IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3239-3252
Journal homepage: http://iieta.org/journals/isi

Object-Size-Aware Dataset Partitioning for Object Detection Evaluation: A Controlled Study of Split Representativeness and Detector Performance

Hussein Ali Murad[1], Ahmed A. Hashim[2*]

[1] Informatics Institute for Postgraduate Studies, University of Information Technology and Communications (UoITC), Baghdad 10001, Iraq
[2] Business Informatics College, University of Information Technology and Communications (UoITC), Baghdad 10053, Iraq

Corresponding Author Email: dr.ahmed.hashim@uoitc.edu.iq

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311001

Received: 5 March 2026
Revised: 20 August 2026
Accepted: 15 September 2026
Available online: 31 October 2026

Keywords:
object detection, dataset partitioning, object-size distribution, partition-induced evaluation bias, split representativeness, data-centric artificial intelligence

ABSTRACT

Object detection evaluation relies on representative training, validation, and test subsets, yet partitioning can produce differences in object-size composition that distort reported performance. This study examines how object-size-aware partitioning affects split representativeness and detector evaluation under a fixed You Only Look Once (YOLO) version 11 nano (YOLOv11n) configuration. Four strategies—natural-order sequential, global size-ordered, Common Objects in Context (COCO)-style size-grouped, and size-aware stratified splitting—were compared across three datasets: two license-plate datasets and WIDER FACE. All experiments used a 640 × 640 input resolution and 100 training epochs; training variability was assessed across three random seeds. Detection performance was measured using mean average precision (mAP)@0.50:0.95, while split representativeness was quantified by a composite score combining Kullback-Leibler divergence and normalized Earth Mover's Distance (KL-EMD). Global size-ordered splitting caused marked performance deterioration in several conditions, most notably on the second license-plate dataset, where mAP@0.50:0.95 fell to 0.005. Size-aware stratified splitting generally reduced object-size distribution mismatch while maintaining competitive detection performance. The COCO-level stratified variant yielded the lowest KL-EMD scores across all three datasets, whereas fine-grained quantized stratification achieved the highest mAP@0.50:0.95 on the first license-plate dataset (0.523). However, the most representative split did not consistently produce the highest detection score; natural-order splitting performed best on WIDER FACE (0.385). Across the three training seeds, standard deviations ranged from 0.002 to 0.011. These findings indicate that split representativeness and detector performance are complementary but distinct evaluation dimensions. Dataset partitioning should therefore be treated as an explicit component of object detection evaluation, with distributional balance assessed alongside detection metrics.

1. INTRODUCTION

Object detection is a fundamental task in computer vision that enables systems to identify and localize objects within images [1, 2]. Traditional sliding window systems and features that are manually developed have been replaced by powerful deep learning-based systems in the past two decades. In any application that lies between autonomous navigation and healthcare diagnostics and monitoring, recent models such as Faster Region-based Convolutional Neural Network (Faster R-CNN), Single Shot MultiBox Detector (SSD), and the You Only Look Once (YOLO) family have improved both real-time speed and detection precision [3, 4].

Despite these developments, an important factor that remains underexplored is the composition of the datasets used for training, validation, and testing. Conventional dataset splitting practices commonly apply fixed ratios, such as 70% for training, 20% for validation, and 10% for testing, without explicitly examining whether important object-level data characteristics are preserved across the generated subsets [5, 6]. In object detection, properties such as object-size distribution, object density, and sample representativeness can influence model learning, evaluation stability, and generalization [7, 8]. Reported performance may partly reflect partitioning bias when small, medium, and large objects are unevenly distributed across the training, validation, and testing subsets.

To address this issue, this study focuses on adopting object-size-aware dataset partitioning as a data-centric evaluation strategy to tackle object detection. You Only Look Once (YOLO) version 11 nano (YOLOv11n) is used as a fixed detector to isolate the effect of dataset partitioning without architectural changes. It compares the different partitioning scenarios for three real-world datasets and examines the splitting representativeness via the distributions of object sizes, distributions of object densities, Kullback-Leibler (KL) divergence, and normalized Earth Mover's Distance (EMD).

The study examines how split composition affects

evaluation bias and detector performance under a fixed YOLOv11n setting. Dataset partitioning can also be interpreted as an information-systems design decision in the context of artificial intelligence (AI)-enabled information systems, as well as a pre-processing step. Managed data resources include training, validation and testing subsets whose composition affects the evidence to validate, compare and select intelligent models. If these subsets are not able to capture important characteristics of the data, performance reported could be misleading and used to inform selection of a model or downstream decision making. Improved split representativeness can support more transparent data management, evaluation, and model-selection decisions, clearer dataset management and governance, and more intelligent use of object detection components in intelligent information systems. This is aligned with the data-centric AI perspective, which provides systematic development, preparation, evaluation, and maintenance of data throughout the AI lifecycle [6].

The aim of this study is to concentrate on the training and validation sets having a different distribution of object sizes from the subset used for testing, which leads to an evaluation bias. The term "bias" is not used for every kind of bias that might exist in a dataset such as class imbalance, annotation noise, labelling inconsistencies, demographic bias, data-acquisition bias and so on, or is architectural or model bias referred to as “bias”. Rather, the study investigates the effect of the differences in object-size representation that are introduced in the case of partitioning the datasets on the evaluation of a fixed object detector. In this sense, YOLOv11n is maintained without any architectural changes and with a fixed training procedure to maintain dataset partitioning as the main experimental parameter. In the light of data-centric AI, the study discusses the factors of evaluation design - split composition and representativeness - which can impact the reliability and interpretation of the results obtained from object detection.

One of the methodological contributions of this study is that it addresses object-size-aware partitioning as an evaluated problem in the design of the study instead of using only the traditional random or coarse size-group splitting. The Common Objects in Context (COCO) small, medium and large categories are a coarse-grained size representation of objects [9]; however, they do not take into account the bounding-box size variation within each of these size categories. The framework therefore compares coarse COCO-level grouping with a finer quantized approach that preserves within-category object-size variation. The contribution is by no means a new detector architecture or a new object-size taxonomy, but rather a controlled data-centric partitioning and evaluation framework to analyze how the composition of the splits impacts the detector evaluation. Its applicability is examined across three datasets, including WIDER FACE, to extend the analysis beyond license-plate detection.

2. RELATED WORKS

In recent years, most advances in object detection have focused on model-centric advances, particularly those that aim at accommodating the challenges posed by detecting small objects and objects of different scales. Architectural techniques such as attention mechanisms, multi-scale feature fusion and precise detection heads have shown to be effective in boosting the sensitivity to fine-grained details and small targets [7, 10]. The approaches are designed to improve feature representations and minimize information loss due to scale and result in better performance on a number of benchmarks. At the same time, a growing body of literature has paid attention to data-centric solutions, suggesting that model architecture is not the only factor influencing detection performance.

Strategies used to increase training diversity and robustness include class-specific dataset splitting [5] and synthetic data augmentation [11]. In other works, it is noted that quality of data, the accuracy of the annotations, and frequency of objects and the uniformity of labelling are key factors, which influence the consistency of the detector and training success [12].

While the significance of data quality and sampling is acknowledged in these studies, most focus on class imbalance, data scarcity, the quality of annotations, and data augmentation, or detector architecture improvement but not on the representativeness of the generated training, validation, and testing subsets. While model-centric advances, such as the recent YOLO-based detectors, are still relevant to enhancing detection capabilities, architectural enhancements can't solve evaluation bias problems that can occur when the data subsets have significantly different underlying object characteristics. This causes a deeper dive into the composition and partitioning of datasets to be more motivating.

It has long been known that visual datasets could have dataset-specific characteristics that could adversely impact the generalization of models across datasets and create dataset bias in computer vision [13]. Recent bias analyses of visual datasets have also demonstrated that missing relevant data attributes can cause dataset bias and the results of their evaluations of perception systems can be unreliable [14]. Sample-selection bias and population distribution shift have been shown to be distinct from algorithmic bias, and it is crucial to ensure that subsets of the data are representative of the population when dealing with different distributions for each [15].

Recent data-centric AI research emphasizes developing, preparing, evaluating, and maintaining data throughout the AI lifecycle rather than focusing only on model architectures [6]. They stress the importance of the traditional operations that have been considered as preprocessing steps, which can directly impact the reliability of AI development and evaluation. For object detection, this view suggests that the issue of evaluating the representativeness of the partitions must be made an explicit design choice and should not be taken for granted, as in the traditional splits.

More generally, dataset shift can make machine learning unreliable when development and evaluation data follow different distributions [16]. Benchmark-reliability research also reveals that detector performance can vary significantly as a function of the distribution with which evaluation data differs from the distributions represented during training. In the context of information systems benchmarking, it was shown that the evaluation of machine learning also depends on its predictive performance, as well as on the protocol for data engineering and splitting, leakage control, reproducibility, and stability assessment of the model [17]. Large-scale benchmark, COUNTS was introduced [8] to evaluate object detectors and multimodal models under natural distribution shifts. They conclude that testing performance of the detection systems is dependent on something other than the distribution of data assumed in the training, and their results demonstrate

that detector performance can be diminished if there are poor alignments between the training and testing conditions. In the present study, the issue of evaluation mismatch is internal, specifically, differences in object-size composition between the training and validation subsets and the differences between the validation and testing subsets, due to partitioning.

In spite of these progresses, the way of partitioning datasets is still relatively little explored as a source of evaluation variability in object detection. Often, the split ratios are predesigned, and it is never explicitly checked if the important data characteristics, especially object-size distributions, are still representative in the resulting training/validation/testing subsets. This issue can become more pronounced when datasets are limited or internally heterogeneous, because small, medium, and large objects may be unevenly concentrated across the generated subsets. The COCO benchmark formalizes small-, medium-, and large-object categories and reports scale-specific evaluation measures, demonstrating the importance of object scale in detection assessment [9]. However, although object scale is considered during evaluation, conventional dataset partitioning procedures do not necessarily control how object-size composition is distributed across the generated subsets. Consequently, reported detector performance may partly reflect partition-induced distribution mismatch rather than detector capability alone.

This gap motivates the present study to treat dataset partitioning as an integral part of evaluation design rather than simply as a data-preparation step. The authors analyze whether it is possible to create more representative partitions of the object-size composition during the training, validation and testing sets, and thus interpret the result of the detector performance more accurately. Therefore, the study is set in a data-centric evaluation setting, where the YOLOv11n detector is kept fixed and the partitioning strategy and split composition are systematically changed. The design enables the influence of the size mismatch induced by partitions to be isolated from its influence on detector architecture.

3. METHODOLOGY

For this study, object size is one of the crucial data characteristics that might affect the learning and evaluation results of the detector. In this study, the assumption of natural representation (as standard data splits) is left open and studied whether the composition of the objects in the dataset (based on size) and the distribution of the objects in the dataset (based on density) is preserved in training, validation and testing. In order to investigate this problem, four partitioning scenarios were developed: Original sequence splitting, Size-ordered splitting, size-group descending splitting (as in COCO), and size-aware stratified sampling. The scenarios were tested with a fixed configuration for YOLOv11n, which enabled the comparison of the different scenarios under the same experimental conditions, with deterministic, size- biased, and size-aware partitioning strategies.

The methodology is divided into seven parts, namely (1) dataset description and preparation, (2) dataset statistics, (3) size-aware dataset partitioning scenarios, (4) split distribution similarity analysis using KL divergence and normalized EMD, (5) model architecture and training, (6) evaluation metrics, and (7) experimental setup.

3.1 Dataset description and preparation

To investigate the impact of partitioning the datasets and the distribution of object sizes on the evaluation of YOLOv11n, this study uses three real-world object detection datasets. This includes two datasets for license plates as well as one face-detection dataset. The choice was made for the following reasons: The first is that it would enable the experimental analysis of a range of object categories that could not have been analyzed if only one object category had been analyzed, and the second is that it would explore the proposed partitioning strategy in different object-scale, object-density, and scene-complexity conditions. For all three datasets, a standard data-pre-processing method was used. Consistency was verified for image and annotation files, and annotations were converted to YOLO format if necessary. Native image dimensions were used for object-size analysis and dataset partitioning, whereas 640 × 640 was applied only during YOLOv11n training.

3.1.1 Car Plate Detection Dataset (Andrew MVD, Kaggle)

Car Plate Detection dataset from Kaggle (Dataset 1): 433 vehicle images containing license plates. The annotations are given in Pascal Visual Object Classes (VOC) format in Extensible Markup Language (XML) format with Bounding Boxes for the license plate class. Variations in viewpoint, image resolution, traffic scene conditions, partial occlusion, and object scale are included in the dataset. Thus, it offers a convenient but informative collection of data for studying the effects of partitioning decisions on the ability to detect license plate images [18].

3.1.2 License Plate Dataset (Fares Elmenshawii, Kaggle)

Dataset 2 contains 5368 images of vehicle license plates in color, and a Pascal VOC XML file for each image. Visible License Plates are annotated in each image in the form of bounding boxes. This dataset offers more illumination, background, and view variation than that of Dataset 1. Since it does not provide a fixed training/validation/testing split, it is suitable for controlled partitioning experiments under different split-ratio and object-size-aware scenarios [19].

3.1.3 WIDER FACE Dataset

Dataset 3 is based on the WIDER FACE benchmark [20] and uses a YOLO-formatted version that has been retrieved from Kaggle [21] and includes 12,900 annotated images. It is included to further validate the experiments outside of the scope of license plate detection and to explore the proposed partitioning framework in another detection context. WIDER FACE contains more diverse scene conditions, greater variation in object density, and a wider range of object scales than the license plate datasets. This is especially important for testing if split-level object-size and object-density composition can affect detection performance in a more cluttered and/or more complex dataset [21].

The three sets of data were analyzed and treated separately and never combined during experiments. Some duplicate samples were identified during data preparation and excluded from the dataset before partitioning. Only valid image-annotation pairs retained after preprocessing were used for the partitioning and the evaluation procedures.

3.2 Dataset statistics

Prior to training, a descriptive analysis was conducted for

the three datasets to examine their internal data characteristics, with particular focus on object-size composition. Before the training of the models, images were categorized according to their size. The images were classified based on their native size into different object sizes prior to model training. In the case of the YOLO-normalized annotation, the width and height of the bounding box were resized to the pixel size by using the original size of the image width and height, and the area of the bounding box was calculated in pixel size. Where images include several annotated objects, the size representation at the image level for dataset-level statistical analysis and partitioning was based on the largest annotated bounding-box area. The 640 × 640 input resolution was applied only during YOLOv11n training; it is not the way by which the pre-training object-size categories are defined as part of the fixed-size YOLOv11n training configuration. This analysis is important because object detection performance can be influenced not only by the detector architecture but also by how object sizes are distributed across the training, validation, and testing subsets. A dataset or a generated split may contain information that is mostly concentrated in some object size category, causing the model to learn biased representations and leading to a variation in performance which may be misinterpreted as being due to the model and not to the composition of the data. Hence, understanding the distribution of object size before partitioning will give a principled approach to choosing appropriate partitioning methods and a better understanding of evaluation results. The dataset statistics after data preparation and cleaning are summarized in Table 1.

As depicted in Figure 1(a), the majority of the samples in Dataset 1 are of medium size (58.6%), followed by small (30.1%) and large (11.3%) object samples. This means that the object-size distribution in Dataset 1 is imbalanced, i.e., medium is the most common object size and large is one of the less common sizes.

The distribution of Dataset 2 is similar, except that it is less concentrated; see Figure 1(b). Medium-object samples make up the largest share (44.5%), small-object samples take up 28.9%, and large-object samples 26.5%. The number of large objects is relatively higher in Dataset 2 than in Dataset 1, indicating a higher object-scale variation among the objects in Dataset 2.

The distribution of object size for Dataset 3 (WIDER FACE), however, is different from the two license plate datasets, as seen in Figure 1(c). Large object samples (48.3%), medium object samples (41.5%), and small object samples (10.2%) have the most samples. The addition of this dataset adds another object category and scene structure to the experimental scope, and enables the proposed partitioning framework to be analyzed under a different set of conditions.

Collectively, these dataset-level statistics support the main assumption of this study: object-size distribution is a relevant data characteristic that should be considered during dataset partitioning. When train, validation, and test subsets do not preserve the global object-size composition of the dataset, the resulting evaluation may become biased toward specific object scales. Therefore, size-aware partitioning provides a more controlled and methodologically consistent evaluation setting for analyzing YOLOv11n performance under different dataset composition conditions.

Table 1. Dataset statistics after data preparation and cleaning

Dataset	Original Images	Final/Cleaned Images	Total Bounding Boxes	Small Images	Medium Images	Large Images
Dataset 1	433	302	334	91	177	34
Dataset 2	5,368	5,368	7,852	1,554	2,390	1,424
Dataset 3	12,900	12,874	159,420	1,319	5,342	6,213

Note: The small, medium, and large counts are reported at the image level (i.e., the count of each size is reported for the largest annotated bounding box of each image). The total bounding-box count is simply a count of all the available object annotations, in contrast. In the case of Dataset 3, the total number of bounding boxes was derived from the available set of annotations, and 12,874 images survived the last data-cleaning step.

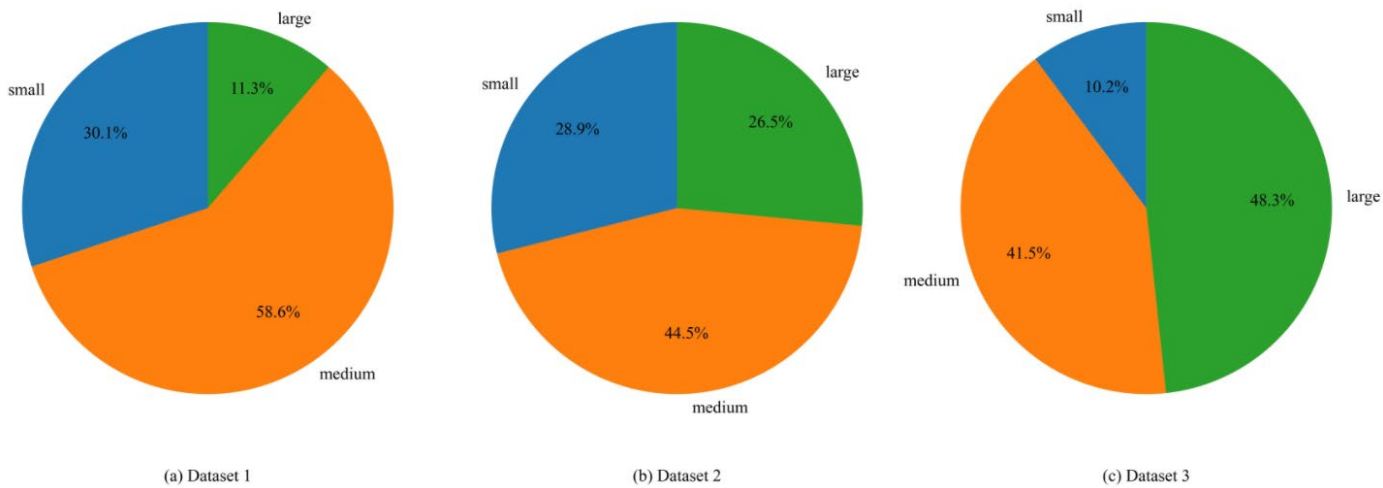


Figure 1. Object-size distribution of the three datasets after size categorization into small, medium, and large groups: (a) Dataset 1, (b) Dataset 2, and (c) Dataset 3 (WIDER FACE)

3.3 Size-aware dataset partitioning scenarios

Four partitioning scenarios were set up to analyze the impact of dataset order and object size composition on the evaluation of YOLOv11n. Each image/sample was partitioned so that all annotations for a given image were contained in a

single subset. To order and stratify by size, each image was encoded with the size of the largest annotated bounding-box area, based on the original image sizes. Two split-direction cases were considered for Scenarios 1–3. The first 70% of the corresponding ordered samples were used for training, the second 20% for validation, and the last 10% for testing in Case 1 (70/20/10). The first 10% of the ordered samples were used for testing, the next 20% for validation, and the remaining 70% for training in Case 2, which was called the reverse-direction case. Case 2 therefore preserves the same 70/20/10 subset sizes but reverses the train/test allocation direction. Two stratified states were evaluated in the case of Scenario 4: fine-grained quantized stratification (State 1) and COCO-level random stratified sampling without quantization (State 2).

Scenario 1: Natural-order sequential splitting

In Scenario 1, the cleaned dataset contained valid image-annotation pairs, which were sorted in ascending natural alphanumeric order by the image names. There was no sorting by size, shuffling, or allocation by randomization. The selected split-direction case was then applied sequentially to this ordered list. This scenario serves as a deterministic baseline representing dataset partitioning without explicit control of object-size composition.

Scenario 2: Size-ordered splitting from large to small

All the areas of the annotated bounding boxes for each image were measured in pixels with the natural dimensions of width and height for the image. If more than one annotated object was in the image, the largest sized bounding-box area was used as the image-level size representation. The whole dataset was then re-ordered from largest to smallest representative object size. The ordered list obtained was then applied to the selected split-direction case in turn. In this situation, there was no randomization.

Scenario 3: COCO-style size-group descending splitting

Images were initially categorized into small, medium, and large using the area of the image-level representative bounding box. Within each size category, samples were independently ordered in descending order according to bounding-box area. The selected split-direction case was then applied separately within each category. Last, the training, validation, and testing subsets for each category were combined into the final subsets. Scenario 3 differs from scenario 2 in that contributions are not only made by all three coarse size contributions, but also deterministic size ordering is retained within each coarse size contribution. No randomization was used.

Scenario 4: Size-aware stratified sampling

State 1: Fine-Grained Quantized Stratified Sampling: Samples were initially classified into the small, medium, and large categories as in COCO. Samples in each category were sorted by their representative bounding-box areas, and then divided into five approximately equal-frequency bins called "quantiles". A compromise at five bins was felt to be viable between the desire for fine-grained control on sample size and the need to have enough samples within each stratum. Samples in each bin were randomly shuffled using a seed of 42 and split into train/validation/test ratio specified. A final rebalancing step was applied within each COCO category before the corresponding category-specific subsets were merged, to account for rounding integer values within the individual bins, which can lead to subtly different numbers of samples in each of the categories in the merged subsets. This state offers a finer control over within-category object-size variation.

State 2: COCO-Level Random Stratified Sampling: Samples were initially classified into the small, medium, and large categories as in COCO. For each category, valid image-annotation pairs were randomly shuffled with a fixed seed value of 42 and assigned directly based on its target train/validation/test ratio. No further quantile subdivision/fine-grained binning was performed. The resulting portions were now combined according to category and the final training, validation and test portions created. State 2 is therefore a coarse baseline (non-quantized random-stratified) that can be compared to the finer quantization used in State 1 within each category.

Comparing State 1 with State 2 isolates the effect of quantization within each COCO size category. In both states, the same category (small, medium, large) hierarchy is maintained at the COCO level, with the categories being randomized within each state. The conventional sampling is used directly in State 2, and in State 1, each of these size categories is subdivided into five approximately equal-frequency bins, and the conventional sampling is applied at the bin level. This is intended to minimize the chances that samples from just one area of the within-category size range will be combined in a subset. The objective of State 1 is not to achieve a higher accuracy of the detector, but to achieve a finer control of the composition of the object-size within each coarse COCO category. The value of the practical use is evaluated based on the detector's performance and the representativeness of the split.

Overall, the four scenarios provide a controlled framework for examining partition-induced evaluation bias. Scenario 1 represents deterministic natural-order sequential splitting without explicit control of object-size composition. Scenario 2 evaluates the effect of global size-ordered splitting from larger to smaller representative objects. Scenario 3 examines COCO-style size grouping combined with deterministic descending size ordering within each category. Scenario 4 evaluates two size-aware stratified approaches: a fine-grained quantized strategy and a coarse COCO-level random-stratified baseline. Together, these scenarios enable a controlled analysis of how differences in object-size composition across training, validation, and testing subsets can influence YOLOv11n evaluation.

3.4 Split distribution similarity analysis using Kullback-Leibler divergence and normalized Earth Mover's Distance

After generating the training, validation, and testing subsets using the proposed partitioning scenarios, the representativeness of each split was analyzed according to object-size distribution and object-density distribution. In this study, split representativeness refers to the extent to which each generated subset preserves a distribution close to the global structure of the full dataset. This analysis is important because a detector trained and evaluated on subsets with different data compositions may produce performance results that partly reflect partition-induced distribution shift rather than detector capability alone.

Object-size representativeness was evaluated at the annotation level. Unlike the image-level representation used during partitioning, where the largest annotated bounding box was used to represent each image, this analysis considered all

annotated bounding boxes within the training, validation, and testing subsets. For each annotation, the normalized bounding-box dimensions were converted to pixel dimensions using the native width and height of the corresponding image, and the bounding-box area was calculated in pixels. Each annotated object was then categorized according to the COCO-style thresholds as small when $A < 32^2$, medium when $32^2 \leq A < 96^2$, and large when $A \geq 96^2$, where A denotes the bounding-box area [9]. Thus, image-level largest-box representation was used for dataset partitioning, whereas annotation-level distributions containing all bounding boxes were used for the split-representativeness analysis. Distribution similarity analysis was done using the KL divergence method and the normalized EMD method. The two levels of representation are used intentionally, as the two analysis types (dataset partitioning, split-representativeness) are based on different units. Each image and all of its annotations have to be in one subset; hence, one scalar value is needed for each image, and the largest bounding-box area of all the annotations is used as the representative size for the image. KL divergence and normalized EMD, on the other hand, measure the object-level composition that is actually contained in each generated subset, which means that the use of all bounding-box annotations is allowed. In multi-object images, the smaller objects in the image, therefore, do not directly impact the partition assignment - and even if there is good control on the image-level allocation, there can be a distribution mismatch at the annotation level. Therefore, the KL-EMD analysis is performed to reveal this mismatch, rather than to make the same representation as is performed in the partitioning process.

In addition, object-density distribution was analyzed at the image level by grouping images into low, medium, and high-density categories according to the number of annotated objects per image. In this study, low-density images contain one annotated object, medium-density images contain two to three annotated objects, and high-density images contain four or more annotated objects. These two representations provide complementary views of split composition: object-size distribution describes the scale of annotated objects, whereas object-density distribution describes how objects are distributed across images. Only the object-size KL divergence and normalized EMD values were used to construct the final composite KL-EMD distribution-balance score, whereas object-density divergence was retained as a supplementary diagnostic measure.

For each generated split, the count values were converted into normalized probability distributions. Let P denote the distribution of a generated split and Q denote the global distribution of the full dataset. Accordingly, P was defined separately for the training, validation, and testing subsets, whereas Q remained the fixed global distribution of the corresponding full dataset. For category i, the normalized probabilities are computed as:

$$p_i = \frac{c_i}{\sum_{j=1}^{k} c_j} \tag{1}$$

$$q_i = \frac{g_i}{\sum_{j=1}^{k} g_j} \tag{2}$$

where, c_i is the count of category i in the generated split, g_i is the count of the same category in the full dataset, and k is the number of ordered categories. In this study, $k = 3$ for both object-size categories (small, medium, and large) and object-density categories (low, medium, and high).

KL divergence was used to measure the disagreement between the split distribution and the global distribution [22, 23]. To avoid zero-probability problems during computation, a small epsilon value ($\varepsilon = 10^{-12}$) was applied before calculating the logarithmic ratio. KL Divergence is computed as:

$$D_{KL}(P||Q) = \sum_{i=1}^{k} p_i log\left(\frac{p_i + \varepsilon}{q_i + \varepsilon}\right) \tag{3}$$

A lower KL value indicates stronger agreement with the global distribution, while a higher value indicates greater distributional mismatch.

EMD was also used because the object-size and object-density categories have an ordered structure [24]. In this study, EMD was computed using the cumulative distribution difference between two normalized ordered distributions. The cumulative distribution of P up to category I is defined as:

$$F_P(i) = \sum_{j=1}^{i} p_j \tag{4}$$

Similarly, the cumulative distribution of Q is defined as:

$$F_Q(i) = \sum_{j=1}^{i} q_j \tag{5}$$

The normalized EMD is computed as:

$$EMD_{\text{norm}}(P, Q) = \frac{1}{k-1} \sum_{i=1}^{k-1} \left| F_{P(i)} - F_{Q(i)} \right| \tag{6}$$

KL divergence measures distributional disagreement, whereas normalized EMD reflects the amount of probability mass that must move across the ordered categories to bring the split distribution closer to the global distribution.

For each partitioning condition, KL divergence and normalized EMD were calculated independently for the training, validation, and testing subsets relative to the global object-size distribution. The three split-level values were then averaged as follows:

$$D_{\text{KL,avg}} = \frac{D_{KL}^{\text{train}} + D_{KL}^{\text{val}} + D_{KL}^{\text{test}}}{3} \tag{7}$$

$$D_{\text{nEMD,avg}} = \frac{D_{\text{nEMD}}^{\text{train}} + D_{\text{nEMD}}^{\text{val}} + D_{\text{nEMD}}^{\text{test}}}{3} \tag{8}$$

Because KL divergence is unbounded whereas normalized EMD is bounded, the average KL divergence was transformed using the following monotonic bounding function:

$$D_{\text{KL,bounded}} = \frac{D_{\text{KL,avg}}}{1 + D_{\text{KL,avg}}} \tag{9}$$

This transformation preserves the ordering of KL divergence values while mapping them to a bounded scale, enabling their combination with normalized EMD in the final composite score. The KL divergence and normalized EMD are

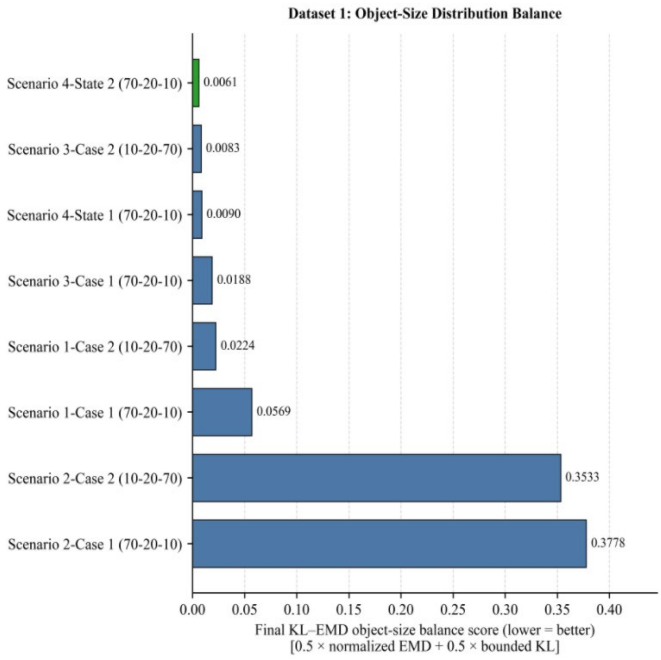
complementary to each other. KL divergence measures disagreement with the probabilities of each split of the space generated and the distribution of probabilities over the global data; the ordered nature of the small, medium, and large categories is taken into account by the normalized EMD, as is the amount of probability mass shifted along the ordered spectrum. These two measures are then calculated independently in advance of being combined. They are not intended to be used in place of the conventional separate divergence measures, but rather in addition to these to provide a single-purpose overall statement about split representativeness. The bounded-KL transformation in Eq. (9) is necessary since KL divergence is unbounded, whereas normalized EMD is bounded, before making an equal-weight combination.

Equal weighting was used so that KL divergence and normalized EMD contributed symmetrically to the final diagnostic score, without assigning greater importance to either measure. Lower values of ($S_{\mathrm{KL-EMD}}$) indicate closer agreement between the generated subsets and the global object-size distribution. Object-density divergence and split-ratio error were retained as supplementary diagnostic measures but were not included in the final composite KL-EMD object-size balance score.

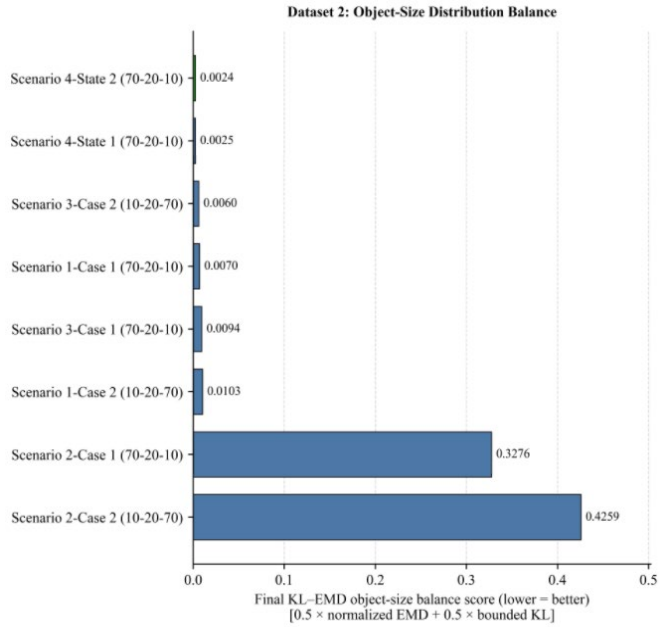
The final composite KL-EMD object-size distribution-balance score was calculated using equal weights:

$$S_{\mathrm{KL-EMD}} = 0.5D_{\mathrm{nEMD,avg}} + 0.5D_{\mathrm{KL,bounded}} \quad (10)$$

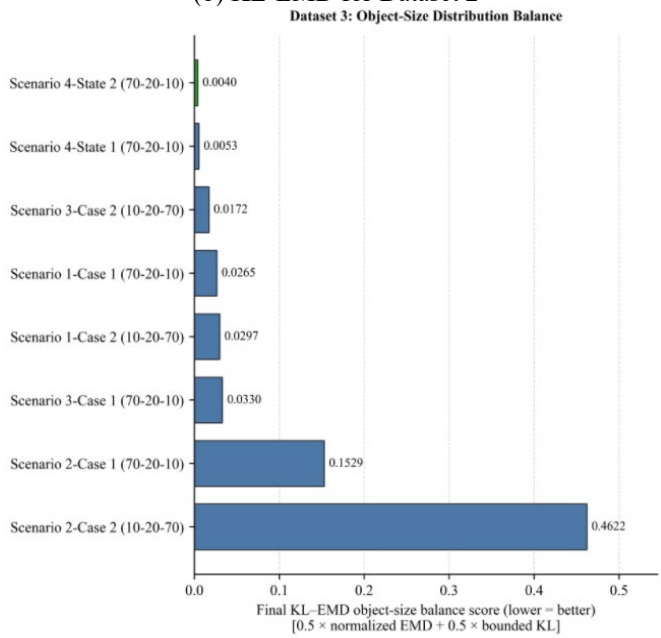
KL divergence, normalized EMD, and the resulting composite KL-EMD score were used only as diagnostic measures of split representativeness and not as substitutes for detector-performance metrics. Consequently, a lower distribution-balance score indicates that the generated subsets more closely preserve the global object-size distribution, but it does not imply that the corresponding experiment must achieve the highest mean average precision (mAP)@0.50:0.95. Detector performance can also be influenced by other data characteristics, including background complexity, occlusion, illumination, object density, viewpoint variation, and annotation quality.



(a) KL-EMD for Dataset 1



(b) KL-EMD for Dataset 2



(c) KL-EMD for Dataset 3

Figure 2. Composite KL-EMD object-size distribution-balance scores for the three datasets: (a) Dataset 1, (b) Dataset 2, and (c) Dataset 3 (WIDER FACE)
Note: Lower scores indicate closer agreement between the generated split distributions and the global object-size distribution. KL = Kullback-Leibler; EMD = Earth Mover's Distance.

Therefore, the distribution-balance analysis is interpreted together with mAP@0.50:0.95 rather than as a direct predictor of detection accuracy. Figure 2 compares the composite KL-EMD object-size distribution-balance scores across the evaluated partitioning conditions for the three datasets, where lower values indicate greater split representativeness.

3.5 Model architecture and training

YOLOv11n was selected as a contemporary one-stage detection framework for the controlled experiments, enabling the same detector architecture to be applied consistently across the large number of dataset-partitioning conditions examined in this study.

The purpose of using YOLOv11n was not to propose or test a new detector architecture, but to provide a fixed experimental model while keeping the architecture, input resolution, training duration, and evaluation protocol consistent. In this controlled environment, partitioning the datasets and composing objects of different sizes are the main experimental variables, and thus the amount of difference between the different conditions can be understood without introducing architectural differences. In the present controlled analysis, however, YOLOv11n does not represent evidence of a unique effect of many object-detection architectures, but instead is a representative modern detector of this type, one-stage. It is therefore important to investigate whether these trends also occur in other detector families.

As the experiments were carried out with one architecture, the amount of the partitioning effect observed should not be taken as a guarantee for other architectures like Faster R-CNN or SSD to be equally applicable without modification. It is therefore crucial to investigate the existence of such trends in other detector families, in order to validate the hypothesis made in this project.

All training, validation, and testing procedures were conducted using a standardized Python-based pipeline to ensure consistency across datasets and partitioning conditions. A fixed input resolution of 640 × 640 pixels and a training duration of 100 epochs were used for all experiments. The same YOLOv11n architecture, training configuration, and evaluation protocol were maintained across all generated dataset splits. This controlled experimental setup was adopted so that differences in detector performance could be interpreted primarily in relation to dataset partitioning and split composition rather than changes in model architecture or training conditions.

Three model-training seeds (20, 42, and 60) were used to evaluate stability when training the model for each partitioning scenario. The dataset partitions were maintained constant throughout runs, and only the stochasticity in the training process was varied. The model-training seed was used independently for Scenario 4, and the partition-allocation seed was fixed at 42. The mAP@0.50:0.95 values obtained are shown as the mean and standard deviation over the three runs.

3.6 Evaluation metrics

The standard object detection metrics, including precision, recall, F1-score, mAP@0.50, and mAP@0.50:0.95, were used to assess the performance of YOLOv11n. Precision is the ratio of the number of true positive detections to the number of positive detections that are predicted, and recall is the ratio of the number of true positive detections to the number of objects that are actually positive. The F1-score is the harmonic mean of precision and recall and is a good measure of the correctness of detection.

mAP@0.50 is the mean Average Precision computed at one specific intersection over union (IoU) threshold (usually 0.50). This metric is helpful to measure if the predicted boxes overlap with the ground- truth boxes enough, but might overestimate performance for bad localization quality in more stringent IoU requirements. The primary evaluation metric is mAP@0.50:0.95, which averages precision across IoU thresholds from 0.50 to 0.95 and provides a stricter assessment of detection and localization quality [25].

This study focuses on split representativeness and the distribution of small, medium, and large objects across subsets rather than scale-specific detector performance. Therefore, mAP@0.50:0.95 is interpreted together with the composite KL-EMD score, which serves only as a diagnostic measure of split representativeness.

3.7 Experimental setup

Experiments were all run on a cloud-based Python environment via Google Colab and using graphics processing unit (GPU) acceleration. All datasets and partitioning conditions were processed using the same experimental pipeline to ensure consistency and reproducibility. The training, testing, and validation subsets were created for each dataset, following the YOLO annotation format, and configuration files were generated for training and evaluation in YOLOv11n.

In all experiments, the detector and training configuration were kept fixed as described in Section 3.5, and the evaluation protocol was kept fixed as described in Section 3.6. The only change in the experiments was the partitioning (and split composition) of the dataset.

4. RESULTS AND DISCUSSION

4.1 Results

The experimental results obtained using YOLOv11n on the three datasets are presented in Tables 2–4. To ensure a fair comparison across the different partitioning conditions, the same detector architecture, input resolution of 640 × 640 pixels, training duration of 100 epochs, and evaluation protocol were maintained across all experiments. The results demonstrate that differences in object-size composition among the training, validation, and testing subsets can substantially influence the reported YOLOv11n performance. Tables 2–4 report the detailed Seed-42 reference-run metrics, while Table 5 summarizes the multi-seed stability of the primary mAP@0.50:0.95 metric.

Table 5 reports the mean mAP@0.50:0.95 and standard deviation across three training seeds for each partitioning condition.

For Dataset 1, Scenario 1 produced relatively stable mAP@0.50:0.95 values of 0.497 and 0.483 for Case 1 and Case 2, respectively. In contrast, Scenario 2, which applies global size ordering from larger to smaller representative objects, resulted in a substantial performance decrease, with mAP@0.50:0.95 values of 0.124 and 0.220. Scenario 3 partially recovered performance by preserving contributions from the COCO-style size categories across the generated subsets. The highest mAP@0.50:0.95 for Dataset 1 was obtained by Scenario 4 State 1 at 0.523, as shown in Figure 3. Therefore, among the evaluated conditions for this dataset, the fine-grained quantized stratified state achieved the highest detector performance.

A related pattern was observed for Dataset 2. Scenario 1 achieved mAP@0.50:0.95 values of 0.451 and 0.514 for Case 1 and Case 2, respectively. Scenario 2 produced substantially lower performance, particularly in Case 2, where mAP@0.50:0.95 decreased to 0.005. This extreme reduction is associated with the strong mismatch in object-size composition introduced by the reverse-direction allocation of the size-ordered sequence.

Table 2. YOLOv11n detection performance for the Seed-42 reference run across partitioning conditions for Dataset 1

Metric	Scenario 1 Case 1	Scenario 1 Case 2	Scenario 2 Case 1	Scenario 2 Case 2	Scenario 3 Case 1	Scenario 3 Case 2	Scenario 4 State 1	Scenario 4 State 2
mAP@0.50:0.95	0.497367	0.483453	0.123639	0.219915	0.366973	0.366674	0.523274	0.420906
mAP@0.50	0.898160	0.918334	0.297605	0.399236	0.674309	0.808457	0.819992	0.785393
F1-score	0.889564	0.913142	0.438280	0.496637	0.704799	0.785248	0.778296	0.815085
Precision	0.900608	0.930753	0.592315	0.973632	0.683673	0.782635	0.799248	0.809728
Recall	0.878788	0.896185	0.347826	0.333333	0.727273	0.787879	0.758415	0.820513

Note: mAP = mean average precision.

Table 3. YOLOv11n detection performance for the Seed-42 reference run across partitioning conditions for Dataset 2

Metric	Scenario 1 Case 1	Scenario 1 Case 2	Scenario 2 Case 1	Scenario 2 Case 2	Scenario 3 Case 1	Scenario 3 Case 2	Scenario 4 State 1	Scenario 4 State 2
mAP@0.50:0.95	0.450905	0.514376	0.130585	0.005411	0.358599	0.312935	0.460224	0.476905
mAP@0.50	0.848614	0.863911	0.464444	0.006659	0.735884	0.637270	0.848278	0.856966
F1-score	0.831557	0.837415	0.564822	0.009990	0.740310	0.710071	0.824865	0.827921
Precision	0.859927	0.870951	0.642951	0.075729	0.767900	0.768063	0.857249	0.845991
Recall	0.805000	0.806366	0.503623	0.005348	0.714634	0.660221	0.794839	0.810606

Note: mAP = mean average precision.

Table 4. YOLOv11n detection performance for the Seed-42 reference run across partitioning conditions for Dataset 3

Metric	Scenario 1 Case 1	Scenario 1 Case 2	Scenario 2 Case 1	Scenario 2 Case 2	Scenario 3 Case 1	Scenario 3 Case 2	Scenario 4 State 1	Scenario 4 State 2
mAP@0.50:0.95	0.385278	0.333491	0.177627	0.185936	0.254463	0.349361	0.371399	0.364110
mAP@0.50	0.673476	0.631743	0.426738	0.356370	0.512727	0.656763	0.677807	0.662562
F1-score	0.695494	0.665922	0.485753	0.426629	0.560070	0.677779	0.694597	0.683297
Precision	0.853218	0.846207	0.682291	0.615630	0.753807	0.841548	0.843078	0.847112
Recall	0.586985	0.548964	0.377121	0.326417	0.445557	0.567368	0.590584	0.572572

Note: mAP = mean average precision.

Table 5. Multi-seed stability of mAP@0.50:0.95 across partitioning conditions

Partitioning Condition	Dataset 1 mAP@0.50:0.95 Mean ± SD	Dataset 2 mAP@0.50:0.95 Mean ± SD	Dataset 3 mAP@0.50:0.95 Mean ± SD
Scenario 1 - Case 1	0.496 ± 0.010	0.451 ± 0.006	0.385 ± 0.006
Scenario 1 - Case 2	0.482 ± 0.010	0.513 ± 0.007	0.333 ± 0.006
Scenario 2 - Case 1	0.123 ± 0.007	0.130 ± 0.004	0.178 ± 0.005
Scenario 2 - Case 2	0.219 ± 0.011	0.005 ± 0.002	0.186 ± 0.005
Scenario 3 - Case 1	0.365 ± 0.011	0.358 ± 0.006	0.254 ± 0.006
Scenario 3 - Case 2	0.366 ± 0.007	0.313 ± 0.007	0.349 ± 0.006
Scenario 4 - State 1	0.521 ± 0.011	0.460 ± 0.007	0.371 ± 0.006
Scenario 4 - State 2	0.420 ± 0.010	0.477 ± 0.007	0.364 ± 0.005

Note: Values are reported as mean ± standard deviation across three model-training seeds (20, 42, and 60). mAP = mean average precision, SD = standard deviation.

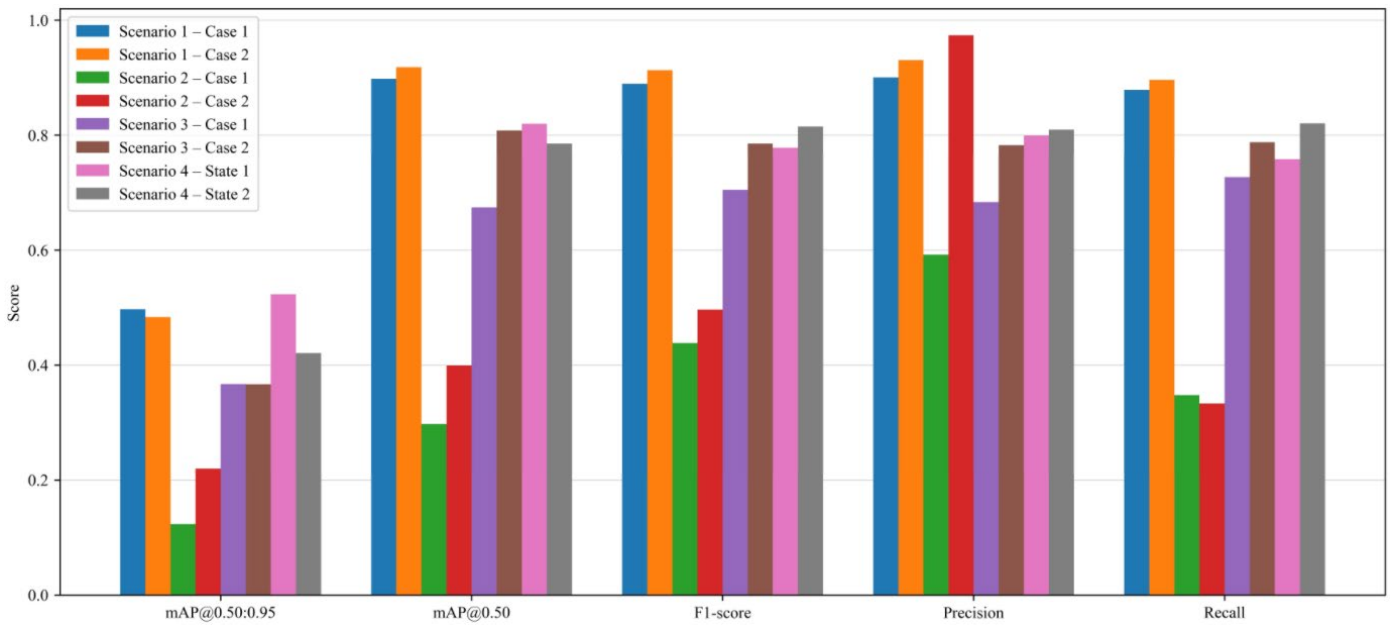


Figure 3. YOLOv11n detection performance across partitioning conditions for Dataset 1

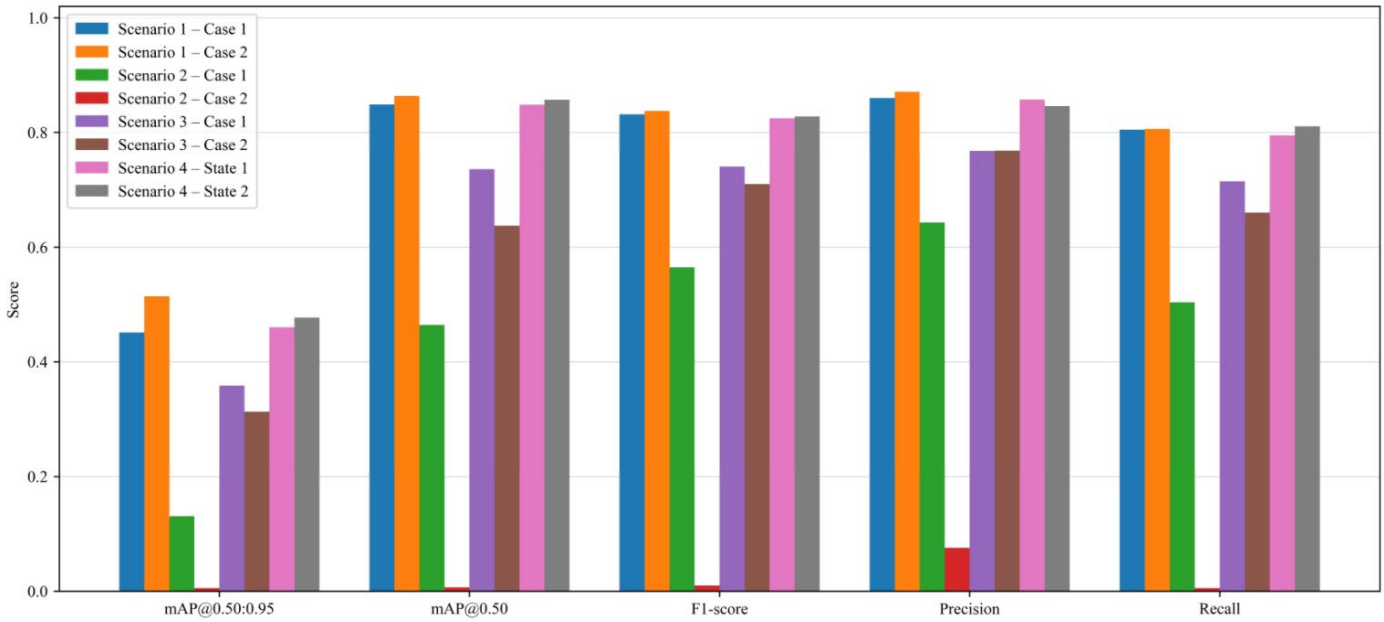


Figure 4. YOLOv11n detection performance across partitioning conditions for Dataset 2

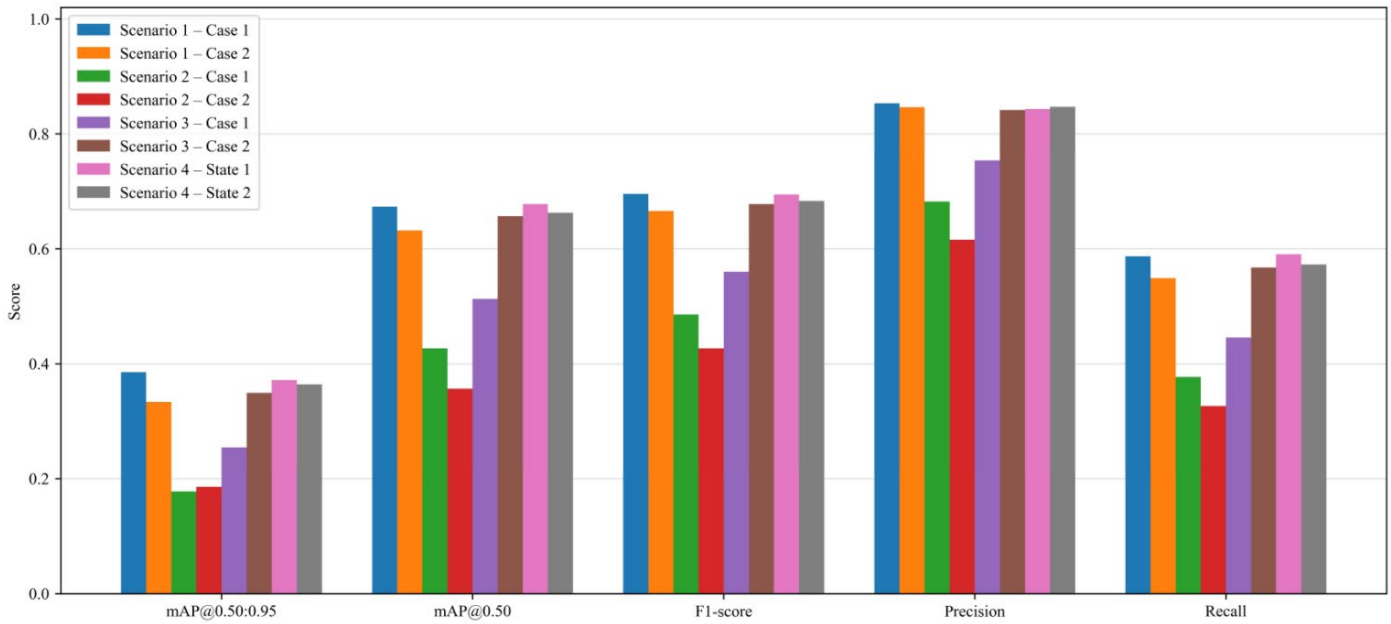


Figure 5. YOLOv11n detection performance across partitioning conditions for Dataset 3

Scenario 3 showed partial recovery relative to Scenario 2, whereas Scenario 4 produced competitive performance, with mAP@0.50:0.95 values of 0.460 for State 1 and 0.477 for State 2, as shown in Figure 4. Although neither Scenario 4 state exceeded Scenario 1 Case 2 in detector performance, both avoided the severe degradation observed under the strongly size-ordered condition.

Dataset 3 (WIDER FACE) was included to extend the experimental evaluation beyond license-plate detection. As shown in Table 4, Scenario 1 Case 1 achieved the highest mAP@0.50:0.95 value of 0.385, followed by Scenario 4 State 1 at 0.371 and Scenario 4 State 2 at 0.364. Scenario 2 again produced the lowest results, with mAP@0.50:0.95 values of 0.178 and 0.186 for Case 1 and Case 2, respectively, as shown in Figure 5. Thus, although Scenario 4 did not outperform Scenario 1 Case 1 on Dataset 3, both size-aware stratified states remained competitive and avoided the marked degradation associated with global size-ordered splitting. The results therefore indicate that sensitivity to partition-induced object-size mismatch is not restricted to the two license-plate datasets. The multi-seed analysis assessed whether the observed differences were stable across training seeds. Across the evaluated conditions, the standard deviations remained relatively small compared with the larger performance differences produced by strongly size-biased partitioning, supporting the interpretation that the main trends were associated primarily with split composition rather than seed variation.

The composite KL-EMD results provide complementary information regarding split representativeness. As shown in Figure 2(a), Dataset 1 achieved the lowest distribution-balance score under Scenario 4 State 2 (0.0061), followed by Scenario 3 Case 2 (0.0083) and Scenario 4 State 1 (0.0090), whereas Scenario 2 Case 1 produced the largest mismatch (0.3778). For Dataset 2, Figure 2(b) shows that Scenario 4 State 2 (0.0024) and State 1 (0.0025) provided the closest agreement with the

global object-size distribution, whereas Scenario 2 Case 2 produced the largest mismatch (0.4259). Similarly, for Dataset 3, Figure 2(c) shows that Scenario 4 State 2 and State 1 achieved the lowest scores of 0.0040 and 0.0053, respectively, while Scenario 2 Case 2 produced the highest mismatch score of 0.4622. These results indicate that the size-aware stratified conditions generally produced more representative object-size distributions than the strongly size-ordered conditions.

Importantly, greater distributional similarity did not necessarily correspond to the highest detector performance. For example, Scenario 1 Case 1 achieved the highest mAP@0.50:0.95 on Dataset 3 despite having a composite KL-EMD score higher than those of both Scenario 4 states. Therefore, the composite KL-EMD score should be interpreted as a diagnostic measure of split representativeness rather than as a direct predictor of detection accuracy. Detector performance may additionally be influenced by other data characteristics, including object density, background complexity, illumination, occlusion, viewpoint variation, and annotation quality.

4.2 Discussion

The three datasets all highlight the influence of the composition of different object sizes in the training, validation, and testing subsets on the results of the YOLOv11n assessment. The "small", "medium", and "large" categories are used in this study mainly to describe the composition and split representativeness of each of the dataset groups and not to compare the performance of the detectors for each individual object scale. The overall performance of the detector is therefore evaluated mostly with respect to the mAP@0.50:0.95, complemented by mAP@0.50, precision, recall, and F1-score.

For Datasets 1 and 2, Scenario 1 produced relatively stable performance across the two cases, indicating that the natural-order sequence used in these datasets did not introduce a severe object-size mismatch between the generated subsets. However, this result should not be interpreted as evidence that natural-order splitting is inherently reliable, because its effectiveness depends on the underlying ordering of the stored samples. In contrast, Scenario 2 produced substantial degradation in both datasets because globally ordering samples from larger to smaller representative objects concentrated different scale ranges in different subsets.

The worst case was in Dataset 2 of Scenario 2 Case 2, with mAP@0.50:0.95 dropping to 0.005. In this “reverse” allocation, the first 10% of the ordered samples were used for testing, the next 20% for validation, and the remaining 70% were allocated to training, resulting in notable differences in scale compositions between the training and test sets. This decrease in recall and localization quality is consistent with the scale-distribution mismatch introduced by partitioning rather than changes in the YOLOv11n architecture or training configuration.

Scenario 3 partially mitigated the degradation observed in Scenario 2 by first separating samples into COCO-style small, medium, and large categories before applying deterministic size ordering within each category. This procedure preserved contributions from all three coarse size groups across the generated subsets and therefore reduced the extreme scale separation produced by global size ordering. However, deterministic ordering within each category may still fail to preserve other data characteristics, including background complexity, illumination, viewpoint, occlusion, and object density.

In general, the degradation that occurs with heavily size-biased partitions is a result of the mismatches between scale characteristics represented during training and scale characteristics during evaluation. The larger the object, the more spatial and structural information that can be obtained, while the smaller the object is, the fewer pixels there are, and the less detail in the features. If the training and testing subsets focus on very different regions of the scale range, the transfer of the learned representations and localization behavior across subsets may be reduced. Consequently, these changes in performance may be caused by differences in scale distribution induced by the partition and manifest themselves as a performance difference without altering the detector architecture and training configuration.

The lower mAP@0.50:0.95 values relative to mAP@0.50 are consistent with the stricter localization requirements of the former metric. The lower mAP@0.50:0.95 values compared with mAP@0.50 reflect the stricter localization requirements imposed by evaluation across multiple IoU thresholds. This is especially true for small objects, where a shift of only a few pixels may represent a substantial fraction of the object extent and reduce IoU. A detector's performance is thus preserved with mAP@0.50, but it is likely to suffer more in terms of mAP@0.50:0.95 when scale-distribution mismatch exists. This is why mAP@0.50:0.95 is considered the major performance index in the present study.

Scenario 4 provided the most controlled size-aware partitioning conditions. For Dataset 1, State 1 achieved the highest mAP@0.50:0.95 value of 0.523, indicating that fine-grained quantized stratification was effective under this dataset configuration. For Dataset 2, both State 1 and State 2 avoided the severe degradation observed in Scenario 2 and achieved competitive mAP@0.50:0.95 values of 0.460 and 0.477, respectively. The highest mAP@0.50:0.95 for Dataset 3 (WIDER FACE) was found for Scenario 1 Case 1 (0.385), and Scenario 4 State 1 (0.371) and State 2 (0.364) were competitive. In general, then, the size-aware stratified conditions offered better control of split composition and did not show a marked drop in detector performance compared to the global size ordering conditions, but did not always yield the best detector performance for the three datasets.

The effectiveness of the within-category quantization is further illustrated by the direct comparison between State 1 and State 2 of Scenario 4. State 1 achieved a higher mAP@0.50:0.95 than State 2 on Dataset 1 (0.523 versus 0.421) and Dataset 3 (0.371 versus 0.364), whereas State 2 achieved a slightly higher value on Dataset 2 (0.477 versus 0.460). State 2, on the other hand, had the lowest KL-EMD composite score for all three datasets. The results do not necessarily mean that the added quantization in State 1 is a better alternative than the normal stratified sampling in general. In the latter case, however, finer control over the distribution of sizes in the categories is possible, and State 2 is a simpler coarse stratification and, in the present experiments, has a greater preservation of the global COCO-level size distribution. It is not a continual improvement in performance but rather a compromise that emerges in a comparison. This distinction is important because the composite KL-EMD score is based on the three COCO size categories and primarily reflects the similarity in the coarse category distribution, while the additional quantile structure in State 1 is within the coarse category distribution. This difference in representation is also

considered as context in interpreting the results of scenario 4. A finer quantization of the representative image may not lead to a better distribution-balance score of the distribution of the object annotations (in the image-level representation), because the distribution-balance score is computed from all of the object annotations. This is due to the different units in which the image was allocated and diagnosed after the partition, and it does not contradict. Another drawback of the largest-bounding-box representation is that there is no simple image-level representation for the smaller objects that might appear together in a scene, and other representations for multi-object scenes could be explored in future work.

The composite KL-EMD results complement the detector-performance metrics. In general, size-aware stratified conditions kept the global object-size distribution closer to the original than the strongly size-ordered conditions across all three datasets and, in some cases, the distribution mismatch was lower, while Scenario 2 showed higher mismatch in several cases. However, it was not always true that the more similar the distribution was, the better the detector would perform. For instance, Scenario 1 Case 1 had a composite KL-EMD score of 0.0265 on Dataset 3, with a very high mAP@0.50:0.95 score, even though both Scenario 4 states had lower KL-EMD scores. Finally, the composite KL-EMD score should be viewed as a diagnostic indicator for split representativeness, rather than an indicator of detection accuracy, as other data characteristics and scene-related factors may affect detector performance.

The results from an information-systems point of view show that partitioning should not simply be seen as a pre-processing step, but it is a part of data governance and evaluation design. Training, validation, and testing subsets are internal data resources that are managed and used to provide the evidence for comparisons, selection, and reliability assessment of models. If these subsets are not evenly distributed in terms of important data characteristics, the performance estimates may lead to misleading model selection or downstream decision-making. Assessing split representativeness can thus aid in improving the transparency of data management, in the evaluation of AI system behavior and in the intelligent deployment of object-detection components in intelligent information systems. This fits the data-centric view of AI where data is a part of the AI lifecycle and needs to be prepared, evaluated, and maintained [6].

In addition, the WIDER FACE dataset is added to the experimental study, and it demonstrates that split composition sensitivity can be present under significantly different object-scale, object-density, and scene conditions. However, the current evidence available is limited to three datasets and two object categories—license plates and faces. It does not, therefore, indicate that partitioning effects reported will be true for other object-detection domains. The framework can be adapted to other databases for which a meaningful scale partitioning effect and bounding-box annotation exists, but for which the partitioning effect size and direction may vary depending on the image resolution, scale range of objects, object density, image complexity, and the quality and architecture of the annotations. The present experiments would then not be able to be used to infer the applications of autonomous driving, surveillance, or medical image detection. The COCO-style size thresholds may require adjustment for other domains where the sizes of the images or objects are very different. Within this context, appropriate transferability is considered a requirement that the proposed framework needs to be validated in the future. To verify the stability of differences in the multi-seed analysis, the stochasticity of the training was added. For all the conditions evaluated, the standard deviations were somewhat small compared to the size-biased partitioning effects that caused larger performance differences; this reinforces the conclusion that the major trends were due to differences in partitioning split composition more than differences in seeds.

Under the evaluated conditions, strongly size-ordered partitioning generally led to marked performance losses in several cases, while size-aware stratified partitioning reduced object-size distribution mismatch while maintaining competitive detector performance. This did not lead to the best mAP@0.50:0.95 for the partitioning condition with the most representative object-size distribution. Thus, the assessment of detector performance and the split representativeness can be seen as complementary because the mAP@0.50:0.95 score reflects detector performance while the composite KL-EMD score reflects the similarity of split distributions to the global object-size distribution. The results consequently justify the joint consideration of both aspects without any rigid reciprocity between increased distributional similarity and increased detection accuracy.

5. CONCLUSION

This study examined how object-size-aware dataset partitioning influences YOLOv11n evaluation under a controlled fixed-detector setting across three real-world object detection datasets, including two license plate datasets and WIDER FACE. Within the evaluated partitioning conditions, the results indicate that the object-size composition of the training, validation, and testing subsets can substantially affect the reported detector performance. Size-ordered splitting was associated with marked performance degradation in several cases, including the severe reduction observed in Dataset 2 under Scenario 2 Case 2, where mAP@0.50:0.95 decreased to 0.005. In contrast, the size-aware stratified conditions generally produced lower object-size distribution mismatch and often maintained competitive detection performance across the evaluated datasets. However, the partitioning condition with the lowest composite KL-EMD object-size distribution-balance score did not necessarily achieve the highest mAP@0.50:0.95, indicating that the proposed distribution measures should be interpreted as diagnostics of split representativeness rather than predictors of detector accuracy. When evaluating under the YOLOv11n setting, partitioning should not be viewed as a common preprocessing step, but rather as an evaluation-design and data-governance consideration. The results are only applicable for the tested conditions and not for the architecture independent behavior. Further research needs to be carried out with other detector architectures, extended and heterogeneous datasets, and other data characteristics including object density, occlusion, viewpoint variation, illumination, and the quality of annotation.

ACKNOWLEDGMENT

The authors would like to thank the University of Information Technology and Communications (UoITC), Baghdad, Iraq for academic support.

REFERENCES

[1] Aamir, M., Rahman, Z., Abro, W.A., Bhatti, U.A., Dayo, Z.A., Ishfaq, M. (2023). A progressive approach to generic object detection: A two-stage framework for image recognition. Computers, Materials & Continua, 75(3): 6351-6373. https://doi.org/10.32604/cmc.2023.038173

[2] Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J. (2023). Object detection in 20 years: A survey. Proceedings of the IEEE, 111(3): 257-276. https://doi.org/10.1109/JPROC.2023.3238524

[3] Jrondi, Z., Moussaid, A., Hadi, M.Y. (2024). Exploring end-to-end object detection with transformers versus YOLOv8 for enhanced citrus fruit detection within trees. Systems and Soft Computing, 6: 200103. https://doi.org/10.1016/j.sasc.2024.200103

[4] Jeba, A.P., David, J.D., Mathew, E.A., Mathew, M., Harish, M., Joel, J.M. (2026). A hybrid approach for transforming retail and customer experience with user motion analytics using YOLOv4 and pattern mining. Journal of Trends in Computer Science and Smart Technology, 8(3): 452-472. https://doi.org/10.36548/jtcsst.2026.3.002

[5] Tadjine, C., Ouafi, A., Taleb-Ahmed, A., El Hillali, Y. (2025). Class-specific dataset splitting for YOLOv8: Improving real-time performance in NVIDIA Jetson Nano for faster autonomous forklifts. In Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025), Porto, Portugal, 1: 788-793. https://doi.org/10.5220/0013308700003905

[6] Zha, D., Bhat, Z.P., Lai, K.H., et al. (2025). Data-centric artificial intelligence: A survey. ACM Computing Surveys, 57(5): 1-42. https://doi.org/10.1145/3711118

[7] Khalili, B., Smyth, A.W. (2024). SOD-YOLOv8—Enhancing YOLOv8 for small object detection in aerial imagery and traffic scenes. Sensors, 24(19): 6209. https://doi.org/10.3390/s24196209

[8] Li, J., Zhang, X., Zou, H., et al. (2025). COUNTS: Benchmarking object detectors and multimodal large language models under distribution shifts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, pp. 9186-9198. https://doi.org/10.1109/CVPR52734.2025.00858

[9] Lin, T.Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common objects in context. Lecture Notes in Computer Science, 8693: 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[10] Chen, S., Peng, Y., Liu, Y., Wang, P., Liu, T. (2025). Enhancing YOLOv11 with large kernel attention and multi-scale fusion for accurate small and multi-lesion bone tumor detection in radiographs. Diagnostics, 15(16): 1988. https://doi.org/10.3390/diagnostics15161988

[11] Mital, N., Malzard, S., Walters, R., De Melo, C.M., Rao, R., Nockles, V. (2025). Improving object detection by modifying synthetic data with explainable AI. In Artificial Intelligence for Security and Defence Applications III, Madrid, Spain, pp. 136790T. https://doi.org/10.1117/12.3069879

[12] Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixé, L., Alvarez, J.M. (2022). Not all labels are equal: Rationalizing the labeling costs for training object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, pp. 14472-14481. https://doi.org/10.1109/CVPR52688.2022.01409

[13] Torralba, A., Efros, A.A. (2011). Unbiased look at dataset bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, pp. 1521-1528. https://doi.org/10.1109/CVPR.2011.5995347

[14] Fernández Llorca, D., Frau, P., Parra, I., Izquierdo, R., Gómez, E. (2024). Attribute annotation and bias evaluation in visual datasets for autonomous driving. Journal of Big Data, 11(1): 137. https://doi.org/10.1186/s40537-024-00976-9

[15] Benmalek, M., Seddiki, A. (2024). Bias in federated learning: Factors, effects, mitigations, and open issues. Ingénierie des Systèmes d'Information, 29(6): 2137-2160. https://doi.org/10.18280/isi.290605

[16] Dockès, J., Varoquaux, G., Poline, J.B. (2021). Preventing dataset shift from breaking machine-learning biomarkers. GigaScience, 10(9): giab055. https://doi.org/10.1093/gigascience/giab055

[17] Alzghoul, M., Baklizi, M., Alkhazaleh, M., Qtaish, O., Alshawabkeh, H. (2026). A leakage-audited prefix-only protocol for session-based consumer navigation prediction in e-commerce clickstream data. Ingénierie des Systèmes d'Information, 31(7): 2215-2229. https://doi.org/10.18280/isi.310708

[18] Larxel. (2020). Car License Plate Detection. Kaggle. https://www.kaggle.com/datasets/andrewmvd/car-plate-detection.

[19] Elmenshawii, F. (2023). License Plate Detection Dataset. Kaggle. https://www.kaggle.com/datasets/fareselmenshawii/license-plate-dataset.

[20] Yang, S., Luo, P., Loy, C.C., Tang, X. (2016). WIDER FACE: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 5525-5533. https://doi.org/10.1109/CVPR.2016.596

[21] LuisMiguelSC. (2023). WIDER FACE Dataset Images with YOLO Label Format for Instant Training. Kaggle. https://www.kaggle.com/datasets/lylmsc/wider-face-for-yolo-training.

[22] Kullback, S., Leibler, R.A. (1951). On information and sufficiency. The Annals of Mathematical Statistics, 22(1): 79-86. https://doi.org/10.1214/aoms/1177729694

[23] Cover, T.M., Thomas, J.A. (2005). Elements of Information Theory. Wiley, Hoboken, NJ, USA. https://doi.org/10.1002/047174882X

[24] Rubner, Y., Tomasi, C., Guibas, L.J. (2000). The Earth Mover's Distance as a metric for image retrieval. International Journal of Computer Vision, 40(2): 99-121. https://doi.org/10.1023/A:1026543900054

[25] Padilla, R., Passos, W.L., Dias, T.L.B., Netto, S.L., da Silva, E.A.B. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics, 10(3): 279. https://doi.org/10.3390/electronics10030279

NOMENCLATURE

A	Bounding-box area

c_i, g_i	Category counts in split/global data	P, Q	Split and global distributions
k, ε	Number of ordered categories; smoothing constant (10^{-12})	p_i, q_i	Category probabilities
		$F_p(i)$, $FQ(i)$	Cumulative distributions
D_{KL}	KL divergence	EMD_{norm}	Normalized EMD
$D_{KL,\ avg}$	Average KL divergence	$D_{nEMD,avg}$	Average normalized EMD
$D_{KL,bounded}$	Bounded KL divergence	$S_{KL\text{-}EMD}$	Composite balance score