IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 November 2022, accepted 29 November 2022, date of publication 2 December 2022,
date of current version 8 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226517

RESEARCH ARTICLE

Tackling Dataset Bias With an Automated Collection of Real-World Samples

VASILEIOS SEVETLIDIS[1,2], (Student Member, IEEE), GEORGE PAVLIDIS[1], (Senior Member, IEEE), SPYRIDON MOUROUTSOS[3], AND ANTONIOS GASTERATOS[2], (Senior Member, IEEE)

[1]Athena Research Center, University Campus at Kimmeria, 67100 Xanthi, Greece
[2]Department Production and Management Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
[3]Department Electrical and Computer Engineering, Democritus University of Thrace, University Campus at Kimmeria, 67100 Xanthi, Greece

Corresponding author: Vasileios Sevetlidis (vasiseve@athenarc.gr)

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-01018).

ABSTRACT The early 21st-century technological advancements tilted the scales towards data-driven learning. Thus, modern machine-learning systems rely heavily on data to learn complex models to efficiently provide relevant predictions. Data-driven learning suffers from overfitting, a situation in which the learning process seems to have converged into a model that, unfortunately, lacks generalization power. One way to withstand overfitting is to expand the training dataset with more diverse samples. Typically, this is implemented (particularly in computer vision research, which is of interest in this study) by multiplying the original sample using several transformations. Although this strategy might seem straightforward, it does not affect any preexisting dataset bias because the initial distribution remains more or less similar. Ideally, new samples of unseen data must be found, but the cost of acquiring them individually is high. This study presents a novel pipeline that combines state-of-the-art modules to automatically create new thematic datasets with low bias. The proposed method was able to acquire and allocate more than 880K previously unseen images to produce a data collection, that InceptionV3 classified it with 72% accuracy and achieved 0.0008 performance variance when testing on similar datasets.

INDEX TERMS AI, dataset bias, domain shift, image datasets, machine learning, web search.

I. INTRODUCTION

Supervised learning is a method for accurately modeling a known generator function that produces data distribution [1]. Often, engineers and AI practitioners that apply supervised learning need to tackle a phenomenon with a significant impact on the learning process, called overfitting. Overfitting occurs when a model captures the underlying pattern and noise of the data to the extent that it cannot generalize the learned concept and make accurate predictions on unseen data. In cases where model alteration techniques, such as simplification, regularization, and ensembling, address this issue only up to a point then the problem lies within the data.

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.

In recent years, another occurrence that hinders model performance on unseen data has been observed. Sometimes, a model performs well on held-out samples of a dataset which simulate an unknown distribution. However, this does not necessarily mean that the model behaves correspondingly well with unseen real-world data. This phenomenon is termed as 'dataset bias' [2].

Dataset bias occurs when the inclusion of samples within a class follows a specific set of rules. Then, even if a model can learn how to recognize the dataset's classes, it is assumed that all distribution instances adhere to the same set of rules. For instance, SUN09 [3] and Caltech101 [4] are generic benchmark datasets that share a class that describes *cars*. Ideally, learning on one side and inferring on the other should work well. However, this was not the case in

practice. According to [5], one dataset portrays the front face of cars, whereas the other depicts them mostly from the aside.

Depending on the application, bias is not harmful by itself. Consider, for example, the case of quality control on a production line, where one needs to ensure that a defect is never missed due to safety implications. Thus, introducing bias towards recognizing defective samples costs less than the misrecognized ones. However, a benchmark should ideally be a representative instance of the natural world, so the computer vision community should focus on trying to deal with the problem instead of modeling the dataset. This observation has been pointed out by Hand [6]: "*However, this also means that there will be some overfitting both to the individual data sets in the collection and the collection as a whole. That is, some methods will do well on data sets in the collection purely by chance. Indeed, the more successful the collection is in the sense that more and more people use it for comparative assessments, the more serious this problem will become*." Acquiring such datasets with real-world distributions is an impossible and practically infeasible endeavor. Perhaps the best course of action would be to sample as many distribution instances as possible to approximate the description of the real world.

Handpicking such samples is cost ineffective, time consuming, and prone to mistakes. Automatic data acquisition from resources such as the web is not a trivial assignment because the lack of a sophisticated methodology is expected to lead to a noisy, imbalanced, and even error-prone assortment of data. A wide range of challenges are likely to be encountered when designing frameworks to undertake this task, such as (a) irrelevant content removal, (b) correct data for class assignment, (c) composition of suitable queries, and (d) rectification of corrupted data. In addition, common concerns exist in dataset formalization procedures, including (a) balancing samples per class, (b) dealing with conflicting content, (c) constructing and leveraging inclusion or mutual exclusion strategies, and (d) handling missing data. These are all issues that need to be considered during the creation of benchmark datasets [7], [8], landmark recognition [9], and in more generic datasets [10]. Therefore, assembling a formal AI-useful dataset from the Web, even to expand existing datasets, remains challenging.

Motivated by the challenges mentioned above, this study proposes a pipeline that acquires samples from the Web to remedy the dataset bias problem. Emphasis is placed on producing a robust and accurate candidate selection method based on removing irrelevant content and preserving the ranking of the most representative samples. The main contribution of this work is a pipeline with novel combinations of modules which

1) effortlessly creates a dataset given a list of labels,
2) expands/augments existing datasets with previously unseen real-world samples,
3) requires minimal to none human involvement, and
4) mitigates dataset bias.

The following sections include a review of the relevant literature, presentation of the novel pipeline, and experimental verification of the proposed procedure.

II. LITERATURE REVIEW

A. DATASET BIAS

The mismatch between the datasets used in the pattern recognition domain for developing better algorithms under laboratory conditions and real applications has been discussed as early as in [6]. It has been illustrated through examples that in many, perhaps most, real classification problems, the data points in the design set are not, in fact, randomly drawn from the same distribution as the data points to which the classifier will be applied, for example, the real world.

In the computer vision domain, this observation became widely known in the work of Torralba and Efros [2], where it was pointed out that the cross-evaluation of categories shared between different datasets led to a lower performance than expected. The authors attributed this phenomenon to the data collection procedure, which can be biased by human and systematic factors, resulting in a distribution mismatch between datasets.

Several studies have focused on overcoming this problem, mainly by developing classifiers with better generalization properties [5]. Datasets with common categories such as SUN [3], LabelMe [11], PascalVOC [12], Caltech101 [4], and ImageNet [13] have been used to cross-evaluate the performance of such classifiers on shared classes. The problem of partially overlapping label sets among different datasets was also considered [14].

The computer vision community has become increasingly aware that existing benchmarks present a characteristic signature that differs from one dataset to the other. This observation spawned the related domain shift problem, for example, performance discrepancies appear on the source and target image datasets with different marginal probability distributions when training on one and inferring on the other. Shared representations to eliminate the original distribution mismatch, such as subspace data embedding [15], [16], metrics [17], [18] and vocabulary learning [19] have been presented. Other studies have demonstrated that deep learning architectures might produce domain-invariant descriptors through a highly nonlinear transformation of the original features [20], [21].

Domain adaptation techniques have been extensively studied [22], [23], [24]. These approaches intrinsically rely on having more than one dataset to observe the shift in the domain; otherwise, the dataset bias problem is concealed in the first place. To this extent, the authors of the present work share the same perspective with [2] that is, "all the datasets are trying to represent the same domain – our visual world", therefore here, it is proposed to construct a visually robust dataset with limited bias from scratch.

B. DATASET CONSTRUCTION

In the early years of image dataset formation, manual acquisition and data annotation were preferred approaches. In situ data acquisition was not an uncommon practice [25], [26]. This involved an expensive overhead prior to analyzing the data and constructing the actual dataset, that is, setting up a camera, removing obstacles from the scene, avoiding capturing humans and human parts to preserve their anonymity, and finally composing the frame [27], [28], [29]. Exploiting the Web as a source for manually gathering images is labor intensive. This task consists of (a) querying the web with a few keywords, (b) downloading the content, and (c) exhaustively manually annotating the downloaded samples to clear out unwanted material. Annotation normalization and alignment must be applied when human annotators collaborate in a task. Benchmark datasets, such as ImageNet [13] (a dataset that took years to complete), CIFAR-10 [10], and ETH-Food-101 [30], are striking examples of datasets constructed by relying extensively on human expertise.

Several frameworks have been proposed to reduce the overhead of manual formulation. In [31], visual information was used to re-rank the retrieved images from the web. However, this method relies on statistical modeling of bootstrapped classifiers to discard unrelated images. Reference [32] used active learning methods to iteratively improve the confidence of unlabelled samples using existing labeled data. Early attempts to automatically construct image datasets involved retrieving images and textual information through a web search. In [33], latent topics were extracted from textual information, and voting classifiers assigned them to related images. In [34], a sequential framework was proposed that takes advantage of the ranking information for the first few results offered by a web search engine. Under the assumption that these images fall correctly into the requested label, a binary classifier was incrementally refined to assess whether those images are relevant or not. Finding content in such an uncontrolled environment might be faster than in other approaches, but introduces additional challenges, viz., the sample per class distribution might not be the desired, and object co-occurrence is often disregarded. As discussed in [35], this problem is ubiquitous in the food image recognition domain, where various image descriptors have been used to extract local and global features to recognize multiple-foods photos considering co-occurrence statistics. In [36], the same authors employed a manifold learning approach to improve results in that domain.

More recent approaches focus on enriching a dataset with subcategories by performing multiple query requests instead of just a single one [37]. To achieve this, vocabulary-based resources have been utilized, such as WordNet [38], ConceptNet [39] and the Google Books Ngram Corpus [40]. Using synonyms or word pairs to multiply facets of a keyword can substantially increase the number of candidate samples [41]. However, unwanted content due to word ambiguity will certainly appear [42], and a mechanism for discarding it must be developed. Because word pairs can be of arbitrary length, word-to-word distance has been applied to reduce candidate queries [43]. Nevertheless, the query composition complexity challenge for unambiguous Web retrieval is only partly addressed.

The Web is regarded as a valuable resource, especially when aiming to use it to acquire extensive collections of data for practical AI applications. However, collecting data blindly without treatment can quickly become futile. To meet the challenges in this task, we propose a pipeline attuned to creating AI-usable datasets with data acquired from the Web. This pipeline consists of several modules that collaborate to construct a dataset. This process begins with defining the aim, requirements, and expectations of the dataset. The source from which the labels of the classes were obtained must be credible. For example, ontologies can be employed to construct a formal list of labels for a specific topic. The Europeana ontology [44], [45] for cultural artifacts, the Amalthea hierarchy [46] for food dishes, and the International Code of Zoological Nomenclature (ICZN) [47] for animal taxonomies are a few examples of said ontologies. As an additional advantage, ontologies usually provide rich information about the relationships among the objects they describe. Thus, keyword or query expansion is possible without introducing text mining or understanding complexities. More than one popular search engine can be used to retrieve image data from the Web, such as engines that provide indexed content according to a query [31]. However, undesired content in the form of duplicate and irrelevant images is expected to appear in the retrieved results.

III. MODELLING THE EFFECT OF DUPLICATE AND IRRELEVANT SAMPLES WITHIN THE TRAINING SET

This section presents an *in vitro* ablation study to demonstrate the need for removing duplicate and irrelevant samples when creating image datasets for pattern recognition tasks. Two benchmark datasets and a robust classifier displayed the effects of including the type of noise at different ratios.

The Food-101 (ETHZ [30]) dataset was considered in this study as the target domain for the classifier to learn. It contains 1000 sample images for each of its 101 food dish classes. The dataset is split into training and testing sets with 75-25% ratios according to the initial publication instructions. ImageNet [13] was used to generate irrelevant samples. A random portion is retrieved regardless of the label and class according to the needs of each noise addition stage.

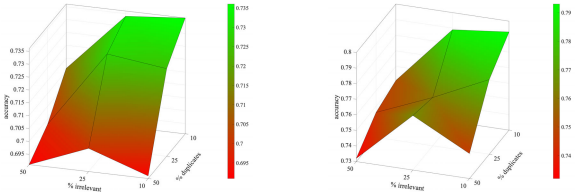
InceptionV3 [48] with its weights initialized on ImageNet, was chosen as the benchmark classifier. This architecture was chosen mainly due to its past state-of-the-art performance, ease of training and parameter configuration, and, last but not least, its wide adoption and usage in the available relevant literature. The top layer was substituted to reflect the 101-class problem, as in [49].

The procedure begins by using the benchmark (Food-101) for training; thus, the baseline classification performance

TABLE 1. Degradation of a classifier's performance when adding duplicate and irrelevant samples: (a) without augmentation; (b) with augmentation.

(a)	Irrelevant (%)		
Duplicate (%)	10	25	50
10	0.736(±0.006)	0.734(±0.004)	0.711(±0.008)
25	0.725(±0.004)	0.728(±0.005)	0.698(±0.003)
50	0.692(±0.009)	0.7(±0.007)	0.691(±0.009)
baseline		**0.747(±0.007)**	

(b)	Irrelevant (%)		
Duplicate (%)	10	25	50
10	0.793(±0.012)	0.789(±0.014)	0.751(±0.012)
25	0.778(±0.005)	0.761(±0.009)	0.746(±0.010)
50	0.746(±0.006)	0.765(±0.013)	0.732(±0.007)
baseline		**0.803(±0.008)**	



(a) Without augmentation (b) With augmentation

FIGURE 1. Model performance with the inclusion of noise at various proportions.

is calculated. Then, the training set is altered to represent an instance in which noise in the form of duplicate and/or irrelevant samples is inserted. In this process, 'noise' samples replace valid ones progressively for each type of noise. The model was reinitialized and trained for each dataset. Throughout these experiments, the test set remained the same with no addition of noise.

As per standard practice, the model was also trained with affine transformations of the training samples to reduce overfitting phenomena. The entire procedure (with and without augmentation via transformations) is repeated ten times, and the mean TOP-1 accuracy and standard deviation are reported in TABLE 1.

FIGURE 1a & 1b depict the mean values of TABLE 1. As anticipated, noise in the combined form of duplicate and irrelevant samples negatively affected the generalization performance of the model. The latter is due to duplicate samples offering no information to the learner other than what has not been seen. On the other hand, irrelevant samples might boost the generalization capacity of a model because it struggles to make up features fitting samples that should not exist in the first place. It should be noted that, in small percentages, both types of noise do not dramatically affect the performance of the model in comparison with those that are absent. However, the greater the noise, the greater the performance degradation, even in cases where their proportion is not the same. Thus, caution should be exercised when acquiring samples from different distributions for a given domain situated in uncontrolled yet rich environments, such as the Web.

IV. PROPOSED PIPELINE

This paper proposes a pipeline to construct image datasets by gathering samples from the Web and assigning them to classes with respect to a list of predefined keywords (i.e., class labels). A high-level graphical overview is shown in FIGURE 2. A list of keywords was used to query the web for the data. The list of query keywords can either be anything that a practitioner provides manually or terms automatically retrieved from an ontology, provided that a parsing mechanism is used. Then the images collected from the Web are processed via modules to filter out irrelevant and duplicate content. Irrelevant images were considered to be outside the scope of the categories described by the query keywords. Duplicate images are exact or geometrically transformed copies of the other images in the dataset. These definitions and related policies were followed throughout the data-collection procedure. Duplicate images were not allowed within a given class or across the dataset.

A. IRRELEVANT DETECTION

A majority voting decision of the binary classifiers was used to reject irrelevant content from the stream of the retrieved samples. The majority voting is similar to unweighted averaging. Nevertheless, instead of averaging the output probability, it counts the votes of all predicted labels from the base learners. It makes a final prediction using the label with the highest number of votes. Equivalently, it takes an unweighted average using the label from the base learners, and chooses the label with the most significant value. This determines whether an image is relevant to the desired dataset or should be discarded.

Three state-of-the-art deep architectures (InceptionV3 [48], MobileNet [50], and ResNet-50 [51]) were used in this study to perform binary classification. All they require is a binary class dataset, which represents the scope of what is considered relevant and what is not. Hence, when a sample reaches this module, the three deep neural network binary classifiers independently vote on whether it is relevant. Subsequently, the majority outcome of the said votes is the final decision: to discard it or to let it move to the next filtering module (deduplication).

B. DUPLICATE DETECTION

A deduplication mechanism was developed to reject duplicate images, leveraging previous work in duplicate image detection, which has been broadly applied in many applications, such as forensics and closed-loop systems. Several methods have been proposed for duplicate image detection, including, but not limited to, relying on textbook image descriptors [52], [53] and bespoke deep neural networks [54], [55], [56]. Typically, a feature extractor combined with a distance measure compares any two images to determine their similarity. The application of a threshold defines the degree of similarity, which in our case, reduces to being a duplicate or not. A reliable method is needed to (a) bring closer those

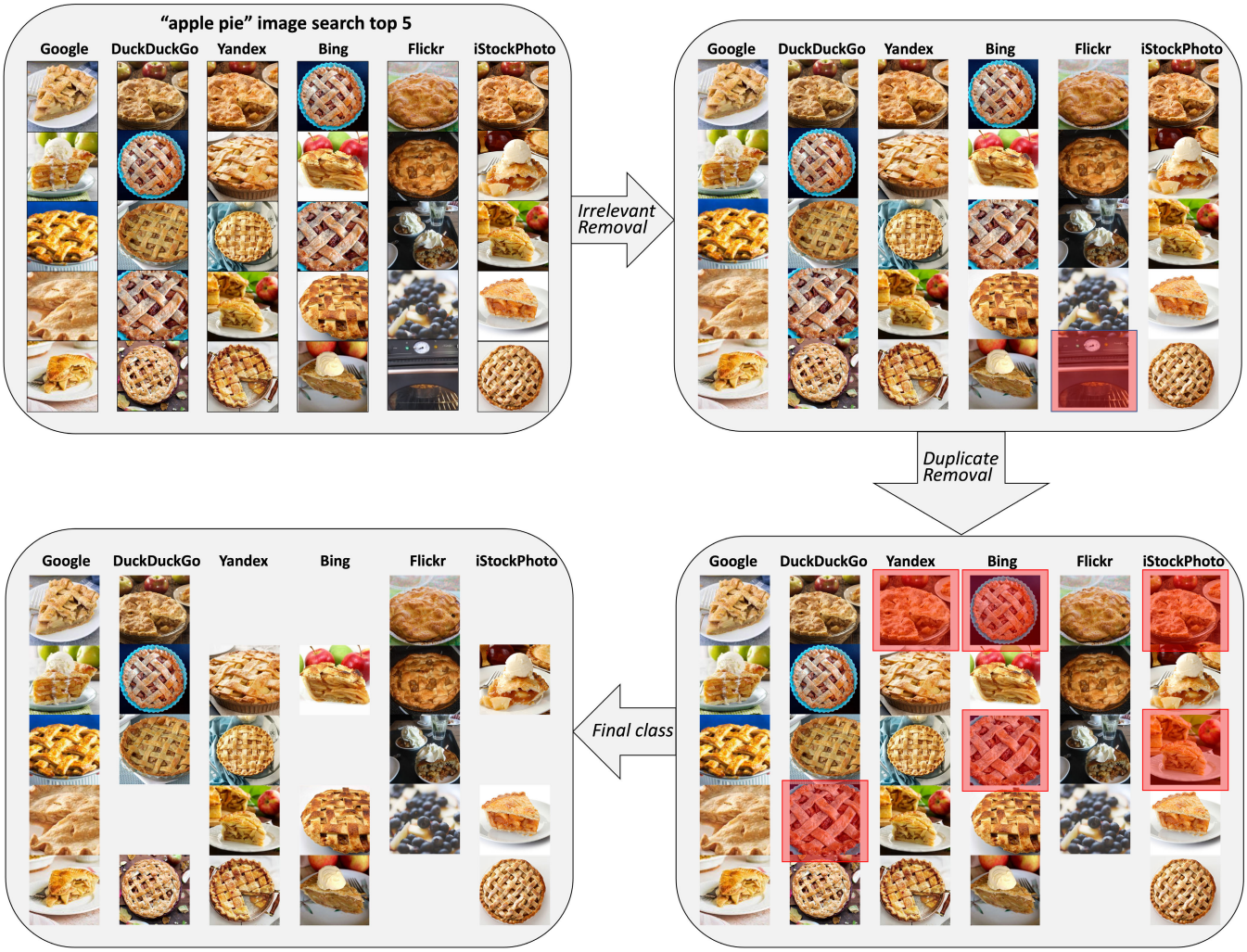


FIGURE 2. The proposed pipeline.

images that are either the same or geometrically transformed copies of one another while at the same time, and (b) separate those that have just a visual resemblance or are entirely different.

Moreover, as this pipeline aims to construct a large-scale dataset, the deduplication method must scale well as more and more data are being collected into the candidate pool. Hence, the manner in which these descriptors are stored affects their retrieval speeds. If these descriptors were put into a simple list, then the search algorithm would be required to query all items individually, resulting in a square search space. Although the triangle inequality holds, the search space is reduced from n^2 to $\frac{n^2-n}{2}$, where n is the number of items. However, the computational complexity still reaches an impractical $O(n^2)$.

To address the computational complexity of the search mechanism, a Burkhar-Keller Tree (BK-Tree) structure [57] has been exploited. It is a tree-based data structure used to find near matches to a string query. The original implementation performed approximate matching in strings, and was used for spell checking [58]. The critical property that constitutes a BK-Tree advantageous as a searching instrument is that it exhibits low complexity. To achieve this, it exploits two functions: triangle inequality and Levenshtein distance. The triangle inequality bounds the solution within the upper and lower limits and allows for the item's ordering. The Levenshtein distance counts the operations required to transform a query word to a match, for example, the next node to visit, while traversing the structure. Finally, the tree structure minimizes the search space and reduces the complexity to $O(nlogn)$.

In this work, we propose two slight modifications to the BK-tree: (a) to store and query the binary feature vectors extracted by locality-sensitive hashing (LSH) algorithms and (b) to use Hamming instead of Levenshtein distance. Hamming distance $\left(\sum_{u_i \neq v_i}^{i=1,\ldots,n} 1\right)$ is a classic metric, as it is a mapping to the set of real numbers ($\mathbb{R}$) with a zero (0) minimum that corresponds to the distance to oneself, which is commutative and satisfies the condition that only equal entities have a zero distance and the triangle

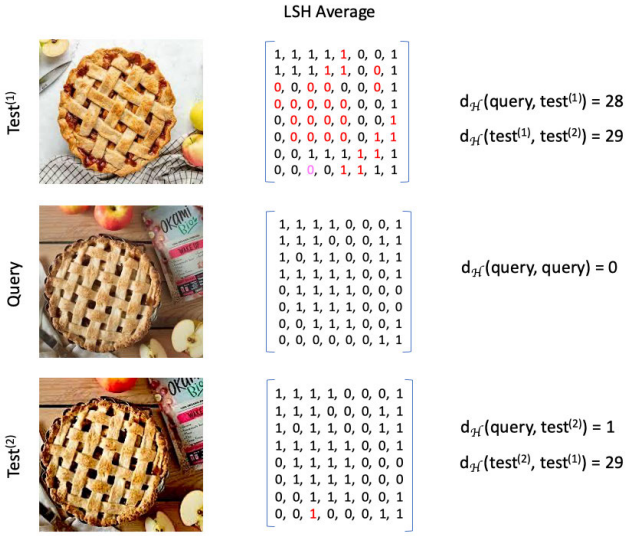


FIGURE 3. A Query image (middle) is compared with two Test images, where only one is a duplicate (bottom). Equation (1) applies.

inequality holds:

$$\begin{aligned} &d(u, v) \geq 0 \\ &d(u, v) = 0 \iff u = v \\ &d(u, v) = d(v, u) \\ &d(u, v) \leq d(u, w) + d(w, v) \end{aligned} \tag{1}$$

It can substitute the original metric in LSH algorithms to perform the same task and can be applied to the setting of binary vectors instead. All the properties of the original search mechanism are guaranteed to continue to exist. This is shown in FIGURE 3 where the feature vectors are extracted from three different images of the same subject (label). In particular, the query image in the middle is different from test$^{(1)}$, whereas test$^{(2)}$ is a manipulated version of the query image (contrast and tone enhancement, slight rotation, and cropping).

Finally, the deduplication module in the proposed pipeline employs three LSH methods [59] which are fast feature extractors that are reliable for detecting most affine transformations, namely, (a) the average image hash, (b) the perceptual hash, and (c) the difference hash. These methods produce binary and compact feature vectors, making them computationally important. The Hamming distance fits nicely as a metric for these descriptors because it measures the number of transforming steps a vector must take to become another. In other words, the steps required for a query to be transformed into a test vector. An exclusive BK-tree is constructed for each descriptor. Each time an image is encountered, its signature is extracted and compared with all candidate image hashes within the tree structures. A majority voting approach decides whether an image is duplicate or not; no tiebreaker is needed because there are three voters with equal voting weights. The query hash is stored in its respective BK-tree if and only if there is no duplicate decision outcome. Additionally, the query image is added to the candidate pool.

V. EXPERIMENTS AND DISCUSSION

This section details the assessment of the performance of the proposed approach in creating an AI-useful image dataset using Web samples. It also details whether sampling from many sources mitigates the dataset bias phenomenon. The domain of food recognition is chosen for the experimental procedure. There are already too many publicly available food datasets [60] for the computer vision community to experiment with and develop approaches for tasks such as dish and ingredient recognition and calorie calculation based on volume. Among these datasets, two versions of Food-101 are available: the originally published (ETHZ [30]) and its twin version (UPMC [61]), which, although share the same categories, they were collected from different sources. Hence, the choice of using these datasets as a case study presents noticeable practical advantages, such as the capacity for (a) validation of the pipeline's ability to construct formal datasets, (b) direct observation of any dataset bias between the published benchmarks, and (c) cross-evaluation of whether sampling from many different sources fixes the dataset bias.

A. IRRELEVANT SAMPLE REMOVER

A binary class dataset was constructed to train three classifiers (InceptionV3, MobileNet, and ResNet-50). Samples were taken from benchmark datasets to populate the relevant/non-relevant (food/non-food) classes. Specifically, for the food images, random samples were taken from FoodX-251 [62] and ISIA Food-200 [63]; thus, 100K images were collected. On the other hand, the non Food samples were taken from benchmark datasets with generic categories, such as PascalVOC, Caltech-101, and ImageNet. Similarly, 100K images were obtained. Special attention was given to the latter case; therefore, the categories and their samples depicted no food.

The training procedure followed the idea of using classifiers pretrained on ImageNet to extract features. A set of classification layers consisting of Global Average Pooling, a Dense Layer of 512 neurons, 20% Dropout, and the final 2-neurons Dense classifier was attached at the top of the model. This model was used to predict whether a sample was relevant. SGD was chosen as an optimizer to minimize the binary cross-entropy loss function. Empirically, it was found that after ten epochs, no significant performance improvement was achieved. Based on the losses, no overfitting was detected, as shown in FIGUER 4.

This approach exploits robust classifiers, which, based on the validation outcome on the test set, should agree on most of their individual decisions, anyway. However, in the case of disagreement, the voting mechanism provides the tiebreaker with a final decision. Based on the proposed approach regarding the irrelevant detection mechanism, discarding unwanted samples is both fast and accurate, as shown in TABLE 2.

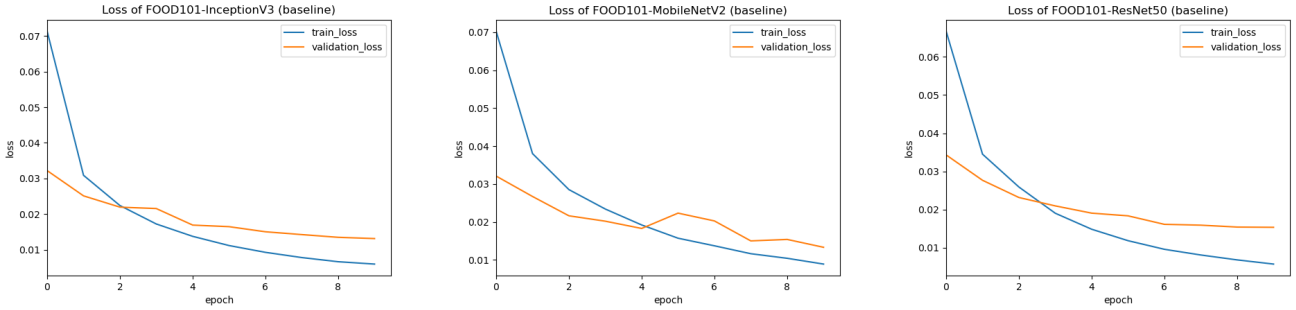


FIGURE 4. Binary cross-entropy for InceptionV3 (left), MobileNet (middle) and ResNet50 (right) while training on the custom food/nonFood dataset.

TABLE 2. Irrelevant detection performances.

InceptionV3	MobileNet	ResNet-50	Accuracy
Yes	No	No	0.9937
No	Yes	No	0.9966
No	No	Yes	0.9963
Yes	No	Yes	0.9941
No	Yes	Yes	0.9965
Yes	Yes	Yes	**0.9968**

B. DUPLICATE SAMPLE REMOVER

An experiment was conducted, similar to that proposed in [64], to determine the optimal threshold for image hashing algorithms. A random subset was used, which comprised 2000 unique images from the CIFAR-10 dataset. For each image, the following transformations were applied to produce 40 images: (a) contrast adjustment; (b) despeckling; (c) flipping; (d) the values of the R, G, and B channels were increased by 10% respectively; (e) cropping by 5%, 10%, 20%, and 30%, preserving the center region of the original image and then resizing to the original size; (f) downsampling the image by 10%, 20%, 30%, 40%, 50%, 70% and 90%; (g) format conversion from JPEG to GIF; (h) an outer frame of random color was added four times to the image, where the size of the frame is 10% of the image, respectively; (i) rotating by 90°, 180° and 270°; (j) scaling up by 2, 4, and 8 times, scaling down by 2, 4, and 8 times; (k) intensity adjustment by 70%, 80%, 90%, 110% and 120%; and (l) saturation adjustment by 70%, 80%, 90%, 110% and 120%. The transformations resulted in a total of 82000 images, of which the signatures were extracted respectively with each image hashing method. Distances and confusion matrices were calculated for all images. The desired threshold θ is the one that minimizes the following function:

$$\arg\min_{\theta} |FN - FP| \tag{2}$$

The point at which the false negative (FN) line intersects with the false positive (FP) one is the threshold θ, as FIGURE 5 illustrates. Another sample of 20000 images was taken from the same dataset, mutually exclusive to the training set, to test the thresholds. Subsequently, 500 were randomly selected and underwent the transformations mentioned above. This resulted in a test set of 40000 images. The achieved cross-validated F1 score for $\theta^{(\alpha)} = 3$ (average image hash), $\theta^{(d)} = 14$ (difference hash) and $\theta^{(p)} = 14$ was 89.8%, as reported in TABLE 3.

TABLE 3. Duplicate detection performance.

Avg	Diff	Perc	Recall	Precision	F1
Yes	No	No	0.807	0.93	0.861
No	Yes	No	0.801	0.973	0.879
No	No	Yes	0.821	0.982	0.894
Yes	Yes	No	0.811	0.982	0.888
Yes	No	Yes	0.828	0.899	0.862
No	Yes	Yes	0.838	0.958	0.894
Yes	Yes	Yes	0.843	0.961	**0.898**

C. CONSTRUCTING FOOD-101 FROM THE WEB

A query list was gathered from the 101 labels composing the Food-101 classes. These queries were given to custom-tailored crawlers to seek content using four search engines and two image repositories. The sample collection consisted of 606 crawling tasks performed in parallel. In total, 885,662 images were obtained. As expected, for popular foods, the samples were far more than those of less-known dishes, as shown in FIGURE 6, which displays the distribution of samples per class. For comparison with the benchmark dataset, the green dashed line at the one thousand mark signifies the FOOD-101 (ETHZ) counts of the samples per class.

Estimating whether an image is relevant to the scope of the dataset requires no memory of previous samples or knowledge of the image label. This procedure begins when the collection tasks report that they have completed their jobs. Thus, the samples were filtered based on their relevance to the dataset. 111.6K (12.6%) were discarded as irrelevant, while the rest were kept as valid.

In contrast with the less demanding relevance filtering, the deduplication mechanism needs to know the label a sample will be assigned to and all unique samples processed before it. A BK-tree structure per class label and descriptor served to store unique samples. Thus, this module extracts its LSH feature vectors for any incoming sample and searches for the

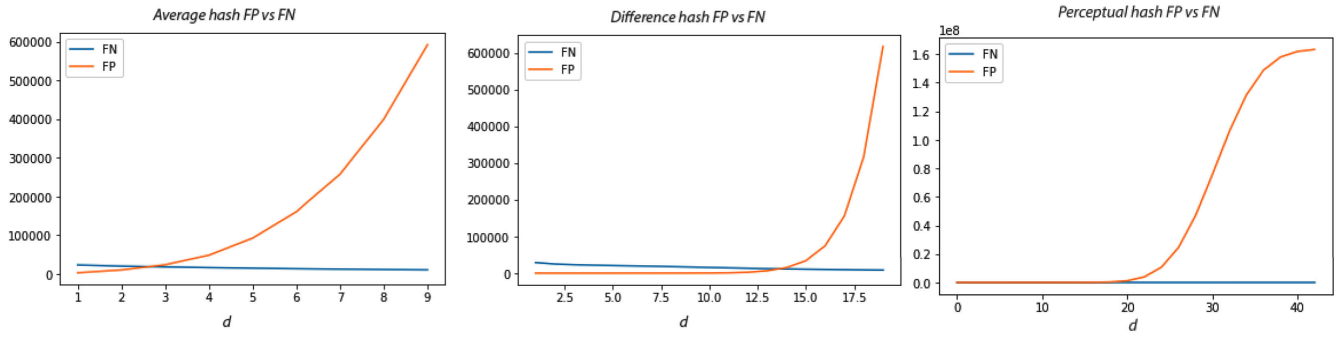


FIGURE 5. Finding the distance that minimizes the parameter θ.

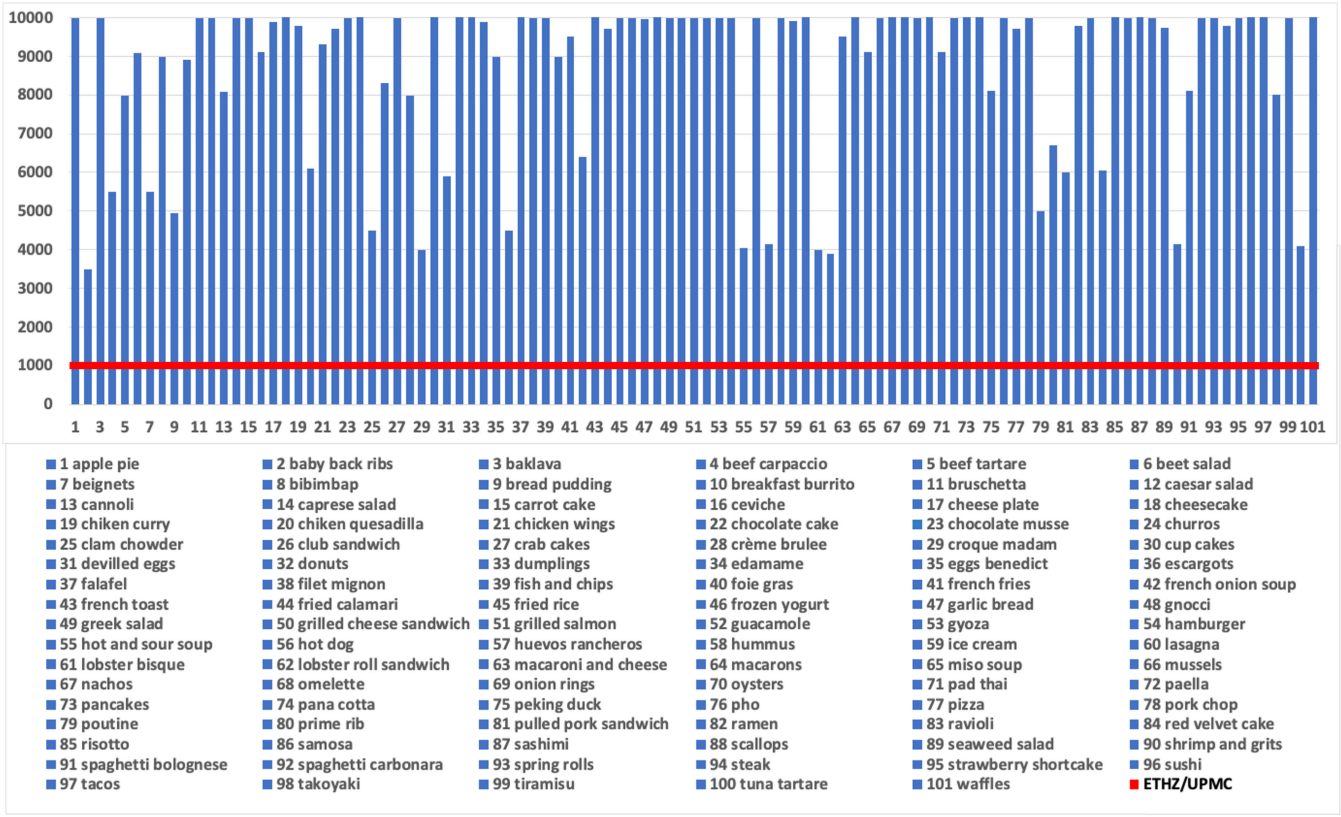


FIGURE 6. The distribution of samples per class. The red vertical line signifies the samples per class of FOOD-101 (ETHZ and UPMC).

respective tree for a match. If none was found, the sample was characterized as unique and stored within the structure; otherwise, it was discarded as a duplicate. Hence, 129.3K (14.6%) images were further rejected.

Finally, after removing the unwanted content, the dataset resulted in 644,800 images (5190.37($\pm$1459.88) samples per class). That is, an increase of **538.3%** over the ETHZ Food-101 dataset (which contains 101K images in total) or **6.3** more samples were collected for every benchmark image. FIGUER 7(right) shows the ratio between the valid, the irrelevant, and the duplicate samples.[1] For practical purposes, the discarded samples being irrelevant or duplicates were kept for further experimentation.

Implementation-wise, the modules were developed as individual software services (SaaS). Containerization and job orchestration are essential design details that contribute to the speed and parallelism of tasks. If multi-processing and multi-threaded practices were neglected, it would have resulted in the collection of samples in more than 1100 hours (approximately 45 days), removal of irrelevant samples in 370 hours (15 days), and rejection of duplicate content in 580 hours (24 days). This approach requires a total of 84 days of operational time. In contrast, the achieved data collection time was **88.25 hours**, a decrease of more than an

[1]The color coding corresponds with FIGURE 2.

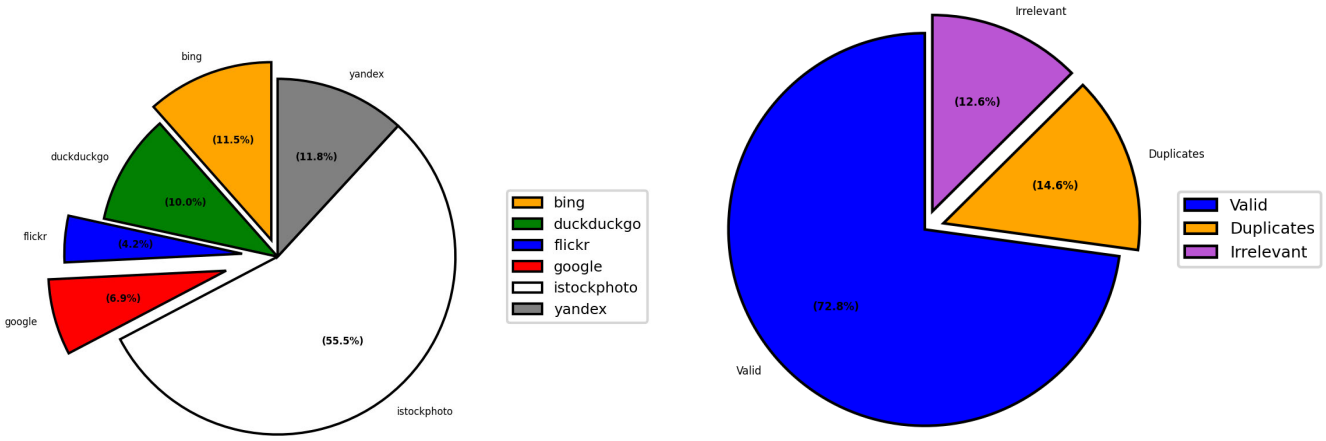


FIGURE 7. The relative contribution of search engines (left). The ratio between the rejected samples and those that were considered valid (right).

order of magnitude. In addition, the achieved irrelevant and duplicate sample rejection lasted **8.5 hours** and **22.75 hours** respectively, a decrease of more than an order of magnitude in both cases, for the exact same infrastructure (computing and networking).[2] The total amount of time needed from start to finish in order to construct the dataset used in the following experiments was **103.43 hours** or slightly more than 4 days, which is a 95.23% decrease in time with respect to a typical serial design (no sophisticated parallel processing whatsoever).

D. ABLATION STUDY

An additional ablation study was performed to establish the requirement for the proposed rejection (filtering) modules. For comparison with an earlier ablation study, which examined the need for filtering out duplicate and irrelevant content, the same deep learning architecture was used before, namely InceptionV3.

The samples acquired from the Web are considered an assortment of data and cannot be applied in typical classification experiments with fairness. A two-step mutual exclusion rule was applied to construct fair training and testing sets as follows:

1) a test set was sampled without including duplicates
2) all samples within the test set were examined for duplicates on the rest of the data. Any existing duplicates that were found during this procedure were removed

Hence, no shared examples exist within the training and test sets for the rest of the experiments.

The four cases this ablation study considers are with regard to the training set, as follows:

- using all the unfiltered data
- using irrelevant-filtered data
- using duplicate-filtered data
- using irrelevant- and duplicate-filtered data

[2]The time reported in the case of irrelevant and duplicate samples rejection is the cumulative operational time since the modules ran asynchronously.

TABLE 4. Evaluating InceptionV3 performance with and without the use of the rejection modules. "Yes" indicates the use of the module; "No" otherwise.

Duplicates	Irrelevant	Accuracy
No	No	0.682
No	Yes	0.691
Yes	No	0.707
Yes	Yes	**0.725**

TABLE 4 presents the performance of InceptionV3 for all dataset instances. These results coincided with the declining trend observed and presented earlier in the *in vitro* study. Having no irrelevant but many duplicate samples granted no performance gains, besides increasing the demand for training resources such as memory and time. However, having irrelevant data decreased the model's performance, as expected, because useful information relevant to the learning problem could not be extracted. The worst and best cases are the two opposites regarding the use of no filtering and rejecting noisy data altogether, respectively.

E. WIDER SAMPLING FIXES DATASET BIAS

Using the labels of the Food-101 dataset to construct another one sampled from a multitude of Web resources presented the practical advantages of (a) having two benchmark dataset versions of the same classes, allowing the study of the dataset bias phenomenon first hand and (b) the effect of more exhaustive sampling on the bias phenomenon.

This section describes a cross-evaluation experiment conducted to assess this discrepancy. For consistency, InceptionV3 was used once more as a classifier. The top layer was substituted as in the earlier experiments to fit the 101-class problem. The architecture was trained for 25 epochs, at which point the performance improvement plateaued. In addition, this experiment explicitly used no transfer-learning techniques at all. Hence, the deep learning architecture was trained for all its layers from scratch.

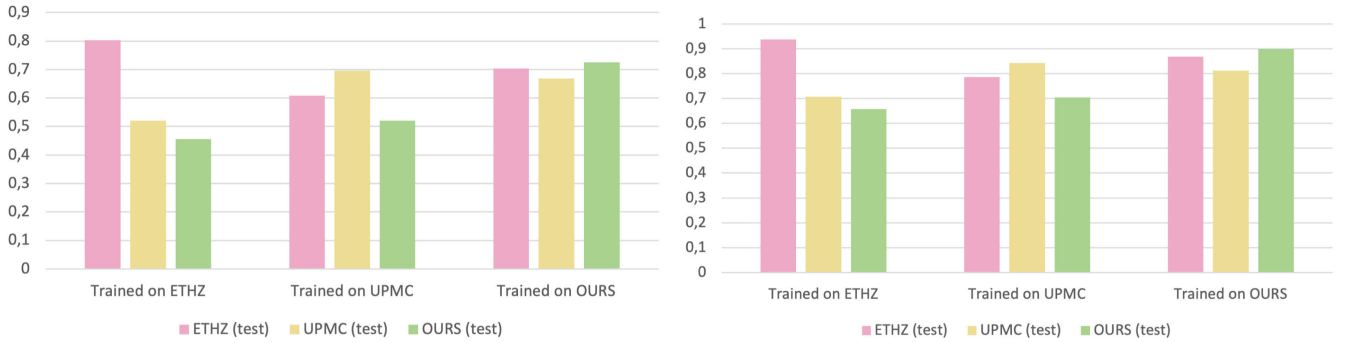


FIGURE 8. (left) Top-1 accuracy cross-evaluation on Food-101 and its variations; (right) Top-5 accuracy cross-evaluation on Food-101 and its variations.

The data split for training and testing was based on the initial publication instructions for both the datasets.

Training on ETHZ and testing on itself performed significantly better than when testing on UPMC. This is not the case when the opposite scenario is addressed. The classifier trained on UPMC and tested on ETHZ performed comparably well, as if it was tested with itself, albeit less accurately. The mismatch in the first case probably occurred because of the selection of samples being biased on ETHZ's behalf of ETHZ. Perhaps the architecture was able to learn representations that easily modeled the sample inclusion rules.

On the other hand, the makers of the UPMC version claimed that they queried a single search engine; thus, the included images contained samples of different origins and wrongly assigned labels, resulting in a comparable yet noisy dataset. In the latter case, looser rules regarding a sample's inclusion resulted in a classifier that learned more general representations. However, the drop in performance regarding the UPMC/ETHZ test suggests that neither version adequately captures the real-world domain shaped by its classes.

Moving forward, the dataset constructed using the proposed approach is introduced into the experimental setup. A 75-25% stratified split was performed on the dataset[3] The same classifier (as in the architecture, hyperparameters, and training procedure) is trained on the newly constructed version and tested on all other versions consecutively. Additionally, the previous models were tested on this version too. The cross-evaluation of TOP-1 and TOP-5 accuracy performances are reported in TABLE 6 and TABLE 5 respectively.

The ETHZ version seems to be the one that suffers the most from dataset bias because testing on all other versions could not match that performance. In particular, testing the new version produced an outcome that could not be considered beneficial: a classification score of less than 50%. As mentioned earlier, training on UPMC and testing on ETHZ resulted in comparable results to testing on itself. However, testing the version produced by the proposed approach resulted in an accuracy of just above 50 % (top-1). The constructed dataset appears to be a useless assortment of data at this point. However, a top-1 of 72.5% accuracy performance when training on the new version and testing on itself suggests otherwise. Moreover, testing the latter on the twin benchmark versions produced meaningful and equivalent to itself results. Note how small the variance between performances is in FIGUER8 (left). The standard deviations in TABLE 6 and TABLE 5 confirm this observation. Similar issues can be noticed when examining the top-5 performances e.g., TABLE 5, FIGUER8 (right). The ETHZ version has been secluded into itself. Meaningful results appear when using the UPMC version, yet the variance of performance is large among versions. Stable performance appears when using the newly constructed dataset.

[3]Common samples with ETHZ and UPMC were also removed: 657 and 4271, respectively. This is proportional to 0.65% and 4,7% of these datasets.

TABLE 5. Cross-evaluation of Top-1 accuracy on Food-101 and its variations.

		Tested			
	TOP-1	ETHZ	UPMC	OURS	±
Trained	ETHZ	0.80324	0.52082	0.45622	0.184
	UPMC	0.6083	0.6958	0.5199	0.087
	OURS	0.70372	0.66851	0.72548	**0.028**

TABLE 6. Cross-evaluation of Top-5 accuracy on Food-101 and its variations.

		Tested			
	TOP-5	ETHZ	UPMC	OURS	±
Trained	ETHZ	0.9367	0.7077	0.658	0.148
	UPMC	0.7855	0.8427	0.7039	0.069
	OURS	0.8684	0.8125	0.8986	**0.043**

VI. CONCLUSION

In this work, an important phenomenon is studied that many benchmark datasets suffer from, the dataset bias, a phenomenon by which a collection of data is not descriptive of the whole domain it represents—learning to transfer knowledge from one domain to a different one deals with a similar but not the same problem. In any case, a dataset

is intended to reflect a real-world domain unless it was designed explicitly not to. Another major challenge is to create large datasets in an automated manner and to guarantee their usability.

To this end, this study proposes a method to automatically construct non-biased datasets by sampling many different sources to handle these issues. The proposed approach uses a list of keywords to query Web-based resources; it then collects samples and discards all those that are duplicates or irrelevant to the scope of the dataset. This study provides evidence that such noisy data degrade the performance of classification tasks.

The advantages of the proposed method for constructing a dataset are as follows: (a) it can build a dataset quickly, (b) it scales well to many more classes, and (c) it allows for looser sample inclusion criteria to occur by imposing a hard rule on excluding duplicate and irrelevant content.

Therefore, the proposed method can be used, for instance, if a classifier has reached its potential given a dataset and no model alteration techniques or virtual augmentations increase its performance. Broader sampling can be used to learn more general representations or a portion of it can be used to augment the dataset with unseen data.

The main weaknesses of the proposed method are: (a) it requires a robust classifier to decide whether a sample is relevant to the scope of the dataset, and, as a consequence, (b) in non-thematically uniform datasets, constructing the non-relevant class to train the relevant/irrelevant classifier can be cumbersome. Thus, the proposed method scales well for thematic datasets such as the food-related ones (that we tested). A different approach is required in other cases where a dataset might contain classes of broad interest.

The ETHZ and UPMC Food-101 (the twin benchmark datasets) were used in the empirical study. They presented a practical advantage: they shared the same class labels, although their data came from different sampling protocols. Thus, they were used as a cross-evaluation testbed. Throughout the experiments, InceptionV3 was used as a classifier mainly for consistency. In most cases, it was trained from scratch using the same procedure.

Three deep learning models are compared with each other. All were trained on Food-101, but one for each version, the ETHZ, the UPMC, and OURS, respectively. The model trained on ETHZ performs remarkably, but only to itself. The model trained on UPMC learned more general representations than the previous model, yet its performance was subpar on the other two datasets. In contrast, based on the procedure proposed in this study, the model trained on the newly constructed dataset performed comparably well on all three datasets.

In conclusion, this work presented a pipeline consisting of a combination of filtering modules that, in an automated manner, can construct a dataset of images with real-world samples acquired from the Web. The experiments presented in this work show that more exhaustive sampling, as in sampling from many different sources, alleviates the dataset bias. We are upgrading this method with sample selectivity and query expansion techniques to balance the number of samples per class without sacrificing the generalization properties of broader sampling.

REFERENCES

[1] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*, vol. 4. New York, NY, USA: AMLBook, 2012.
[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. CVPR*, Jun. 2011, pp. 1521–1528.
[3] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 129–136.
[4] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 178.
[5] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 158–171.
[6] D. J. Hand, "Classifier technology and the illusion of progress," *Stat. Sci.*, vol. 21, no. 1, pp. 1–14, Feb. 2006.
[7] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 343–347.
[8] R. Rothe, R. Timofte, and L. V. Gool, "DEX: Deep EXpectation of apparent age from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 10–15.
[9] T. Weyand and B. Leibe, "Visual landmark recognition from internet photo collections: A large-scale evaluation," *Comput. Vis. Image Understand.*, vol. 135, pp. 1–15, Jun. 2015.
[10] A. Krizhevsky, "Learning multiple layers of features from tiny images," MIT, NYU, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[11] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008.
[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and W. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.
[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[14] T. Tommasi, N. Quadrianto, B. Caputo, and C. H. Lampert, "Beyond dataset bias: Multi-task unaligned shared knowledge transfer," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 1–15.
[15] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.
[16] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[17] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 213–226.
[18] T. Tommasi and B. Caputo, "Frustratingly easy NBNN domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 897–904.
[19] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 631–645.
[20] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.
[21] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Domain Adaptation in Computer Vision Applications*. Berlin, Germany: Springer, 2017, pp. 37–55.
[22] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 289–296.
[23] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," *Inf. Fusion*, vol. 24, pp. 84–92, Jul. 2015.

[24] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*. Berlin, Germany: Springer, 2021, pp. 877–894.
[25] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 722–729.
[26] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 289–292.
[27] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 285–288.
[28] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2010, pp. 296–301.
[29] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 3–17.
[30] L. Bossard, M. Guillaumin, and L. V. Gool, "Food-101–mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 446–461.
[31] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for Google images," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2004, pp. 242–256.
[32] B. Siddiquie and A. Gupta, "Beyond active noun tagging: Modeling contextual interactions for multi-class active learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2979–2986.
[33] T. L. Berg and D. A. Forsyth, "Animals on the web," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 1463–1470.
[34] L.-J. Li and L. Fei-Fei, "OPTIMOL: Automatic online picture collection via incremental model learning," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 147–168, 2010.
[35] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2012, pp. 25–30.
[36] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 2017–2020.
[37] Y. Yao, J. Zhang, F. Shen, X. Hua, J. Xu, and Z. Tang, "Automatic image dataset construction with multiple textual metadata," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2016, pp. 1–6.
[38] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[39] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4444–4451.
[40] Y. Lin, J.-B. Michel, E. A. Lieberman, J. Orwant, W. Brockman, and S. Petrov, "Syntactic annotations for the Google books Ngram corpus," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 169–174.
[41] Y. Yao, J. Zhang, F. Shen, X.-S. Hua, J. Xu, and Z. Tang, "Exploiting web images for dataset construction: A domain robust approach," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1771–1784, Aug. 2017.
[42] Y. Yao, J. Zhang, F. Shen, L. Liu, F. Zhu, D. Zhang, and H. T. Shen, "Towards automatic construction of diverse, high-quality image datasets," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1199–1211, Jun. 2020.
[43] Y. Yao, X.-S. Hua, F. Shen, J. Zhang, and Z. Tang, "A domain robust approach for image dataset construction," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 212–216.
[44] M. Doerr, S. Gradmann, S. Hennicke, A. Isaac, C. Meghini, and A. H. V. D. Sompel, "The Europeana data model (EDM)," in *Proc. World Library Inf. Congr., 76th IFLA Gen. Conf. Assembly*, vol. 10, 2010, p. 15.
[45] G. Pavlidis and V. Sevetlidis, "Demystifying publishing to Europeana: A practical workflow for content providers," *Sci. Culture*, vol. 1, no. 1, pp. 1–8, 2015.
[46] S. Markantonatou, K. Toraki, P. Minos, A. Vacalopoulou, V. Stamou, and G. Pavlidis, "Amalthea: A dish-driven ontology in the food domain," *Data*, vol. 6, no. 4, p. 41, Apr. 2021.
[47] W. Ride, *International Code of Zoological Nomenclature*. Singapore: International Commision of Zoological Nomenclature, 1999.
[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[49] C. Kiourt, G. Pavlidis, and S. Markantonatou, "Deep learning approaches in food recognition," in *Machine Learning Paradigms*. Berlin, Germany: Springer, 2020, pp. 83–108.
[50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[52] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucl. Acids Res.*, vol. 31, no. 13, pp. 3812–3814, 2003.
[53] T. Sikora, "The MPEG-7 visual standard for content description-an overview," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 696–702, Jan. 2001.
[54] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.
[55] I. Kansizoglou, N. Santavas, L. Bampis, and A. Gasteratos, "HASeparator: Hyperplane-assisted softmax," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 519–526.
[56] I. Kansizoglou, L. Bampis, and A. Gasteratos, "Deep feature space: A geometrical perspective," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6823–6838, Oct. 2022.
[57] W. A. Burkhard and R. M. Keller, "Some approaches to best-match file searching," *Commun. ACM*, vol. 16, no. 4, pp. 230–236, 1973.
[58] R. Baeza-Yates and G. Navarro, "Fast approximate string matching in a dictionary," in *Proc. String Process. Inf. Retrieval, South Amer. Symp.*, 1998, pp. 14–22.
[59] L. Chi and X. Zhu, "Hashing techniques: A survey and taxonomy," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 1–36, 2017.
[60] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, "ISIA food-500: A dataset for large-scale food recognition via stacked global-local attention network," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 393–401.
[61] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMEW)*, Jun. 2015, pp. 1–6.
[62] P. Kaur, K. Sikka, W. Wang, S. Belongie, and A. Divakaran, "FoodX-251: A dataset for fine-grained food classification," 2019, *arXiv:1907.06167*.
[63] W. Min, L. Liu, Z. Luo, and S. Jiang, "Ingredient-guided cascaded multi-attention network for food recognition," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1331–1339.
[64] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Multimedia*, vol. 4. NY, USA: Citeseer, 2004, p. 5.

VASILEIOS SEVETLIDIS (Student Member, IEEE) received the undergraduate and postgraduate degrees (summa cum laude) and the M.Sc. degree in applied archaeological sciences. He is currently pursuing the Ph.D. degree with the Department of Production and Management Engineers, Democritus University of Thrace, under the supervision of Prof. Antonios Gasteratos. He has an academic progression characterized by diversity and distinction. He started as an undergraduate in informatics engineering, where he was taught, among other topics, the concepts of information theory, algorithmics, image analysis, and software engineering. He started a collaboration with Dr. George Pavlidis at the Athena Research Center as a Research Associate. During this time, he got exposed to emerging technologies applied to Cultural Heritage.

GEORGE PAVLIDIS (Senior Member, IEEE) received the Diploma and Ph.D. degrees in electrical and computer engineering. His doctoral research focused on digital image processing, particularly the optimal segmentation and compression of mixed document images, for which he received the Ericsson Award of Excellence in Telecommunications. In 2002, he joined with the Athena Research Center, where he is the Research Director (Researcher A') with the Institute for Language and Speech Processing. He is the Head of the Media Department and the Head of research with the Clepsydra-Cultural Heritage Digitization Center. His research interests include digital image and multimedia technologies, content analysis and retrieval, machine learning and artificial intelligence, human–machine interaction, intelligent interactive environments, multi-sensory environments and ubiquitous and ambient intelligence, 3-D digitization, and cross-reality applications. He is a member of Scientific, Technical, and Management committees. He is a member of the Technical Chamber of Greece and CAA and a Board Member of CAA-Gr. He is the Editor-in-Chief of the *International Journal on Computational Methods in Heritage Science*.

SPYRIDON MOUROUTSOS received the Diploma degree in electrical engineering and the Ph.D. degree in systems automation from the Democritus University of Thrace, Greece, in 1981 and 1986, respectively. In 1986, he joined as an Assistant Professor at the Electrical and Computer Engineering Department, Democritus University of Thrace, Greece. Currently, he is a Professor of mechatronics, systems automation, and standards. He has acted as an Evaluator for National and EU Research Grant Applications. His research interests include applications in mechatronics, systems automation and robotics, intelligent and autonomous robots (humanoids, animated, underwater, flying, etc.), data fusion–sensors with applications in robotics and automation, computer architectures–microprocessors and their applications, and standards and certification. He is a referee, a committee member, or a member of the Editorial Board for numerous international scientific and technical journals and conferences.

ANTONIOS GASTERATOS (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees from the Department of Electrical and Computer Engineering, Democritus University of Thrace (DUTh), Greece. He is the Director of the Laboratory of Robotics and Automation, DUTh. His research interest includes robotics. He has served as a reviewer for numerous scientific journals and international conferences. He is a Subject Editor with *Electronics Letters*.