Pattern Analysis and Applications
https://doi.org/10.1007/s10044-023-01147-x

ORIGINAL ARTICLE

Kurcuma: a kitchen utensil recognition collection for unsupervised domain adaptation

Adrian Rosello[1] · Jose J. Valero-Mas[1] · Antonio Javier Gallego[1] · Javier Sáez-Pérez[1] · Jorge Calvo-Zaragoza[1]

Received: 30 August 2022 / Accepted: 24 January 2023

Abstract
The use of deep learning makes it possible to achieve extraordinary results in all kinds of tasks related to computer vision. However, this performance is strongly related to the availability of training data and its relationship with the distribution in the eventual application scenario. This question is of vital importance in areas such as robotics, where the targeted environment data are barely available in advance. In this context, domain adaptation (DA) techniques are especially important to building models that deal with new data for which the corresponding label is not available. To promote further research in DA techniques applied to robotics, this work presents Kurcuma (Kitchen Utensil Recognition Collection for Unsupervised doMain Adaptation), an assortment of seven datasets for the classification of kitchen utensils—a task of relevance in home-assistance robotics and a suitable showcase for DA. Along with the data, we provide a broad description of the main characteristics of the dataset, as well as a baseline using the well-known domain-adversarial training of neural networks approach. The results show the challenge posed by DA on these types of tasks, pointing to the need for new approaches in future work.

Keywords Deep learning · Domain adaptation · Robotics · Computer vision

1 Introduction

Image classification represents the task within the broad Computer Vision field that aims at automatically labeling a given image out of a set of possible categories [32]. While historically tackled with a broad range of Pattern Recognition strategies, current proposals in the area generally resort to neural models based on deep learning (DL) as they represent the state of the art in terms of recognition performance [25].

✉ Antonio Javier Gallego
jgallego@dlsi.ua.es

Adrian Rosello
arp129@alu.ua.es

Jose J. Valero-Mas
jjvalero@dlsi.ua.es

Javier Sáez-Pérez
javisaezua@gmail.com

Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es

[1] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

However, DL-based solutions are also characterized by requiring large amounts of labeled data to be adequately trained [2]. Such a constraint typically entails the need for performing prohibitively labor-expensive and error-prone labeling campaigns to collect and label high-quality data [22]. Furthermore, these models are known to remarkably tailor to the data used in the training process, hence not adequately performing when addressing—even subtly—different distributions in the prediction phase [11]. From a theoretical perspective, these limitations may be tackled by transferring the knowledge gathered from a source domain of labeled elements to a target one where data are labeled—completely or partially—or even unlabeled [24].

In response to this, the research community has proposed diverse practical mechanisms to perform such knowledge transference, being semi-supervised learning, weakly labeled learning, transfer learning, and domain adaptation some of the most representative ones [29]. Among them, one of the particular interests is that of domain adaptation (DA), which addresses the case when *source* and *target* domains deal with the exact same task but the distribution or appearance of the data differs (e.g., classifying the same type of labels but on a different dataset) [13]. This particular transference strategy

Published online: 17 February 2023

is now further discussed as it constitutes the gist of the presented work.

Attending to the amount and nature of the information in the target domain, DA proposals are further divided into three different categories [21]: (1) Supervised DA, in which the target domain is completely labeled; (2) Semi-supervised DA, in which some labels of the target space are known; and (3) Unsupervised DA (UDA) that stands for the case in which the target domain is completely unlabeled. Out of these three families, the latter category stands as a relevant, yet challenging case as it allows knowledge to be exploited without having to label the target space [9]. However, given its inherent difficulty, existing UDA methods still show limited robustness, thereby remaining open to new proposals [15].

Since its proposal, the DA field has been largely considered in the robotics area as a manner of mitigating the disparate differences commonly found between controlled training scenarios and real-life applications [33]. Some use cases are its application to unmanned aerial vehicles [20], medical and surgery tasks [35], or autonomous home assistants [36]. As it will be introduced, this latter case stands of particular relevance to the presented work since both of them frame the use of DA strategies in computer vision in the context of housework.

Considering all the above, this work expands the initial proof-of-concept effort by Sáez-Pérez et al. [28] and presents a novel collection of image data suited for UDA tasks. More precisely, we introduce the *Kurcuma set*—acronym for Kitchen Utensil Recognition Collection for Unsupervised doMain Adaptation—which stands for the largest collection of kitchen utensil image data specifically devised for UDA tasks. The goal is to provide a benchmark dataset to assess and compare novel methodological proposals. Kurcuma comprises a total of seven corpora, corresponding to 6,869 labeled images, distributed in nine different categories. Each corpus constitutes a specific and unrelated data domain, being hence of particular interest for research on robotic home-assistance tasks. In addition, we provide thorough baseline experimentation considering the state-of-the-art domain-adversarial training of neural networks (DANN) UDA method [15] that may serve as reference for future works that consider Kurcuma. Note that while kitchen utensil identification may be deemed as a very specific use case, current research works such as that by Karungaru [19]—which tackles the recognition task by resorting to transfer learning and fine-tuning strategies—state its relevance in support systems for impaired people and robot-based assistance systems, among other scenarios.

The rest of the paper is structured as follows: Section 2 contextualizes the UDA field within the broader machine learning area; Sect. 3 introduces and thoroughly describes the Kurcuma collection proposed in the work; Sect. 4 details the experimental setup together with the particular UDA strategy considered to benchmark the data collection; Sect. 5 presents and discusses the results obtained; and finally, Sect. 6 concludes the work and poses future research lines to tackle.

2 Background in domain adaptation

This section formally describes and provides the necessary background in UDA for the rest of the work.

Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the spaces of image data and associated labels, related by the underlying function $h : \mathcal{X} \rightarrow \mathcal{Y}$. Within supervised learning, this relation is typically approximated as $\hat{h}(\cdot)$ by considering a set of source data $\mathcal{S} = \left\{(x_i, y_i) : x_i \in \mathbf{X}, y_i \in \mathbf{Y})\right\}_{i=1}^{|N|}$, where $\mathbf{X} = \left\{x_1, ..., x_N\right\} \in \mathcal{X}$ represents a set of images with associated labels $\mathbf{Y} = \left\{y_1, ..., y_N\right\} \in \mathcal{Y}$ and N stands for the cardinality of the collections. Eventually, given a set of target data $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$, inference is performed by considering the estimated $\hat{h}(\cdot)$ function.

Most commonly, general learning frameworks assume that these source $\mathcal{S}$ and target $\mathcal{T}$ sets are mutually independent and identically distributed [31]. Nevertheless, this statement does not generally hold in practice since the underlying distributions in each case—namely, *domains*—typically differ, being hence additional mechanisms required to adequate or adapt the estimated $\hat{h}^S(\cdot)$ with set $\mathcal{S}$ to the data in $\mathcal{T}$ as $\hat{h}^T(\cdot)$ [27]. In this context, DA represents the research field that aims at obtaining a model based on the knowledge from a certain *source domain* that properly performs on a different, but related, *target domain* [13]. We now introduce additional notation to describe the DA field.

Formally, a data domain may be defined as the duple $\mathcal{D} = \{\mathcal{X}, P(\mathbf{X})\}$, where $P(\mathbf{X})$ represents the marginal probability of the previously considered $\mathbf{X} \in \mathcal{X}$ set of images. Additionally, let $\Gamma = \{\mathcal{Y}, P(\mathbf{Y} \mid \mathbf{X})\}$ denote the image classification task, where $P(\mathbf{Y} \mid \mathbf{X})$ represents the conditional probability distribution of the set of labels $\mathbf{Y} \in \mathcal{Y}$ on the image collection $\mathbf{X} \in \mathcal{X}$.

Given two data domains—source $\mathcal{D}^S$ and target $\mathcal{D}^T$ with their respective Γ^S and Γ^T classification tasks—the most general DA formulation assumes that the set of labels is the same in both cases ($\Gamma^S = \Gamma^T$), hence exclusively differing in their respective domain distributions, i.e., $\mathcal{D}^S \neq \mathcal{D}^T$ [34]. While this divergence may be due to the fact that these domains consider different representation spaces—namely, heterogeneous DA—this work focuses on homogeneous DA in which the feature space is shared across domains ($\mathcal{X}^S = \mathcal{X}^T$) with different data distributions, i.e., $P\left(\mathbf{X}^S\right) \neq P\left(\mathbf{X}^T\right)$ [26].

It must be noted that the availability of labeled data in the target domain may further impose additional restrictions on

Table 1 Summary of the characteristics of the seven corpora comprising the Kurcuma collection in terms of the nature of the objects and background, the number of samples per class, and their total size

Corpus	Description		Samples per class			Total size
	Objects	Background	Min	Max	Average	
EKUD	Real	Uniform	15	165	68.7 ± 51.9	618
EKUD-M1	Real	Synthetic				
EKUD-M2	Real†	Synthetic				
EKUD-M3	Real†	Uniform				
AKUD	Real	Real	84	596	186.8 ± 157.2	1681
RENDER	Synthetic	Synthetic	149	449	183.0 ± 94.1	1647
CLIPART	Synthetic	None	73	214	118.8 ± 47.5	1069

Symbol † denotes that colormaps have been artificially altered

the DA task. Among the different possibilities, the particular case of UDA represents the most challenging—as well as general—approach since labels are only provided for the source domain, being the target one fully unlabeled [9].

Within UDA, there exist three broad ways of approaching the problem. One of the most used is based on the search for characteristics common to both source and target domains—namely, domain-invariant features—so that they are not affected by domain changes. Some relevant algorithms that apply this technique are: domain-adversarial neural networks (DANN) [15], which retrieves such type of descriptors by forcing the network to resort to those that do not allow domain differentiation; visual-adversarial domain adaptation (VADA) [30], which proposes a loss function to penalize differences between the internal representations learned by the method for these domains; deep reconstruction-classification network (DRCN), which forces data from both domains to be represented in the same way by reconstructing instances using a common architecture; or domain separation network (DSN) [3], which is trained to classify input data into two subspaces—domain-specific and domain-independent—with the goal of improving the way domain-invariant features are learned.

The second family of approaches relies on the use of generative adversarial networks (GAN) to *translate* data from an initial domain to a target one [18]. Two of the most representative proposals within it are: pixel domain adaptation (PixelDA) [4], which transforms the images from the source domain to resemble those from the target domain and deep joint distribution optimal transport (DJDOT) [8], which proposes a loss function based on optimal transport theory to learn an aligned representation between the source and target domains.

The third UDA strategy focuses on transforming and aligning the internal representations learned by the methods. Two of the most representative works based on this premise are subspace alignment (SA) [12], which transforms features from one domain to another by using subspaces modeled through eigenvectors and Deep CORelation ALignment (DeepCORAL) [31], which learns a nonlinear transformation to correlate the layers' activations of both domains.

Considering all the above, this work resorts to the use of homogeneous UDA based on domain-invariant representations—more precisely, the DANN method that is presented in Sect. 4—to benchmark the Kurcuma kitchen utensil collection, thoroughly described below. In formal terms, DANN will be used to retrieve a $\hat{h}^T(\cdot)$ classification function for a target data domain $\mathcal{D}^T$ by adequately *adapting* the homologous $\hat{h}^S(\cdot)$ one obtained from a related source data distribution characterized by the $\mathcal{D}^S$ domain and the Γ^S task.

3 The Kurcuma collection for kitchen utensil recognition

This section describes the Kurcuma collection of kitchen utensil images presented in this work, which can be freely downloaded for the reproducibility of the results as well as for any other research purposes.[1] Given that this corpus has been specifically devised for research on DA tasks in the context of robotic home-assistance scenarios, the images within have been compiled and/or created to exhibit different characteristics that represent possible domain shift variations. However, given that the collection is entirely labeled, it may be further used for other tasks that fall within supervised, unsupervised or semi-supervised learning frameworks.

The collection comprises seven individual kitchen utensil corpora—four of them built by the authors—with a varying number of color images with a spatial resolution of 300×300 pixels distributed in the same nine classes for all corpora. These sets, which represent the actual data domains of the Kurcuma corpus, essentially differ in that the objects and background in each of the cases may be deemed as being either real or synthetic. A comprehensive description of these seven corpora is now facilitated, being a summary of

[1] https://www.dlsi.ua.es/~njgallego/datasets/kurcuma.

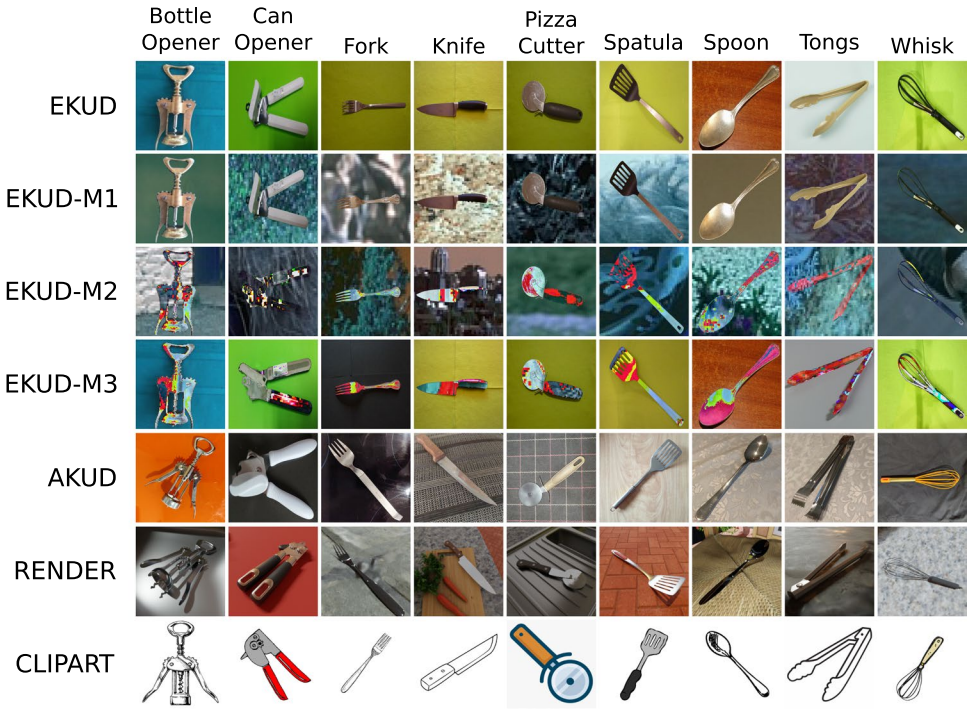


Fig. 1 Examples of the nine different classes for the different image corpora comprising the Kurcuma collection. Note that all derivatives from the EKUD set consider different random seeds to guarantee non-repeated backgrounds among them

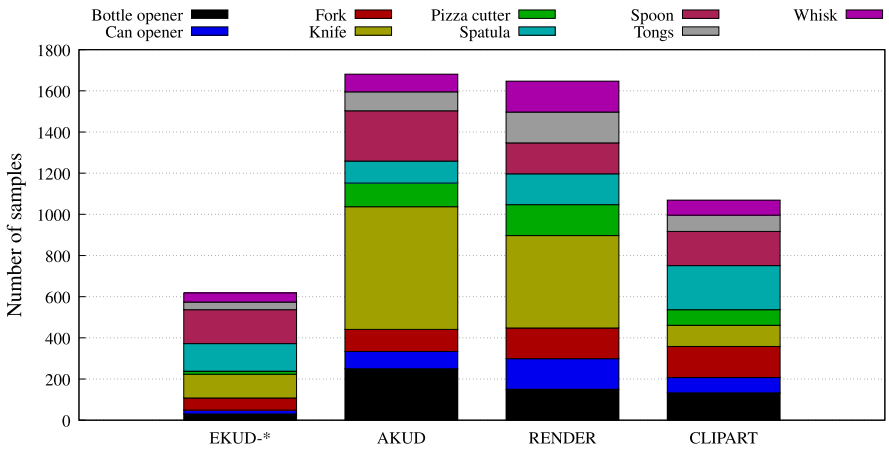
their main features provided in Table 1 together with some image examples in Fig. 1:

- Edinburgh Kitchen Utensil Database (EKUD) [16]: Corpus created to train domestic assistance robots that comprises 897 real-world pictures of utensils with uniform backgrounds originally distributed in 12 kitchen utensils with close to 75 images per class. Note that when compiled for the Kurcuma collection, a data curation process based on merging the most confusing classes, removing under-represented categories, and discarding low-quality examples was applied, hence resulting in the nine labels of the presented assortment with a total of 618 images.
- EKUD Real Color (EKUD-M1): Corpus generated by the authors combining images of EKUD with patches from the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) [1], following a similar approach to that used to develop the MNIST-M collection [14]. In this case, only the background of the EKUD images was modified, keeping the original color of the objects.
- EKUD Not Real Color (EKUD-M2): Extension to the EKUD-M1 set in which the color of the objects was altered by being mixed with those of the background patches.
- EKUD Not Real Color with Real Background (EKUD-M3): Third variation proposed by the authors in which the distortion process devised for the EKUD-M2 case is directly applied to the initial EKUD corpus.
- Alicante Kitchen Utensil Database (AKUD): Collection developed by the authors by manually taking 1480 photographs of the nine kitchen utensil categories considered for the Kurcuma collection. It depicts real objects in different real-world backgrounds covering a wide range of lighting conditions and perspectives.
- RENDER: Corpus generated by the authors by rendering synthetic images using different base public models of utensils and backgrounds from the Internet with the Blender tool.[2] To ensure variability in the data, these images were created considering different camera perspectives and illumination cases as well as a varied range of focal lengths of the virtual camera.
- CLIPART: Set of draw-like images gathered by the authors from the Internet representing each of the classes in the collection. No background is attached to the samples, showing the figure representation over a plain white background.

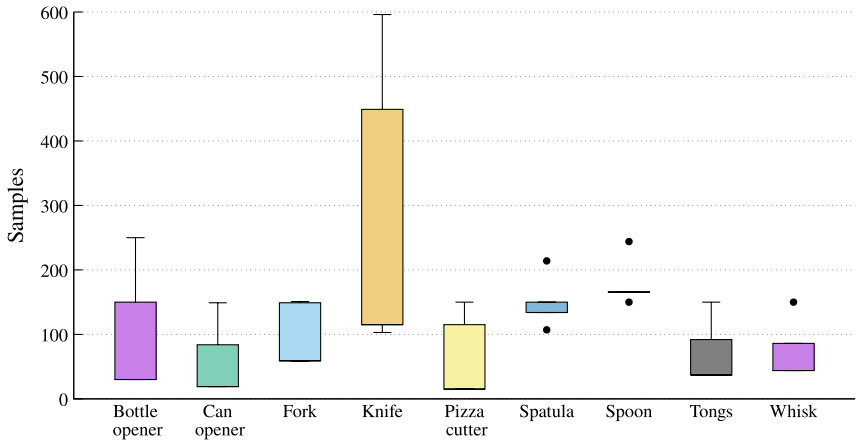
In addition to the previous explanation, Fig. 2 details the number of samples per class for each of the individual corpora comprising Kurcuma (Fig. 2a) as well as in overall terms (Fig. 2b). Note that as observed in the graphs, this collection presents a remarkable imbalance—both at the individual corpus level and at the overall assortment—among the different possible classes.

[2] https://www.blender.org.

Fig. 2 Class distribution of the Kurcuma collection



(a) Class histogram of the individual corpora of the collection among the 9 existing classes. Note that all EKUD-based sets are represented in the same column—namely, EKUD-*—as they all depict the same class distribution.



(b) Boxplot representing the overall class distribution of the collection.

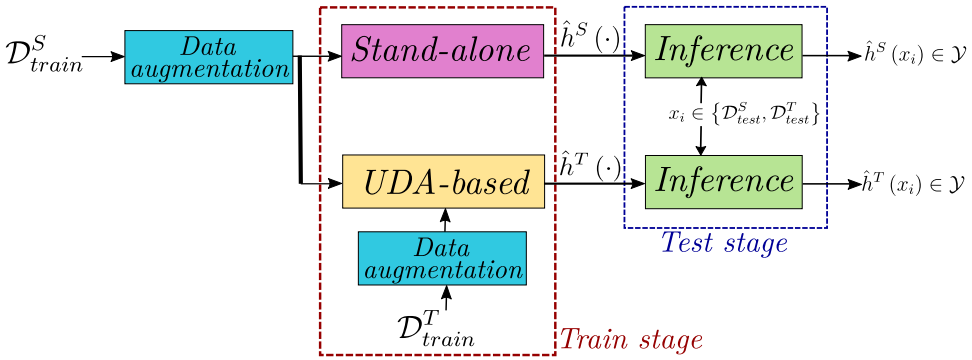


Fig. 3 Experimental procedure proposed for the benchmarking of the Kurcuma kitchen utensil image collection. A classification model is inferred and tested out of a source data domain $\mathcal{D}^S$—denoted as *Stand-alone* case—or adapted to a target data distribution $\mathcal{D}^T$—namely, *UDA-based* scenario. Both source $\mathcal{D}^S$ and target $\mathcal{D}^T$ domains contemplate non-overlapping train and test partitions for the corresponding stages

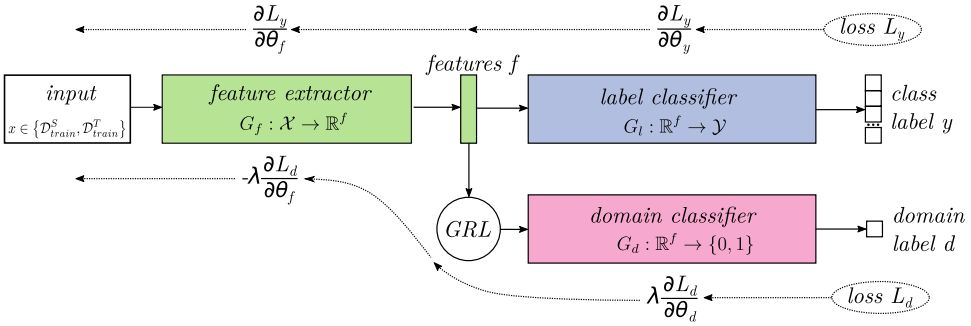


Fig. 4 Outline of the DANN architecture

4 Experimental setup

This section presents the proposed scheme to benchmark the introduced Kurcuma image collection as well as the neural classification architecture and the evaluation procedures considered for finally presenting the state-of-the-art UDA strategy contemplated, the domain-adversarial training of neural networks (DANN) [15]. Figure 3 graphically illustrates the devised procedure for easier comprehension.

As it may be observed, the proposal considers the train partition of a given source domain of data $\mathcal{D}^S_{\text{train}}$ to either infer a classification function $\hat{h}^S(\cdot)$ exclusively considering that source distribution of data—namely, *Stand-alone* case—or adapting it to an unlabeled target domain $\mathcal{D}^T_{\text{train}}$ as $\hat{h}^T(\cdot)$ using a UDA method—denoted as *UDA-based* scenario. Note that the use of data augmentation procedures is also contemplated and experimentally studied in Sect. 5 to artificially increase the amount of training data. During the inference phase, the estimated recognition functions are assessed on different test partitions $\mathcal{D}^S_{\text{test}}$ and $\mathcal{D}^T_{\text{test}}$ corresponding to the aforementioned source and target data domains involved. In addition to this one-to-one comparison, we also contemplate the case of combining several source domains—i.e., $\mathcal{D}^S_{\text{train}} = \bigcup_{i=1}^{m} \mathcal{D}^{S_i}_{\text{train}}$, where $\mathcal{D}^{S_i}$ represents the i-th source domain out of the m sets to be combined—to assess its influence in the adaptation process.

In relation to the data augmentation stage, we have considered typical transformations used for these purposes in the context of image classification tasks. More precisely, this set comprises horizontal and vertical flips, image shifts along the X- and Y-axes corresponding to the 10% of its size, shearing operations in a range of 10%, zoom transformations—both in and out with respect to the initial position—by 10% of the size of the element, and random image rotations of up to 5°.

Regarding the classification architecture, we have contemplated the particular ResNet-50 convolutional neural network by He et al. [17] pre-trained with the ImageNet corpus [10]. It must be highlighted that this model was selected as it achieved the best results in preliminary experimentation compared to other state-of-the-art convolutional classifiers as well as its large usage in other DA schemes as, for instance, DeepCORAL [31]. All neural architectures in the work were trained for 200 epochs with a batch size of 32 images considering the Stochastic Gradient Descent optimizer [23] with a Nesterov momentum parameter of 0.9, a learning rate of 10^{-2}, and a decay factor of 10^{-6}.

In terms of performance assessment, we resorted to the F-measure (F_1) figure of merit to avoid possible biases toward any particular class given the aforementioned label imbalance of the Kurcuma collection. Assuming a binary classification scenario, this metric is computed as the harmonic mean of the Precision (P) and Recall (R) indicators. These figures of merit are defined as:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} \tag{3}$$

where TP, FP, and FN, respectively, denote the True Positives or correctly classified elements, False Positives or type I errors, and False Negatives or type II errors.

Finally, due to the non-binary nature of the Kurcuma collection, we consider the use of the macro-averaged F_1 score—typically denoted as F_1^M—that extends the binary F_1 to multiclass scenarios as:

$$F_1^M = \frac{1}{|\mathcal{Y}|} \sum_{i=1}^{|\mathcal{Y}|} F_1^{(i)} \tag{4}$$

where $F_1^{(i)}$ denotes the F_1 score obtained for the i-th class assuming a one-versus-all evaluation framework.

4.1 Domain-adversarial training of neural networks (DANN)

As aforementioned, we have considered the state-of-the-art domain-adversarial training of neural networks (DANN) algorithm by Ganin et al. [15] as a representative example of UDA method. The gist of this strategy, which is illustrated in Fig. 4, is that of deriving a set of features that adequately performs the recognition task independently of the domain of the query data, i.e., domain-invariant descriptors.

Attending to this scheme, it may be observed that the DANN architecture comprises two branches that perform different classification tasks on the input image $x \in \mathcal{X}$: (1) the label classification that assigns the corresponding label $y \in \mathcal{Y}$; and (2) the domain classification part that estimates its domain, either source or target. Note that while depicting different goals, both parts share a common feature extraction stage $G_f : \mathcal{X} \rightarrow \mathbb{R}^f$ that maps the input image $x \in \mathcal{X}$ into an f-dimensional space. This representation constitutes the aforementioned domain-invariant space suitable for the $\mathcal{Y}$ set of labels that the DANN method is meant to obtain.

To achieve this goal, this architecture relies on the use of the gradient reversal layer (GRL) during the backpropagation phase of the training process. More precisely, GRL reverses the gradient of the domain classifier loss L_d and scales it with a certain $\lambda \in \mathbb{R}$ coefficient—hyperparameter to be experimentally adjusted—to weight its contribution in the overall learning process. This forces the G_f feature extractor to derive a set of domain-invariant descriptors capable of adequately addressing the label classification task disregarding the domain of the input datum.

In terms of practical models, our experiments consider the backbone of the aforementioned ResNet-50 model as the G_f feature extractor, which maps the initial image into a space of $f = 2048$ elements. Regarding the label G_l and domain G_d classifiers, both rely on the same architecture that only differs on the output space: an initial fully connected layer with 256 neurons with rectified linear unit activation and a dropout factor of 50% followed by a second fully connected layer with a *softmax* activation, respectively, depicting $|\mathcal{Y}|$ and two output neurons corresponding to the cardinality of their classification spaces.

Based on other works addressing image processing tasks with DL methods [5], images are mapped from the initial $[0, 255]^{h \times w \times c}$ representation space to a $[-1, 1]^{h \times w \times c}$ one—without any precision loss—to favor the convergence of the models as:

$$\bar{x} = \frac{x - 255}{255} \tag{5}$$

where x and $\bar{x}$ respectively represent the initial and normalized images and h, w, and c denote the height, width, and number of channels of the image.

Table 2 Average $F_1^M(\%)$ recognition rates considering the same-domain source and target sets in terms of the input image sizes (in pixels) and data augmentation scenarios

Image size ($h \times w$)	Data augmentation	
	No	Yes
64×64	47.05	54.23
128×128	67.46	75.19
224×224	70.94	79.35

In all cases, the initial images consider the original colormap ($c = 3$ channels)

5 Results

This section analyzes and benchmarks the Kurcuma collection based on the proposed experimental scheme. More precisely, we discuss three different evaluation scenarios based on the possible training alternatives in Fig. 3: a first one framed on the *Stand-alone* procedure of the training stage in which we configure the neural recognition model and examine its achieved performance disregarding any DA mechanism; a second one corresponding to the *UDA-based* case of the training phase that extends the former case by introducing the DANN strategy as a means of solving cross-domain issues; and a third case which analyses the use of multiple source domains—both disregarding and considering the DANN mechanism—to improve the overall recognition performance. In addition to these scenarios, we also analyze the effect of class imbalance in the recognition performance focusing on the most relevant scenarios out of the posed experiments. Finally, the main insights observed in these cases are summarized and discussed in the last part of this section.

Regarding data partitioning for each domain, train and test divisions were created with no overlapping between them corresponding to the 80% and 20% of the total amount of data, respectively.

5.1 Stand-alone recognition

The first scenario, as commented, performs an initial assessment of the Kurcuma collection by resorting to the ResNet-50 classification model and disregarding the use of

Table 3 Results in terms of the $F_1^M(\%)$ metric when assessing the recognition model trained on source distribution $\mathcal{D}^S$ (row) against the $\mathcal{D}^T$ target one (column)

Source (D^S)	Target (D^T)							AVG
	EKUD	EKUD-M1	EKUD-M2	EKUD-M3	AKUD	RENDER	CLIPART	
EKUD	**76.58**	56.50	45.14	66.67	39.61	22.10	37.65	44.61
EKUD-M1	61.02	**65.19**	48.01	48.52	41.14	24.36	37.59	43.44
EKUD-M2	62.15	58.02	**62.26**	59.90	38.56	36.27	41.31	49.37
EKUD-M3	67.33	44.51	50.38	**78.05**	35.69	26.68	47.01	45.27
AKUD	44.84	44.00	33.44	41.25	**89.39**	40.98	30.09	39.10
RENDER	33.79	41.04	29.42	32.59	51.94	**92.10**	38.66	37.91
CLIPART	38.04	31.92	39.85	34.36	44.16	31.37	**91.87**	36.62
AVG	51.19	46.00	41.04	47.22	41.85	30.30	38.72	–

Average results—denoted as AVG—fixing the source (row-wise) or the target domains (column-wise) are also provided. Note that these average values exclude the score obtained when the source and target spaces represent the same domain, i.e., the bold figures in the diagonal of the matrix

Table 4 Recognition rates in terms of the $F_1^M(\%)$ metric when adapting source distribution $\mathcal{D}^S$ (row) to each possible target $\mathcal{D}^T$ (column) considering the DANN method with $\lambda = 10^{-4}$

Source (D^S)	Target (D^T)							AVG
	EKUD	EKUD-M1	EKUD-M2	EKUD-M3	AKUD	RENDER	CLIPART	
EKUD		59.74	59.03	70.19	42.60	28.40	45.55	50.92
EKUD-M1	90.48		62.70	64.20	49.56	30.25	54.96	58.69
EKUD-M2	80.74	68.50		70.84	42.29	40.33	48.30	58.50
EKUD-M3	85.72	52.06	54.90		41.62	33.36	50.43	53.01
AKUD	62.00	53.66	52.03	48.98		48.74	53.25	53.11
RENDER	50.98	56.21	53.79	48.67	62.00		45.01	46.59
CLIPART	50.35	48.36	44.10	46.16	51.84	38.76		46.59
AVG	70.05	56.42	54.42	58.17	48.32	36.64	49.58	

The heatmap color indicates the absolute improvement with respect to the non-adaptive case depicted in Table 3, with darker colors representing greater improvement. Average results—denoted as AVG—fixing the source (row-wise) as well as the target domains (column-wise) are provided

any DA-based strategy. This experimentation is performed with a twofold aim: on one side, this stage serves as a preliminary study to retrieve the adequate image size for the neural model as well as to quantify the influence of the data augmentation procedure; on the other side, these experiments also provide the base performance results to compare within the forthcoming sections.

Regarding the commented preliminary parameter assessment, Table 2 shows the results in terms of the F_1^M score obtained considering different scaling factors—always maintaining their initial squared aspect ratio—as well as either including or omitting the data augmentation procedures. Note that these figures represent the average score of exclusively addressing the same-domain source and target sets.

As it may be observed, image scaling remarkably influences the overall recognition performance. More precisely, focusing on the case in which no augmentation procedure is applied, the smallest considered size ($h = w = 64$ pixels) achieves the lowest classification rate of all cases ($F_1^M = 47.05\%$) that improves up to $F_1^M = 67.46\%$ by doubling the size of the image. The case of $h = w = 224$ pixels, which matches the size used by He et al. in the ResNet-50 model [17], further improves the performance in, roughly, a 3%. Regarding the data augmentation procedures, it may be observed a steady improvement around 8% to 9% in the figure of merit, independently of the image scaling factor. According to this analysis, the rest of the experiments consider the configuration that maximizes the recognition performance: squared images with $h = w = 224$ pixels including data augmentation.

Considering the obtained set of parameters, Table 3 provides the base recognition results obtained for all possible pairs of source and target domains in the Kurcuma collection. As aforementioned, these figures represent the base performance that may be achieved when tackling the possible cross-domain scenarios disregarding any DA strategy.

Attending to these figures, there exists a remarkable gap between the cases in which source and target distributions depict the same domain and those in which the domains differ. More precisely, the recognition performance in the former case varies in the $[62.26\%, 92.10\%]$ range with an average value of 79.35%, whereas in the latter case, this range decreases to $[22.10\%, 67.33\%]$, with a mean score of 42.33%. Such a fact suggests that despite sharing the same

label space, the stand-alone neural model is unable to obtain an adequate set of features capable of addressing the underlying data distributions across the different domains in the Kurcuma collection. In response to this, the use of DA-based mechanisms is expected to tackle this limitation since, in principle, it should be able to infer a set of descriptors that perform well on both source and target spaces.

5.2 Single-source UDA-based training

This second scenario introduces the DANN architecture in the experimental pipeline to address the limitations of the former scheme: one single source domain that is assessed on a target one. As commented in Sect. 4.1, this method requires the computation of a gradient score—done at the GRL layer—scaled by a $\lambda \in \mathbb{R}$ coefficient that must be experimentally tuned. In our case, preliminary testing studied values in the range $\lambda \in \left[10^{-4}, 1\right]$ providing the best performance when $\lambda = 10^{-4}$.

Table 4 details the results obtained when considering the DANN method with a weight parameter of $\lambda = 10^{-4}$ for all possible pairs of source and target domains of the Kurcuma collection. Note that no values are provided when both domains match as the mechanism is disregarded in those scenarios.

The inclusion of the DANN method in the pipeline considerably changes the results obtained in the first experimental scenario. As it may be observed, all cross-domain cases show a consistent improvement in their recognition rates in the range [3%, 30%]. Some particular examples of this statement are the case of adapting EKUD-M1 (source domain, $\mathcal{D}^S$) to EKUD (target domain, $\mathcal{D}^T$), which shows a boost of 29.46% by going from an initial score of 61.02% in the former case to a 90.48% in the latter one or the case of EKUD ($\mathcal{D}^S$) to AKUD ($\mathcal{D}^T$), which improves the performance in just a 3% (recognition rates from the non-adaptive and DANN-based schemes were 39.61% and 42.60%, respectively).

Overall, the use of the DANN mechanism achieves an average performance score of $F_1^M = 53.37\%$, which remarkably improves that of $F_1^M = 42.33\%$ obtained in the previous non-adaptive scenario in an 11%.

Finally, while the scheme does prove the relevance of the adaptation stage, still the recognition rates may be deemed as far from being practical. In this regard, we consider that the use of additional mechanisms that adjust their performance in terms of the domain similarity [6] may be contemplated to further improve the performance of the scheme.

5.3 Multiple-source UDA-based training

The last scenario poses the case in which we may consider the combination of several source domains of data to address a single target one. Such an experiment is based on the premise that in the context of DA, a larger and more varied collection of source data should better represent the underlying distribution $\mathcal{X} \times \mathcal{Y}$, hence increasing the resemblance of source $\hat{h}^S(\cdot)$ and target $\hat{h}^T(\cdot)$ recognition functions.

Given the number of possible combinations of source and target domains when addressing the Kurcuma collection, we focus on a specific use case that may be deemed as the most relevant in terms of a practical application: those scenarios in which both source and target distributions depict real elements. In this regard, we exclusively consider the adaptation either from AKUD to EKUD or from EKUD to AKUD, being the rest of the domains—those including synthetic parts—exclusively considered as additional data sources. Table 5 shows the results obtained in these two scenarios with the possible additional source domains both when considering and disregarding the DANN adaptation method. Note that the AKUD-to-EKUD case does not contemplate the EKUD-derived synthetic corpora as additional sources of data to avoid including any information related to the target during the training phase of the scheme.

Focusing on the AKUD-to-EKUD scenario, it may be observed that the inclusion of additional domains in the training stage does remarkably boost performance. More

Table 5 Average $F_1^M(\%)$ recognition rates for the contemplated source-to-target adaptation scenarios of real data considering the use of additional source domains

Scenario ($\mathcal{D}^S$ - $\mathcal{D}^T$)	Additional source(s)†	DANN adaptation	
		No	Yes
AKUD - EKUD	None	44.84	62.00
	C	56.82	69.18
	R	**61.42**	**69.73**
	C + R	59.63	65.81
EKUD - EKUD	–	76.58	
EKUD - AKUD	None	39.61	42.60
	C	50.36	55.19
	R	57.00	63.66
	C + R	58.59	67.00
	E	49.07	52.58
	E + C	51.39	55.92
	E + R	60.55	64.44
	E + C + R	**61.74**	**71.95**
AKUD - AKUD	–	89.39	

The figures provided depict the cases in which the DANN adaptation method is both used and disregarded from the experimentation. Elements highlighted in bold represent the best-performing cases for each data scenario and DA configuration. Same-domain recognition rates (already presented in Table 3) are provided for reference purposes.

†For compactness, letters C, R, and E, respectively, denote the CLIPART, RENDER, and all EKUD-M* sets

Table 6 Class-wise recognition results in terms of the P, R, and F_1 figures of merit for the two best-performing multiple-source UDA cases—AKUD+R→EKUD and EKUD+E+C+R→AKUD. Recognition figures for same-domain scenarios—namely, EKUD→EKUD and AKUD→AKUD—are provided for completeness in the analysis

Source ($\mathcal{D}^S$)	AKUD (+R)			EKUD			EKUD (+E+C+R)			AKUD		
	↓			↓			↓			↓		
Target ($\mathcal{D}^T$)	EKUD			EKUD			AKUD			AKUD		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Class label												
Bottle opener	47.3	62.4	53.8	55.5	59.2	57.3	67.6	57.0	61.8	89.3	84.0	86.6
Can opener	45.0	61.9	52.1	55.0	57.9	56.4	51.6	56.3	53.8	86.2	87.2	86.7
Fork	91.4	76.7	83.4	92.3	87.7	89.9	79.2	85.4	82.1	89.9	90.9	90.4
Knife	71.1	86.3	78.0	87.5	91.3	89.4	83.6	77.3	80.4	94.3	95.5	94.9
Pizza cutter	74.6	76.7	75.6	90.0	89.4	89.7	68.6	73.9	71.1	90.3	90.3	90.3
Spatula	72.7	68.2	70.4	85.9	73.8	79.4	76.9	89.0	82.5	89.3	90.5	89.9
Spoon	88.5	69.7	78.0	90.9	89.9	90.4	77.4	83.7	80.4	90.7	91.8	91.3
Tongs	59.9	61.1	60.5	58.6	61.4	60.0	68.8	61.1	64.7	86.4	87.4	86.9
Whisk	74.0	77.8	75.8	75.8	77.8	76.8	70.6	70.6	70.6	87.9	87.1	87.5
Average	69.7	64.5	69.7	76.4	78.7	76.6	67.1	69.9	71.9	91.8	92.0	89.4

Average results are also included for reference purposes

precisely, this procedure reports an increase in terms of the F_1^M metric between 7 and 17% depending on the additional domains and the use of the DANN architecture. The best results are achieved when exclusively adding the RENDER collection to the source data with figures of $F_1^M = 61.42\%$ and $F_1^M = 69.73\%$ when omitting and including the DANN architecture, respectively. This fact states that the CLIPART set is inadequate for this scenario, possibly due to being remarkably different from the target data distribution.

The EKUD-to-AKUD case shows similar behavior to the previous one as the use of additional sources of information reports an increase in the performance of up to 30%. In contrast to the other scenario, and disregarding the use of the EKUD-based collections, the joint use of the CLIPART and RENDER sets achieves better performance rates than their stand-alone use. The inclusion of the EKUD-based corpora further improves the results obtained, being the best performance achieved when considering all additional sources of information—case denoted as $E+C+R$ in the results—with performance scores of $F_1^M = 61.74\%$ and $F_1^M = 71.95\%$ respectively obtained for the non-adaptive and DANN-based schemes.

Finally, comparing these results to the same-domain values in each case, it may be observed certain performance gap to fill. In this regard, while the adaptation strategy and the use of additional domains do prove to benefit the recognition rates, some additional mechanisms—e.g., the ones already commented about domain similarity—should be studied to narrow this difference or even surpass these reference values.

5.4 Class imbalance analysis

The previous scenarios have studied the capabilities of the DANN adaptation strategy in the context of kitchen utensil recognition based on the presented Kurcuma assortment. However, given the remarkable class imbalance that this particular collection presents (see Sect. 3), this section extends the previous analyses to obtain some additional insights related to this particularity. In addition to the macro-averaged F_1^M score previously assessed, we now additionally consider the Precision (P) and Recall (R) (see Eqs. 1 and 2) metrics as well as the F_1 (see Eq. 3) at the class level to obtain additional conclusions about the performance of the scheme.

Given the large number of experiments in the work, we selected a set of representative cases to perform this study. More precisely, we focus on the two best-performing configurations of the multiple-source UDA scenario (Sect. 5.3): (1) adapting from AKUD to EKUD considering the RENDER set as additional source domain—henceforth, AKUD+R→EKUD—; and (2) adapting from EKUD to AKUD considering all EKUD-related sets, CLIPART, and RENDER as additional source domains—subsequently referred to as EKUD+E+C+R→AKUD.

Table 6 provides the results obtained for the two considered cases. Note that same-domain recognition rates—EKUD→EKUD and AKUD→AKUD—are also provided to facilitate the comparison with the reference classification rates that may be achieved for the respective target domains when no DA process is required.

Attending to the results obtained, it may be observed that the model does not exhibit the same recognition performance for all categories. However, since there is no clear trend in these figures, we now analyze these results focusing on the particular target domain ($\mathcal{D}^T$) of the scenario.

Focusing on the $\mathcal{D}^T$ = EKUD case, it may be observed that when both the source and target collections are gathered from the same domain—*i.e.,* $\mathcal{D}^S = \mathcal{D}^T$ = EKUD—, the recognition rates achieved for the *Bottle opener*, *Can opener*, and *Tongs* categories are noticeably lower than those for the rest of the labels (particularly, the *P* score are remarkably low, suggesting the presence of a large number of FP errors). This is possibly due to the fact that these three classes constitute the most underrepresented ones of the particular set (see Fig. 2a), hence highlighting the difficulties of the model to deal with class-imbalance cases.

Regarding the DA-oriented AKUD+R→EKUD case, it may be observed that, as reported in other experiments in the work, the recognition rates generally decrease with respect to those of the same-domain scenario. However, the provided class-wise analysis shows that the least recognition rates are achieved by the same categories as in the non-DA scenario, i.e., the *Bottle opener*, *Can opener*, and *Tongs* labels. Such a fact suggests that the DA scheme is able to adapt the features from the source collection (AKUD with RENDER) to the target domain (EKUD), but is not able to cope with the imbalanced label distribution of the assortment.

In relation to the $\mathcal{D}^T$ = AKUD case, the figures obtained show a rather steady recognition rate for all labels in which all categories show a remarkably competitive performance. Note that while the AKUD set also shows a remarkable class imbalance, no category shows a very scarce amount of examples, being hence possible for the neural model to distinguish among the different labels within the collection.

Focusing on the EKUD+E+C+R→AKUD case, it may be observed that, similar to the adaptive and non-adaptive cases with $\mathcal{D}^T$ = EKUD, there is a performance decrease compared to the same-domain scenario. While this decrease constitutes an expected behavior, results show that the least performing categories match those in the previous experiment, i.e., the *Bottle opener*, *Can opener*, and *Tongs* labels. This effect suggests that the imbalanced EKUD set used as the source domain biases the model to favor the most represented classes and that the DANN mechanism with the AKUD target domain is not able to compensate for that issue.

The presented analysis proves that the class imbalance constitutes one of the main limitations of the considered DA approach as it generally biases the model toward the recognition of the majority class. In this regard, we consider the future exploration of manners to palliate this issue as, for instance, the work by Cui et al. [7] that compensates the class imbalance during the learning stage by introducing the effective number of samples per class as part of the model loss.

5.5 Discussion

Once presented and analyzed the different experimental scenarios, we provide below an additional discussion about the different insights and conclusions obtained, as well as the limitations observed.

Given the proneness of DL models to overfit the distribution of the training data, recognition rates generally result in a decrease in the performance when addressing a target corpus that differs from the source one. DA mechanisms have been typically considered to deal with this issue in general image classification tasks. To contribute to this research area, this work introduces the Kurcuma corpus of kitchen utensil recognition for robotic home-assistance tasks that presents 9 different data domains. Note that this corpus depicts an imbalanced class distribution to resemble as much as possible to a real-world scenario in which there may not exist an equal number of examples for all categories of the utensils to be differentiated.

The results obtained prove that the particular UDA strategy studied in the work—the DANN mechanism—is able to cope with the performance drops observed when addressing cross-domain data collections. This conclusion is of particular relevance considering the end-user scenario of robotic assistants as the recognition model would be able to adapt to any instance of kitchen utensil that has not been used for training the model. Moreover, in the absence of real train data, UDA stands as an alternative to alleviate a possible cold-start problem by training the model with a synthetic assortment (e.g., the RENDER one) and adapting it to an unlabeled set of real elements.

In relation to the use of additional source domains, the results show that such a configuration yields more competitive recognition rates than the use of a single one. While this observation may be deemed as expected, it must be highlighted that the best performance is not always achieved when considering the highest number of source collections since some combination may hinder the overall recognition rate of the model (e.g., the AKUD collection benefits from the use of the RENDER one as additional source set but is negatively affected when the CLIPART assortment is contemplated as well).

Finally, the underlying class imbalance of the Kurcuma set highlights the main limitation of the approach. As it has been studied, the unequal class distribution of the collection hinders the learning process as the model tends to bias toward the majority cases. Moreover, while the DANN mechanism succeeds in achieving a domain-invariant feature space, the issues related to the imbalanced distribution of the classes are not adequately managed by the model. Such

a disadvantage needs to be further studied to achieve robust recognition models that may address general scenarios disregarding this imbalance data limitation.

6 Conclusions

Within the Computer Vision field, deep learning (DL) strategies are generally deemed to achieve state-of-the-art performances when addressing recognition tasks. Nevertheless, DL-based schemes are generally constrained by their large requirements as regards the amount of annotated data to learn from as well as being remarkably tailored to the underlying distribution of the training elements. Among the different strategies in the scientific literature, the so-called domain adaptation (DA) paradigm stands as a promising framework for tackling these issues. Such strategies work on the basis that a model trained on a source domain could be adjusted to transfer its knowledge to a new and different target domain, as long as it shares the same set of categories.

This work focuses on the particular case of kitchen utensil recognition for robotic home-assistance tasks. More precisely, we present a novel collection of image data suited for research on DA tasks: the *Kurcuma set*—acronym for Kitchen Utensil Recognition Collection for Unsupervised doMain Adaptation. This collection contains seven different corpora, corresponding to a total of 6,869 labeled images, distributed in nine different categories. Each corpus constitutes a particular and unrelated data domain that might be either synthetic or real. To our best knowledge, this corpus constitutes the largest set of kitchen utensil images specifically devised to assess and gauge DA methods. Furthermore, we provide a reference benchmark of the collection by evaluating the inter- and cross-domain recognition rates attained by state-of-the-art models without any adaptation, as well as thorough experimentation considering the state-of-the-art domain-adversarial training of neural networks DA method.

Future work considers extending this study to other recent DA proposals such as visual-adversarial domain adaptation, deep correlation alignment or deep joint distribution optimal transport. In addition, we also aim at studying different mechanisms to address the issues that the class imbalance in the Kurcuma collection introduces as well as considering other adaptive methodologies based on the similarity of the domains involved in the process. Finally, the last point to address is the case of under-resourced classes in which there might be a very scarce amount of examples to perform the adaptation task.

Author Contributions Conceptualization was done by all authors; methodology was done by JJV-M and AJG; formal analysis and investigation were done by AR, JS-P and JC-Z; writing—original draft preparation were done by AR, JJV-M and JS-P; writing—review and editing were done by AJG and JC-Z; funding acquisition was done by JC-Z and JJV-M; resources were done by AJG and JC-Z; and supervision was done by AJG and JC-Z

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was supported by the I+D+i project TED2021-132103A-I00 (DOREMI), funded by MCIN/AEI/10.13039/501100011033. Some of the computing resources were provided by the Generalitat Valenciana and the European Union through the FEDER funding program (IDIFEDER/2020/003). The second author is supported by grant APOSTD/2020/256 from "Programa I+D+i de la Generalitat Valenciana".

Data availability The Kurcuma dataset is available for research purposes at https://www.dlsi.ua.es/~jgallego/datasets/kurcuma.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Arbelaez P, Maire M, Fowlkes C et al (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
2. Bolte JA, Kamp M, Breuer A, et al (2019) Unsupervised domain adaptation to improve image segmentation quality both in the source and target domain. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops
3. Bousmalis K, Trigeorgis G, Silberman N et al (2016) Domain separation networks. Adv Neural Inf Process Syst 29:343–351
4. Bousmalis K, Silberman N, Dohan D, et al (2017) Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3722–3731
5. Castellanos FJ, Gallego AJ, Calvo-Zaragoza J (2020) Automatic scale estimation for music score images. Expert Syst Appl 158(113):590
6. Castellanos FJ, Gallego AJ, Calvo-Zaragoza J (2021) Unsupervised neural domain adaptation for document image binarization. Pattern Recogn 119(108):099
7. Cui Y, Jia M, Lin TY, et al (2019) Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9268–9277
8. Damodaran BB, Kellenberger B, Flamary R, et al (2018) Deepjdot: deep joint distribution optimal transport for unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision, pp 447–463

9. Das D, Lee CG (2018) Sample-to-sample correspondence for unsupervised domain adaptation. Eng Appl Artif Intell 73:80–91
10. Deng J, Dong W, Socher R, et al (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 248–255
11. Fang T, Lu N, Niu G et al (2020) Rethinking importance weighting for deep learning under distribution shift. Adv Neural Inf Process Syst 33:11996–12007
12. Fernando B, Habrard A, Sebban M, et al (2013) Unsupervised visual domain adaptation using subspace alignment. In: IEEE International Conference on Computer Vision, pp 2960–2967
13. Gallego AJ, Calvo-Zaragoza J, Fisher RB (2021) Incremental unsupervised domain-adversarial training of neural networks. IEEE Trans Neural Netw Learn Syst 32(11):4864–4878
14. Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp 1180–1189
15. Ganin Y, Ustinova E, Ajakan H et al (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096
16. Goel A, Fisher RB (2016) Classification of kitchen cutlery using a visual recognition algorithm. Tech rep, University of Edinburgh
17. He K, Zhang X, Ren S, et al (2016) Identity mappings in deep residual networks. In: European Conference on Computer Vision, pp 630–645
18. Huang SW, Lin CT, Chen SP, et al (2018) Auggan: cross domain adaptation with gan-based data augmentation. In: Proceedings of the European Conference on Computer Vision, pp 718–731
19. Karungaru S (2019) Kitchen utensils recognition using fine tuning and transfer learning. In: Proceedings of the 3rd International Conference on Video and Image Processing, pp 19–22
20. Kishida I, Chen H, Baba M, et al (2021) Object recognition with continual open set domain adaptation for home robot. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 1517–1526
21. Kouw WM, Loog M (2021) A review of domain adaptation without target labels. IEEE Trans Pattern Anal Mach Intell 43(3):766–785
22. Le T, Nguyen T, Ho N, et al (2021) Lamda: label matching deep domain adaptation. In: International Conference on Machine Learning, pp 6043–6054
23. Mitchell TM (1997) Machine learning, vol 1. McGraw-hill, New York
24. Murez Z, Kolouri S, Kriegman D, et al (2018) Image to image translation for domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4500–4509
25. O'Mahony N, Campbell S, Carvalho A, et al (2019) Deep learning vs. traditional computer vision. In: Science and Information Conference, Springer, pp 128–144
26. Patel VM, Gopalan R, Li R et al (2015) Visual domain adaptation: a survey of recent advances. IEEE Signal Process Mag 32(3):53–69
27. Ramponi A, Plank B (2020) Neural unsupervised domain adaptation in NLP—A survey. In: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, pp 6838–6855
28. Sáez-Pérez J, Gallego AJ, Valero-Mas JJ, et al (2022) Domain adaptation in robotics: a study case on kitchen utensil recognition. In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, pp 366–377
29. Sener O, Song HO, Saxena A, et al (2016) Learning transferrable representations for unsupervised domain adaptation. Adv Neural Inf Process Syst 29
30. Shu R, Bui H, Narui H, et al (2018) A DIRT-t approach to unsupervised domain adaptation. In: International Conference on Learning Representations
31. Sun B, Saenko K (2016) Deep CORAL: correlation alignment for deep domain adaptation. In: European Conference on Computer Vision, pp 443–450
32. Szeliski R (2010) Computer vision: algorithms and applications. Springer Science & Business Media
33. Tahmoresnezhad J, Hashemi S (2017) Visual domain adaptation via transfer feature learning. Knowl Inf Syst 50(2):585–605
34. Wang M, Deng W (2018) Deep visual domain adaptation: a survey. Neurocomputing 312:135–153
35. Xu M, Islam M, Lim CM, et al (2021) Learning domain adaptation with model calibration for surgical report generation in robotic surgery. In: IEEE International Conference on Robotics and Automation, pp 12350–12356
36. Ye J, Fu C, Zheng G, et al (2022) Unsupervised domain adaptation for nighttime aerial tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8896–8905