Leveraging FINCH and K-means for Enhanced Cluster-Based Instance Selection

Panagiota Zotou[1,2], Konstantinos Bacharidis[1,2], and Antonis Argyros[1,2]

[1] Computer Science Department, University of Crete, Heraklion, Greece
[2] Institute of Computer Science, FORTH, Heraklion, Greece
{pzotou,kbach,argyros}@ics.forth.gr

Abstract. We introduce a novel instance selection method, which integrates FINCH and the K-means clustering algorithms into a unified process for improved instance selection. Initially, FINCH is employed to perform a first dataset clustering. Representatives of the resulting clusters are used as seeds for K-means to refine clustering. The samples that are closer to the centers of the tuned clusters form the set of selected instances. Our experiments show that our method outperforms established instance selection techniques. We also showcase the practical benefits of our approach by applying it to reduce the size of augmented training datasets in a case-study that involves ship detection in aerial and satellite images. The results demonstrate that our method leads to significant dataset size reduction with minimal impact on ship detection accuracy.

Keywords: Dataset Distillation · Instance Selection · FINCH · K-means

1 Introduction

In the era of big data and deep learning, the computational and storage requirements of neural networks are proportional to the size of the datasets used for their training. As datasets are becoming increasingly larger [23,29], it is highly important to be able to select subsets of them that, despite their reduced size, consist of representative training instances/examples that can support accurate training. In that direction, methods from the research fields of dataset distillation [38,44] have emerged as essential preprocessing steps in a wide range of machine learning and classification tasks.

Instance selection is a particular form of data distillation [39,27,28,26] that aims to preserve informative samples that accurately represent the original dataset, while eliminating redundant or irrelevant data. By decreasing the dataset size, instance selection algorithms reduce runtime during both the classification and training stages. Instance selection seeks for the smallest subset of a training dataset that does not compromise training accuracy. At the same time, instance selection itself needs to be computationally efficient. Currently, no instance selection algorithm is a clear winner in all these fronts. Therefore, searching for robust, effective and efficient approaches to instance selection remains an active area of research.

In this work, we present a new approach to the problem of instance selection. The proposed algorithm capitalizes on a careful integration of two widely accepted clustering algorithms, FINCH [30] and K-means [25]. Specifically, a variant of the FINCH algorithm is initially employed on an input dataset to perform a preliminary clustering of the data. Representatives of these clusters initialize a variant of K-means that refines them. Finally, the dataset instances that are closer to the cluster centers as those are determined by K-means, are declared as the selected dataset instances.

A number of experiments were performed to evaluate variants of the proposed instance selection mechanism on six different standard datasets in different domains. These datasets differ with respect to important characteristics such as their number of classes, the number of samples and their dimensionality, etc. The evaluation is performed on the basis of three criteria, (a) number of selected samples, (b) classification accuracy when training with the selected instances and (c) running time of the instance selection algorithm. We also compare the proposed variants with several established top-performing instance selection algorithms. The comparison is performed on the basis of each of the above criteria, but also on the basis of a new criterion that aggregates them into a single metric. The obtained results demonstrate that the proposed solutions outperform the existing algorithms in the great majority of tests and datasets.
Overall, the contributions of this work can be summarized as follows:

- We introduce a novel algorithm for instance selection that integrates variants of FINCH and K-means.
- We evaluate thoroughly the proposed instance selection approach on varied datasets and in comparison with existing instance selection algorithms. The results show that the proposed algorithms offer the best balance of running time, number of selected instances, and achieved classification accuracy.
- We illustrate the practical advantages of the proposed instance selection approach in augmented dataset size reduction, demonstrating experimentally that it achieves significant reductions with minimal loss in model accuracy.

2 Related Work

General methods for instance selection: Instance selection algorithms have a long-standing history. Based on the instance selection strategy that they follow, they can be classified into three main categories: (a) *condensation*, (b) *edition* and (c) *hybrid* methods [13,4]. Condensation-based algorithms aim to retain the instances that are closer to the decision boundaries. Since border instances are usually fewer than internal ones, these methods often achieve significant reduction; however, sometimes this may have a negative impact on the classification accuracy. Edition-based algorithms focus on discarding noisy instances or instances that do not align with their neighbors, resulting in the removal of border instances and the retention of internal ones, which may not necessarily contribute to classification accuracy. Thus, these methods typically achieve a low

reduction rate. Finally, hybrid algorithms allow for the retention of both border and internal instances.

One of the first and widely explored condensation-based algorithms is the Condensed Nearest Neighbor (CNN) [16] algorithm, based on the Nearest Neighbor (NN) [9] method. The initial step of CNN is to randomly select one instance from each class of the training set, TS, and put them in a subset, S. Next, each instance in TS is classified using only the instances in S. If an instance is misclassified, it is added to S, ensuring that all instances in TS are classified correctly. According to this criterion, noisy instances are likely to be retained because they are commonly misclassified by their k-NN. Based on this criterion, CNN may include noisy instances in S, which may affect classification accuracy.

The Edited Nearest Neighbor (ENN) [40] method is perhaps one of the first edition methods. In comparison to CNN, ENN begins with $S = TS$ and removes noisy and border instances from S. An instance is discarded when its class differs from the majority class among its k nearest neighbors. ENN retains instances that are more centrally located within their respective classes, aiming to create smoother, more distinct boundaries among different classes. The Repeated ENN (RENN) [34] is an extension of ENN that repeatedly applies ENN until all instances in S belong to the same class as the majority of their k Nearest Neighbors. This iterative procedure results in an even wider gap between classes.

A more recent edition-based method, the Local Density-based Instance Selection (LDIS) [8] algorithm exhibits relatively low time complexity and is one of the most popular density-based algorithms in the literature [26]. The algorithm independently analyzes each class by computing local density, identifying local nearest neighbors, and retaining only the densest instances from each class. XLDIS [7], an extension of LDIS, selects instances with the highest local density ordering (LDO) in their neighborhood, analyzing them in descending LDO order. To reduce processing time, XLDIS maintains a list of instances excluded from further analysis for inclusion in the final set. Finally, the Local Set-based Smoother (LSSm) [21] is an edition-based algorithm that removes instances when their harmfulness outweighs their usefulness. Although the method achieves high accuracy, it does not significantly reduce the number of instances.

Hybrid methods include, IB3 [2], an early well-known approach that retains effective classifiers while discarding noisy instances by tracking their correct and incorrect classifications. In similar design to IB3, Wilson et al. [39] introduced a set of five methods which are based on the Decremental Reduction Optimization Procedure (DROP). DROP methods are based on the concept of associates. According to [27], DROP3 is the most widely recognized variant as it outperforms several well-known previous methods in both classification and dataset reduction. DROP3 first filters out noisy instances and then discards any instance that can be correctly classified without its associates. However, the iterative k-neighborhood search to decide on removal leads to high computational cost. [8].

The more recent Clustering-based Instance Selection (CIS) algorithm [28] is a hybrid instance selection method that initially employs the unsupervised K-means clustering algorithm to identify clusters within the training instances,

given predefined cluster number and selection rate values. The algorithm then selects instances from both cluster centers and borders using distance-based measures. Central instances represent core cluster characteristics, while border instances protect clusters from each other. The key drawback of this method is the need to pre-determine the number of clusters.

In [36] the authors introduced a novel hybrid under-sampling approach called cluster-based instance selection (CBIS). The CBIS method combines clustering analysis with three well-known instance selection algorithms (IB3, DROP3, Genetic Algorithm [6]). As a clustering method, the CBIS employs the Affinity Propagation (AP) [12] algorithm, which has the advantage of not requiring a predetermined number of clusters. However, calculating the similarity between all samples, results in increased computational cost as the dataset size grows.

Instance selection in computer vision datasets: Although many of the aforementioned methods have been extensively applied to real datasets across various domains, research on instance selection methods in the field of computer vision remains relatively limited [19,3,35,5]. For example, in [19], the authors suggest the concept of a training value, according to which a classifier is trained using one particular positive example along with all the negative examples. The average performance of the classifier across the training set defines the training value of an example. The algorithm selects the most valuable training examples by using a greedy forward method, which incrementally includes examples from a ranked list based on their training value. In [3], the authors introduce a Random Mutation Hill Climbing (RMHC) [33] based method. Here, the authors modified the RMHC algorithm to reduce its computational cost, computing the accuracy only for neighboring instances affected by adding or removing an instance.

3 Proposed Method

Our method integrates the strengths of FINCH [30] and K-means [25] algorithms. In a first step, a novel modified version of FINCH is employed to automatically determine a number of clusters in data. In a second step, these clusters are refined using the K-means algorithm. Finally, the samples that are most similar to the refined cluster centers are selected to constitute the final proposed instances.

3.1 Class Membership-aware FINCH

FINCH (First Integer Neighbor Clustering Hierarchy) [30] is an agglomerative clustering method that follows the first nearest neighbor clustering principle. It operates hierarchically, progressively merging data points into clusters based on their similarity to their closest nearest neighbor. Initially, each data point is considered as a cluster. The algorithm then iteratively merges clusters by examining their first nearest neighbor, repeating this process until no further merges are possible. FINCH is efficient (has a computational complexity of $O(Nlog(N)))$, and allows the generation of a series of partitions, each representing a different level in the clustering hierarchy. These partitions enable exploration of the

data at various levels of granularity (fine to coarse), making FINCH particularly suitable for tasks requiring multi-level data analysis and exploration.

Due to its unsupervised and class-agnostic nature, FINCH may struggle to accurately discriminate between samples with similar characteristics belonging to different classes, particularly in borderline or edge cases. These cases are crucial for instance selection, as the goal is to maximize classification performance while minimizing the amount of data used. However, the class labels of training sets are already available as ground truth in the datasets used to train neural networks. Therefore, we incorporate this knowledge of class membership into the clustering process. In more detail, we propose two design approaches to achieve this: (a) applying FINCH to the data points of each class independently (Class-wise FINCH, CW-FINCH), and (b) enforcing class purity during the merging process across the entire dataset (Class-Purity FINCH, CP-FINCH).

CW-FINCH: In order to integrate class-specific homogeneity into the outcomes of FINCH, we investigate the feasibility of applying it separately to each class. This approach involves assessing the partitions generated and selectively preserving clusters from FINCH partition 0, which encapsulate the most coherent and tight groupings within each class. As a result, the final outcome of this modified methodology comprises the amalgamation of partition 0 clusters obtained from applying FINCH to each class individually. Formally, the union $\mathcal{C}_{CW}$ of partition 0 clusters across all classes is defined as:

$$\mathcal{C}_{CW} = \bigcup_i C_i^{(0)}, \tag{1}$$

where $C_i^{(0)} = \{c_{i,1}, c_{i,2}, ..., c_{i,M}\}$ denote the set of clusters in partition 0, obtained by applying the FINCH algorithm to the samples of class i from the original dataset which results in M clusters for the samples of this class ($c_{i,j}$ is the jth cluster, $1 \leq j \leq M$, of the ith class of the dataset).

CP-FINCH: Enforcing class purity during the merging process of FINCH facilitates improved class distinction by ensuring that clusters remain homogeneous with respect to class labels, thereby supporting the accurate identification of representative instances. To achieve our objective, we initially apply FINCH to the entire dataset, focusing specifically on the first partition generated by FINCH, which represents a single merging process capturing the primary cluster structures in the data. This decision is grounded on the observation that the initial partitioning phase of FINCH (partition 0) identifies the most prominent cluster patterns, forming a sturdy basis for subsequent instance selection. As FINCH progresses through additional partitions, the merging processes lead to centroids that progressively abstract away detailed data characteristics, potentially diminishing cluster specificity. We denote as $C^{(0)} = \{c_1, c_2, ..., c_K\}$, the set containing the derived clusters, c_i, from the partition 0 of FINCH, with K referring to the number of clusters that FINCH identified. Each data point $x \in \mathbb{R}^d$ is assigned to (a) a class label $y(x) \in \{1, 2, \ldots, L\}$, and (b) a cluster $c_i, i \in \{1, 2, \ldots, K\}$. For each cluster c_i, we identify *outlier elements* as the samples whose class label $y(x)$

differs from the majority class label $y_{\text{maj}}(c_i)$ within the cluster. The majority class label for cluster c_i is given by:

$$y_{\text{maj}}(c_i) = \arg \max_{l \in \{1,\ldots,L\}} \sum_{x \in c_i} \mathbb{I}(y(x) = l), \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The set $\mathcal{O}_i$ of outliers of cluster c_i is:

$$\mathcal{O}_i = \{x \in c_i \mid y(x) \neq y_{\text{maj}}(c_i)\}. \tag{3}$$

These outliers are then extracted to form new, distinct clusters. Specifically, if multiple outlier elements share the same class identity $y(x)$, they are aggregated into a singular new cluster. Let N_j denote a new cluster formed by aggregating outliers with the same class label l:

$$N_j = \{x \in \mathcal{O}_i \mid y(x) = l \text{ with } l \in \{1, \ldots, L\}\}. \tag{4}$$

The set of clusters $\mathcal{C}_{CP}$ for partition 0 is constructed by removing the outliers from the initial clusters and incorporating the newly formed clusters N_j:

$$\mathcal{C}_{CP} = \left(\bigcup_{i=1}^{K} (c_i \setminus \mathcal{O}_i) \right) \cup \left(\bigcup_{j} N_j \right). \tag{5}$$

This refined clustering, $\mathcal{C}_{CP}$ ensures higher class purity, which facilitates in more accurate and meaningful data representation.

3.2 Integrating the modified FINCH with K-means

K-means is a widely used non-hierachical clustering algorithm characterized by its simplicity and effectiveness in partitioning a dataset into a given number k of distinct clusters. The process begins by randomly selecting k initial data points as the starting cluster centroids. Subsequently, each data point is assigned to the nearest centroid, forming clusters. The centroids are then recalculated as the mean of all points in their respective clusters. This iterative process of assignment and centroid recalculation continues until convergence. Despite its popularity, it is sensitive to the initial selection of centroids [32], which can lead to suboptimal clustering and poor convergence, particularly in the presence of complex data distributions or outliers.

To address the limitations of K-means initialization (knowledge of k, sensitivity to centroid initialization), we exploit the proposed FINCH variants to initialize K-means and obtain more informed and strategically placed initial centroids that reflect the inherent structure of the data. Specifically, the K-means is initialized with the centers of the clusters, as those were identified by the two FINCH variants ($\mathcal{C}_{CW}$ by CW-FINCH or $\mathcal{C}_{CP}$ by CP-FINCH).

To perform instance selection, we utilize the final centroids obtained upon convergence of the K-means algorithm. Specifically, we select samples that are closest to these final centroids, as these samples are considered representative

of their respective clusters. The similarity metric employed is the Euclidean distance between data points and the centroids. In essence, upon convergence, for each cluster j, we identify the data point x_i that minimizes the Euclidean distance, $d(x_i, c_j)$, to the centroid c_j:

$$x_{\text{sel},j} = \underset{x_i \in \text{cluster } j}{\arg\min}\ d(x_i, c_j). \tag{6}$$

We identify two instance selection variants, (a) **KM-F**$_{cw}$, and (b) **KM-F**$_{cp}$ depending on whether K-means is applied to $\mathcal{C}_{CW}$ or $\mathcal{C}_{CP}$, respectively.

4 Experimental Setup

We demonstrate the robustness and potential benefits of the proposed instance selection method in assisting neural network models to achieve high performance with minimal data. Our evaluation spans a variety of domains, including biological data and image datasets depicting digits, objects, and animals, highlighting the versatility and applicability of our approach in various real-world scenarios.

We conduct experiments to benchmark the proposed instance selection approach against top-performing competitors that share similar design principles, and also demonstrate how the proposed variants further increase performance.

4.1 Datasets

We tested the method in three image (MNIST 28x28, CIFAR-10, CALTECH-101) and three biological data-centered dataset (Wine, Iris, Breast Cancer). For the image datasets, the images are flattened into 1-dimensional vectors to serve as data samples for the instance selection process. Conversely, for the biological datasets, the provided feature vectors are directly utilized in their existing form.

With respect to dataset splits, we diverge from the conventional approach commonly adopted in the literature by employing a stratified split scheme. We report the used split ratios at the end of each dataset description.

Wine Dataset [1]: is a multivariate dataset consisting of attributes depicting the chemical properties of wines. It comprises 178 instances of wine samples, each described by 13 attributes, such as alcohol content, malic acid, etc. The task is to classify the wines into 3 categories corresponding to different cultivars. For Wine we follow a 90-10 train/test split leading to 168 train & 18 test samples.

Iris Dataset [11]: consists of 150 instances of iris flowers, with each instance described by four continuous attributes: sepal length, sepal width, petal length, and petal width, measured in centimeters. These attributes are used to classify the flowers into three species: Iris setosa, Iris versicolor, and Iris virginica, with each species containing 50 samples. We follow a 90-10 train/test split for this dataset, amounting in 135 train and 15 test samples.

Breast Cancer Wisconsin [41]: comprises of 569 instances of breast cancer cases, each described by 30 continuous attributes derived from digitized images

of fine needle aspirate (FNA) of breast masses. It supports binary classification with the samples labeled to indicate whether the tumor is malignant or benign. For this dataset we follow a 90-10 train/test split (512 train, 57 test samples).

MNIST 28x28: is a subset of the original MNIST dataset [20], designed for digit recognition tasks. It contains 70,000 grayscale images of handwritten digits, each image sampled to a 28x28 pixel resolution, resulting in 784 features per image. Each image represents one of the digits from 0 to 9, with the original dataset being balanced across the 10 classes. For MNIST 28x28, we follow a 70-20-10 train/val/test split scheme to construct an unbalanced variant.

CIFAR-10 [18]: consists of 60,000 color images divided into ten distinct classes. Each image is of size 32x32 pixels. The ten classes include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. For CIFAR-10, we follow a 60-20-20 train/val/test split scheme.

CALTECH-101 [10]: containing 9,146 images categorized into 101 distinct object classes and one additional background class. Each object class contains between 40 to 800 images, with most classes containing around 50 images. The images vary in size, however, for our experiments we resize all images to a $200x300$ resolution. We follow a 70-30 train/test split for this dataset.

4.2 Baseline Methods & Classifiers

Methods: We compare the proposed variants of FINCH with other prominent instance selection methods, representative of the basic categories of existing instance selection methods. Specifically, we include edition (ENN [40], RENN [34], LSSm [21], LDIS [8], XLDIS [7]), condensation (CNN [16], AP [12]), and hybrid (DROP3 [39], IB3 [2]) methods. **Classification Models**: To evaluate the quality of the selected instances of each method in each dataset, we trained classification models specific to each dataset. For the MNIST and CIFAR-10 datasets, we employed a lightweight convolutional neural network. The network was trained for {10 epochs MNIST, 20 epochs CIFAR-10} with a batch size of 64, using the Adam optimizer (learning rate: 0.001). In contrast, for the Iris, Wine, Breast Cancer, and CALTECH-101 datasets, we utilized a Support Vector Machine (SVM) classifier with a linear kernel. In all cases, the models were trained exclusively on the selected instances and evaluated on the test set.

4.3 Evaluation metrics

We assess the methods based on three key performance criteria: (a) *classification accuracy* A when training is performed on the reduced dataset resulting from instance selection, (b) *Running time* T of the instance selection algorithm, and, (c) *number of instances* I selected by the algorithm. We provide two different rankings of the methods, (a) *Criterion-based ranking* and (b) *Total ranking*.

Criterion-based ranking: This ranking evaluates each method against an individual criterion (A, T, I) in all employed datasets. Initially, we rank each method according to each criterion within each dataset. Then, we aggregate

[**Top-1 Accuracy** A **(%) — Rank] per method, per dataset**													
Method	**Wine**		**Iris**		**B.C.**		**MNIST**		**CIFAR10**		**CTECH**		R_A
RENN [34]	100.00	1	100.00	1	92.98	1	98.76	1	48.67	8	34.88	7	**2**
ENN [40]	100.00	1	100.00	1	91.23	2	98.76	1	48.67	8	34.88	7	**3**
LSSm [21]	100.00	1	100.00	1	89.47	6	98.36	3	70.14	1	46.50	1	**1**
XLDIS [7]	94.44	7	86.67	10	91.23	2	95.11	11	25.95	12	31.20	10	**10**
LDIS [8]	88.89	11	86.67	10	89.47	6	95.93	10	26.15	11	31.16	11	**12**
IB3 [2]	94.44	7	93.33	8	89.47	6	97.61	5	67.88	2	44.93	2	**6**
CNN [16]	94.44	7	93.33	8	89.47	6	96.07	9	63.83	3	35.79	6	**9**
DROP3 [39]	94.44	7	100.0	1	87.72	10	96.24	8	53.65	4	35.90	5	**8**
AP [12]	88.89	11	86.67	10	91.23	2	95.03	12	36.55	10	27.15	12	**10**
FINCH [30]	100.00	1	100.00	1	87.72	10	97.24	7	49.46	7	32.73	9	**7**
KM-F$_{cw}$	100.00	1	100.00	1	91.23	2	97.74	4	53.01	6	36.01	4	**4**
KM-F$_{cp}$	100.00	1	100.00	1	85.96	12	97.54	6	57.65	4	41.11	3	**6**

Table 1. Evaluation of test accuracy of the methods across all considered datasets. B.C. is an abbreviation for the Breast Cancer dataset and CTECH for CALTECH-101. R_A refers to the accuracy ranking aggregated over all datasets via averaging.

these rankings to determine the performance on that criterion across all datasets by computing the average ranking. This results in the rankings R_A, R_T and R_I of a certain method with respect to A, T and I, respectively, across all datasets.

Total ranking: Holistic ranking aims at evaluating the best algorithm by considering all criteria, simultaneously. To do so, we compute a normalized aggregation function that scores a method over all three criteria on each dataset. Specifically, given a criterion $X \in \{A, T, I\}$, and a dataset, we first perform min-max normalization of the scores of the method i with respect to that criterion so that its normalized value X_N is brought in the range $[0, 1]$:

$$X_N = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{7}$$

In equation 7, X is the score of a method in a dataset and X_{min}, X_{max} the minimum and maximum scores of all methods on this dataset, respectively. The total score S of a method is defined based on the following aggregation function:

$$S = \exp\left([A_N + (1 - T_N) + (1 - I_N)] - 3\right). \tag{8}$$

S increases with normalized accuracy, and decreases with normalized running time and normalized number of instances. Subtraction of 3 in the exponential normalizes S in the range $[0, 1]$. Using the normalized scores guarantees that the three criteria contribute equally to the total score S, implementing the default choice of all of them being considered equally important. If not, equation 8 can be adjusted appropriately by adopting a proper weighting scheme. More elaborate multi-criteria ranking methods can be used (e.g., the Condorcet method [37]) but such an investigation is beyond the scope of this paper. Finally, all methods are ranked according to S, giving rise to their total ranking R_S.

5 Experimental Results

We evaluate the proposed instance selection framework by first considering each criterion individually in Section 5.1, followed by total ranking results in Section 5.2. Finally, Section 5.3 presents a case study showing how the proposed instance selection method significantly reduces the training dataset size for a ship detector with minimal impact on detection accuracy.

5.1 Performance evaluation on individual criteria

Classification Accuracy A: Table 1 presents the top-1 accuracy A of a classifier when trained on datasets (columns) where instance selection is applied based on a variety of algorithms (rows). In the table, B.C. is an abbreviation for the Breast Cancer dataset and CTECH for CALTECH-101. The last three rows show the performance of FINCH alone, as well as of the two proposed variants KM-F$_{cw}$, KM-F$_{cp}$. Besides accuracy, the table also reports the ranking of each algorithm per dataset. The last column of the table presents the ranking R_A with respect to accuracy of all algorithms over all datasets. We can observe that KM-F$_{cw}$ consistently performs at a high level, achieving the highest ranks in the Wine and Iris datasets and maintaining competitive ranking across MNIST and CIFAR10. Its aggregate rank of 4 underscores its overall robustness and reliability compared to other methods such as RENN and LSSm, which, while also performing well, exhibit lower ranks on more complex datasets like CIFAR10. KM-F$_{cp}$ similarly excels in the Wine and Iris datasets but shows low performance on the Breast Cancer dataset, leading to a slightly lower aggregate rank of 6. This variability highlights its less consistent performance across diverse datasets compared to KM-F$_{cw}$. FINCH, while competitive with an aggregate rank of 7, demonstrates high accuracy in simpler datasets and maintains a balanced performance across others. Despite its lower performance in the Breast Cancer dataset, FINCH remains effective but does not surpass KM-F$_{cw}$ in overall consistency.

Number of selected instances I: In Table 2 we present the number of selected instances for each method across the examined datasets. KM-F$_{cw}$ provides substantial reductions in instance selection (mean reduction is $\geq 80\%$), particularly in image datasets. KM-F$_{cp}$, while effective, shows slightly less effectiveness, with a mean reduction of 77% across all datasets. Comparatively, methods like XLDIS and LDIS achieve the highest reduction, with mean reductions of 95% and 93%, respectively. Other methods, such as IB3, CNN, and DROP3, show moderate effectiveness with mean reductions ranging from 80% to 90%, but exhibit inconsistency across different datasets. Lower-performing methods (RENN, ENN, LSSm) select a larger number of instances, resulting in lower reductions.

Overall, the results indicate that KM-F$_{cw}$ lies in the above average range in terms of instance selection efficiency across diverse datasets. Its performance is constant across both biological and image datasets indicating that it can be considered as a reliable data reduction choice for diverse domains.

Execution Time T: The results in Table 3 provide a detailed comparison of execution times across various instance selection methods. KM-F$_{cw}$ and KM-

[**Selected samples I — Rank**] **per method, per dataset**													
Method	**Wine) (160)**		**Iris (135)**		**B.C. (512)**		**MNIST (48k)**		**CIFAR10 (35k)**		**CTECH (6.4k)**		R_I
RENN [34]	140	10	115	11	488	11	42620	10	6387	7	1524	8	**10**
ENN [40]	146	11	115	10	493	12	42620	10	6387	7	1524	8	**11**
LSSm [21]	156	12	124	12	498	12	45490	11	28788	11	4912	11	**12**
XLDIS [7]	15	2	10	2	25	1	3172	1	298	1	314	2	**1**
LDIS [8]	18	4	10	2	26	2	3650	3	366	2	352	3	**3**
IB3 [2]	27	1	20	5	43	3	6103	5	24739	10	4499	10	**7**
CNN [16]	54	9	57	9	233	9	7592	6	20173	9	1406	7	**8**
DROP3 [39]	15	2	16	4	43	3	4778	4	3967	6	596	5	**4**
AP [12]	14	1	8	1	45	5	3270	2	638	3	168	1	**2**
FINCH [30]	39	6	36	7	109	7	8944	7	1962	4	451	4	**5**
KM-F$_{cw}$	39	6	34	6	108	6	9127	8	2640	5	789	6	**5**
KM-F$_{cp}$	45	8	47	8	125	8	11089	9	6784	8	2211	9	**9**

Table 2. Evaluation of the instance selection outcome of each method across all datasets. R_I refers to the ranking aggregated over all datasets via averaging. For each dataset in the headline, we include the original sample pool size.

[**Execution Time T (sec) — Rank**] **per method, per dataset**													
Method	**Wine**		**Iris**		**B.C.**		**MNIST**		**CIFAR10**		**CTECH**		R_T
RENN [34]	0.388	10	0.444	12	0.674	8	286	7	504	6	867	7	**8**
ENN [40]	0.482	11	0.422	10	1.114	10	273	6	529	7	865	6	**8**
LSSm [21]	0.269	8	0.247	7	0.663	7	5444	10	3119	10	6416	11	**10**
XLDIS [7]	0.231	6	0.214	4	0.251	5	136	5	086	4	155	1	**4**
LDIS [8]	0.227	5	0.216	5	0.242	4	109	3	85	3	174	2	**3**
IB3 [2]	0.248	7	0.231	6	0.263	6	828	8	7449	9	5552	10	**7**
CNN [16]	1.802	12	0.436	11	4.213	11	3784	9	8275	11	3520	9	**11**
DROP3 [39]	0.336	9	0.294	9	0.733	9	135267	12	126039	12	162219	12	**11**
AP [12]	0.225	4	0.247	7	10.623	12	18941	11	1037	8	180	3	**6**
FINCH [30]	0.077	1	0.081	1	0.100	2	124	4	71	1	376	4	**1**
KM-F$_{cw}$	0.087	2	0.092	2	0.079	1	86	2	79	2	502	5	**1**
KM-F$_{cp}$	0.145	3	0.153	3	0.155	3	81	1	174	5	1790	8	**4**

Table 3. Evaluation of the execution time (secs) of each method across all considered datasets. R_T refers to the ranking aggregated over all datasets via averaging.

F$_{cp}$ demonstrate impressive performance, securing top positions in execution efficiency across multiple datasets. KM-F$_{cw}$ consistently ranks in the top two for all datasets, showing particularly strong performance in the MNIST and CIFAR10 datasets. KM-F$_{cp}$ also performs exceptionally well, achieving the fastest execution time for MNIST and maintaining competitive execution times for other datasets. In comparison, methods such as LSSm, CNN, and DROP3 show much higher execution times, particularly in large datasets like MNIST and CIFAR10, where their times can be several orders of magnitude higher. For instance, LSSm records an execution time of 5444 seconds for MNIST and 3119 seconds for CIFAR10, while DROP3 exhibits the highest execution times overall.

5.2 Total Ranking

We evaluate the overall performance of the proposed methods according to the total score S (see equation 8). The obtained results are presented in Table 4, which reveals several key insights regarding the performance of the methods.

The proposed methods, KM-F$_{cw}$ and KM-F$_{cp}$, demonstrate superior performance across all datasets. Specifically, the class-wise application of FINCH variant, KM-F$_{cw}$, achieved the highest aggregated rank ($R_S = 1$), consistently securing top positions across the majority of datasets, including the first rank in CIFAR10 and CALTECH-101 datasets. KM-F$_{cp}$ also performed exceptionally well, attaining an overall rank of 3, highlighting its robustness, particularly in biological datasets like Wine and Iris. In comparison, the baseline methods exhibited varied performance. FINCH, from which KM-F$_{cw}$ and KM-F$_{cp}$ are derived, secured the second overall rank, indicating the foundational strength of the FINCH method. Methods such as IB3 and XLDIS showed competitive performance with overall ranks of 4 and 5, respectively, suggesting their utility in specific scenarios but not consistently across all datasets. Conversely, methods like LSSm and ENN ranked lower ($R_s = 12$ and 10, respectively), indicating potential limitations in their applicability or efficiency across diverse datasets. RENN, while showing moderate performance ($R_s = 7$), demonstrated variability across different datasets, reflecting a more specialized than general approach.

5.3 Case Study: Instance Selection on Augmented Datasets

To further evaluate the proposed methodology and to highlight its potential impact, we employ the method proposed by Savathrakis and Argyros [31] as a case study. In this work, the authors proposed an automated method to convert Horizontal Bounding Boxes (HBBs) into Oriented Bounding Boxes (OBBs) in ship detection datasets, specifically for HRSC2016 [24] and ShipRSImageNet [43]. Their approach leverages the Segment-Anything Model (SAM) [17] for object segmentation, followed by morphological filtering and contour detection to accurately compute OBBs from the segmented masks. Given the computed OBBs, they perform data augmentation by synthesizing unseen ship views at various orientations. The contribution of this proposed augmentation scheme is evaluated on these two datasets, with a set of top-performing oriented object detectors.

Setup: We apply the proposed instance selection variants to the Increased Size Object-wise (ISO) augmentation variant from [31], which resulted in the largest improvements in detection accuracy. Utilizing our proposed methods, KM-F$_{cw}$ and KM-F$_{cp}$, we perform instance selection in these augmented datasets to train the object detectors referenced in the original study. We adhere to the training configurations reported in the original paper, and similar to [31], we use mean Average Precision (mAP) to evaluate the performance of the detectors. For our experiments we examined the following detectors, R^{3}Det [42], ReDet [15], Rotated-RetinaNet [22], and, S^{2}A-Net [14].

In our experiments, we focus exclusively on the ISO augmentation variant of HRSC2016 [24]. To create the reduced augmented ISO variants, we combine

[Total Score S — Rank] per method, per dataset													
Method	**Wine**		**Iris**		**B.C.**		**MNIST**		**CIFAR10**		**CTECH**		R_S
RENN [34]	0.344	7	0.146	12	0.355	7	0.393	8	0.495	4	0.425	8	**7**
ENN [40]	0.312	11	0.155	10	0.263	11	0.393	7	0.494	5	0.426	7	**10**
LSSm [21]	0.329	9	0.233	9	0.211	12	0.317	11	0.359	11	0.354	11	**12**
XLDIS [7]	0.551	4	0.251	6	0.767	1	0.376	9	0.368	10	0.442	5	**5**
LDIS [8]	0.328	10	0.249	7	0.596	3	0.463	5	0.369	9	0.437	6	**6**
IB3 [2]	0.501	6	0.361	5	0.573	4	0.682	1	0.395	8	0.402	9	**4**
CNN [16]	0.168	12	0.149	11	0.264	10	0.426	6	0.404	7	0.448	4	**8**
DROP3 [39]	0.518	5	0.519	4	0.428	5	0.180	12	0.223	12	0.197	12	**8**
AP [12]	0.338	8	0.233	8	0.275	9	0.319	10	0.459	6	0.368	10	**10**
FINCH [30]	0.839	1	0.786	1	0.395	6	0.580	4	0.622	2	0.465	3	**2**
KM-F$_{cw}$	0.834	2	0.775	2	0.654	2	0.661	2	0.634	1	0.518	1	**1**
KM-F$_{cp}$	0.773	3	0.586	3	0.296	8	0.598	3	0.600	3	0.515	2	**3**

Table 4. Method ranking, under the proposed total score S (Equation 8). R_S refers to the total score ranking aggregated over all datasets via averaging.

two sample definition strategies: (a) we consider the entire scene in the image samples of ISO, denoted as S_{im}, and (b) we consider the cropped ship-containing regions within the images as samples, denoted as S_{cr}. We apply the proposed methods to both S_{im} and S_{cr} and generate two sets of selected instances, S'_{im} and S'_{cr}. The final, reduced set, S_{fin}, is then union of S'_{im} and the dataset images containing the samples from S'_{cr}. The rationale for this sample formulation is to select samples that are representative of both the overall scene and those that encompass characteristic ship appearances and orientations.

Results: The results in Table 5 demonstrate that KM-F$_{cw}$ successfully reduces the dataset size by approximately 33%. This substantial data reduction results in minimal loss of detection accuracy, with decreases of less than 3% for the R^3Det and S^2A-Net detectors. Despite the significant reduction in data volume, the ReDet and Rotated-RetinaNet detectors still perform with accuracy drops of less than 9%. The second proposed variant, KM-F$_{cp}$, achieves an 8% reduction in dataset size while maintaining higher detection accuracy with minor drops of approximately 1% or even leading to marginal accuracy improvements, as observed in the case of the Rotated-RetinaNet detector.

When comparing the performance of the instance selection algorithm that integrates the original FINCH with K-means for final instance subset creation to our proposed variants, we observe that the combined approach results in a substantially smaller number of selected instances—approximately 38% fewer than those selected by the KM-F$_{cw}$ method. However, this reduction in instance count is accompanied by a notable decline in detection accuracy, with a decrease of approximately 10% relative to KM-F$_{cw}$. As a final observation, the reduction in dataset size significantly affects the training duration of the models. Specifically, models trained on the selected subset generated by KM-F$_{cw}$ converge in approximately half the time required for training on the original ISO dataset.

HRSC2016 dataset variants (#samples), mAP (%)				
Method	**ISO [31]** (1455)	**ISO**$_{KM-F_{cw}}$ (571)	**ISO**$_{KM-F_{cp}}$ (1158)	**ISO**$_{KM-F}$ (351)
R^3Det [42]	88.33	86.12	87.86	76.56
ReDet [15]	87.71	78.73	86.52	75.94
Rotated-RetinaNet [22]	78.43	67.42	78.72	56.15
S^2A-Net [14]	89.23	87.53	88.79	79.33

Table 5. mAP scores for object detectors, trained on (a) ISO [31], (b) ISO + KM-F$_{cw}$, referred to as ISO$_{KM-F_{cw}}$, (c) ISO + KM-F$_{cp}$, referred to as ISO$_{KM-F_{cp}}$, and (d) ISO + (FINCH + K-means), referred to as ISO$_{KM-F}$. Each dataset is an augmented variant of HRSC2016 [24]. The value in (·) refers to sample size.

6 Summary

In this work, we introduced two novel instance selection methods, KM-F$_{cp}$ and KM-F$_{cw}$, which integrate the FINCH and K-means clustering algorithms into a unified process specifically designed for the task at hand. Our empirical evaluations reveal that FINCH emerges as an effective instance selection method exhibiting high overall performance. Consequently, FINCH can also serve as a base model for the development of more sophisticated instance selection strategies. Our results indicate that such variants yield improved instance selection performance relative to other established methods. Finally, we demonstrate the practical utility of our proposed methods in the context of augmented dataset size reduction, specifically by eliminating redundant augmentations. In a case study focusing on an augmentation technique for ship detection in aerial and satellite imagery, we illustrated that our methods substantially decrease the augmented dataset size while preserving detection accuracy across various detectors.

Acknowledgements

This work was co-funded by the European Union (EU - HE Magician – Grant Agreement 101120731) and by the Hellenic Foundation for Research and Innovation (HFRI) under the "1st Call for HFRI Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment", project I.C.Humans, no 91. The authors also gratefully acknowledge the support for this research from the VMware University Research Fund (VMURF). The authors are grateful to Giorgos Savathrakis, for generously providing the data and implementation detailed in [31] as well as for his valuable guidance in adapting it for the experiments of this work.

References

1. Aeberhard, S., Forina, M.: Wine. UCI Machine Learning Repository (1991), DOI: https://doi.org/10.24432/C5PC7J

2. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine learning **6**, 37–66 (1991)
3. Albelwi, S., Mahmood, A.: Analysis of instance selection algorithms on large datasets with deep convolutional neural networks. In: 2016 IEEE Long island systems, applications and technology conference (LISAT). pp. 1–5. IEEE (2016)
4. Blachnik, M., Kordos, M.: Comparison of instance selection and construction methods with various classifiers. Applied Sciences **10**(11) (2020). https://doi.org/10.3390/app10113933, https://www.mdpi.com/2076-3417/10/11/3933
5. Branikas, E., Papastergiou, T., Zacharaki, E.I., Megalooikonomou, V.: Instance selection techniques for multiple instance classification. In: 2019 10th international conference on information, intelligence, systems and applications (IISA). pp. 1–7. IEEE (2019)
6. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. IEEE transactions on evolutionary computation **7**(6), 561–575 (2003)
7. Carbonera, J.L.: An efficient approach for instance selection. In: Bellatreche, L., Chakravarthy, S. (eds.) Big Data Analytics and Knowledge Discovery. pp. 228–243. Springer International Publishing, Cham (2017)
8. Carbonera, J.L., Abel, M.: A density-based approach for instance selection. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 768–774. IEEE (2015)
9. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967). https://doi.org/10.1109/TIT.1967.1053964
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop. pp. 178–178 (2004). https://doi.org/10.1109/CVPR.2004.383
11. Fisher, R.A.: Iris. UCI Machine Learning Repository (1988), DOI: https://doi.org/10.24432/C56C76
12. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007). https://doi.org/10.1126/science.1136800, https://www.science.org/doi/abs/10.1126/science.1136800
13. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE transactions on pattern analysis and machine intelligence **34**(3), 417–435 (2012)
14. Han, J., Ding, J., Li, J., Xia, G.S.: Align deep features for oriented object detection. IEEE transactions on geoscience and remote sensing **60**, 1–11 (2021)
15. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: A rotation-equivariant detector for aerial object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2786–2795 (2021)
16. Hart, P.: The condensed nearest neighbor rule (corresp.). IEEE transactions on information theory **14**(3), 515–516 (1968)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

19. Lapedriza, À., Pirsiavash, H., Bylinskii, Z., Torralba, A.: Are all training examples equally valuable? ArXiv **abs/1311.6510** (2013), https://api.semanticscholar.org/CorpusID:1019326
20. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
21. Leyva, E., González, A., Pérez, R.: Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. Pattern Recognition **48**(4), 1523–1537 (2015)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, Y., Cao, J., Liu, C., Ding, K., Jin, L.: Datasets for large language models: A comprehensive survey. arXiv preprint arXiv:2402.18041 (2024)
24. Liu, Z., Yuan, L., Weng, L., Yang, Y.: A high resolution optical satellite image dataset for ship recognition and some new baselines. In: International conference on pattern recognition applications and methods. vol. 2, pp. 324–331. SciTePress (2017)
25. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489
26. Malhat, M., El Menshawy, M., Mousa, H., El Sisi, A.: A new approach for instance selection: Algorithms, evaluation, and comparisons. Expert Systems with Applications **149**, 113297 (2020)
27. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artificial Intelligence Review **34**, 133–143 (2010)
28. Saha, S., Sarker, P.S., Al Saud, A., Shatabda, S., Newton, M.H.: Cluster-oriented instance selection for classification problems. Information Sciences **602**, 143–158 (2022)
29. Salari, A., Djavadifar, A., Liu, X., Najjaran, H.: Object recognition datasets and challenges: A review. Neurocomputing **495**, 129–152 (2022). https://doi.org/https://doi.org/10.1016/j.neucom.2022.01.022, https://www.sciencedirect.com/science/article/pii/S092523122200039X
30. Sarfraz, S., Sharma, V., Stiefelhagen, R.: Efficient parameter-free clustering using first neighbor relations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8934–8943 (2019)
31. Savathrakis, G., Argyros, A.: An automated method for the creation of oriented bounding boxes in remote sensing ship detection datasets. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 830–839 (2024)
32. Sieranoja, S.: How much k-means can be improved by using better initialization and repeats? Pattern Recognition **93**, 04 (2019)
33. Skalak, D.B.: Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Cohen, W.W., Hirsh, H. (eds.) Machine Learning Proceedings 1994, pp. 293–301. Morgan Kaufmann, San Francisco (CA) (1994). https://doi.org/https://doi.org/10.1016/B978-1-55860-335-6.50043-X, https://www.sciencedirect.com/science/article/pii/B978155860335650043X
34. Tomek, I.: An experiment with the edited nearest-nieghbor rule. (1976)
35. Tsai, C.F., Chen, Z.Y.: Towards high dimensional instance selection: An evolutionary approach. Decision Support Systems **61**, 79–92

(2014). https://doi.org/https://doi.org/10.1016/j.dss.2014.01.012, https://www.sciencedirect.com/science/article/pii/S016792361400013X
36. Tsai, C.F., Lin, W.C., Hu, Y.H., Yao, G.T.: Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Information Sciences **477**, 47–54 (2019)
37. Wallis, W.D.: Simple Elections II: Condorcet's Method, pp. 19–32. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-09810-4_3, https://doi.org/10.1007/978-3-319-09810-4_3
38. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
39. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine learning **38**, 257–286 (2000)
40. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics (3), 408–421 (1972)
41. Wolberg, William, M.O.S.N., Street, W.: Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository (1995), DOI: https://doi.org/10.24432/C5DW2B
42. Yang, X., Yan, J., Feng, Z., He, T.: R3det: Refined single-stage detector with feature refinement for rotating object. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 3163–3171 (2021)
43. Zhang, Z., Zhang, L., Wang, Y., Feng, P., He, R.: Shiprsimagenet: A large-scale fine-grained dataset for ship detection in high-resolution optical remote sensing images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 8458–8472 (2021)
44. Zhao, B., Mopuri, K.R., Bilen, H.: Dataset condensation with gradient matching. arXiv preprint arXiv:2006.05929 (2020)