Qualitative attention maps generated with EigenCAM for the best hyperparameter configuration returned by each optimizer, CADICA dataset. Columns correspond to optimizers; the top three rows show the base YOLOv8 backbones (*v8l*, *v8m*, *v8s*), and the bottom three replicate the layout for DCA-YOLOv8. In every panel, the EigenCAM heat-map is super-imposed on the original ICA frame; blue rectangles mark ground-truth stenosis locations, red rectangles indicate the detector's prediction. The color-bar below all images maps cool (0) to warm (1) hues.

3.3.3. Attention analysis (GradCAM)

Fig. 9 overlaps EigenCAM saliency for the ARCADE benchmark. In the base YOLOv8 rows (upper block) CMA-ES delivers the most anatomically precise heat-maps for the *v8l* and *v8m* backbones: high-confidence reds cling to the stenotic lumen with minimal spill-over, mirroring its superior quantitative scores on these capacities. TPE assumes that role for the lightweight *v8s*, localizing sharply around the lesion while CMA-ES and GPSAMPLER spread their attention over adjacent side branches. Random Search alternates between accurate and diffuse maps from frame to frame, reflecting its larger performance variance, whereas Ultralytics-ES routinely fires on background tissue or remote vessel segments, signaling the absence of a stable diagnostic focus.

In the DCA-YOLOv8 block (lower panel), saliency from the leading optimizers becomes even more condensed along the coronary centerline, and the distinction between well-tuned and poorly tuned configurations is accentuated. CMA-ES again produces the crispest localization for *v8l* and *v8m*, and TPE maintains its edge on *v8s*. GPSAMPLER clusters close to CMA-ES but shows slightly broader lobes.

3.4. Overall performance

Figs. 4 and 7 (Sections 3.2.1 and 3.3.1) report the peak F_1-Score attained during hyperparameter search, with whiskers denoting the 95% confidence interval computed from the stratified 3-fold cross-validation (CV) of the best-performing configuration. Complementing these plots, Table 4 lists the *mean* F_1-Score at an IoU threshold of

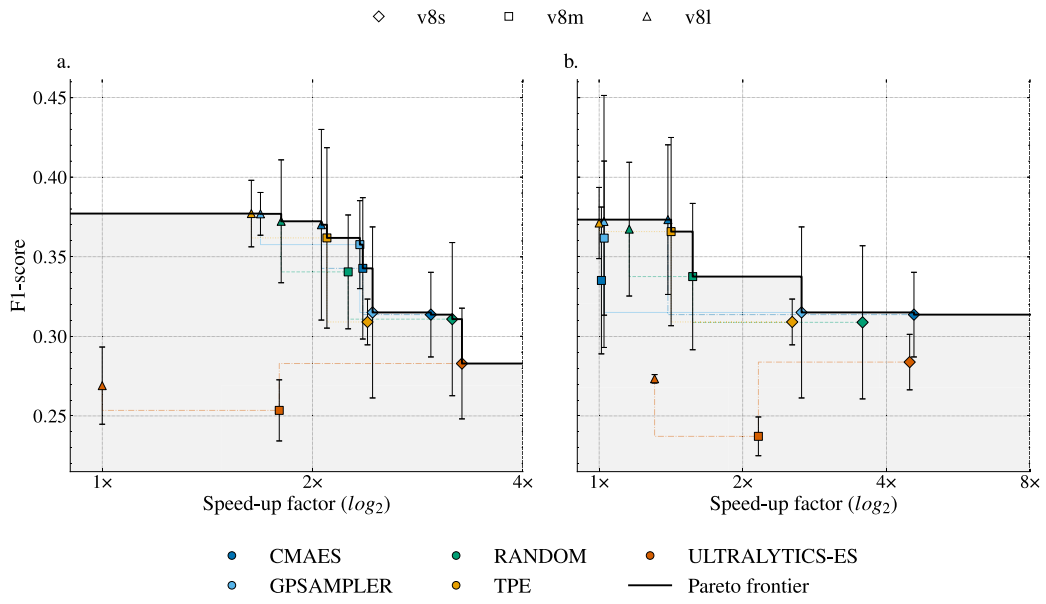

Fig. 7. Trade-off between optimization efficiency and detection accuracy for YOLOv8 (panel a) and DCA-YOLOv8 (panel b) for the ARCADE dataset. Each marker corresponds to the best trial returned by one hyperparameter–search strategy (color) on one model size (shape). The shape is the maximum F1 observed during the hyperparameter search. Vertical bars show the 95% confidence interval (CI) obtained by re-training the model with the same hyperparameters under a 3-fold cross-validation protocol.

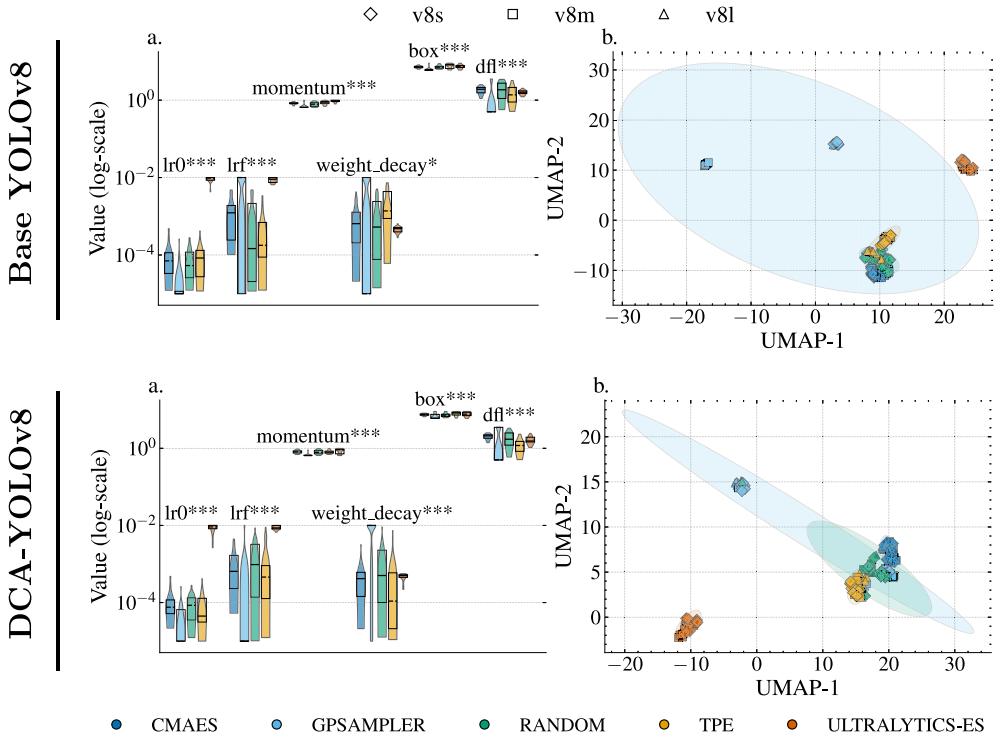

Fig. 8. Hyperparameter decision landscapes for the base YOLOv8 (top) and DCA-YOLOv8 (bottom) models, for the ARCADE dataset. Each sub-figure combines (a) violin–box plots of the three most influential hyperparameters with Kruskal–Wallis statistical significance ($p < 0.05 \rightarrow$ *, $p < 0.01 \rightarrow$ **, $p < 0.001 \rightarrow$ ***) and (b) a UMAP embedding of the elite 10% trials, colored by optimization strategy and shaped by network size. Shaded ellipses represent the 95% confidence region of each optimizer in the embedded space.

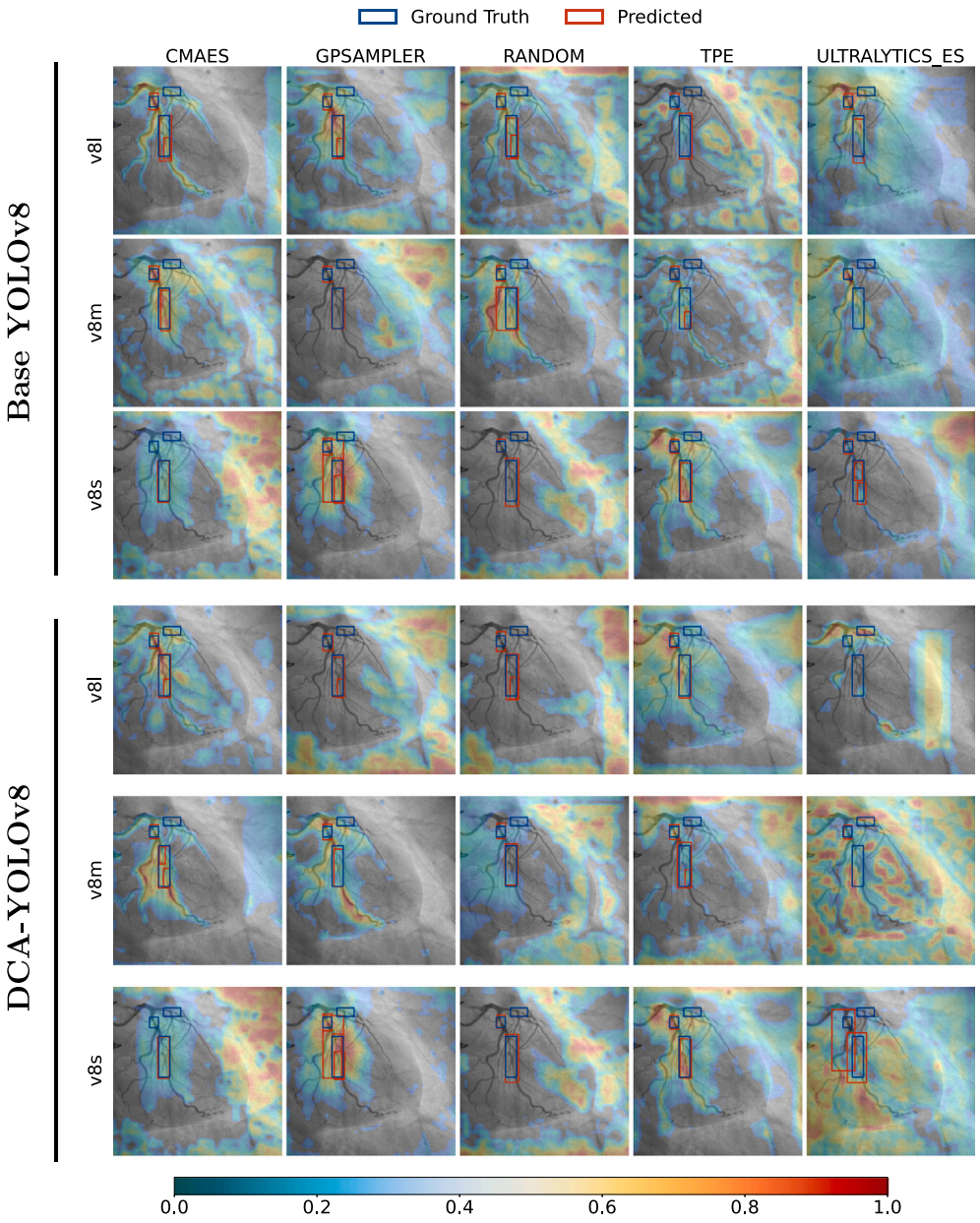

Fig. 9. Qualitative attention maps generated with EigenCAM for the best hyperparameter configuration returned by each optimizer, ARCADE dataset. Columns correspond to optimizers; the top three rows show the base YOLOv8 backbones (*v8l*, *v8m*, *v8s*), and the bottom three replicate the layout for DCA-YOLOv8. In every panel the EigenCAM heat-map is super-imposed on the original ICA frame; blue rectangles mark ground-truth stenosis locations, red rectangles indicate the detector's prediction. The color-bar below all images maps cool (0) to warm (1) hues.

0.5, averaged over 3-fold CV and accompanied by the corresponding 95% confidence bounds, thereby providing a more global view of each optimizer generalization performance on both datasets.

4. Discussion

The present study set out to determine whether model–based hyperparameter optimizers can outperform the mutation-only approach offered by the `ultralytics` package creators when tuning the YOLOv8 detector for coronary-stenosis detection. Experiments were conducted on two complementary benchmarks: CADICA, which offers full angiography sequences with heterogeneous visibility, and ARCADE, which provides a single, high-quality frame per patient. Across both datasets and for both the canonical and domain-calibrated (*Double Coordinate Attention*, DCA) versions of YOLOv8, the evidence consistently supports our working hypothesis.

Pareto frontiers in Figs. 4 and 7 show that CMA-ES, GPSAMPLER, and TPE occupy the upper envelope of F_1-Score versus speed-up, while Ultralytics-ES remains strictly inferior and Random Search touches the frontier only sporadically. These patterns hold for both datasets, indicating that the performance gains are not an artefact of a particular imaging protocol but rather stem from the optimizers' ability to exploit structure in the hyperparameter search space. A more granular look reveals that the choice of optimizer should be aligned with backbone capacity. CMA-ES delivers the highest mean F_1 on the large *v8l* network, presumably because its full-covariance adaptation excels with larger models, where the search space could be more anisotropic, and, therefore, complex. For the medium backbone (*v8m*), the Bayesian approaches (i.e., GPSAMPLER, TPE) match or slightly surpass CMA-ES, and, for the lightweight *v8s* model, TPE becomes the clear leader, reflecting its superior sample efficiency when the dimension is lower and, probably, the signal-to-noise ratio of the search space is weaker.

Table 4
Mean F_1-score at 0.5 IoU under 3-fold CV (± 95% CI) for two datasets. Top-performing optimizers are highlighted in bold for each backbone size and YOLOv8 model.

		CADICA		ARCADE	
		DCA-YOLOv8	YOLOv8	DCA-YOLOv8	YOLOv8
CMA-ES	v8l	**0.225 ± 0.060**	**0.215 ± 0.063**	**0.355 ± 0.079**	**0.348 ± 0.060**
	v8m	**0.215 ± 0.104**	0.171 ± 0.061	0.329 ± 0.046	0.327 ± 0.044
	v8s	0.187 ± 0.097	0.155 ± 0.019	0.269 ± 0.027	0.268 ± 0.027
Gaussian Process	v8l	0.216 ± 0.104	0.199 ± 0.078	0.347 ± 0.047	0.342 ± 0.013
	v8m	0.199 ± 0.083	0.177 ± 0.001	**0.346 ± 0.048**	0.318 ± 0.028
	v8s	0.187 ± 0.152	0.177 ± 0.039	**0.304 ± 0.054**	**0.303 ± 0.054**
TPE	v8l	0.217 ± 0.161	0.198 ± 0.006	0.349 ± 0.022	0.338 ± 0.021
	v8m	0.202 ± 0.095	**0.188 ± 0.023**	0.321 ± 0.059	**0.344 ± 0.057**
	v8s	**0.196 ± 0.116**	**0.181 ± 0.037**	0.285 ± 0.014	0.285 ± 0.014
Random	v8l	0.213 ± 0.109	0.197 ± 0.070	0.339 ± 0.042	0.337 ± 0.039
	v8m	0.213 ± 0.148	0.180 ± 0.056	0.301 ± 0.046	0.333 ± 0.036
	v8s	0.152 ± 0.070	0.158 ± 0.041	0.305 ± 0.048	0.305 ± 0.048
Ultralytics λ+1-ES	v8l	0.198 ± 0.109	0.160 ± 0.004	0.247 ± 0.003	0.251 ± 0.024
	v8m	0.176 ± 0.053	0.154 ± 0.046	0.229 ± 0.012	0.229 ± 0.019
	v8s	0.176 ± 0.153	0.165 ± 0.016	0.256 ± 0.017	0.251 ± 0.035

Violin–box plots in Figs. 5a and 8a show highly significant Kruskal–Wallis statistics for most variables, an effect possibly driven by Ultralytics-ES sampling a disjoint numerical regime, claim supported by the UMAP representations in Figs. 5b and Fig. 8b, where CMA-ES and TPE cluster tightly in all the embedded space representations, Random Search forms a diffuse halo around them, and Ultralytics-ES occupies a remote, low-density region. By contrast, `weight_decay` frequently exhibits non-significant differences, suggesting that all methods converge on a similar magnitude for this regularizer. The finding implies that mutation-only search fails to discover the stable corridors identified by model-based optimizers, lending statistical weight to the performance gaps reported above.

EigenCAM visualizations in Fig. 6 and Fig. 9 further support the numerical results. On both datasets, CMA-ES produces the most lesion-centered maps for *v8l*, TPE yields the crispest localization for *v8s*, and attention maps for *v8m* vary. Ultralytics-ES fails to lock its attention onto diagnostically relevant segments, and Random Search alternates between focused and diffuse activations. The qualitative agreement between attention and accuracy strengthens confidence that the measured gains translate into clinically meaningful predictions.

Table 4 shows that CMA-ES is better suited for large backbones (*v8l*), TPE for lightweight variants (*v8s*), and the GPSAMPLER as a balanced option for medium models (*v8m*). Yet each optimizer carries non-trivial costs. CMA-ES performs full covariance adaptation, and therefore scales cubically with the search dimensionality, $\mathcal{O}(d^3)$, and quadratically in memory; deployments with limited GPU RAM or more than ∼60 tunables may find this prohibitive. Bayesian samplers scale more gently, with $\mathcal{O}(dn\log n)$ for TPE, and $\mathcal{O}(n^3)$ for the GP, but both can stall in local optima and typically benefit from multiple restarts and careful bandwidth selection [39].

We acknowledge the following limitations in our study: First, the present study caps every optimizer at 100 trials, sufficient to expose clear rankings but below the 300–500 evaluations often recommended for CMA-ES and GP models [39]. Second, we optimize a single performance metric; extending the search to *multi-objective* settings (accuracy, latency, energy) with frameworks like Hyperband or NSGA-II could benefit algorithmic performance [42,43]. Finally, only "static" training variables were tuned; incorporating dynamic schedules (early stopping, mixed precision, structured pruning) may yield further gains and could be tackled with hybrid strategies that combine covariance adaptation with surrogate modeling.

Despite these drawbacks, such model-based engines are necessary for YOLO-style detectors, whose hyperparameters form an anisotropic landscape that the mutation-only Ultralytics-ES approach fails to explore effectively. Choosing an optimizer should thus balance backbone capacity, available compute, and the need for either global exploration or fine-grained exploitation.

5. Conclusions

This study confirms that pairing YOLOv8 with principled, model-based HPO delivers substantial and reproducible gains in automated coronary-stenosis detection. Across two angiography benchmarks of contrasting visual complexity and clinical context, the model-based engines: CMA-ES, TPE and GPSAMPLER; consistently achieved higher F_1-Scores, converged more tightly in both marginal and joint hyperparameter spaces, and produced anatomically sharper saliency maps than the default Ultralytics tuning strategy. A clear backbone-size hierarchy emerged, where CMA-ES excels on large backbones, TPE dominates lightweight models, and GPSAMPLER provides a balanced compromise. These findings endorse adaptive, probabilistic search as a practical tool for tuning high-dimensional YOLO-based models in stenosis detection, motivating researchers to adopt such optimization techniques and obtain reproducible performance improvements.

CRediT authorship contribution statement

Mario Pascual-González: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ariadna Jiménez-Partinen:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation. **Esteban J. Palomo:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation. **Ezequiel López-Rubio:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Almudena Ortega-Gómez:** Writing – review & editing, Writing – original draft, Validation, Investigation, Conceptualization.

Ethics statement

The authors declare that they do not need an Ethics statement because this research uses an open-access dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the Autonomous Government of Andalusia (Spain) under project UMA20-FEDERJA-108, project name Detection, characterization and prognosis value of the non-obstructive coronary disease with deep learning, and also by the Ministry of Science and Innovation of Spain, grant number PID2022-136764OA-I00, project name Automated Detection of Non Lesional Focal Epilepsy by Probabilistic Diffusion Deep Neural Models. It includes funds from the European Regional Development Fund (ERDF). It is also partially supported by the Fundación Unicaja under project PUNI-003_2023, project name Intelligent System to Help the Clinical Diagnosis of Non-Obstructive Coronary Artery Disease in Coronary Angiography. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of a RTX A6000 GPU with 48Gb. The authors also thankfully acknowledge the grant of the Universidad de Málaga, Spain and the Instituto de Investigación Biomédica de Málaga y Plataforma en Nanomedicina-IBIMA Plataforma BIONAND. Funding for open access charge: Universidad de Málaga/CBUA, Spain.

Data availability

CADICA dataset [21] is open-access available at the Mendeley Data repository with the data identification number: https://doi.org/10.17632/p9bpx9ctcv.2.

ARCADE dataset [22] is an open-access dataset available at Zendodo repository with the DOI: https://doi.org/10.5281/zenodo.7981244.

References

[1] J.P. Duggan, A.S. Peters, G.D. Trachiotis, J.L. Antevil, Epidemiology of coronary artery disease, Surg. Clin. 102 (3) (2022) 499–516.

[2] N. Townsend, D. Kazakiewicz, F. Lucy Wright, A. Timmis, R. Huculeci, A. Torbica, C.P. Gale, S. Achenbach, F. Weidinger, P. Vardas, Epidemiology of cardiovascular disease in europe, Nat. Rev. Cardiol. 19 (2) (2022) 133–143.

[3] K.D. Kochanek, S.L. Murphy, J. Xu, E. Arias, Mortality in the United States, 2022, US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2024.

[4] R.A. Byrne, X. Rossello, J. Coughlan, E. Barbato, C. Berry, A. Chieffo, M.J. Claeys, G.-A. Dan, M.R. Dweck, M. Galbraith, et al., 2023 ESC guidelines for the management of acute coronary syndromes: developed by the task force on the management of acute coronary syndromes of the European society of cardiology (ESC), Eur. Hear. J.: Acute Cardiovasc. Care 13 (1) (2024) 55–161.

[5] G. Rigatelli, F. Gianese, M. Zuin, Modern atlas of invasive coronary angiography views: a practical approach for fellows and young interventionalists, Int. J. Cardiovasc. Imaging 38 (5) (2022) 919–926.

[6] G. Litjens, F. Ciompi, J.M. Wolterink, B.D. de Vos, T. Leiner, J. Teuwen, I. Išgum, State-of-the-art deep learning in cardiovascular image analysis, JACC: Cardiovasc. Imaging 12 (8 Part 1) (2019) 1549–1565.

[7] K.K. Wong, G. Fortino, D. Abbott, Deep learning-based cardiovascular image diagnosis: a promising challenge, Future Gener. Comput. Syst. 110 (2020) 802–811.

[8] M.N. Menezes, J. Lourenço-Silva, B. Silva, T. Rodrigues, A.R.G. Francisco, P.C. Ferreira, A.L. Oliveira, F.J. Pinto, Development of deep learning segmentation models for coronary X-ray angiography: Quality assessment by a new global segmentation score and comparison with human performance, Rev. Port. Cardiol. 41 (12) (2022) 1011–1021.

[9] J. Park, J. Kweon, Y.I. Kim, I. Back, J. Chae, J.-H. Roh, D.-Y. Kang, P.H. Lee, J.-M. Ahn, S.-J. Kang, et al., Selective ensemble methods for deep learning segmentation of major vessels in invasive coronary angiography, Med. Phys. 50 (12) (2023) 7822–7839.

[10] H. Ling, B. Chen, R. Guan, Y. Xiao, H. Yan, Q. Chen, L. Bi, J. Chen, X. Feng, H. Pang, et al., Deep learning model for coronary angiography, J. Cardiovasc. Transl. Res. 16 (4) (2023) 896–904.

[11] Y.I. Kim, J.-H. Roh, J. Kweon, H. Kwon, J. Chae, K. Park, J.-H. Lee, J.-O. Jeong, D.-Y. Kang, P.H. Lee, et al., Artificial intelligence-based quantitative coronary angiography of major vessels using deep-learning, Int. J. Cardiol. 405 (2024) 131945.

[12] W. Wu, J. Zhang, H. Xie, Y. Zhao, S. Zhang, L. Gu, Automatic detection of coronary artery stenosis by convolutional neural network with temporal constraint, Comput. Biol. Med. 118 (2020) 103657.

[13] D.L. Rodrigues, M.N. Menezes, F.J. Pinto, A.L. Oliveira, Automated detection of coronary artery stenosis in X-ray angiography using deep neural networks, 2021, arXiv preprint arXiv:2103.02969.

[14] T. Han, D. Ai, X. Li, J. Fan, H. Song, Y. Wang, J. Yang, Coronary artery stenosis detection via proposal-shifted spatial-temporal transformer in X-ray angiography, Comput. Biol. Med. 153 (2023) 106546.

[15] T. Wang, X. Su, Y. Liang, X. Luo, X. Hu, T. Xia, X. Ma, Y. Zuo, H. Xia, L. Yang, Integrated deep learning model for automatic detection and classification of stenosis in coronary angiography, Comput. Biol. Chem. 112 (2024) 108184.

[16] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, et al., Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 13 (2) (2023) e1484.

[17] M.A.K. Raiaan, S. Sakib, N.M. Fahad, A. Al Mamun, M.A. Rahman, S. Shatabda, M.S.H. Mukta, A systematic review of hyperparameter optimization techniques in convolutional neural networks, Decis. Anal. J. (2024) 100470.

[18] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, A. Gertych, Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization, Heliyon 10 (5) (2024).

[19] L. Ramos, E. Casas, E. Bendek, C. Romero, F. Rivas-Echeverría, Hyperparameter optimization of YOLOv8 for smoke and wildfire detection: Implications for agricultural and environmental safety, Artif. Intell. Agric. 12 (2024) 109–126.

[20] D. Wahyudi, I. Soesanti, H.A. Nugroho, Optimizing hyperparameters of YOLO to improve performance of brain tumor detection in MRI images, in: 2023 6th International Conference on Information and Communications Technology, ICOIACT, IEEE, 2023, pp. 413–418.

[21] A. Jiménez-Partinen, M.A. Molina-Cabello, K. Thurnhofer-Hemsi, E.J. Palomo, J. Rodríguez-Capitán, A.I. Molina-Ramos, M. Jiménez-Navarro, CADICA: A new dataset for coronary artery disease detection by using invasive coronary angiography, Expert Syst. (2024) e13708.

[22] M. Popov, A. Amanturdieva, N. Zhaksylyk, A. Alkanov, A. Saniyazbekov, T. Aimyshev, E. Ismailov, A. Bulegenov, A. Kuzhukeyev, A. Kulanbayeva, et al., Dataset for automatic region-based coronary artery disease diagnostics using x-ray angiography images, Sci. Data 11 (1) (2024) 20.

[23] J. Wang, H. Zhu, S. hua Wang, Y. Zhang, A review of deep learning on medical image analysis, Mob. Netw. Appl. 26 (2020) 351–380, URL https://api.semanticscholar.org/CorpusID:228826220.

[24] E. Ovalle-Magallanes, J.G. Aviña-Cervantes, I. Cruz-Aceves, J. Ruiz-Pinales, Transfer learning for stenosis detection in X-ray coronary angiography, Mathematics (2020) URL https://api.semanticscholar.org/CorpusID:225266229.

[25] M. Akgül, H.İ. Kozan, H.A. Akyürek, Ş. Taşdemir, Automated stenosis detection in coronary artery disease using yolov9c: Enhanced efficiency and accuracy in real-time applications, J. Real Time Image Process. 21 (5) (2024) 177.

[26] J. Li, X. Tang, X. Wang, LT-YOLO: long-term temporal enhanced YOLO for stenosis detection on invasive coronary angiography, Front. Mol. Biosci. 12 (2025) 1558495.

[27] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2016 IEEE Conf. Comput. Vis. Pattern Recognition (CVPR) (2015) 779–788, URL https://api.semanticscholar.org/CorpusID:206594738.

[28] P.F. Felzenszwalb, R.B. Girshick, D.A. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627–1645, URL https://api.semanticscholar.org/CorpusID:3198903.

[29] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 2014 IEEE Conf. Comput. Vis. Pattern Recognit. (2013) 580–587, URL https://api.semanticscholar.org/CorpusID:215827080.

[30] G. Jocher, Ultralytics YOLOv5, 2020, http://dx.doi.org/10.5281/zenodo.3908559, URL https://github.com/ultralytics/yolov5.

[31] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLOv8, 2023, URL https://github.com/ultralytics/ultralytics.

[32] T.-Y. Lin, M. Maire, S.J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision, 2014, URL https://api.semanticscholar.org/CorpusID:14113767.

[33] J.R. Terven, D.M.C. Esparza, J.-A. Romero-González, A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS, Mach. Learn. Knowl. Extr. 5 (2023) 1680–1716, URL https://api.semanticscholar.org/CorpusID:258823486.

[34] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, 2019, ArXiv, arXiv:1911.08287.URL https://api.semanticscholar.org/CorpusID:208158250.

[35] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, 2020, ArXiv, arXiv:2006.04388, URL https://api.semanticscholar.org/CorpusID:219531292.

[36] H. Duan, S. Yi, Y. Ren, DCA-YOLOv8: A novel framework combined with AICI loss function for coronary artery stenosis detection, Sensors 24 (24) (2024) 8134.
[37] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (1) (2012) 281–305.
[38] A.H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization, Evol. Syst. 12 (2020) 217–223, URL https://api.semanticscholar.org/CorpusID:219734460.
[39] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.
[40] T. Bartz-Beielstein, J. Branke, J. Mehnen, O. Mersmann, Evolutionary algorithms, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 4 (3) (2014) 178–195.
[41] N. Hansen, The CMA evolution strategy: A tutorial, 2016, arXiv preprint arXiv:1604.00772.
[42] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, J. Mach. Learn. Res. 18 (185) (2018) 1–52.
[43] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.