

USC-DCT: A Collection of Diverse Classification Tasks

Adam M. Jones ^{1,†}, Gozde Sahin ^{2,†}, Zachary W. Murdock ^{1,†}, Yunhao Ge ², Ao Xu ², Yuecheng Li ², Di Wu ², Shuo Ni ², Po-Hsuan Huang ¹, Kiran Lekkala ² and Laurent Itti ^{1,2,*}

¹ Neuroscience Graduate Program, University of Southern California, Los Angeles, CA 90007, USA

² Department of Computer Science, University of Southern California, Los Angeles, CA 90007, USA

* Correspondence: itti@usc.edu

† These authors contributed equally to this work.

Abstract: Machine learning is a crucial tool for both academic and real-world applications. Classification problems are often used as the preferred showcase in this space, which has led to a wide variety of datasets being collected and utilized for a myriad of applications. Unfortunately, there is very little standardization in how these datasets are collected, processed, and disseminated. As new learning paradigms like lifelong or meta-learning become more popular, the demand for merging tasks for at-scale evaluation of algorithms has also increased. This paper provides a methodology for processing and cleaning datasets that can be applied to existing or new classification tasks as well as implements these practices in a collection of diverse classification tasks called USC-DCT. Constructed using 107 classification tasks collected from the internet, this collection provides a transparent and standardized pipeline that can be useful for many different applications and frameworks. While there are currently 107 tasks, USC-DCT is designed to enable future growth. Additional discussion provides explanations of applications in machine learning paradigms such as transfer, lifelong, or meta-learning, how revisions to the collection will be handled, and further tips for curating and using classification tasks at this scale.

Dataset: <https://github.com/iLab-USC/USC-DCT>

Dataset License: CC-BY-NC

Keywords: machine learning; data sharing; classification; computer vision; visual classification; dataset collection; dataset organization; data cleaning



Citation: Jones, A.M.; Sahin, G.; Murdock, Z.W.; Ge, Y.; Xu, A.; Li, Y.; Wu, D.; Ni, S.; Huang, P.-H.; Lekkala, K.; et al. USC-DCT: A Collection of Diverse Classification Tasks. *Data* **2023**, *8*, 153. <https://doi.org/10.3390/data8100153>

Academic Editor: Jamal Jokar Arsanjani

Received: 15 July 2023

Revised: 6 September 2023

Accepted: 21 September 2023

Published: 12 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The last decade has seen machine learning techniques flourish, with many targeting the problem of classification [1–3], using the ever-increasing depth of learning models [4–7] to meet this challenge. During this time, many datasets have been developed to form benchmarks in the field of computer vision for the training and evaluation of classification models [1–3]. ImageNet [1] has become the most common choice for both evaluating the performance of and pre-training of deep neural networks for visual recognition; however, as the accessibility of deep learning has grown, so too has the variety of classification datasets collected for both academic and real-world applications. Furthermore, as interest in paradigms such as lifelong learning and meta-learning continues to build, which explore multiple tasks and their corresponding datasets, collections of these datasets have begun to be utilized in an effort to evaluate model performance across varying classification scenarios and to encourage stronger algorithmic generalizability [8].

As these collections become more comprehensive, a major issue has become glaringly apparent—a near-total lack of standardization in dataset construction. Methods of collection and distribution can vary wildly between datasets and strategies taken are often not well documented. These issues result in a multitude of barriers preventing the use of resources

available due to the possibility of errors and overlap. Additionally, the mechanisms for using different datasets provide another point of contention. Without a standard convention for the general organization and labeling of datasets, most need their own unique procedures or require a complete overhaul in the organization. And as the practice of creating sets of datasets builds, the effort required of an end-user to encompass a wide range of tasks results in a lower likelihood of use and, thus, reduces the rate of advancement. While it is too late to enforce every past unique dataset fitting within certain criteria, the distribution of current and future classification tasks can be unified into a single pipeline for ease of future accessibility. This paper describes how to create a large-scale collection of diverse classification tasks (USC-DCT) and presents it as a fully standardized benchmark that can easily be acquired and used for many different paradigms in visual classification. Along with the full benchmark, a data processing pipeline is presented that is designed to help adapt any classification task into the standard format used in this benchmark. Therefore, the benchmark can grow easily as new classification datasets are collected by the machine learning community. Not only can it be used in a piecemeal fashion for applications like lifelong learning [9], but it could also be used in its entirety to provide a diversified pre-training backbone. Outside of the specific sets used, USC-DCT can also serve as a framework-independent dataset repository for those interested in working on classification problems across domains.

The initial release of USC-DCT (v1.0) contains 107 diverse classification tasks obtained from 94 datasets. Each dataset in the collection has been processed by the proposed pipeline to standardize the utilization of these diverse tasks, which includes a clean-up procedure to remove duplicates within and across datasets. Additionally, extensive statistics on dataset diversity, common issues found when processing a large number of datasets (e.g., invalid images, non-dataset images, etc.), and the exact and near duplicates identified are presented. In addition to proposing the benchmark and a transparent pipeline to standardize distribution of a large-scale collection like USC-DCT, we also present discussions on how USC-DCT can be utilized by the machine learning community, suggestions on future additions to USC-DCT and how we foresee its growth, as well as our recommendations on the best ways to create and publish a dataset.

2. Related Works

Data collection is one of the more challenging aspects of learning for deep models. The cost of collecting and annotating large amounts of data can be very high, and various approaches have been used over time to achieve this. Some have been organized from data collected from subjects directly [3,10], while others have utilized more indirect collection methods. For example, ImageNet queries and retrieves images from search engines for data collection [1]. Since search engine results are extremely noisy, clean-up is often performed while annotating using services like Amazon Mechanical Turk (AMT). Even with the human effort to retrieve a mostly clean dataset, a variety of issues within datasets have been identified [11,12], which also impact model robustness [13]. The CIFAR-10 and CIFAR-100 datasets were also collected by querying search engines and underwent removal of exact duplicates [2]. More recently, there have been efforts to remove images from the CIFAR datasets using nearest-neighbors and further manual annotation [14].

When data collection strategies in visual classification tasks are surveyed [1,2], it becomes apparent that no globally accepted methodology has been employed for structuring, collecting, and cleaning vision datasets. Since collection methods can vary greatly, this is not surprising. Standardization can make it possible to train stronger models, as well as aid further research in paradigms like transfer learning [15], continual learning [16], meta-learning [17], and task adaptation [18], where collections of datasets can be crucial for the generalization of algorithms. Some collections that have previously been proposed include the 8-dataset sequence [8] for continual learning, fine-grained 6-task collection for task adaptation [19], the Visual Domain Decathlon [20], and the Visual Task Adaptation Benchmark [21]. While [20] re-distributes tasks and their annotations, most of the collec-

tions like these require users to set up each dataset separately to evaluate their algorithms on them. This can be feasible for collections such as these (up to 20 datasets). However, for large-scale collections, this can become unfeasible due to the non-standardized ways in which classification datasets are distributed. Examples include SVHN [22] images being encoded in .npy files instead of the common image format or in a text file, like some distributions of MNIST [3]. This overhead can easily be eliminated if datasets are distributed in a standardized manner in the future, and paradigms like continual learning and task adaptation can be easily evaluated at scale. Some standardization in the distribution has been achieved by datasets archived using common machine learning frameworks like PyTorch [23], TensorFlow [24], the Hugging Face dataset hub [25], and more. These dataset repositories are a great resource for the standardized distribution of datasets. However, they are not consistent about which datasets appear in the repositories, nor do they address issues with duplicates (i.e., removing duplicates that exist in the original datasets), and therefore, forming a collection of datasets for a task like continual learning can become framework dependent.

Regardless of how a dataset is initially processed and distributed, it might still contain some errors that can have some effect during model training. These include noisy labels, ambiguity, or other similar issues. To study the impact of cleaning datasets, many data-driven approaches have been proposed. Some studies have used outlier detection to achieve this task [26]. CleanML evaluates the impact of data cleaning in classification tasks using 14 real-world non-vision datasets, considering how the database community approaches the problem [27]. ActiveClean proposes an interactive data cleaning framework that uses a user-defined cleaning operator—which can be expensive depending on the dataset [28]. While there is the cleaning of exact duplicates in this collection, this effort is orthogonal to the studies to further clean datasets in a more data-driven way, as well as to algorithms that are robust to noisy labels [29].

3. Methodology

The building of USC-DCT can be divided into three stages. The data aggregation stage was performed by the authors to find the initial large and diverse collection of tasks, and can be performed by others in the service of future expansions and revisions to the collection. The pre-processing and validation stage is the intra-dataset procedure that records the image information for each dataset and contains both manual and automated steps. Finally, the post-processing stage is a combination of duplication removal, set generation, and building the final database. This section expands on this methodology and gives details for each component of the pipeline, as visualized in Figure 1.

Although choosing the datasets to be included in the collection was and will be a purely subjective and manual process, the rest of the USC-DCT pipeline is meant to be transparent and reproducible. For each dataset, a script was (and will be) created to handle the downloading, extraction, conversion, and parsing of the dataset. This script handles the peculiarities of each dataset in order to fit every dataset into a common framework. There are several reasons for performing this process, the most important of which are, first, it allows all down-stream users to see exactly what decisions were made on the raw dataset (instead of just trusting that individuals performed this step correctly), and second, it allows anyone to inspect the scripts and correct any mistakes they find.

Hence, DCT is (1) a collection of scripts that can be applied to raw dataset archives, and (2) a resulting database that allows end-users to obtain any image from any dataset in the collection through a unified interface.

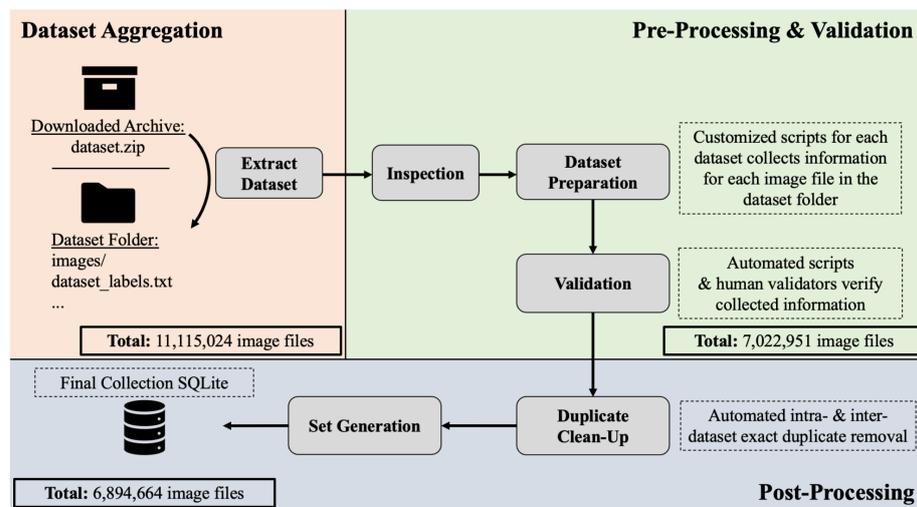


Figure 1. Overview of the dataset collection and preparation pipeline applied for USC-DCT.

3.1. Dataset Aggregation

The first stage of building USC-DCT involved collecting diverse datasets, then downloading their source files and extracting the files for the images and labels.

3.1.1. Collection

Since one of the driving factors in creating a collection such as USC-DCT is to support research domains like lifelong and meta-learning, the acquisition of computer vision datasets that span a wide variety of different subject matters is the key concern during the dataset-gathering phase. An increased variance of tasks in the collection promotes research for more generalizable or adaptable algorithms that could be used for any problem in this domain. Following this principle, we chose to omit common pre-training or evaluation datasets (such as ImageNet [1] or CIFAR-10/100 [2]) to focus on increasing the accessibility to other diverse data. Further limiting the scope, we only consider classification or classification-adjacent datasets during this process.

As mentioned, one of the bigger challenges in forming a large-scale collection such as this is the varying practices of distribution and the overhead it creates to set up a collection of this size. To assess what is lacking in standardization practices, previously curated collections—such as those found in [8] for use in the lifelong learning domain, as well as the “Visual Decathlon Challenge” [20]—were thoroughly examined. While both groupings provide methods to use all of the datasets included, they do not propose a methodology for unifying them, outside of some overlap with what is presented by [30]. Moreover, there is a notable overlap in the subject matter covered between these two collections. Therefore, a collection of individual datasets can better encompass the larger diversity of classification tasks currently available, especially if their distribution is streamlined.

For USC-DCT, following these insights, the primary metric used for determining diversity between datasets was the subject matter, followed by pixel resolution and overall dataset size. The first aimed to ensure that collections that used natural images, medical imaging, artificial visuals, etc., would become more accessible. The latter two factors are used to ensure that any sort of standardization methods developed could be used for any newly developed datasets in the future, regardless of their size.

With the scope and limitations established, the discovery and collection of these various datasets were conducted using dataset repositories such as PapersWithCode.com, (<https://paperswithcode.com/datasets>) and others mentioned in Section 2, making sure they were commonly used or cited in current research over the past few years. Basic assessments such as the ease of interpretation, data formatting and organization, and labeling systems were completed on the first 32 datasets. Others consisted of datasets

published on the open-source Kaggle platform (<https://www.kaggle.com/datasets>), an important resource for datasets that span a diverse number of classification tasks. Adding Kaggle as a source greatly increased the diversity in this large-scale collection, at the cost of added complexity to the standardization process in the form of some additional implemented limitations. Most notably, selected datasets could not have fewer than 128 total images, images could not be smaller than 16×16 pixels, and the dataset could not be targeting a visual task that could not be reduced to a binary or multi-class classification task. (This process is visualized in Supplementary Materials S1.) The datasets also required manual investigation to ensure a minimum image quality and label correctness. Using these two primary resources, the final raw collection comes to a total of just under 100 datasets.

Further taking inspiration from the use of datasets like [31–33], they are split into sub-datasets based on assigned super-classes to create increasingly diverse tasks. This also establishes a standard for creating subdivisions in the future. In the end, 94 datasets were collected and then further split into 107 tasks, with a total of 11,115,024 images among 13,272 classes before any processing.

During the dataset collection effort, a brief inspection saw that many of the datasets maintained some internal consistency, but that there was weak translation to other sets. The myriad of issues ranged from differences in image types to broken files to varied organizational structures. How these and other issues were addressed is further detailed throughout the subsequent sections of the proposed dataset processing pipeline.

3.1.2. Download and Extraction

Every dataset identified during the collection is downloaded automatically by the pipeline. This process is customized for each dataset for two reasons. (1) Every dataset is hosted differently (e.g., Kaggle, Google Drive, author websites, etc.), which changes how they should be downloaded. (2) The number of archives/files to be downloaded for each dataset is not standardized. Initial inspection found that some datasets came in a single archive, while others were distributed in multiple archives. Moreover, annotations are not always distributed in the archives and sometimes need to be downloaded separately. For each of the necessary files downloaded, the md5sum values are stored as constants within the pipeline scripts and verified for every fresh install of USC-DCT. This is to ensure the reproducibility of the efforts taken to put USC-DCT together. Due to the many different ways these datasets are distributed, archive extraction must be customized for each dataset as well. The archive formats encountered were .zip, .tar, and other multi-archive distributions. Extraction produces a dataset folder that contains images and other annotation files for each dataset in the collection.

3.2. Pre-Processing and Validation

The next stage involved inspecting the files that were extracted for each dataset, preparing the script that will allow the pipeline to ingest those files and explain to others what the files are, and then validating all of the images that were to be included in USC-DCT.

3.2.1. Inspection

Inspection is a common but overlooked step every user needs to perform to use a published dataset in their work. It consists of inspecting the dataset folder and the distributed files to understand the actions necessary to include the dataset in a classification pipeline. Are the data distributed as image files or in another format, like pixel values in a .csv format? Are the labels given in a file, inferred from the folder structure, or part of the naming convention for the images? Are there any sets given (e.g., training, validation, test)? Are there any confusing folders/files with no clear documentation? An inspection helps to answer questions such as these and helps to identify how to prepare the dataset for a machine learning pipeline.

A handful of flawed practices were found numerous times while reviewing the 94 datasets: these occurrences were documented in the form of status codes to ensure that they were addressed during data preparation. The codes that are assigned are a mixture of objective designations (e.g., too small, invalid file, etc.) and subjective designations (e.g., near-duplicate folders). A list of these status codes is given in Table 1, with the shorthand used for each of them in the rest of this paper, along with their detailed explanations. An important note is that not all of these status codes represent outright mistakes or errors made during the distribution of the dataset; instead, they were developed to help document which images in the downloaded archives should and should not be used, as well as why any images were excluded (i.e., assigned to a non-zero status code). Finally, these codes were developed in parallel with the processing of all of the datasets and, therefore, were added to organically.

Table 1. Explanation of status codes that every image within the USC-DCT collection was assigned during the second stage of the pipeline. A shorthand for each code is presented as well as longer explanations.

Status Code	Shorthand	Explanation
0	problem-free	Image has no identified problem.
1	invalid file	The image does not open, is corrupted, or does not load. This is tested by attempting to load every image using the Pillow package.
2	non-dataset image	These are images found in the dataset folder that are completely unlike images in the dataset. Examples can be title images, sample image collages, logos, etc.
3	duplicate folders	These are images exactly duplicated in the downloaded dataset folder and can occur if a class folder or the full dataset itself has been duplicated in the distributed archive by mistake. To prevent removing valid images, during validation, it is checked if every image with this status also exists elsewhere, using a matching file hash.
4	no label	These are dataset images with no provided labels, e.g., images that were used for demonstration or test sets with labels intentionally left out.
5	near-duplicate folders	These differ from <i>status</i> = 3; these folders contained a mixture of exact duplicate images as well as images that were often crops or re-scaled variants. Since the folder does not contain exact duplicates of other folders, they received a different code. An example can be raw medical scans before crops were made to classify different parts.
6	different task	Some datasets are distributed with multiple possible tasks, not all of which were used in USC-DCT. These tasks might still have images in the distributed archives, which are eliminated using this status code. For instance, provided segmentation mask images for actual dataset samples would fall into this category.
7	multiple labels	These are images marked with more than one label.
8	too small	These images are smaller than 16 pixels in at least one dimension. We chose this threshold to ensure enough data in any given image.

3.2.2. Preparation

Once the dataset has been inspected, it can be prepared to be added to the collection. For some datasets that instead present the images in .npy, .csv, or other non-image formats, a conversion step is performed to save them as .png files. This standardizes the data loading across all tasks.

Once all necessary image data are in a readable image format, dataset preparation is performed through an automated script (within the function “parse_dataset”) for transparency, where any necessary information provided in the downloaded archives is recorded across all tasks. Supplementary Materials S2 contains samples of customized code for some chosen tasks, where more information can be found on these automated scripts. The information stored for each image includes the following:

- Relative paths to the dataset folder (or to the converted data);
- Integer labels;
- Set (training/validation/testing);
- Subject ID (for any medical or biometric datasets);
- Status code;
- Image hash.

This function (“parse_dataset”) also creates a mapping between classes (i.e., written English—if they exist) to integer labels. If no specific integer labels are given for a dataset, one is created for them (0-indexed) by sorting the class names alphabetically. If no set information is distributed with the dataset, all images are designated as belonging to the training set. The images with a non-zero status code associated with them have their set and integer label recorded as -1 . If no subject/patient/object ID was provided, or it is not applicable for the dataset/image, the subject ID is set to -1 . It has to be noted that status codes 1 and 8 are set automatically by the automated validation script for each task, and are the only non-zero statuses that are not manually provided by the writer of the function “parse_dataset”. As an output, this script generates a list of image files with the above information collected for each image. This list is then used to run the validation for each dataset, providing a standardized, reproducible, and transparent method of integration into the collection.

3.2.3. Validation

Validation was then performed both by framework-provided scripts and human oversight. Each dataset-specific preparation script and results were assigned to one or two other members of our team, who re-inspected dataset folders and processing scripts to confirm the results for the dataset.

The framework-provided validation script also aids users by providing extensive statistics on the processed dataset to help them to verify that any image marked with a non-zero status code actually exhibits that issue. In addition to reviewing the dataset folders and preparation code, validators inspect the validation script summary to verify that (1) the number of images in different sets and/or classes makes sense, (2) class names and integer labels are recorded correctly, and (3) images marked as invalid or exact duplicates are independently confirmed by the script. The script also verifies every image with a non-zero status code has its image label and set assigned to -1 . This process ensures a standardized output across all 107 tasks. Examples of validation script outputs for selected datasets are shown in Supplementary Materials S2.

3.3. Post-Processing

The final stage is performed once the full collection of—or new addition to—USC-DCT is ready. This stage removes both intra- and inter-dataset duplicates, as well as generating the training, validation, and testing sets on a class-by-class basis.

3.3.1. Duplicate Removal

In addition to the above stages that allow us to standardize the collection, a round of clean-up was performed on the datasets. This removed any remaining duplicates that were not easily identified during the dataset preparation (e.g., because the datasets were too large).

During this clean-up, two different hashes were utilized. The file hash was computed using the xxh64sum of the image file, whereas the image hash was the mds5sum of only the final image pixel data. Image pixel data were separately hashed in case there were duplicate images (i.e., the visual representation) that were stored in different file formats or otherwise had been saved in a different manner, which would produce different file hash values, as the file's bytes would now differ. This would also be true if some of the image's metadata (or header) had changed (e.g., the software used, internally stored timestamps, etc.). These two different hashes were used to identify any image duplication issues in USC-DCT datasets and to mark images with further status codes to handle them accordingly. Finally, when removing images based on duplicate files or image hashes, the script first removed those images with duplicate file hashes, and then those with duplicate image hashes.

The status codes related to intra- and inter-dataset duplicates can be seen in Table 2, where we give a shorthand and more detailed explanation for each code that we assign to images at the third stage of the proposed pipeline. In summary, any image that appears in 2+ classes or datasets is excluded from the collection. If an image appears 2+ \times for the same class, only one of the duplicated samples is kept in the collection.

Table 2. Explanation of status codes that every image is assigned with in the USC-DCT collection during the third stage of the pipeline. We present a shorthand for each code, as well as longer explanations.

Status Code	Shorthand	Explanation
100/102	file/image hash in 2+ classes	Either the image had more than one label (see status = 7), or it was improperly placed in multiple classes.
101/103	remaining file/image hash dup.	All images should be unique for each class. Therefore, only one of the file/image hashes is kept.
200/201	file/image hash in 2+ datasets	Sometimes the dataset and class these appear in could make sense, and other times not. However, it is easier to just remove them on the assumption that duplicates should not exist.
300	class size too small	Fewer than three images in a class, which makes it impossible to assign one image to each set.

3.3.2. Set Generation

During the creation of DCT, it was noted that while a few datasets provided suggested sets (training/validation/testing), most did not. And even among the datasets that provided suggested sets, most only provided two: training and testing. Moreover, some testing sets were provided without labels. This is a problem for building models and for comparing final results to those reported by others. As part of this effort, standardized splits were created for every class of every dataset: $\geq 10\%$ testing, $\geq 10\%$ validation, and

$\leq 80\%$ training. This allows anyone reporting results on these datasets to work on the same subsets of data and promotes reproducibility.

Before creating the splits, it is verified that each class has enough images; if any class has fewer than three images left in it (one each for training, validation, and testing sets), it is eliminated by setting the status code for the images of that class to 300; also detailed in Table 2 with other status codes assigned to images during the third stage of the pipeline.

To calculate the splits for most datasets, a simple method was used to randomize the image placement in a thoroughly deterministic way. For every dataset, the images were sorted by their file hash. Then, on a class-by-class basis, the first 10% of images were marked for the validation set, the next 10% of images for the testing set, and the remaining images (80%) for the training set. It should be noted that since this was performed on a class-by-class basis, the class distributions were unchanged from the source datasets—except when a class was eliminated because there were fewer than three images remaining.

For the five datasets that provided us with the subject ID [34–38], the algorithm was complicated by the desire to have every individual subject, regardless of the number of classes they were associated with, appear in only one of the three sets. This prevented any information leaks between the different sets. Therefore, an iterative algorithm was used that sorted the subjects in decreasing order of the number of classes they appeared in and put the subject into the set that was currently most “behind” the desired ratio. The next phase, now working on subjects that appeared in only one class, sorted the subjects in decreasing order of the number of images and put them into sets using the same method. In both phases, a subject was not allowed to be placed into a set if the count of images in any given class exceeded the desired count.

It should be noted that our policy is that new duplicates found when adding future datasets will be removed from those new datasets but kept in the datasets that are already in the collection. This topic is discussed further in Section 5.2.

After the sets are generated, the data are inserted into an SQLite database for distribution. The database includes the relative path for each image, the file hash, image details (e.g., dimensions, type, hash), as well as the classes—both a numeric ID and their text representation.

3.4. Overview

USC-DCT consists of 107 classification tasks collected from 94 diverse datasets, which were acquired and processed by the proposed pipeline in Section 3.2. The additional tasks that bring the collection up to 107 are generated by separating the iNaturalist [33] and Office-Home [32] datasets into multiple tasks. For iNaturalist, this was achieved by separating tasks by defined superclasses in the dataset, while for Office-Home, different domains (clipart images vs. real-world images of dataset classes) were considered different tasks.

For other datasets that provide more than one task or a more complicated classification task, their sub-tasks were added to USC-DCT instead. For example, CelebA [34] is a multi-label classification dataset of facial attributes. Since USC-DCT includes only multi-class classification tasks, the sub-task of hair color was carefully sub-sampled from the multi-label problem in CelebA. For datasets that present multiple tasks like VOC [39] and CORe50 [40], one appropriate classification task was chosen depending on how it contributes to diversity.

Across the 107 tasks included in the collection, USC-DCT boasts 13,269 classes and 6,894,664 images in total. Table 3 shows the minimum, maximum, and median values for the number of classes and images, which shows how diverse the collection is in terms of these values. Moreover, images range in size from 16×16 (WxH) to $15,530 \times 9541$. Table 4 shows each of the 107 tasks with their names, number of classes, and number of total images after they were processed by the proposed pipeline.

Table 3. Statistics on the number of classes and images across the 107 classification tasks in the USC-DCT collection.

		Minimum	Maximum	Median	Average
Number of classes per task		2	4271	15	124
Number of images per task	All	148	1,189,798	15,524	64,436
	Training	112	950,216	12,383	51,493
	Validation	18	119,791	1587	6471
	Test	18	119,791	1587	6471

Table 4. Overview of 107 datasets in the USC-DCT collection version 1.0 with their respective number of classes and the number of total samples used in the benchmark. See Supplemental Materials S3 for visual exemplars of each dataset.

Dataset Name	# of Classes	# of Images	Dataset Name	# of Classes	# of Images
100 Sports [41]	100	14,558	iNaturalist: Mollusca [33]	169	46,239
7000 Pokemon [42]	150	6803	iNaturalist: Plantae [33]	4271	1,189,798
Apparel Images [43]	24	11,372	iNaturalist: Reptilia [33]	313	89,803
APTOS 2019 [44]	5	3504	Intel Image Classification [45]	6	17,003
Art Images Type [46]	5	7110	IP102 Dataset [47]	102	75,217
ASL Alphabets [48]	29	87,000	Kannada MNIST [49]	10	60,000
Blood Cell Images [50]	4	12,513	Kvasir Capsule Dataset [51]	14	37,776
Boat Types [52]	9	1460	Labeled Surgical Tools [53]	5	3004
Book Covers 30 [54]	30	56,975	Land-Use Scene Classification [55]	21	10,499
Brain Tumor Dataset [56]	4	2870	Large-Scale Fish Dataset [57]	9	9408
Brazilian Coins [58]	5	3059	LEGO Bricks [59]	50	40,000
Breast Ultrasound [60]	3	778	Malacca Historical Buildings [61]	3	162
Cataract Dataset [62]	4	601	Manga Facial Expressions [63]	7	455
CelebA [34]	5	124,803	Mechanical Tools [64]	8	7335
Chars74k [65]	62	11,202	MIT Indoor Scenes [66]	67	15,524
Chest X-Ray [67]	2	5824	Monkey Species [68]	10	1267
CLEVR v1.0 [69]	8	85,000	Multi-Class Weather Dataset [70]	4	1099
Colorectal Histology MNIST [71]	8	5000	NEU Surface Defect [72]	6	1799
Concrete Cracks [73]	2	38,402	NWPU-RESISC45 [74]	45	31,492
CORE50 [40]	10	163,006	Office-Home: Art [32]	65	2297
CUB-200 [75]	200	11,787	Office-Home: Clipart [32]	65	4190
DeepVP-1M [76]	9	74,288	Office-Home: Product [32]	65	4269
DeepWeedsX [77]	9	17,508	Office-Home: Real World [32]	65	4317
DermNet Dataset [78]	23	17,826	OnePiece Dataset [79]	18	11,503
Describable Textures [80]	47	5623	Oregon Wildlife [81]	20	7076
Diabetic Retinopathy [36]	5	35,126	OrigamiSet 1.0 [82]	3	1495
Dragon Ball Super Saiyan [83]	6	148	Oxford Buildings [84]	11	845
Electronic Components [85]	36	10,153	PAD-UFES-20 [38]	6	2269
EuroSAT [86]	10	27,000	PatchCamelyon [87]	2	277,483
FaceMask Dataset [88]	3	13,755	Planets And Moons [89]	11	1634
Facial Expression 2013 [90]	7	33,977	PlantDoc Dataset [91]	27	2547
Fashion Product Images [92]	43	43,653	Polish Craft Beer Labels [93]	100	7971
Fine-Grained Aircraft [94]	70	10,000	Retinal-OCT 2017 [37]	4	76,677
Flowers [95]	102	8185	Rice Image Dataset [96]	5	74,703
Food-101 [97]	101	100,938	Rock Classification [98]	7	2032
Freiburg Groceries [99]	25	4933	Russian Letter Dataset [100]	33	37,667
Galaxy10 [101]	10	17,615	RVL-CDIP [102]	16	398,388
Garbage Classification [103]	12	15,493	Satellite Images African Poverty [104]	4	25,571
GTSRB [105]	43	51,831	Simpsons Characters [106]	42	21,882
HistAerial [107]	7	137,427	Sketches [108]	250	19,999

Table 4. Cont.

Dataset Name	# of Classes	# of Images	Dataset Name	# of Classes	# of Images
House Room Images [109]	5	5174	Skin Cancer MNIST [35]	7	10,013
Hurricane Damage [110]	2	21,050	Stanford Cars [111]	196	16,109
iFood2019 [112]	251	130,468	Stanford Online Products [113]	12	117,983
iLab 80m [114]	15	15,000	SVHN [22]	10	630,417
iLab Atari [115]	67	368,870	Texture Dataset [116]	64	8530
iMaterialist Fashion 2019 [117]	46	45,191	UIUC Sports Event Dataset [118]	8	1578
iNaturalist: Actinopterygii [33]	183	46,641	UMIST FaceDatabase [119]	20	1012
iNaturalist: Amphibia [33]	170	47,850	Vegetable Images Dataset [120]	15	20,996
iNaturalist: Animalia [33]	142	38,349	VOC2012 - Human Action [39]	11	4294
iNaturalist: Arachnida [33]	153	42,164	Watermark Classification [121]	2	31,147
iNaturalist: Aves [33]	1486	428,775	WikiArt Dataset [122]	27	78,677
iNaturalist: Fungi [33]	341	93,359	Yoga-82 [123]	82	19,450
iNaturalist: Insecta [33]	2526	687,823	Zolando Clothing/Models [124]	6	10,667
iNaturalist: Mammalia [33]	246	71,276			
			Total:	13,269	6,894,664

4. Results

This section includes the USC-DCT collection characteristics, ranging from an overview of the raw datasets themselves to the distribution of challenges faced when processing these datasets and forming the collection.

4.1. Task Diversity

Task diversity was one of the main aims during the collection of USC-DCT. To show how diverse the USC-DCT tasks are, we assign a semantic label to each task from 16 different semantic areas. These labels include nature, vehicles, art, media, medical, scenes, facial, food, fashion, letters/numbers, astronomy, instruments, actions, objects, satellite imagery, and counting. We mark the datasets that do not fit into these 16 semantic labels as 'other'. A visual representation of the diversity of images is provided in Supplementary Materials S3.

Figure 2 shows how these semantic labels are distributed across different-sized tasks in the collection and compares USC-DCT to other popular collections of datasets used in the literature such as the 8-dataset collection [8], Visual Domain Decathlon [20], and fine-grained 6-tasks [19]. This analysis displays how diverse USC-DCT is compared to these other collections of datasets, which shows how useful it can be to the machine learning community. We note that there is no comparison of these different collections using t-SNE (see Figure 3), because the method does not allow direct comparisons from such wildly different data (e.g., cluster locations and shapes, even for the same data, will be completely different).

Moreover, to further identify how similar these tasks are to each other in the feature space, a sub-sample of USC-DCT was generated to have five images per class from every task in the collection. Then, a deep neural network [125] was trained on ImageNet [1] to extract features from this sub-sampled version of the datasets. Figure 3 shows the visualization of these features through a t-SNE [126] plot, and shows the diversity of the collection as well as how different tasks intersect with each other.

When viewed in the feature space, some tasks are overlapping and appear close to each other, which is to be expected. Examples of this are Food-101 [97] and iFood2019 [112], as well as partial overlaps between iNaturalist [33] sub-tasks Aves and Plantae with tasks CUB-200 [75] and Flowers [95], respectively. The Office-Home sub-tasks that contain the same objects in different domains, like real world, product, and clipart, also fall into similar regions in the t-SNE plot.

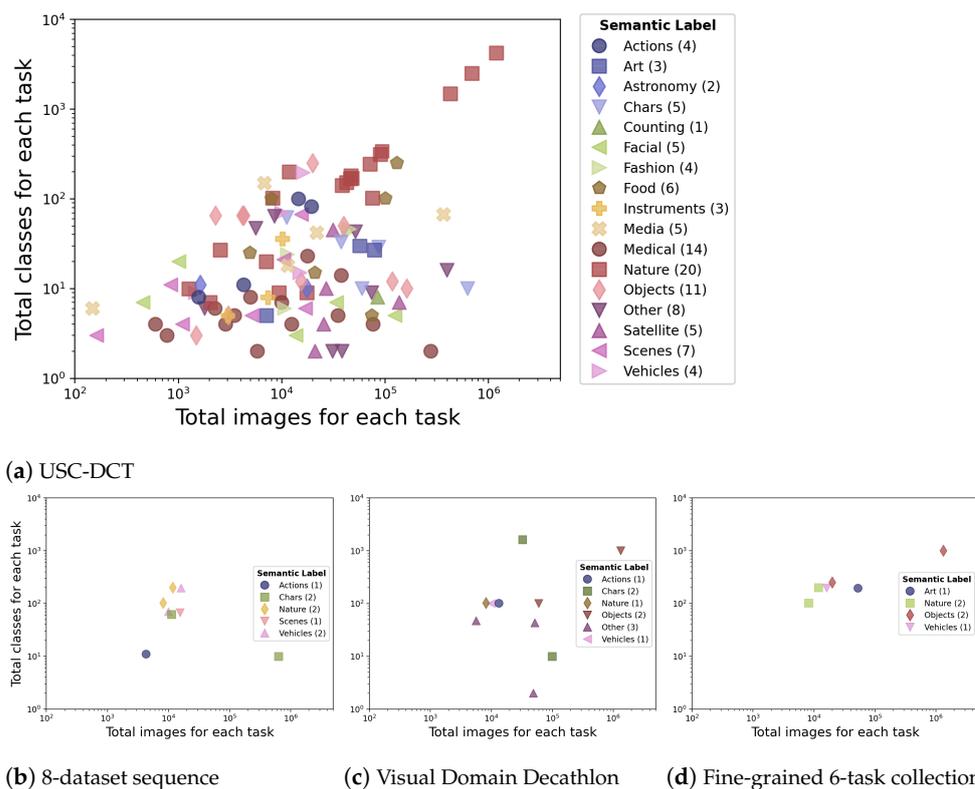


Figure 2. Semantic labels vs. task sizes of common collections used in the literature to develop and evaluate different learning paradigms compared to those of USC-DCT. Every task was hand-labeled to show the level of diversity in the 107 tasks in DCT.

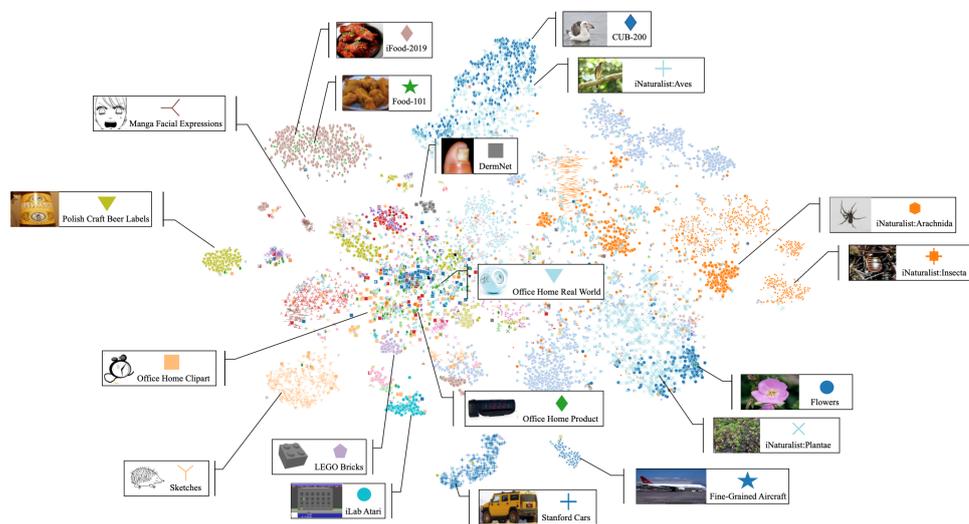


Figure 3. t-SNE visualization of the feature space for a sub-sample of images from each of the 107 tasks in USC-DCT.

Meanwhile, in line with the design goals of the collection, a large variety of other tasks can be viewed in their separate clusters. These include the unique datasets Polish Craft Beer Labels [93], LEGO Bricks [59], and iLab Atari [115], as well as the more well known Stanford Cars [111] and Fine-Grained Aircraft [94]. These experiments further show the diversity in the USC-DCT collection.

4.2. Status Code Statistics

As mentioned in Section 3.2, one of the challenges that was encountered during the creation of this large-scale collection was the issue with standardization in structuring and publishing datasets, as well as issues with exact duplicates in and across datasets. This section summarizes some of the more important points that were discovered by applying the proposed dataset pipeline to these 107 tasks. Table 5 presents the exact numbers of images and datasets affected by each identified status, and then further discussion of some of these results follows.

Table 5. Breakdown of how many images and datasets were affected by their identified status during collection creation using the proposed pipeline.

Stage	Status Code	# of Images	# of Datasets
Pre-Processing and Validation	1 (invalid file)	225	6
	2 (non-dataset image)	11	5
	3 (duplicate folders)	260,707	17
	4 (no label)	143,829	14
	5 (near-duplicate folders)	10,801	2
	6 (different task)	3,674,847	8
	7 (multiple labels)	284	1
	8 (image too small)	1369	3
Post-Processing	100 (file hash in 2+ classes)	13,610	48
	101 (remaining file hash duplicates)	113,819	81
	102 (image hash in 2+ classes)	20	4
	103 (remaining image hash duplicates)	328	14
	200 (file hash in 2+ datasets)	504	15
	201 (image hash in 2+ datasets)	2	2
	300 (class size too small)	4	2

First, some key statistics needed to be identified: Across the several million images that comprise USC-DCT, only a small fraction were invalid files that could not be loaded (a total of 225). Only five datasets included summary or other non-dataset images discovered in the archives which we had to identify and eliminate from the collection. Comparatively, dataset images that appeared in archives without an accompanying label appeared around 143k times, across 14 of the tasks. These images were mostly test sets with unreleased labels or other demonstration images distributed with the archives.

Other images that were not included in the USC-DCT collection consisted of images that were marked with multiple labels in the dataset (affecting only one dataset and 284 images), and images that were too small to reliably use for classification (around 1k images appearing in three datasets).

We also investigated datasets where a large percentage of images in the downloaded archives were excluded from the final collection. It was found that high image-elimination rates from these datasets were caused by two main factors: The removal of images due to task selection, and the removal of images due to partially, or entirely, duplicated folders. The former is a design choice for creating this collection, and affected datasets, e.g., CelebA [34], HistAerial [107], were those we chose a sub-task for to add to USC-DCT. The latter implies distribution errors made while publishing the datasets (e.g., one dataset contained a full copy of itself in a class-level sub-folder); these require careful inspection and can lead to mistakes in training and evaluation if uncaught.

There were 20 datasets that contained a total of 3052 images that had a mismatch between the file extension and the actual image stored within. This count is small in comparison to the total collection but possibly indicates that some datasets used a naive conversion and renaming pipeline (either before the image was uploaded to the web (in the case that the dataset was collected from the web), or during the processing (for images collected in other ways)).

In addition to computing the statistics for defined status codes for the proposed processing pipeline, an analysis of how many images and which tasks were affected by intra- and inter-dataset exact duplicates was completed. Thousands of images were found multiple times in different class folders, as well as many that were duplicated in the same class folder. Images that appeared multiple times in different datasets were also identified. These were much less frequent than intra-dataset duplications; however, for the health of the collection, they were also eliminated from the database.

Figure 4 gives a summary of the number of images and datasets affected by each status code identified in Section 3.2. Overall, there were 9 tasks out of the 107 that had no images removed for any reason. These were DeepVP-1M [76], Diabetic Retinopathy Detection [36], EuroSAT [86], Sketches [108], iLab 80M [114], Fine-Grained Aircraft [94], UMIST Face Database [119], Manga Facial Expressions [63], and Malacca Historical Buildings [61]. This shows how challenging it can be to form a collection of this scale.

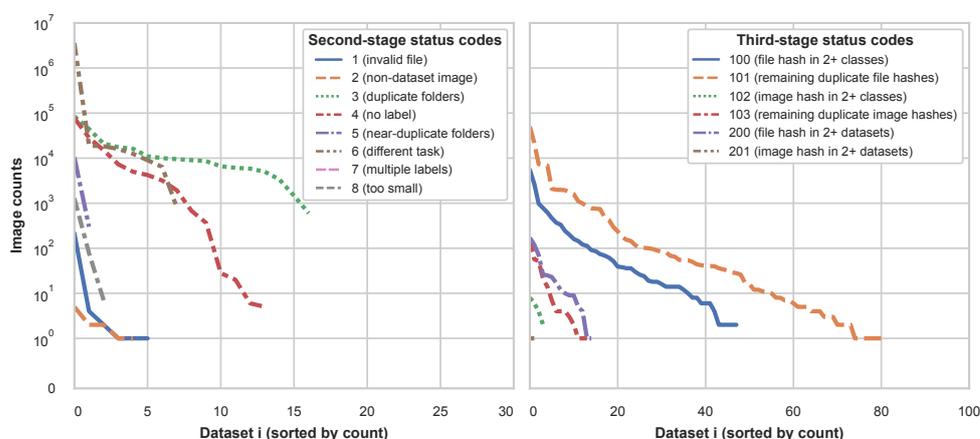


Figure 4. The distribution of status codes sorted by the total count of images in the dataset with that code.

5. Discussion

5.1. Utility of USC-DCT

As previously mentioned in Sections 1 and 2, a large-scale collection like USC-DCT can benefit many different learning paradigms in the classification domain. Some frameworks of note are transfer learning, lifelong learning, meta-learning, and task adaptation. In this section, further discussion on how USC-DCT can be utilized for the aforementioned frameworks, as well as others in the field, is presented.

Transfer learning and pre-training have been very important for innovations in machine learning during the last decade, where a model is first trained on a large-scale dataset to learn general discriminative features. The pre-trained model can then be fine-tuned for a specific task using fewer data and fewer resources [15,127]. USC-DCT used as a whole can be an important asset for developing and validating new approaches to transfer learning. Moreover, the semantic labels we provide for each dataset can allow subsets of the collection to be used for this purpose as well.

Collections like USC-DCT have already been widely utilized for lifelong learning [16] and domain adaptation [18], even though the collections have been up to 20 datasets. Having a large-scale collection like USC-DCT can help test these algorithms' generalization abilities at scale, either by evaluating subsets of sequences from USC-DCT or evaluating all tasks in it. Since USC-DCT has been designed to contain a diverse number of tasks, it also makes interdisciplinary research on these frameworks easier. Using this benchmark, one can just as easily evaluate a lifelong learning algorithm developed on object classification on a subset of USC-DCT that contains medical tasks. Moreover, it can inspire new research directions, such as shared lifelong learning, proposed in [9].

On the other hand, in paradigms like meta-learning [17], the large-scale variety of tasks is what is crucial. Many meta-learning algorithms require some base knowledge to train, and having a collection of at least 100 datasets can unlock the development of algorithms that can learn how to manage a variety of new tasks. Having a large variety in the base knowledge can also benefit the training and evaluation of zero-, one-, or few-shot learning frameworks. These paradigms generally assume good existing representations and attempt to learn new concepts using this knowledge. While most of these studies use partitioned versions of popular datasets like ImageNet [1] and CIFAR [2], having a large collection like USC-DCT can truly test how generalizable these algorithms can be.

While USC-DCT can be used with all the learning paradigms mentioned above, it is not necessary to use it as a complete benchmark. The individual datasets in USC-DCT can easily be used on their own for many different small-scale projects since they are distributed in a standard way, independent of a deep learning framework like PyTorch or TensorFlow. This allows anyone interested in classification problems to experiment with many different tasks with minimal effort to integrate a dataset into their code, allowing for easier cross-domain collaboration.

5.2. Revisions to DCT

USC-DCT is distributed in a fully standardized way, where every dataset was processed using the proposed pipeline. This makes it feasible for this collection to grow and change as more classification tasks are collected. New datasets can easily be added to the collection, which can grow to be a popular repository of diverse classification tasks in the future. We expect each future addition to be inspected and cleaned in the same manner as above, and to be distributed with pre-defined splits that can be used across the community for a fair evaluation of all tasks included.

The version that is released with this paper is considered to be version 1.0. Although there is full support for adding additional datasets, in order for others to use the same collection, three things are necessary: (1) there needs to be an authoritative source for any new additions, (2) any new datasets must be publicly available under an open-source license, and (3) the new revisions should not add or remove datasets already in any previous versions. To that end, new additions to the collection will go through a slightly different pipeline:

- The fully intra-dataset status codes will still be handled by the individuals who write the scripts to parse each new dataset and the uniform first-pass quality checks. Additionally, the analysis of the “post-processing” status codes 100–103, which are concerned with intra-dataset duplication removal, will remain the same.
- The post-processing status codes that are concerned with inter-dataset duplication removal (i.e., status codes 200 and 201) are handled in a slightly different manner. Now, instead of all file and image hash duplicates being removed from all datasets, only those duplicates that appear in the new datasets that are being added will be removed. This means that no images will be removed from the previous revisions of USC-DCT.
- Finally, the set building for the new datasets will follow the same pipeline. If any class contains fewer than three images, status code 300 will be applied.

In summary, the current version 1.0 and all of the images and set assignments—barring future bug fixes that affect the current datasets—are considered immutable. When new revisions are added (with one or more new datasets), the inter-dataset duplication removal process will only remove duplicates found in the new datasets (i.e., the inter-dataset duplicates that exist between new and existing datasets will only be removed from the new datasets). Upon the release of the updated version, the consideration of immutability will be extended to all images and set assignments from the then-current list of datasets.

5.3. Tips for Usage

Depending on each particular use case, using some or all of the datasets contained with USC-DCT may be appropriate. For instance, if one is only interested in training a model on naturalistic scenes, it would make sense to exclude datasets containing histology, graphic art, etc. In general, we recommended the following:

- Train on the complete training set of the desired datasets.
- Evaluate models on the validation set during the hyperparameter tuning phase.
- Only evaluate final models on the testing set, and only once. When comparing results against others that used the same datasets, compare against the results on the testing set of the same version.

Although we strongly recommend against it, if the sets are too large for the available time or computing constraints, sub-sampling of the training/validation/testing sets could be a solution. However, in order to guarantee that others will be able to make apples-to-apples comparisons with the results, one would need to make sure that the data were sub-sampled in a deterministic manner. The simplest and most foolproof way to achieve this is to take the first X samples in the training/validation/testing sets. Because the images have been sorted using their file hash, they are already randomly ordered in the database. Therefore, if one chooses the first X entries (of either the entire dataset or of each of the classes in a dataset), they will be guaranteed a deterministic random sample that others can reproduce—knowing that they will be using the same set of sub-sampled images. It should be noted that we do not consider a sub-sampling of USC-DCT to be synonymous with the complete dataset, nor do we make any claims that a sub-sample is representative of the complete dataset or datasets. If sub-sampling from USC-DCT, the methodology used should be clearly stated, and if it deviates from the above proposed one, we recommend storing a list of the hashes used for the generated subsets, so that others can exactly reproduce the same sub-sample. For an example of creating a sub-sampled subset, the authors developed one for the work completed in our associated paper [9].

5.4. Recommendations

During the gathering and processing of USC-DCT, over 100 datasets were scrutinized. In this section, we discuss some recommendations for anyone who plans to gather and publish a new dataset based on the experiences with forming USC-DCT:

- ✓ Documentation on both the data and distribution structure. Build these before populating them with data. The documentation should match the rest of the built dataset.
- ✓ Structuring data in sensible ways. Class folders seem to be the best method.
- ✓ Choosing the annotation distribution format carefully. Any additional annotations could be in .txt/.csv format. Annotations should not be replicated in multiple areas.
- ✓ Not leaving any extraneous files in the final distribution. This includes hidden files as well.
- ✓ Basic checks for exact duplicates.

These recommendations are meant to act as a checklist for anyone wishing to publish data in an accessible manner. While they are good properties to have, not all will represent the optimal choice for every dataset. However, the more datasets that are published using a similar approach for structuring and distributing their data, the more accessible they will be for future research or other applications.

5.5. DCT Reconstitution

Our repository (USC-DCT, <https://github.com/iLab-USC/USC-DCT>) has the code necessary to reconstitute, or rebuild, part of or the entire USC-DCT. This allows users to then train and evaluate the collection on their own machines, without us having to store, and provide for download, the entire collection. The steps for rebuilding the USC-DCT are:

- Clone the git repository to a drive.

- Run `reconstitute_dct.py`, which will allow one to choose the entire collection, or just a subset, and what version to use.
- This will download the original archive files for each desired dataset. These archives will then be expanded and processed.
- Finally, the script will download an individualized mapping of all files, which set they are included in, and what their status codes are.

6. Conclusions

This work presents USC-DCT, a diverse collection of classification tasks compiled from publicly available classification datasets. USC-DCT is a standardized database of 107 classification tasks, as well as a set of scripts that are transparent and reproducible with minimum logistical overhead for setting up a large collection. In the paper, several items are presented: the proposed pipeline for generating USC-DCT, a methodology for easy future growth, and statistics on the collection. Subsequent discussion then focuses on its use cases and importance in machine learning research, tips for usage, as well as our recommendations for publishing a dataset based on our experiences with creating the collection.

USC-DCT is designed to easily ingest new classification tasks, and can further scale as algorithms become more generalizable and adaptable. Moreover, it can further evolve via future efforts that focus on handling images that are similar but not exactly the same, label ambiguities, and similar classes that might exist across datasets. These additions can make USC-DCT into the largest fully unified classification benchmark in the future.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/data8100153/s1> or the USC-DCT Github Repo, <https://github.com/iLab-USC/USC-DCT>: USC-DCT Database Inclusion Decision Tree, S1; Dataset Preparation Code Examples, S2; and Visual Overview of USC-DCT v1.0, S3.

Author Contributions: G.S., A.M.J. and Z.W.M. wrote the manuscript and prepared the figures. A.M.J., G.S., Z.W.M., A.X., Y.G., Y.L., D.W., S.N. and K.L. wrote and verified the individual dataset scripts. Z.W.M. and P.-H.H. assembled the raw collection of datasets. A.X. computed the t-SNE results. G.S., A.M.J., Z.W.M. and L.I. were involved in reviewing and editing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by DARPA (HR00112190134), C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), and the Army Research Office (W911NF2020053). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: USC-DCT is a large-scale collection formed of publicly available image classification datasets. We do not re-distribute datasets that are included in USC-DCT. Instead, through the toolkit published at <https://github.com/iLab-USC/USC-DCT>, these datasets are automatically downloaded from their respective links and processed to set up the final collection database. The database files are included in the toolkit repository to track version changes as the collection grows in the future.

Conflicts of Interest: The authors declare no conflict of interest. Additionally, the funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
2. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.

3. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 20 June 2022).
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; Volume 3385, pp. 770–778.
6. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.
7. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
8. Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2018; pp. 139–154.
9. Ge, Y.; Li, Y.; Wu, D.; Xu, A.; Jones, A.M.; Rios, A.S.; Fostiropoulos, I.; Wen, S.; Huang, P.H.; Murdock, Z.W.; et al. Lightweight Learner for Shared Knowledge Lifelong Learning. *arXiv* **2023**, arXiv:2305.15591.
10. Cohen, G.; Afshar, S.; Tapson, J.; Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 international joint conference on neural networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2921–2926.
11. Beyer, L.; Hénaff, O.J.; Kolesnikov, A.; Zhai, X.; Oord, A.v.d. Are we done with imagenet? *arXiv* **2020**, arXiv:2006.07159.
12. Ekambaram, R.; Goldgof, D.B.; Hall, L.O. Finding label noise examples in large scale datasets. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 2420–2424.
13. Rolnick, D.; Veit, A.; Belongie, S.; Shavit, N. Deep learning is robust to massive label noise. *arXiv* **2017**, arXiv:1705.10694.
14. Barz, B.; Denzler, J. Do We Train on Test Data? Purging CIFAR of Near-Duplicates. *J. Imaging* **2020**, *6*, 41. [[CrossRef](#)] [[PubMed](#)]
15. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [[CrossRef](#)]
16. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [[CrossRef](#)]
17. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [[CrossRef](#)]
18. Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **2018**, *312*, 135–153. [[CrossRef](#)]
19. Mallya, A.; Davis, D.; Lazebnik, S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 67–82.
20. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Learning multiple visual domains with residual adapters. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1.
21. Zhai, X.; Puigcerver, J.; Kolesnikov, A.; Ruyssen, P.; Riquelme, C.; Lucic, M.; Djolonga, J.; Pinto, A.S.; Neumann, M.; Dosovitskiy, A.; et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv* **2019**, arXiv:1910.04867.
22. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011.
23. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
24. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: [tensorflow.org](https://www.tensorflow.org) (accessed on 20 June 2022).
25. Lhoest, Q.; Villanova del Moral, A.; Jernite, Y.; Thakur, A.; von Platen, P.; Patil, S.; Chaumond, J.; Drame, M.; Plu, J.; Tunstall, L.; et al. Datasets: A Community Library for Natural Language Processing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Punta Cana, Dominican Republic, 7–12 November 2021; pp. 175–184.
26. Ng, H.W.; Winkler, S. A data-driven approach to cleaning large face datasets. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 343–347. [[CrossRef](#)]
27. Li, P.; Rao, X.; Blase, J.; Zhang, Y.; Chu, X.; Zhang, C. CleanML: A study for evaluating the impact of data cleaning on ML classification tasks. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 13–24.
28. Krishnan, S.; Franklin, M.J.; Goldberg, K.; Wang, J.; Wu, E. ActiveClean: An Interactive Data Cleaning Framework For Modern Machine Learning. In Proceedings of the 2016 International Conference on Management of Data SIGMOD '16, New York, NY, USA, 18–21 November 2016; pp. 2117–2120. [[CrossRef](#)]
29. Bernhardt, M.; Castro, D.C.; Tanno, R.; Schwaighofer, A.; Tezcan, K.C.; Monteiro, M.; Bannur, S.; Lungren, M.P.; Nori, A.V.; Glocker, B.; et al. Active label cleaning: Improving dataset quality under resource constraints. *Nat. Commun.* **2022**, *13*, 1161. [[CrossRef](#)] [[PubMed](#)]

30. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
31. Goodfellow, I.J.; Mirza, M.; Xiao, D.; Courville, A.; Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv* **2013**, arXiv:1312.6211.
32. Venkateswara, H.; Eusebio, J.; Chakraborty, S.; Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5018–5027.
33. Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; Belongie, S. The inaturalist species classification and detection dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8769–8778.
34. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
35. Tschandl, P.; Rosendahl, C.; Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data* **2018**, *5*, 1–9. [[CrossRef](#)] [[PubMed](#)]
36. Dugas, E.; Jorge, J.; Cukierski, W. Diabetic Retinopathy Detection. 2015. Available online: <https://kaggle.com/competitions/diabetic-retinopathy-detection> (accessed on 20 June 2022)
37. Kermany, D.; Zhang, K.; Goldbaum, M. Large dataset of labeled optical coherence tomography (oct) and chest X-ray images. *Mendeley Data* **2018**, *3*, 17632.
38. Pacheco, A.G.; Lima, G.R.; Salomão, A.S.; Krohling, B.; Biral, I.P.; de Angelo, G.G.; Alves, F.C., Jr.; Esgario, J.G.; Simora, A.C.; Castro, P.B.; et al. PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Data Brief* **2020**, *32*, 106221. [[CrossRef](#)] [[PubMed](#)]
39. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. Available online: <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html> (accessed on 20 June 2022).
40. Lomonaco, V.; Maltoni, D. Core50: A new dataset and benchmark for continuous object recognition. In Proceedings of the Conference on Robot Learning (PMLR), Auckland, New Zealand, 14–18 December 2017; pp. 17–26.
41. 100 Sports Image Classification. 2022. Available online: <https://www.kaggle.com/datasets/gpiosenska/sports-classification> (accessed on 20 June 2022).
42. 7000 Labeled Pokemon. 2019. Available online: <https://www.kaggle.com/datasets/lantian773030/pokemonclassification> (accessed on 20 June 2022).
43. Apparel Images Dataset. 2020. Available online: <https://www.kaggle.com/datasets/trolukovich/apparel-images-dataset> (accessed on 20 June 2022).
44. Karthik, M., Sohler, D. The Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset, 2019. Available online: <https://www.kaggle.com/c/aptos2019-blindness-detection/overview> (accessed on 20 June 2022).
45. Intel Image Classification. 2019. Available online: <https://www.kaggle.com/datasets/puneet6060/intel-image-classification> (accessed on 20 June 2022).
46. Art Images: Drawing/Painting/Sculptures/Engravings. 2018. Available online: <https://www.kaggle.com/datasets/thedownhill/art-images-drawings-painting-sculpture-engraving> (accessed on 20 June 2022).
47. Wu, X.; Zhan, C.; Lai, Y.; Cheng, M.M.; Yang, J. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In Proceedings of the IEEE CVPR, Long Beach, CA, USA, 15–19 June 2019; pp. 8787–8796.
48. ASL Alphabet. 2018. Available online: <https://www.kaggle.com/datasets/grassknotted/asl-alphabet> (accessed on 20 June 2022).
49. Prabhu, V.U. Kannada-MNIST: A new handwritten digits dataset for the Kannada language. *arXiv* **2019**, arXiv:1908.01242.
50. Blood Cell Images. 2018. Available online: <https://www.kaggle.com/datasets/paultimothymooney/blood-cells> (accessed on 20 June 2022).
51. Smedsrud, P.H.; Thambawita, V.; Hicks, S.A.; Gjestang, H.; Nedrejord, O.O.; Næss, E.; Borgli, H.; Jha, D.; Berstad, T.J.D.; Eskeland, S.L.; et al. Kvasir-Capsule, a video capsule endoscopy dataset. *Sci. Data* **2021**, *8*, 42. [[CrossRef](#)] [[PubMed](#)]
52. Boat Types Recognition. 2018. Available online: <https://www.kaggle.com/datasets/clorichel/boat-types-recognition> (accessed on 20 June 2022).
53. Labeled Surgical Tools and Images. 2018. Available online: <https://www.kaggle.com/datasets/dilavado/labeled-surgical-tools> (accessed on 20 June 2022).
54. Iwana, B.K.; Raza Rizvi, S.T.; Ahmed, S.; Dengel, A.; Uchida, S. Judging a Book by its Cover. *arXiv* **2016**, arXiv:1610.09204.
55. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
56. Bhuvaji, S.; Kadam, A.; Bhumkar, P.; Dedge, S.; Kanchan, S. Brain Tumor Classification (MRI). 2020. Available online: <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri> (accessed on 20 June 2022).
57. Ulucan, O.; Karakaya, D.; Turkan, M. A Large-Scale Dataset for Fish Segmentation and Classification. In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020; pp. 1–5.

58. Moneda, L.; Yonekura, D.; Guedes, E. Brazilian Coin Detection Dataset. *IEEE Dataport* **2020**, *2020*, 809. [CrossRef]
59. Images of LEGO Bricks. 2019. Available online: <https://www.kaggle.com/datasets/joosthazelzet/lego-brick-images> (accessed on 20 June 2022).
60. Al-Dhabyani, W.; Goma, M.; Khaled, H.; Fahmy, A. Dataset of breast ultrasound images. *Data Brief* **2020**, *28*, 104863. [CrossRef]
61. Historical Building (Malacca, Malaysia). 2021. Available online: <https://www.kaggle.com/datasets/joeylimzy/historical-building-malacca-malaysia> (accessed on 20 June 2022).
62. Cataract Dataset. 2019. Available online: <https://www.kaggle.com/datasets/jr2ngb/cataractdataset> (accessed on 20 June 2022).
63. Manga Facial Expressions. 2021. Available online: <https://www.kaggle.com/datasets/mertkkl/manga-facial-expressions> (accessed on 20 June 2022).
64. Hossain, S.; Komol, J.; Raidah, M.M. Mechanical Tools Classification Dataset. 2020. Available online: <https://www.kaggle.com/datasets/salmaneeunus/mechanical-tools-dataset> (accessed on 20 June 2022).
65. de Campos, T.E.; Babu, B.R.; Varma, M. Character recognition in natural images. In Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, 19–21 February 2009.
66. Quattoni, A.; Torralba, A. Recognizing indoor scenes. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 413–420.
67. Kermany, D.; Zhang, K.; Goldbaum, M. Labeled optical coherence tomography (oct) and chest X-ray images for classification. *Mendeley Data* **2018**, *2*, 2.
68. 10 Monkey Species. 2018. Available online: <https://www.kaggle.com/datasets/slothkong/10-monkey-species> (accessed on 20 June 2022).
69. Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–27 July 2017; pp. 2901–2910.
70. Zhang, Z.; Ma, H.; Fu, H.; Zhang, C. Scene-free multi-class weather classification on single images. *Neurocomputing* **2016**, *207*, 365–373. [CrossRef]
71. Kather, J.N.; Weis, C.A.; Bianconi, F.; Melchers, S.M.; Schad, L.R.; Gaiser, T.; Marx, A.; Zöllner, F.G. Multi-class texture analysis in colorectal cancer histology. *Sci. Rep.* **2016**, *6*, 27988. [CrossRef] [PubMed]
72. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
73. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
74. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
75. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *Caltech-UCSD Birds-200-2011*; Technical Report CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.
76. Chang, C.K.; Zhao, J.; Itti, L. DeepVP: Deep Learning for Vanishing Point Detection on 1 Million Street View Images. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 4496–4503. [CrossRef]
77. Lammie, C.; Olsen, A.; Carrick, T.; Azghadi, M.R. Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge. *IEEE Access* **2019**, *7*, 51171–51184. [CrossRef]
78. Dermnet. 2020. Available online: <https://www.kaggle.com/datasets/shubhamgoel27/dermnet> (accessed on 20 June 2022).
79. One Piece Image Classifier. 2021. Available online: <https://www.kaggle.com/datasets/ibrahimserouis99/one-piece-image-classifier> (accessed on 20 June 2022).
80. Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; Vedaldi, A. Describing Textures in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
81. Oregon Wildlife. 2019. Available online: <https://www.kaggle.com/datasets/virtuallavid/oregon-wildlife> (accessed on 20 June 2022).
82. Ma, D.; Friedland, G.; Krell, M.M. OrigamiSet 1.0: Two New Datasets for Origami Classification and Difficulty Estimation. *arXiv* **2021**, arXiv:2101.05470.
83. Dragon Ball—Super Saiyan Dataset. 2020. Available online: <https://www.kaggle.com/datasets/bhav09/dragon-ball-super-saiyan-dataset> (accessed on 20 June 2022).
84. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Object retrieval with large vocabularies and fast spatial matching. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–27 July 2007; pp. 1–8.
85. Electronic Components and Devices. 2018. Available online: <https://www.kaggle.com/datasets/aryaminus/electronic-components> (accessed on 20 June 2022).
86. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2019**, *12*, 2217–2226. [CrossRef]

87. Veeling, B.S.; Linmans, J.; Winkens, J.; Cohen, T.; Welling, M. Rotation equivariant CNNs for digital pathology. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Granada, Spain, 16–20 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 210–218.
88. Mask Dataset. Available online: <https://makeml.app/datasets/mask> (accessed on 20 June 2022).
89. BULUT, E. Planets and Moons Dataset—AI in Space: A Public Dataset for Large-Scale Multi-Label and Multi-Class Image Classification. 2022. Available online: <https://www.kaggle.com/datasets/emirhanai/planets-and-moons-dataset-ai-in-space> (accessed on 20 June 2022).
90. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.H.; et al. Challenges in representation learning: A report on three machine learning contests. In Proceedings of the Neural Information Processing: 20th International Conference (ICONIP 2013), Daegu, Republic of Korea, 3–7 November 2013; Proceedings, Part III 20.
91. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: a dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, New York, NY, USA, 5–7 January 2020; pp. 249–253.
92. Fashion Product Images Dataset. 2019. Available online: <https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset> (accessed on 20 June 2022).
93. Galla, Z. Polish Craft Beer Labels. 2021. Available online: <https://www.kaggle.com/datasets/zozolla/polish-craft-beer-labels/> (accessed on 20 June 2022).
94. Maji, S.; Kannala, J.; Rahtu, E.; Blaschko, M.; Vedaldi, A. *Fine-Grained Visual Classification of Aircraft*; Technical Report; University of Oxford: Oxford, UK, 2013. Available online: <http://xxx.lanl.gov/abs/1306.5151> (accessed on 20 June 2022).
95. Nilsback, M.E.; Zisserman, A. Automated flower classification over a large number of classes. In Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 722–729.
96. Koklu, M.; Cinar, I.; Taspinar, Y.S. Classification of rice varieties with deep learning methods. *Comput. Electron. Agric.* **2021**, *187*, 106285. [\[CrossRef\]](#)
97. Bossard, L.; Guillaumin, M.; Gool, L.V. Food-101—mining discriminative components with random forests. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 446–461.
98. Hossain, S.; Uddin, J.; Nahin, R.A. Rock Classification Dataset. 2021. Available online: <https://www.kaggle.com/datasets/salmaneunus/rock-classification> (accessed on 20 June 2022).
99. Jund, P.; Abdo, N.; Eitel, A.; Burgard, W. The freiburg groceries dataset. *arXiv* **2016**, arXiv:1611.05799.
100. Classification of Handwritten Letters. 2021. Available online: <https://www.kaggle.com/datasets/olgabelitskaya/classification-of-handwritten-letters> (accessed on 20 June 2022).
101. Walmsley, M.; Lintott, C.; Tobias, G.; Kruk, S.J.; Krawczyk, C.; Willett, K.; Bamford, S.; Keel, W.; Kelvin, L.S.; Fortson, L.; et al. Galaxy Zoo DECaLS: Detailed Visual Morphology Measurements from Volunteers and Deep Learning for 314,000 Galaxies. *Mon. Not. R. Astron. Soc.* **2020**, *509*, 3966–3988. [\[CrossRef\]](#)
102. Harley, A.W.; Ufkes, A.; Derpanis, K.G. Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015.
103. Garbage Classification Dataset, 2018. Available online: <https://www.kaggle.com/datasets/asdasdasdas/garbage-classification> (accessed 20 June 2022).
104. Satellite Images to Predict Poverty. 2021. Available online: <https://www.kaggle.com/datasets/sandeshbhat/satellite-images-to-predict-povertyafrica> (accessed on 20 June 2022).
105. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [\[CrossRef\]](#) [\[PubMed\]](#)
106. The Simpsons Characters Data. 2018. Available online: <https://www.kaggle.com/datasets/alexattia/the-simpsons-characters-dataset> (accessed on 20 June 2022).
107. Ratajczak, R.; Crispim-Junior, C.F.; Faure, É.; Fervers, B.; Tougne, L. Automatic Land Cover Reconstruction From Historical Aerial Images: An Evaluation of Features Extraction and Classification Algorithms. *IEEE Trans. Image Process.* **2019**, *28*, 3357–3371. [\[CrossRef\]](#)
108. Eitz, M.; Hays, J.; Alexa, M. How Do Humans Sketch Objects? *ACM Trans. Graph.* **2012**, *31*, 1–10. [\[CrossRef\]](#)
109. House Rooms Image Dataset. 2020. Available online: <https://www.kaggle.com/datasets/robinreni/house-rooms-image-dataset> (accessed on 20 June 2022).
110. Cao, Q.D.; Choe, Y. Detecting Damaged Buildings on Post-Hurricane Satellite Imagery Based on Customized Convolutional Neural Networks. *IEEE Data Port* **2018**, *2018*, e56. [\[CrossRef\]](#)
111. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2–8 December 2013.
112. Kaur, P.; ; Sikka, K.; Wang, W.; Belongie, s.; Divakaran, A. FoodX-251: A Dataset for Fine-grained Food Classification. *arXiv* **2019**, arXiv:1907.06167.
113. Song, H.O.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep Metric Learning via Lifted Structured Feature Embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
114. Leksut, J.T.; Zhao, J.; Itti, L. Learning visual variation for object recognition. *Image Vis. Comput.* **2020**, *98*, 103912. [\[CrossRef\]](#)

115. MultiClassAtari. 2023. Available online: <https://www.kaggle.com/datasets/kiranlekkala/multiclassatari> (accessed on 20 June 2022).
116. Huang, Y.; Qiu, C.; Wang, X.; Wang, S.; Yuan, K. A Compact Convolutional Neural Network for Surface Defect Inspection. *Sensors* **2020**, *20*, 1974. [[CrossRef](#)] [[PubMed](#)]
117. Shi, D.; Maggie, M.J.; Sirotenko, M. The iMaterialist Fashion Attribute Dataset. In Proceedings of the Workshop on Fine-Grained Visual Categorization at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Long Beach, CA, USA, 15–19 June 2019. Available online: <https://www.kaggle.com/competitions/imaterialist-fashion-2019-FGVC6> (accessed on 20 June 2022).
118. Li, L.J.; Fei-Fei, L. What, where and who? Classifying events by scene and object recognition. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
119. Graham, D.B.; Allinson, N. Characterizing virtual Eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications*; NATO ASI Series F, Computer and Systems Sciences (163); Springer: Berlin/Heidelberg, Germany, 1998; pp. 446–456.
120. Ahmed, M.I.; Mamun, S.; Asif, A. DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study. In Proceedings of the 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 24–25 May 2021; pp. 235–243. [[CrossRef](#)]
121. Watermarked/Not Watermarked Images. 2020. Available online: <https://www.kaggle.com/datasets/felicepollano/watermarked-not-watermarked-images> (accessed on 20 June 2022).
122. Tan, W.R.; Chan, C.S.; Aguirre, H.; Tanaka, K. Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork. *IEEE Trans. Image Process.* **2019**, *28*, 394–409. [[CrossRef](#)] [[PubMed](#)]
123. Verma, M.; Kumawat, S.; Nakashima, Y.; Raman, S. Yoga-82: A New Dataset for Fine-grained Classification of Human Poses. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 4472–4479.
124. Clothing & Models. 2018. Available online: <https://www.kaggle.com/datasets/dqmonn/zalando-store-crawl> (accessed on 20 June 2022).
125. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [[CrossRef](#)]
126. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
127. Shaha, M.; Pawar, M. Transfer Learning for Image Classification. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 656–660. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.