How Much Is Enough: A Temporal Approach for Deep Learning Training and Dataset Splitting

Abeer Alsafran and Noura Aljeri
Computer Science Department, Kuwait University, Kuwait
alsafran.abeer@ku.edu.kw, aljeri@cs.ku.edu.kw

***Abstract*—Sequential image collections often contain similar frames, making the amount of data required for manual annotation an important consideration when developing deep learning models. Image metadata, particularly temporal information, offers a way to guide the selection of an initially labeled subset. In this paper, we introduce How Much Is Enough (HMIE), a temporal approach to image subset selection for model-assisted annotation. Using the AU-AIR aerial image dataset, we select nine subsets ranging from 10% to 90% of the complete collection. Each subset is selected based on temporal criteria, and each subset is divided into training, validation, and internal test partitions for developing and evaluating a YOLO11n detector. This experiment examines the use of a selected labeled subset to support the annotation of the remaining images, while identifying class-specific detection limitations. The reported results provide an initial assessment of the workflow; optimal annotation budgets and reductions in manual correction time remain subjects for further evaluation.**

***Index Terms*—Deep learning, metadata, temporal subset selection, model-assisted annotation, unmanned aerial vehicles.**

I. Introduction

DESPITE significant advances in deep learning methodologies, the preparation of annotated image datasets remains an important part of model development. For object detection, annotators must identify objects, assign their classes, and draw bounding boxes. In sequential image collections, this process can involve many frames that contain similar objects and backgrounds. Selecting a portion of the collection for initial annotation therefore raises the practical question: how much labeled data can support a model that generates useful annotations for the remaining images?

Dataset preparation also involves challenges such as class imbalance, spatial imbalance, and objective imbalance, as discussed in [1]. These challenges can affect model training and evaluation. Resampling, transfer learning, and other methods have been investigated to address aspects of imbalance [2]. Image metadata provides another source of information for organizing a collection. This information may include camera settings, geographic location, altitude, and capture time. Previous work has shown that camera metadata, including EXIF information, can support learning useful image representations [3].

Understanding temporal relationships is particularly important when images are extracted from video or captured during UAV flights. Consecutive frames may share objects, backgrounds, and viewpoints, as illustrated in Fig. 1. These similarities motivate the selection of images across a collection rather than treating every frame as equally informative for initial annotation.

The purpose of an evaluation also affects how such relationships should be interpreted. Similar frames assigned to different partitions may give an optimistic impression of the performance of new recordings [4]. However, for model-assisted annotation, the intended application can be to predict annotations for the remaining images of the same collection. In that setting, the distinction between images used to develop the selected model and images subsequently annotated by it must be made explicit. Performance on the remaining frames does not, by itself, establish a generalization to new flights or locations.

In this paper, we propose HMIE (How Much Is Enough), which uses temporal metadata to select image subsets for detector development. The selected portion provides the annotations used for training and internal evaluation. The resulting detector can then generate candidate bounding boxes and class labels for unseen images. We investigate this workflow using AU-AIR dataset [5], considering nine selected proportions ranging from 10% to 90%. We subsequently evaluated the model associated with the best selected subset on its remaining unseen data.

The primary contributions of this paper are as follows.

- Implementation of HMIE, a pioneering temporal approach for training deep learning models.
- Development of a semi-automated labeling system for image datasets.
- Experimental evidence illustrating the enhancements achieved through HMIE in the context of the YOLOv11 model.

The remainder of this paper is organized as follows: Section II will review the related literature. Section III will outline the preliminary concepts and provide an in-depth examination of the problem statement. Subsequently, Section IV will present a comprehensive discussion of the HMIE algorithm. Section V will describe the performance evaluation and associated results. Finally, we will conclude our findings in Section VI.

II. Related Work

Previous work has focused on improving random splitting and label distribution, while the dependency on image metadata has received comparatively little attention. We can classify these issues into two groups: image dataset split

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

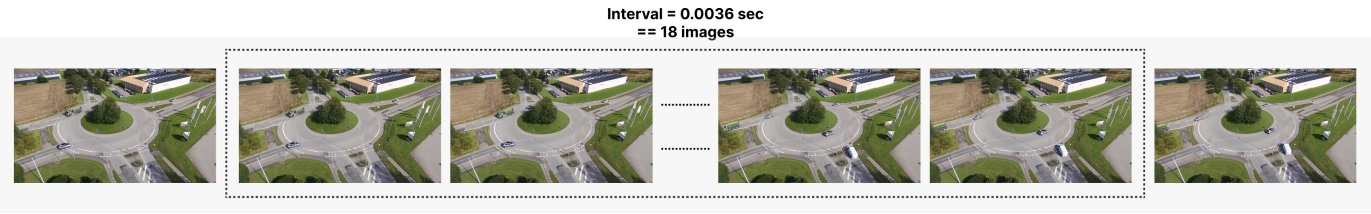


Fig. 1. Temporal Difference in Consecutive Frames

(random vs. selective) and image metadata utilization, such as EXIF files.

The authors in [3] have used the EXIF information as text and fused it with the images into the model to predict the properties of the images. The model takes two inputs: 1) EXIF as text, 2) image. The two inputs are used to train the model for tasks such as predicting image properties, estimating radial distortion, and evaluating forensic features. Furthermore, when tested on other image properties, it can generalize to images not included in the EXIF data.

Random splits often lead to unrepresentative sets, resulting in bias and imbalance. The authors in [6] believe that methods already used for splitting, such as iterative strategies, greedy search, and genetic algorithms, cannot be applied directly to image segmentation tasks. They introduced WDES, a genetic-algorithm-based stratification method for image segmentation tasks that minimizes the Wasserstein distance between label distributions. Using the Wasserstein distance to split the datasets ensures that the training, validation, and test sets are siblings rather than strangers, yielding a reliable predictor of real-world performance.

Images are just video frames; these frames may not be identical, but may be highly similar, which may indicate a significant spatial redundancy in most frames. In sequential images, all elements, including the background and the camera perspective and angle, are the same; i.e., the spatial and the temporal correlation is high. This observation suggests that existing benchmark datasets may display significant biases in how moment locations are distributed, making it difficult to determine if the methods currently in use are merely accommodating these biases or genuinely learning the semantic alignment between the two modalities. In reference to this, the authors in [4] adopted an alternative strategy focused on detecting information leakage across training, validation, and testing sets created through random splitting. They utilized t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimensionality of the feature vector. The core idea is to cluster the images extracted from videos in order to achieve the maximum number of clusters, M, where increasing M enhances the randomness in the segmentation of the sub-datasets. The algorithms used are Dimensionality Reduction, Clustering Algorithms, and Object Detection.

The authors in [7] also propose an algorithmic approach for automatic bias detection. Their approach is designed to divide the dataset in a way that the predictors trained on the training split are unable to generalize to the testing split. It comprises two main components: the Splitter and the Predictor. The Splitter is responsible for categorizing each input-label pair into either the training or testing split. The Predictor utilizes the training split to learn how to derive the label from the input. The performance of the predictions on the testing split is leveraged to help the Splitter create a more difficult split. Their method for training the Splitter relies on the observation that the Predictor's capacity to generalize will decline if correctly predicted examples are removed from the testing split, leaving only those that were predicted incorrectly.

On the other hand, statistics motivated the authors in [8] to attain a balanced representation of the dataset through a conventional algorithmic method that minimizes the unpredictability of random partitioning. Initiating with dimensionality reduction, clustering is utilized to allocate data points into clusters, dividing the clustered dataset into three distinct ranges (median range, quartile range, extreme range). Consequently, constructing a subset by individually selecting data from these groups and subsequently merging them will decrease randomness. Ultimately, preparing the data designated for model training. Feature-aware dataset division, in contrast, represents another dimension of dataset partitioning; [9] suggests implementing one-shot, near-optimal data arrangements that enhance traditional training optimization. This approach aims to fully leverage the potential of inference models while managing the computational demands. It introduces a lightweight exploration strategy that is feature-aware. By addressing three components of dataset splitting through feature awareness: 1) a diversity-focused data splitting method that combines feature engineering with dataset division, 2) feature clustering distance-based image selection (FCDIS) using the K-means algorithm, and 3) feature space center-based image selection (FSCIS), which utilizes a sorted list of cluster distances.

Researchers have investigated the problem of bias in dataset splits and in dataset balancing. However, work related to utilizing the metadata of the images and camera settings information needs more attention.

III. Preliminaries and Problem Statement

Metadata plays a crucial role in our methodological framework, as it has become apparent that a considerable portion of benchmark image datasets lacks comprehensive metadata inclusion. Metadata typically includes essential attributes such as camera settings, geographic location (longitude and latitude), date and time of capture, and altitude, particularly for aerial imagery captured by drones. It is imperative to assert

that a more extensive collection of information is necessary for a robust analysis.

Current practices in dataset partitioning often involve random or sequential splits that neglect both temporal and spatial considerations. In random splitting, images may include duplicates or visually similar scenes, resulting in the model retraining on the same subset of images repeatedly. Consequently, when validating the model on the test set, performance may be compromised due to the presence of scenes that differ significantly from those encountered during training. A similar issue arises with sequential splitting, wherein the training set may comprise images from a single scene with varying objects, leading to model bias. This methodological oversight highlights the need for more sophisticated approaches to dataset partitioning that account for these factors.

TABLE I
AU-AIR METADATA ATTRIBUTES AND THE DERIVED TIME DIFFERENCE

Attribute	**Description**
Image name	Image identifier
Image width and height	Image dimensions
Platform	UAV used to record the data
Time and date	Recording time information
Longitude and latitude	UAV location
Altitude	Recorded UAV altitude
Linear x, y, and z	UAV linear velocity components
Angles ϕ, θ, and ψ	UAV orientation angles
Time difference	Derived difference between successive entries

A. Preprocessing and Attributes

As part of our data cleaning and preparation process, we have introduced an additional attribute: the *time difference*. This attribute captures the temporal aspect of HMIE, providing insight into the intervals between consecutive images. Our analysis reveals that the most common time difference between each image and the subsequent one is 0.0002 seconds. This important information indicates that the scenes depicted in the images are nearly identical, allowing us to make more informed decisions in our analysis.

The AU-AIR dataset [5] comes with its own JSON metadata format. Table I shows the attribute list that comes within the JSON file, along with the annotation and our proposed time difference.

IV. How Much Is Enough

A. Dataset

The AU-AIR dataset, as referenced in [5], encompasses multimodal sensor data, including altitude, location, and temporal information, collected in a real-world outdoor setting. This multifaceted data composition makes it particularly suitable for the proposed HMIE algorithm, which relies fundamentally on the aforementioned metadata associated with each image.

Comprising 32,823 images captured by a DJI Phantom 4 drone, the dataset categorizes instances into eight classes: human, car, van, truck, motorbike, bicycle, bus, and trailer, yielding a total of 132,034 annotated instances. As illustrated in Fig. 2, there is a notable discrepancy in the distribution of these classes, with the Car class exhibiting a significantly higher count of instances than the others.

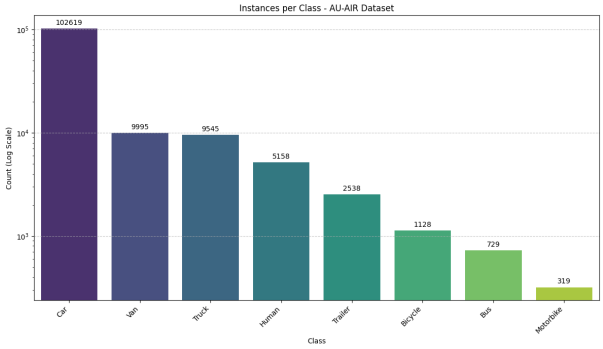


Fig. 2. Total Instances Distribution Over Classes

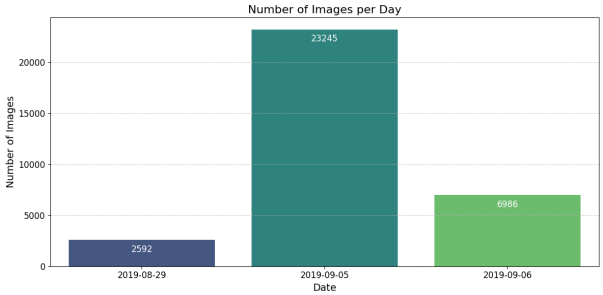


Fig. 3. Images Captured Per Day for AU-AIR Dataset

The preliminary exploration of the dataset, as shown in Fig. 3, reveals the frequency of image captures over three distinct days. Given that the HMIE algorithm operates on the premise of analyzing temporal relationships between images, it is imperative to examine the time intervals between consecutive images.

B. HMIE Algorithm

How Much Is Enough is an efficient algorithm that achieves two main goals. As discussed earlier, HMIE enhances the splitting mechanism of the image dataset by taking into consideration the temporal aspect. This leads to a better and more efficient deep learning training process that helps annotators know how much is enough to annotate a subset of images in a dataset to enable the model to complete the remaining annotation process based on the highest accuracy possible. In particular, the HMIE is applied before the train/test/validation split.

We assume that the images have been captured sequentially and that the consequent images have the same repeated scenes or sometimes overlapping scenes. Taking this into account, we need to exclude repeated scenes in order to give an opportunity to other different scenes to be included in the subset to be trained on by the model. For that, we need to have a certain

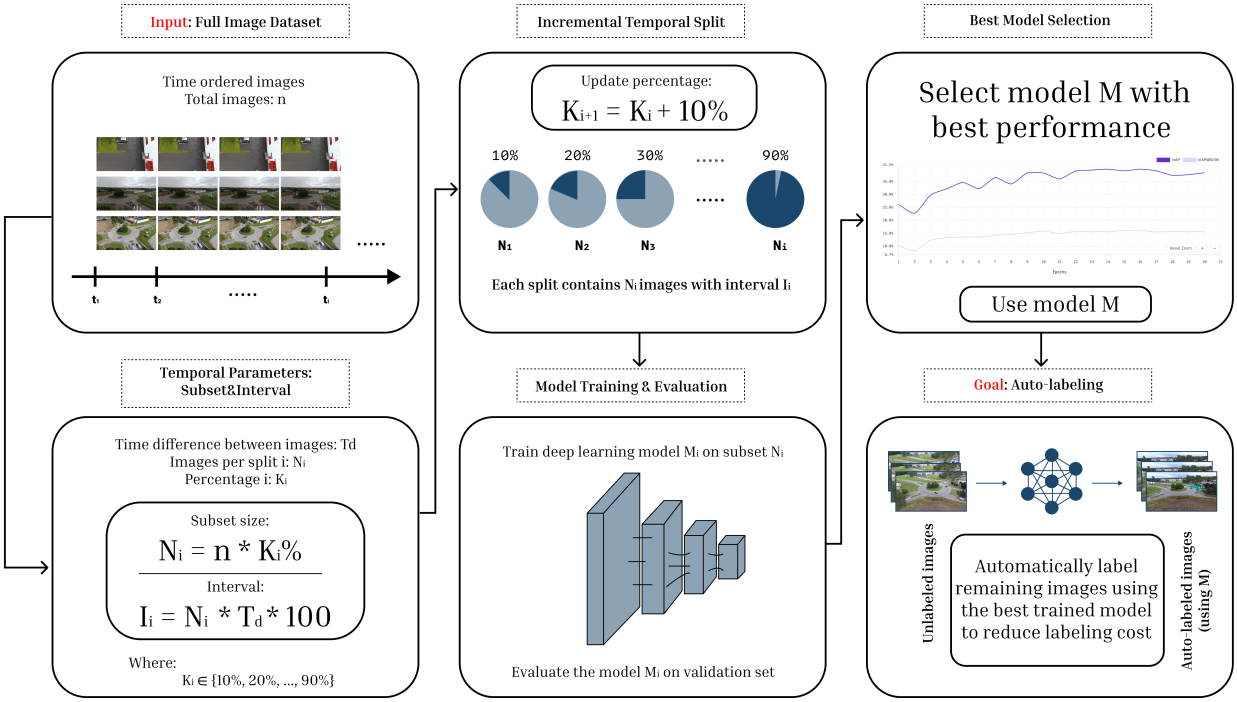


Fig. 4. Proposed Temporal Approach (HMIE)

interval between each image in the sorted list of images and the next one to avoid having the model train on the same scene over and over, and to let the annotator focus on different scenes.

As illustrated in Fig. 4, the overview of our model highlights the crucial steps necessary to process the image dataset. Initially, we begin with the complete image dataset, which we sort based on date and time. This sorting allows us to establish a chronological order, vital for accurate temporal analysis. Following this, we computed the time difference for each image in relation to the subsequent image, providing us with essential parameters for our subsequent calculations. At this stage, we have all the required data to proceed with the temporal computations. The sorted dataset serves as our primary input. From the sorted images, we extract key parameters: first, the time difference indicated as (T_d); second, the dynamically calculated number of images per split, represented as (N_i); and third, the interval between each selected frame, indicated as (I_i). With these parameters in hand, we proceed to generate data splits. Specifically, we create subsets of the dataset ranging from 10% to 90%, as delineated in Algorithm 1. Up to this point, we have meticulously prepared the data for deep learning training. The subsequent phase involves both model training and evaluation. We train the model on each percentage split and meticulously record the results for comparative analysis. The ultimate objective of this process is to identify and utilize the best-performing model for the automatic labeling of future UAV datasets, thereby improving our ability to efficiently and accurately process aerial imagery.

The subset starts at 10% of the total dataset and increases by

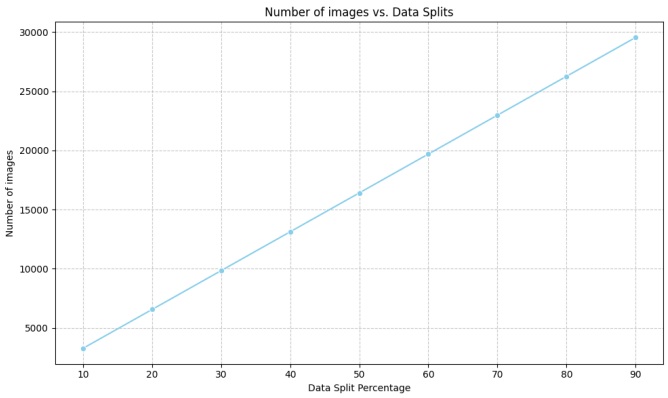


Fig. 5. Total Images Per Each Split

10% each time. First, we apply the first equation to compute the subsets. We apply it 9 times, from 10% to 90%. The first equation is the following.

$$N = n * K\% \tag{1}$$

where N is the new subset, n is the total number of images in the dataset, and K is an array of percentages (initially 10).

To calculate the interval between each image in the sub-dataset (N), we rely entirely on the temporal metadata of the image. So we start by sorting the images based on the first image that has been captured using a timestamp. We then calculate the interval size for all subsets. To calculate the interval, the following equation is applied:

Algorithm 1 HMIE

Require: Total image list D of size n, time differences T_d, percentages $K = \langle 10, 20, \ldots, 90 \rangle$
Ensure: Subsets $\mathcal{M} = \{L_1, \ldots, L_{|K|}\}$

$T_d \leftarrow \text{mode}(T_d)$
if $T_d \geq 1$ **then**
  $T_d \leftarrow T_d / 10^4$
end if
for $a \leftarrow 1$ **to** $|K|$ **do**
  $N[a] \leftarrow \lfloor n \cdot K[a]/100 \rfloor$
  $I[a] \leftarrow \lfloor T_d \cdot N[a] \cdot 100 \rfloor$
end for
Sort I in descending order
for $a \leftarrow 1$ **to** $|K|$ **do**
  $L_a \leftarrow \langle\,\rangle$
  $i \leftarrow 0$
  while $|L_a| < N[a]$ **do**
    if $D[i] \in L_a$ **then**
      $i \leftarrow i + 1$
    end if
    Append $D[i]$ to L_a
    $i \leftarrow i + I[a]$
    if $i \geq n$ **then**
      $i \leftarrow (i \bmod n) + 1$
    end if
  end while
  $\mathcal{M}[a] \leftarrow L_a$
end for

$$I = T_d * N * 100 \tag{2}$$

Applying equation (2) nine times will produce an array of intervals. Following that, equation (3) will reverse the array.

$$I = sort(I, reverse = True) \tag{3}$$

Where T_d is the time difference between the current image and the next image in the sorted list of images, *N* is the new subset. We reverse the list of intervals so that the larger interval comes first for the smaller percent.

These two equations guide the selection of training data in temporal order. Finally, we will have a size 9 list for *N* and *I*. These two lists will contain the subset size (N) and the interval size (*I*) – accordingly – for all K percents.

Now comes the heavy lifting of the algorithm, where progressive temporal splitting takes place, for example, on the AU-AIR dataset. The total number of images (*n*) is 32,823 frames; the first *K* is 10, so the subset (*N*) will be 3,282. We calculated the time difference (T_d) as 0.0002 seconds. After we reverse the interval list (*I*), we obtain the dedicated interval for the subset of 10% at the first index: 656. This number is dynamic for different datasets, depending on the variability of the time difference. In summary, this study proposes a systematic approach for generating nine distinct splits of any given image dataset, with careful consideration of temporal information. The methodology begins with the sorting of the dataset, followed by the calculation of time intervals between images. Subsequently, the size of each subset is determined relative to the overall dataset, using a predefined percentage. A critical component of this approach is the calculation of the interval, which is dynamically adjusted based on both the subset size and the temporal differences. The result is an optimized framework for dataset splitting that facilitates an unbiased representation across the generated subsets, thereby enhancing the integrity and reliability of subsequent analyzes. Furthermore, as illustrated in Fig. 5, there is a steady linear increase in the number of images corresponding to the increasing percentage of the dataset. For example, selecting 10% of the dataset yields 3,282 images, while selecting 20% yields 6,564. This linear growth continues consistently, reaching 29,540 images at the 90% percentage.

V. Performance and Evaluation

We are now in a position to address the question of *How much is enough?* Our temporal approach was evaluated using the AU-AIR dataset, a well-regarded benchmark in the domain. For this, we used [5] as our reference dataset, given its comprehensive coverage of the essential components that are imperative for the HMIE algorithm to achieve optimal performance, as referenced in Table I.

A. Training Setup

The training was performed using the Roboflow platform, known for its compatibility with various models. For this project, we utilized the YOLOv11 nano [10] with the COCOn checkpoint [11]. To enhance our dataset, we used Roboflow's range of augmentation techniques, specifically auto-orient and mosaic augmentation. This preprocessing step plays a crucial role in balancing the dataset and improving its robustness for model training. In particular, [12] also implemented mosaic augmentation as a key component in identifying the optimal data augmentation strategies. Each subset of the 9 subsets has been allocated according to the following distribution: 85% designated for training, 10% for testing, and 5% for validation.

B. Evaluation metrics

1) Mean Average Precision: Performance is quantified using standardized computer vision metrics. Mean Average Precision (mAP) is one of the most widely used metrics to evaluate model performance in computer vision. Mean Average Precision is the industry standard that provides a comprehensive view of the accuracy of an object detector in all classes of the dataset [13].

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

where N represents the total number of classes and AP_i is the Average Precision for class i. This leads us to the following. Average Precision (AP):

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{5}$$

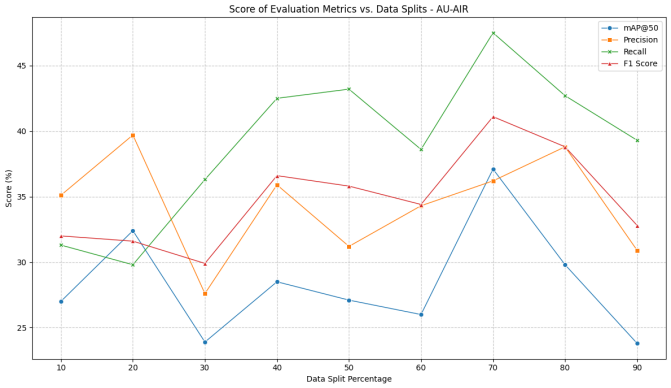


Fig. 6. Results of the Nine Splits Based on the HMIE Algorithm

where P_n and R_n denote precision and recall at the n-th threshold.

2) Intersection over Union: One of the most important quantitative measures in computer vision is Intersection over Union (IoU), which is the overlap area between two bounding boxes and evaluates how well the predicted bounding box fits the ground truth. It can be calculated using the following function: given that A is the area of the predicted bounding box, and B is the area of the ground truth bounding box.

$$IoU = \frac{A \cap B}{A \cup B} \tag{6}$$

3) Precision: To account for the relevance of positivity, we employ precision. It is the measurement of the correct predictions among all ground truths.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

4) Recall: Recall is highest when the model minimizes false negatives in bounding-box predictions; in other words, recall increases as more true positives correctly match the ground-truth objects.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

TP represents True Positives, FP represents False Positives, and FN represents False Negatives.

5) F1 Score:

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

C. Model Performance and Evaluation

To evaluate our model performance, we have relied mainly on four evaluation metrics, including Mean Average Precision (mAP) @ 50, which is the average precision across all classes at Intersection over Union (IoU) with 50% spatial overlap. Precision, recall, and F1 score. mAP is the standard metric for object detection in computer vision.

Fig. 6 shows the results for the evaluation matrix for each split. The 70% split has achieved the highest accuracy in all metrics, except precision, with mAP@50 at 37.1%, precision at 36.2%, recall at 47.5%, and F1 score at 41.1%. The following splits show that the performance has increased and decreased several times; the second-best mAP@50 is 32.4% in the 20% split.

TABLE II
INTERNAL-TEST DETECTION PERFORMANCE FOR THE NINE HMIE-SELECTED SUBSETS. EACH SELECTED SUBSET IS DIVIDED INTO 85% TRAINING, 5% VALIDATION, AND 10% INTERNAL TESTING. THE PERCENTAGES AND IMAGE COUNTS INDICATE THE SELECTED PORTION OF THE COMPLETE DATASET. EACH MODEL IS EVALUATED ON ITS CORRESPONDING INTERNAL TEST PARTITION

Split (%)	Images	Precision	Recall	F1-Score	mAP@0.5
10	3,282	35.1	31.3	32.0	27.0
20	6,564	**39.7**	29.8	31.6	32.4
30	9,846	27.6	36.3	29.9	23.9
40	13,129	35.9	42.5	36.6	28.5
50	16,411	31.2	43.2	35.8	27.1
60	19,693	34.3	38.6	34.4	26.0
70	22,976	36.2	**47.5**	**41.1**	**37.1**
80	26,258	38.8	42.7	38.8	29.8
90	29,540	30.9	39.3	32.8	23.8

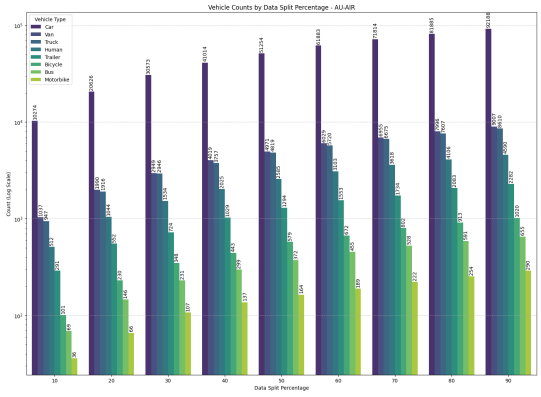


Fig. 7. Class Distribution Over Each Split

Table II reports detection performance as a function of the size of the training subset, from 10% to 90% of the full dataset. Across the nine splits, mAP@0.5 ranges from 23.8 to 37.1 (mean 28.4 ± 4.2), precision from 27.6 to 39.7 (34.4 ± 3.9), recall from 29.8 to 47.5 (39.0 ± 5.8), and F1-score from 29.9 to 41.1 (34.8 ± 3.6). Performance does not improve monotonically with subset size: the best mAP@0.5 (37.1), Recall (47.5), and the best F1-score (41.1) are all obtained at 70%, whereas the largest subset (90%) yields the lowest mAP@0.5 (23.8). Precision is essentially uncorrelated with subset size, while recall increases moderately.

As shown in Fig. 7, the differences between the multiple classes provide insight into the results obtained by HMIE. In the 10% split of the dataset, the sample size is relatively small, resulting in a minor gap between classes. However, as we increase the size of the split, the divergence between the classes begins to widen significantly, highlighting the impact of the size of the dataset on the classification outcomes.

In this study, we assess the performance of the detection of

TABLE III
PER-CLASS OBJECT DETECTION PERFORMANCE. TEST UNSEEN IMAGES FROM THE TOTAL DATASET USING THE MODEL THAT HAS BEEN TRAINED ON 70% OF THE DATASET WITH A 0.25 CONFIDENCE THRESHOLD AND IOU AT 0.50.

Class	Precision	Recall	F1-score	AP_{50}	AP_{50-95}
Human	61.90	10.97	18.64	32.81	10.87
Car	67.65	26.66	38.25	40.59	14.43
Truck	53.20	56.69	54.89	50.02	22.79
Van	54.37	30.49	39.07	35.92	15.08
Motorbike	00.00	00.00	00.00	18.52	6.18
Bicycle	00.00	00.00	00.00	13.42	3.89
Bus	70.63	50.25	58.72	58.78	29.85
Trailer	52.94	1.12	2.19	10.82	0.38

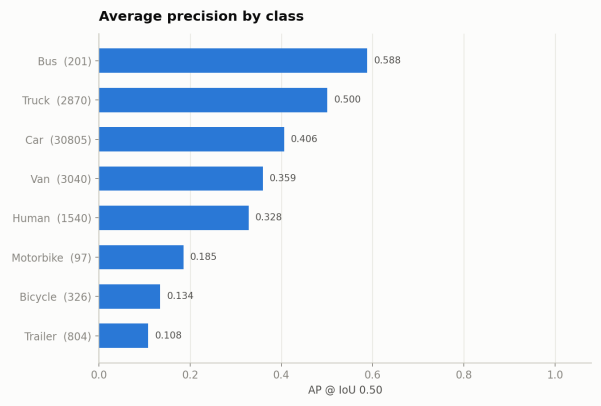


Fig. 8. Average Precision Per Class Result For The Unseen Images

objects per-class as shown in Table III by evaluating a model trained on 70% of the total dataset. We utilized a separate set of unseen images from the complete dataset to test the model's efficacy in accurately detecting and classifying objects across various categories. This approach allows for a comprehensive analysis of the model's ability to generalize to unseen data. The overall test results indicate that the model, trained on the specified dataset subset, achieved a mean Average Precision (mAP@50) of 31.79, a precision metric of 63.24, a recall rate of 25.92, and an F1-score of 36.77. These findings underscore the model strengths in performing object detection across various categories. As previously illustrated in Fig. 4, the proposed temporal approach for HMIE seeks to automate the labeling of the remaining images in the dataset by leveraging the best-performing model; in this instance, the model trained on the 70% dataset split.

From Fig. 8, we can confirm that larger instances do not mean better detection. Hence, the car has more instances (30,805) than the bus (201), yet the AP of the bus (0.588) is greater than the AP of the car (0.406). This implies that the model managed to detect bus objects efficiently even with a small number of instances. Fig. 9 studies the relationship of the Precision-Recall curve, showing that objects such as

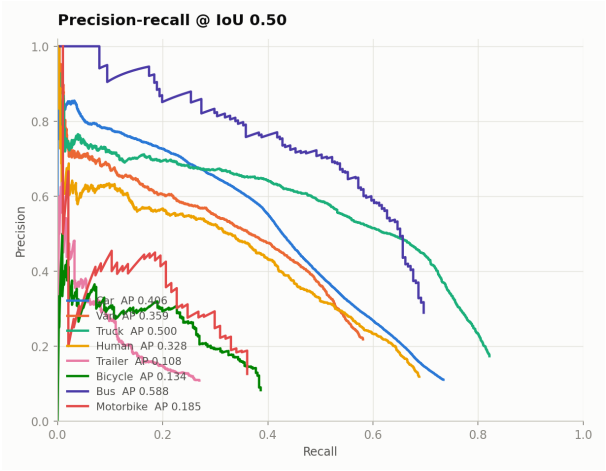


Fig. 9. Precision-Recall @ IoU 50 For The Unseen Images

a trailer, Bicycle, and Motorbike plateau in markedly lower recall, consistent with the known challenges in the detection of small objects from UAV images.

Finally, Fig. 10 shows qualitative YOLOv11 detections on a representative sample from the test set. Despite the challenges posed by UAV imagery, such as the size of small objects, scale variation, and occlusion. The model effectively navigated these obstacles, achieving reliable detections with a confidence threshold equal to 0.20.

The results suggest that it is unnecessary to train on the entire dataset to achieve the highest level of accuracy. This implies that even with a smaller and more carefully chosen subset of the data, we can achieve comparable performance. This finding highlights the potential for more efficient training practices, enabling faster model development and reduced computational costs without compromising result quality.

VI. CONCLUSION

Temporal metadata provides a practical source of information for selecting images from sequential collections. In this paper, we proposed HMIE to select labeled subsets for detector development and subsequent model-assisted annotation. We examine nine selected proportions of AU-AIR, from 10% to 90%, with separate internal training, validation, and test partitions for each configuration. The model associated with the selected subset 70% achieved the highest internal-test mAP_{50} reported of 37.1% and was selected to generate candidate annotations for the complementary 30%.

Evaluation against existing annotations showed differences in detection performance across classes and identified limitations in the completeness of the generated annotations. The study provides an initial assessment of the use of a selected labeled subset to support the annotation of the remaining images. Future work will investigate matched sampling comparisons, repeated experiments, additional datasets, and spatial information, together with the manual correction effort needed to turn candidate predictions into completed annotations.

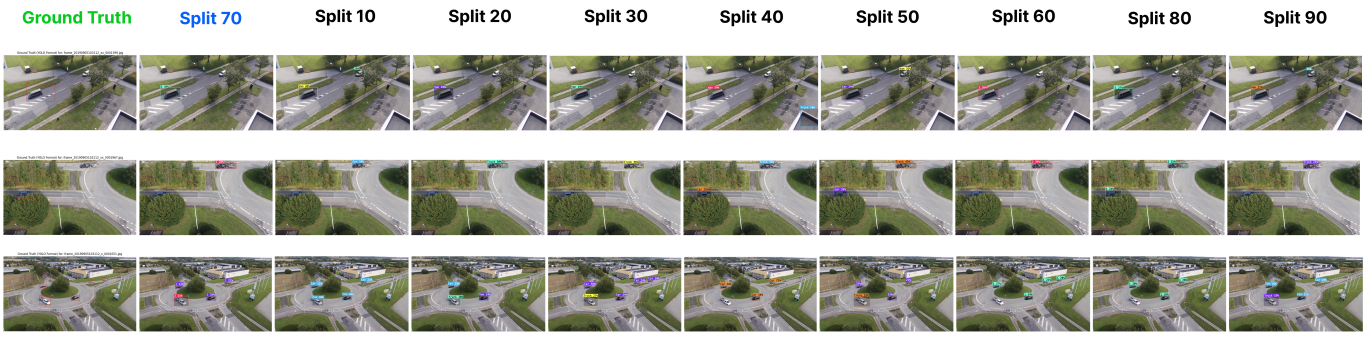


Fig. 10. YOLOv11 Detection Results On The AU-AIR Dataset On Test Sets from All the Splits. The Confidence Threshold is 0.20, The Overlap Threshold is 0.50, and The Opacity Threshold is 0.75.

ACKNOWLEDGMENT

This work was financially supported by the Kuwait Foundation for the Advancement of Sciences (KFAS).

REFERENCES

[1] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3388–3415, 2021.

[2] M. Tomaszewski and J. Osuchowski, "Effectiveness of data resampling in mitigating class imbalance for object detection," in *The 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP-2023)*, 02 2024.

[3] C. Zheng, A. Shrivastava, and A. Owens, "Exif as language: Learning cross-modal associations between images and camera metadata," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6945–6956.

[4] R. B. D. Figueiredo and H. A. Mendes, "Analyzing information leakage on video object detection datasets by splitting images into clusters with high spatiotemporal correlation," *IEEE Access*, vol. 12, pp. 47 646–47 655, 2024.

[5] I. Bozcan and E. Kayacan, "Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8504–8510.

[6] N. V. S. J. Jami, T. Altstidl, J. Mueller, J. Li, D. Zanca, B. Eskofier, and H. Leutheuser, "Stratify or die: Rethinking data splits in image segmentation," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] Y. Bao and R. Barzilay, "Learning to split for automatic bias detection," in *2023 The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[8] K. M. Kahloot and P. Ekler, "Algorithmic splitting: A method for dataset preparation," *IEEE Access*, vol. 9, pp. 125 229–125 237, 2021.

[9] S. Shahrabadi, T. Adão, E. Peres, R. Morais, L. Magalhães, and V. Alves, "Automatic optimization of deep learning training through feature-aware-based dataset splitting," *Algorithms*, vol. 17, pp. 1–29, 02 2024.

[10] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024.

[11] Roboflow, "Yolo11 object detection model: What is, how to use," https://roboflow.com/model/yolo11, accessed: Jun. 7, 2026.

[12] M. Jenkins, K. A. Franklin, M. A. C. Nicoll, N. C. Cole, K. Ruhomaun, V. Tatayah, and M. Mackiewicz, "Improving object detection for time-lapse imagery using temporal features in wildlife monitoring," *Sensors*, vol. 24, no. 24, 2024.

[13] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.