Towards Universal Image Embeddings: A Large-Scale Dataset and Challenge for Generic Image Representations

Nikolaos-Antonios Ypsilantis[1] Kaifeng Chen[2] Bingyi Cao[2] Mário Lipovský[2]
Pelin Dogan-Schönberger[2] Grzegorz Makosa[2] Boris Bluntschli[2]
Mojtaba Seyedhosseini[2] Ondřej Chum[1] André Araujo[2]

[1]VRG, FEE, Czech Technical University in Prague [2]Google

Abstract

*Fine-grained and instance-level recognition methods are commonly trained and evaluated on specific domains, in a model per domain scenario. Such an approach, however, is impractical in real large-scale applications. In this work, we address the problem of **universal image embedding**, where a single universal model is trained and used in multiple domains. First, we leverage existing domain-specific datasets to carefully construct a new large-scale public benchmark for the evaluation of universal image embeddings, with 241k query images, 1.4M index images and 2.8M training images across 8 different domains and 349k classes. We define suitable metrics, training and evaluation protocols to foster future research in this area. Second, we provide a comprehensive experimental evaluation on the new dataset, demonstrating that existing approaches and simplistic extensions lead to worse performance than an assembly of models trained for each domain separately. Finally, we conducted a public research competition on this topic, leveraging industrial datasets, which attracted the participation of more than 1k teams worldwide. This exercise generated many interesting research ideas and findings which we present in detail. Project webpage:* https://cmp.felk.cvut.cz/univ_emb/

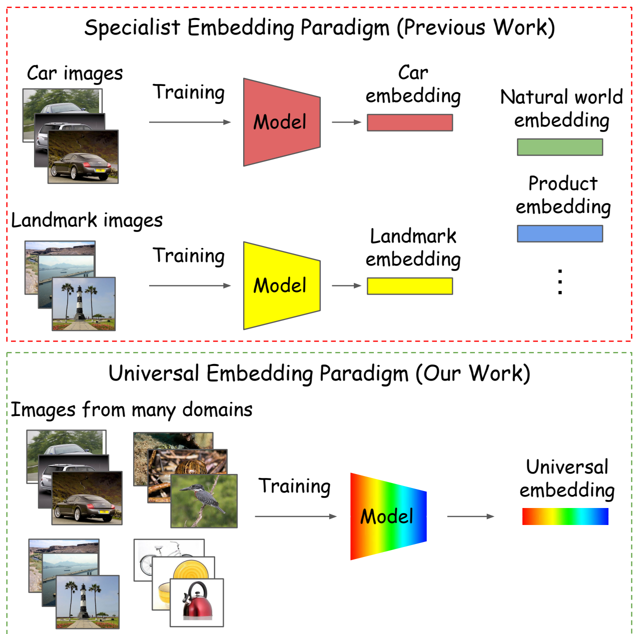

Figure 1: Previous work (top) on learning image embeddings has mainly focused on representations that are specialized for narrow domains, such as cars, landmarks, natural world, products, among others. While this may lead to high performance, it cannot scale to meet demands of modern general purpose visual search systems, which are required to identify objects in many domains. In this work, we consider the problem of learning universal image embeddings (bottom), which are representations that can encode fine-grained visual information about multiple domains. We propose the first large-scale dataset for research on universal image embeddings and additionally present results from a public industrial challenge in this area.

1. Introduction

The past decade has witnessed significant progress in image representations that are capable of discriminating objects at a fine-grained or instance level. Several techniques [54, 49, 58, 19] have been demonstrated to learn such image embeddings when given data from a specific domain, for example images of different birds or images of different landmarks. Recently, there has been growing interest in general purpose visual search systems that can identify objects from many domains [70, 7, 9]. The use of per-domain models in general-purpose systems is very expensive and generally impractical, since a large number of models would need to be developed and maintained. The holy grail for this kind of application is a unified model that can discriminate fine-grained objects across several domains, which we refer

*Correspondence: ypsilnik@fel.cvut.cz, francischen@google.com.

to as an *universal image embedding*, as per Fig. 1. Such an universal embedding is a challenging goal as different domains provide different visual cues that are essential for fine-grained and instance-level recognition. It can be seen as an evolution of generalization in training: generalization beyond training instances in classification with fixed classes, beyond training classes within one domain in an open set problem (such as landmark retrieval), and finally, beyond training classes in multiple domains in the universal embedding problem. We believe that the field of image embedding learning needs to continue moving forward by considering universal representations as a critical direction of future work.

The main reason holding back research explorations on universal embeddings is the lack of a standard, large-scale dataset – only small datasets have so far been proposed [63, 53], or related medium ones that have been constructed with different objectives [14]. Today, there are no established strategies to train such models, and their effectiveness on industrial applications is not well-studied either. We set out to bridge this gap by introducing the following contributions.

Contributions. (1) The first large-scale dataset for research on universal image embeddings, referred to as Universal Embedding Dataset (UnED). The dataset contains more than $4M$ images from $349k$ classes in 8 different domains, representing diverse & real use cases: food, cars, online products, clothing, natural world, artworks, landmarks and retail products. We leverage already-existing public datasets to construct UnED, carefully combining them into a common format, with standard splits and metrics. (2) A comprehensive benchmarking and reference implementations of models for research in this area, highlighting that specialized models on average outperform universal models trained with simple strategies; nevertheless, the universal models achieve promising results and pave the way for further improvements. (3) The first public competition in this area, the Google Universal Image Embedding Challenge[1], focusing on industrial applications, which attracted more than $1k$ researchers and $21k$ submissions in total. We report learned lessons from this challenge, which helped open up new research directions.

2. Related Work

Image embedding research. Traditionally, academic research on image embedding learning has been conducted with a focus on models which are specialized for a given domain, *i.e*., a specific object type (*e.g*., birds, cars, landmarks, etc). Generally, researchers propose embedding learning techniques which are applied to different domains separately, rather than developing (universal) embedding models which could be applied to all domains combined. There are three main computer vision sub-communities working in this area, and we review their work in the following paragraphs.

(1) *Deep Metric Learning* – generally focused on the domains of cars [35], products [58] and birds [62]. Recent papers focus on improved benchmark methodology [44], leveraging intra-batch relations [56], enhanced sampling [37] and integrating language guidance [52].

(2) *Instance-level Retrieval* – generally focused on the domain of landmarks [64, 50]. Recent work reports improvements to models [18, 67] and re-ranking strategies [60, 36]. A recent survey can be found in [19].

(3) *Person Recognition/Re-Identification* – focusing on person-related data such as face [33, 41] or full-body [38, 74]. Recent research introduces quality-adaptive margins [34], joint optimization of data/architecture/loss [71], cross-domain learning [72] and improved pre-training [75].

Dataset	Year	# Images	# Domains	# Classes
INSTRE [63]	2015	$28k$	3	250
GPR1200 [53]	2021	$12k$	6	$1.2k$
MRT [14]	2022	$267k$	6	$23k$
UnED (ours)	2023	**4.1M**	**8**	**349k**

Table 1: Comparison of our dataset against existing ones. Our dataset is significantly larger, with one order of magnitude more images and classes.

Universal embedding datasets. To the best of our knowledge, no truly large-scale datasets for unified embedding model evaluation exist. Tab. 1 compares our new dataset against the three existing related datasets we are aware of. INSTRE [63] contains $1k$ query images and $27k$ index images of 250 classes, covering 3 domains: landmarks, planar objects and other daily objects. More similar in spirit to our work, GPR1200 [53] introduces an evaluation set containing $12k$ images in total (from $1.2k$ classes), constructed by collecting images from existing public collections in 6 domains: landmarks, sketches, natural world, products, planar objects and faces. The recent MRT [14] focuses on adapting pre-trained models using unlabeled data from 6 different domains, also reusing images from existing collections: aircrafts, cars, birds, flowers, food and products. In its standard setup, MRT's training set discards class labels to address how well models are able to adapt in the absence of supervision, but the same splits could potentially be reused in a supervised setup. MRT's evaluation is performed for each domain separately.

Our newly-introduced benchmark differs substantially from these three, most notably on the scale aspect, comprising images from 8 domains: food, cars, online products, clothing, natural world, artworks, landmarks and retail products. With $241,986$ query images, $1,397,126$ index images and $2,831,222$ training images, we provide $15\times$ the number of images and $15\times$ the number of classes compared to previ-

[1] https://www.kaggle.com/competitions/google-universal-image-embedding

Dataset	Domain	Train images	Train classes
Food2k [42]	food	472,349	900
CARS196 [35]	cars	6,346	78
SOP [58]	online products	48,942	9,054
InShop [40]	clothing	20,897	3,198
iNaturalist (2018) [61]	natural world	273,929	4,552
Met [68]	artworks	397,121	224,408
GLDv2 [64]	landmarks	1,422,914	73,182
Rp2k [48]	retail products	188,724	1,074
Total	All	2,831,222	316,446

Table 2: The different domains the UnED subsets span, along with the training splits and the corresponding training classes.

ous datasets[2]. Differently from INSTRE and GPR1200, we provide a large training set, enabling researchers to directly compare different models under the same standard training data. In contrast to MRT, we provide labels in the training set to allow models to better adapt to the domains of interest – in our experience, this setting is more common in practice. Additionally, we consider the more challenging evaluation setup where the index contains images from all domains, requiring the model to distinguish images from multiple domains simultaneously. Finally, we would like to emphasize that the public research competition we conducted was the first to assess performance of universal image embeddings; previous embedding learning challenges focused only on per-domain evaluations [4, 2, 3, 1].

Universal embedding techniques have not been thoroughly investigated in previous work. An early attempt by Feng *et al*. [25] used data from five domains in a distillation approach to combine the knowledge of specialized models, three at a time, into a universal one. Our effort differs from theirs as we introduce a much larger dataset where a significant number of domains must be jointly considered. A more recent method relevant to this problem is Grappa [14], which aims to adapt a pre-trained model using unlabeled data from several domains combined – their setup is different from ours since we consider supervision to be available for the different domains in our task.

Other relevant literature. [16] propose to learn a unified embedding for recognition at different levels of granularity; models are trained on ImageNet with both classification and ranking losses, and testing is done both on ImageNet and instance-level retrieval datasets. [29] propose metric learning knowledge transfer between datasets, but those are within the same broad modality of person recognition. The OmniBenchmark dataset [73] aims to unify image representation efforts for classification tasks, gathering $1M$ images from $7k$ classes and 21 domains. Evaluation in OmniBenchmark is per-domain, with linear classifier probing.

[2]In the context of this work, the term class is used interchangeably for either fine-grained categories or instances, since the proposed benchmark consists of fine categories at levels of granularity which make the most sense for each domain.

3. The Universal Embedding Dataset

The Universal Embedding Dataset (UnED) contains images that come from several publicly available datasets and benchmarks used to evaluate the performance in various tasks, including image retrieval, instance-level and fine-grained recognition. The new dataset covers popular visual domains listed in Table 2. These domains were selected to simulate the environment of universal recognition application, and each of them already has its own commercial applications [6, 5, 12, 11, 10, 8]. Visual cues used for fine-grained or instance level recognition in the selected domains differ substantially, which pronounces the need for a universal embedding approach and makes the dataset challenging. At the same time, none of the domains addresses or benefits from methods exploiting any type of biometric information (such as faces), that would allow for human identification.

3.1. Datasets and splits details

Each of the datasets comes with predefined training and testing splits, some of them, but not all, also provide a validation set. For the sake of fair comparison in the future, for each of the domains (datasets) we define train, validation and test sets. These sets do not necessarily exactly correspond to the original splits. The specific statistics for the training splits are given in Table 2, while those for the validation and test splits are given in Table 3. The train and validation sets are used for training and hyperparameter tuning, while the test sets are used for evaluation. The test and validation sets are further divided into two subsets. The *index set* contains database images to be retrieved, while the *query set* are images that serve to query by example, details are summarized in Table 3. The index and query set splits in validation are provided in order to allow a retrieval validation to have a better proxy for the final task. For some domains, the query and index sets are identical. This is indicated by the $\star$ symbol in Table 3. In such cases, the query itself is always excluded from the retrieved results.

The **food domain** is represented by the recently published Food2k dataset [42], the largest to date fine-grained food classification dataset. New training and testing splits are defined so that there are disjoint train and test classes. Further, for the test classes, the image instances are subsampled (including near-duplicate removal) to have no more than 10 images per class. Such a reduction is conducted to keep the dataset challenging and to avoid saturated scores (as many embeddings would obtain high perfomance in the presence of near-duplicated images). For the **car domain**, the Stanford Cars [35] (known as CARS196) is used. This dataset is widely used for the task of metric learning and image retrieval. The standard training set is split into training and validation sets (78 and 20 categories respectively), while the original test set of 98 categories is preserved. The Stanford Online Products dataset [58] is leveraged as the **online prod-**

Dataset	Val split: query images	query classes	index images	index classes	unseen classes	Test split: query images	query classes	index images	index classes	unseen classes
Food2k	49,323	100	⋆	⋆	✓	9,979	1,000	⋆	⋆	✓
CARS196	1,708	20	⋆	⋆	✓	8,131	98	⋆	⋆	✓
SOP	10,609	2,264	⋆	⋆	✓	60,502	11,316	⋆	⋆	✓
InShop	4,985	799	⋆	⋆	✓	14,218	3,985	12,612	3,985	✓
iNaturalist (2018)	51,917	1,138	⋆	⋆	✓	136,093	2,452	⋆	⋆	✓
Met	129	111	38,307	33,501	✗	1,003	734	397,121	224,408	✗
GLDv2	157,556	8,131	⋆	⋆	✓	1129	318	761,757	101,302	✓†
Rp2k	17,185	120	⋆	⋆	✓	10,931	1,186	⋆	⋆	✓
Total	293,412	12,683	331,590	46,073	-	241,986	21,089	1,397,126	345,747	-

Table 3: Query and index subsets for the validation and test sets of each subset of UnED. If index images are missing for a specific subset, it means that for this dataset queries are used as index as well. ⋆ means that the index set statistic matches the corresponding query set statistic for the same dataset split. † means that some classes are seen during training, but not all of them.

uct domain. The original training set is split into training and validation sets, the test set remains as defined in [58]. For the **domain of clothing**, the InShop retrieval part of the DeepFashion dataset [40] is used. The original training set is divided into train and validation, while the test set is kept identical, including the separate query and index sets. The **natural world domain** is represented by the large-scale iNaturalist dataset (2018 version) [61]. The retrieval splits of [17] are adopted, the training set is further divided into train and validation. The **artwork domain** is covered by the Met dataset [68]. In this domain, the test classes are a subset of the train ones. The training set of catalog images of artworks is used as an index set, while queries are given as user photographs taken in the exhibition by mobile devices. For the Met dataset, all the original splits are preserved. The GLDv2 Dataset [64] serves as a source for the **landmark domain**. The train-clean version of the dataset is used to provide the train and the validation sets. The public and private queries are merged to define the query part of the test set, while the original index set is used as the index part of the test set. The last **domain of retail products** is given by the Rp2k dataset [48], originally designed for image classification. Similarly to the Food2k dataset, new splits are created to have disjoint training and test classes. The test classes are also reduced to up to 10 instances per class.

3.2. Evaluation protocol

Each image is described by a 64-dimensional embedding. Low dimensionality is a crucial factor for practical large-scale applications, which is a natural target for fine-grained recognition of many classes from a number of different domains. The index contains embeddings of index images from **all** domains. It is not allowed to exploit the information about the query and/or result domains. The evaluation is performed by Euclidean-distance retrieval between the query embedding and the embeddings of images in the index.

The embedding of the query is compared against the embeddings of the merged index set, producing a ranked list of images. The first metric used to quantify the quality of this ranked list is the commonly used Recall@1 ($R@1$), which is equivalent to the nearest neighbour accuracy. For a given query, it only takes into account the predicted top neighbor, being equal to 1 if it comes from the same class as the query and 0 if it doesn't, and finally it is averaged across all queries Q. Mathematically, it is defined as:

$$R@1 = \frac{1}{Q}\sum_{q=1}^{Q} rel_q(1), \tag{1}$$

where $rel_q(j)$ denotes the relevance of image at rank j for query q (binary indicator). Additionally, the precision of the top-5 neighbors list is calculated, *i.e.* how many of the 5 neighbors come from the same class. If the number of index images of the same class (n_q) as the query are less than 5, the precision at n_q is calculated instead. This metric averaged across all queries Q is called modified Mean Precision at 5 (mMP@5) and has the following definition:

$$mMP@5 = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{\min(n_q,5)}\sum_{j=1}^{\min(n_q,5)} rel_q(j), \tag{2}$$

where j is the index of the neighbors of q, sorted in descending order by their similarity. In the case of a query that has multiple classes assigned to it (as for some GLDv2 queries), each of them is considered a correct prediction.

Let us also highlight that, in contrast to some image embedding benchmarks [50, 64, 44], we do not include mean Average Precision (mAP) as one of our core evaluation metrics. We find that mAP has many drawbacks, for example, capturing differences in scores even for changes in sorting of low-ranked positions that in practice do not matter. Besides, mAP is unintuitive, being difficult to interpret exactly what a given value means (the AP meaning changes for each query, depending on the number of expected results for each). For these reasons, we find it more suitable in our case to rely on simple and practical metrics such as the above, which are easily interpretable and capture well the desired system behavior: rank relevant images high, and focus mainly on the very top positions. Despite this, recognizing that the community still relies on mAP in many cases, we include results using it in the Supplementary Material.

4. Benchmarking

In this section we describe the baseline approaches that are evaluated on the proposed benchmark, in order to offer a

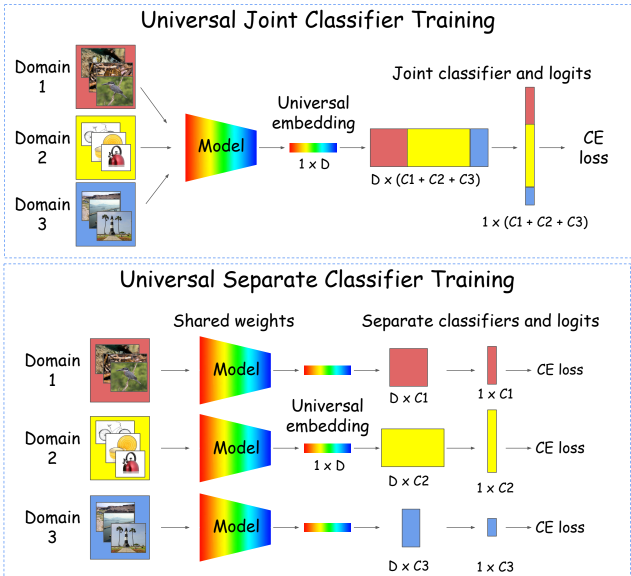

Figure 2: Universal embedding training methods used as baselines in this work. In joint classifier training (top), a single classifier head on top of the backbone is trained on all combined classes. In separate classifier training (bottom), classifier heads for different domains are trained separately, given a shared backbone. For the sake of compactness, only 3 of the 8 domains are presented.

testbed for future comparisons.

4.1. Baseline approaches

Pretrained models. First, different pretrained models are evaluated by extracting off-the-shelf embeddings. For this evaluation, the original dimensionality is used. More specifically, we benchmark standard ImageNet AugReg pretraining (IN) [59], image-text foundation model OpenAI CLIP [51] and recent DINOv2 [45] that has shown to produce very strong generic features, all using a ViT-B/16 backbone (768-D). We also benchmark MultiGrain [16] embeddings from a ResNet50 backbone (2048-D), as they have been trained for a relevant task. For the ones that utilise the ViT backbone, the [CLS] token of the last layer is used as the embedding; for the Multigrain, the pooled representation before the classifier. In both cases they undergo ℓ_2 normalization. For the IN and CLIP models which are later used as initialization for finetuning, we perform PCA-Whitening [32] to additionally reduce the dimensionality to 64-D, trained on the union of a subset of random training images from each domain.

Training on UnED. The ViT-B/16 [24] backbone initialized with either IN or CLIP is further finetuned on the UnED training set. In particular, the [CLS] token of the last layer is ℓ_2 normalized and then projected to 64-D using a (trainable) linear layer. The 64-D embedding is ℓ_2 normalized again, as is common in end-to-end image search architectures that include a projection layer [28], forming the embedding used in the search. For learning the embedding, we use the Softmax Cross Entropy loss (CE) on top of linear layer with no bias and ℓ_2 normalized rows (Normalized Softmax Loss [69]), which is a commonly used classification based objective in the metric learning literature.

Given that the ultimate goal is to achieve (or even overcome) specialist performance with only one universal embedding, we first train one model on each domain (specialists), in order to get an estimate of the specialist performance that can be achieved on that domain. Then, we train the universal model on all domains at the same time to examine how far direct generalizations of the specialist training methods are from achieving specialist performance in each specific domain.

Specialist embedding training. For each domain, a specialist model is trained using only training samples from the particular domain. The validation set of that domain is used in order to prevent overfitting by early stopping at the epoch that maximizes validation $R@1$.

Universal embedding training. Universal models are trained on the union of the training sets of the domains, with the total number of training classes being equal to the sum of the training classes of the different domains. The validation set during universal training consists of the union of the validation sets across all domains, *i.e.* the index set corresponds to the merged index sets and the query set corresponds to the merged query sets. It is used to perform early stopping at the epoch that maximizes the balanced average $R@1$ across all domains. We choose this way of performing validation when training universal models as it matches the final evaluation.

We examine two different approaches for the final classification layer of the universal model, *i.e.* a Joint (common) classifier for all classes of the UnED training set, or a Separate classifier for each domain; both are visualized in an example in Figure 2. For the latter, we take into account the domain that the training sample comes from, and only forward it through the corresponding classifier to produce the loss.

When training on multiple domains, the sampling strategy of the domains has to be taken into account, as imbalances are inevitable. The model is trained with batches that contain samples from only one domain at a time, perform an optimisation step after every batch, and we examine the following schemes: (i) sampling domains with probability that is proportional to their frequency in the training set (Dataset Size (DS) sampling), (ii) sampling each next batch in a Round-Robin manner (RR sampling), in a cyclic order, resulting in a balanced sampling and (iii), following [39], sampling according to the steps needed for the corresponding Specialist to reach maximum performance in its domain (Specialist Steps (SS) sampling).

Implementation details. We use the following standard metric learning training augmentations: resizing the image

	Food2k		CARS196		SOP		InShop		iNat		Met		GLDv2		Rp2k		Mean	
Model	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1
Off-the-shelf																		
IN (**768-D**)	31.1	44.1	41.4	54.1	43.7	65.6	35.5	53.9	67.1	74.2	21.1	30.8	14.8	25.2	52.9	74.3	38.4	52.8
CLIP (**768-D**)	29.4	42.9	74.7	82.2	44.2	65.4	37.2	56.0	52.4	61.9	21.4	28.5	20.4	31.0	38.6	59.9	39.8	53.5
DINOv2 (**768-D**)	39.9	51.4	67.1	79.5	35.6	56.0	17.4	33.4	71.2	77.6	38.3	48.1	35.4	51.7	46.6	67.8	43.9	58.2
MultiGrain (**2048-D**)	14.8	24.2	27.1	39.7	37.7	59.5	19.8	34.3	34.9	43.4	12.0	16.1	8.6	16.7	51.5	73.3	25.8	38.4
IN+PCAw	19.1	29.1	29.0	37.8	30.5	51.2	19.6	31.6	50.9	57.9	8.0	11.0	8.3	13.2	37.6	57.8	25.4	36.2
CLIP+PCAw	23.4	34.6	62.8	72.7	36.5	57.0	27.0	41.8	42.7	51.1	12.1	15.8	11.9	17.6	32.0	51.8	31.0	42.8
Specialists																		
IN+Oracle	49.9±0.5	62.8±0.5	61.9±0.6	71.8±0.3	60.9±0.5	78.1±0.3	66.3±0.2	85.9±0.2	**70.1±0.1**	**75.2±0.1**	20.4±1.0	24.9±0.7	31.2±0.5	43.1±0.5	73.6±0.2	**87.1±0.1**	54.9±0.4	66.6±0.3
CLIP+Oracle	**51.5±0.3**	**63.7±0.2**	**83.4±0.8**	**88.5±0.3**	65.8±0.3	81.2±0.2	68.0±0.3	86.2±0.3	67.3±0.1	73.0±0.2	**27.6±1.4**	**32.9±1.5**	**35.1±0.7**	**46.6±0.4**	69.7±0.2	84.4±0.2	**59.6±0.3**	**70.4±0.2**
Universal models																		
IN+UJCDS	49.1±1.7	60.6±1.2	50.9±1.7	60.9±1.9	56.9±1.4	74.4±0.9	60.3±2.8	78.0±2.1	67.6±0.7	72.5±0.8	3.8±0.3	4.6±0.4	29.3±1.2	42.0±1.3	71.6±2.3	85.1±1.4	48.7±0.3	59.8±0.3
CLIP+UJCDS	50.9±0.6	62.2±0.7	77.4±1.1	83.6±0.9	59.1±0.3	75.8±0.2	62.6±0.7	80.3±0.6	64.2±0.1	70.0±0.1	2.5±0.1	3.3±0.3	32.8±0.4	44.7±0.6	69.6±0.4	83.8±0.3	52.4±0.3	63.0±0.2
IN+UJCRR	48.6±0.2	60.3±0.3	62.9±0.2	71.3±0.4	64.7±0.3	80.2±0.3	74.0±0.4	89.9±0.2	68.3±0.1	73.3±0.1	5.5±0.2	7.0±0.6	21.1±0.2	31.6±0.3	**74.1±0.3**	86.8±0.1	52.4±0.1	62.6±0.1
CLIP+UJCRR	50.1±0.1	62.0±0.2	80.0±0.2	85.4±0.7	**68.6±0.2**	**82.7±0.1**	**77.0±0.3**	**91.1±0.1**	63.7±0.2	69.5±0.3	4.6±0.5	5.8±0.7	25.5±0.3	36.0±0.4	70.1±0.3	84.1±0.2	55.0±0.1	64.6±0.1
IN+USCRR	48.3±1.3	60.9±0.9	58.9±1.3	69.7±1.2	61.9±0.7	78.7±0.4	70.4±0.7	88.3±0.5	69.1±0.1	74.2±0.1	7.3±0.7	9.7±0.9	21.3±1.0	31.4±1.6	**74.1±0.4**	**87.1±0.4**	51.4±0.2	62.5±0.3
CLIP+USCRR	49.5±0.5	61.4±0.3	79.0±0.8	84.9±0.8	65.6±0.3	81.1±0.1	73.1±0.1	89.4±0.1	64.4±0.6	70.5±0.5	8.6±0.2	10.8±0.1	25.3±0.3	36.5±0.2	71.1±0.8	85.1±0.4	54.6±0.4	64.9±0.3
IN+USCSS	49.0±0.2	61.7±0.2	53.4±2.5	64.3±2.2	62.0±0.5	78.8±0.3	67.6±0.1	87.2±0.1	68.3±0.4	73.5±0.3	8.4±1.2	10.7±1.7	28.0±0.2	40.6±0.1	73.5±0.5	**87.1±0.4**	51.3±0.3	63.0±0.1
CLIP+USCSS	49.8±0.7	62.0±0.7	76.4±2.0	83.4±1.4	65.8±1.1	81.3±0.6	71.0±1.2	88.5±0.9	65.3±1.0	71.4±0.8	9.9±1.8	12.7±2.0	31.5±1.3	42.8±1.9	70.1±0.9	84.8±0.5	55.0±0.9	65.9±0.8

Table 4: Model evaluation on UnED test set, all results for 64-D (unless stated otherwise) ℓ_2 normalized descriptors. PCAw : Projection to 64-D by PCA-Whitening learned on a subset of the UnED training set. UJCDS: Universal Joint Classifier Dataset Size sampling, UJCRR: Universal Joint Classifier Round Robin sampling, USCRR: Universal Separate Classifier Round Robin sampling, USCSS: Universal Separate Classifier Specialist Steps sampling, For 64-D embeddings, we highlight with: Blue: best unified model for that domain, **Bold**: best for that domain across all baselines. The evaluation is averaged across (i) queries of each domain separately and (ii) across all domains, *i.e.* balanced average ("Mean" column) of the UnED test set. Note that all queries are compared against the merged index set that contains all domains.

to 256×256 followed by random cropping to 224×224, random horizontal flipping with probability of 0.5 and normalization with pretraining statistics of the backbone. During test time, we resize to 224×224 and normalize with pretraining statistics. The projection and the classification layers are randomly initialized. Optimization uses the AdamW algorithm with a batch size of 128 images; the scale parameter of the Normalized Softmax loss is kept fixed at the value of 16 and we use the following learning rate schedule: for the first 2 epochs, only the classifier is trained with a learning rate of $1e-3$, while the embedding is kept fixed, and weight decay is introduced with a value of $1e-6$. Then, everything is trained until 30 epochs are reached, with the backbone being updated with a learning rate of $1e-5$. Those hyperparameters are common to all experiments, for both specialists and universal models, chosen by performing the best on average across all domains. All the experiments (unless deterministic) are run for 3 different seeds; the mean and standard deviation of each experiment is reported. Training and validation for a universal embedding model takes approximately 16 hours in a Google Cloud TPU v4, using the Scenic framework [21].

4.2. Experimental results and discussion

The experimental results are summarized in Table 4, where we report performance of each baseline both across queries of each domain, as well as the balanced average of all domains ("Mean" column). We discuss our findings.

Different pretrainings. For the dimensionality of 768-D produced by the ViT-B backbone, DINOv2 is the best performing pretraining method compared to CLIP and IN. We can partially attribute this to DINOv2 having also been trained on parts of our training set (see [45] for details of DINOv2 pretraining data). MultiGrain, that employs a CNN ResNet50 backbone, underperforms the others, despite the much higher dimensionality. PCA-Whitening further harms performance; in the Supplementary Material we also compare it with a random linear projection to 64-D, showing that it performs on par with PCA-Whitening. Overall, even the much higher dimensional off-the-shelf embeddings underperform our finetuned 64-D embeddings.

Oracle Specialist embedding. In this setting, 8 models were trained, one for each domain. The domain of the query image at test time is used, so the corresponding specialist of that domain is used to extract its embedding, as well as the embeddings for the entire merged index set. The best average performance is achieved, but it only consitutes an unrealistic setting, since Oracle is used to select the correct specialist.

Universal embedding models. The baseline techniques of universal embedding training examined in this work are direct extensions of specialist training methods. During training no expert knowledge was exploited (*e.g.* the Met dataset would clearly benefit from strong geometric augmentations). Still, the final performance is close to the Specialists performance, or even surpasses the Specialists in the SOP and InShop domains. This is remarkable given that 8 times less parameters are used compared to the corresponding Specialist Oracle, and also that no knowledge of the test time domain is utilized.

We observe the following: (i) The universal models reach the same performance of validation retrieval metrics (R@1 and mMP@5) on most domains faster than the corresponding specialist (in terms of total optimization steps performed for training samples of that domain); we attribute this to the sharing of useful features across domains, (ii) Different domains overfit at different rates, as observed on the validation retrieval metrics; this has also been reported in [25], (iii) Different domains benefit differently from the pretraining method, *i.e.* CARS196 benefits a lot more from CLIP pretraining,

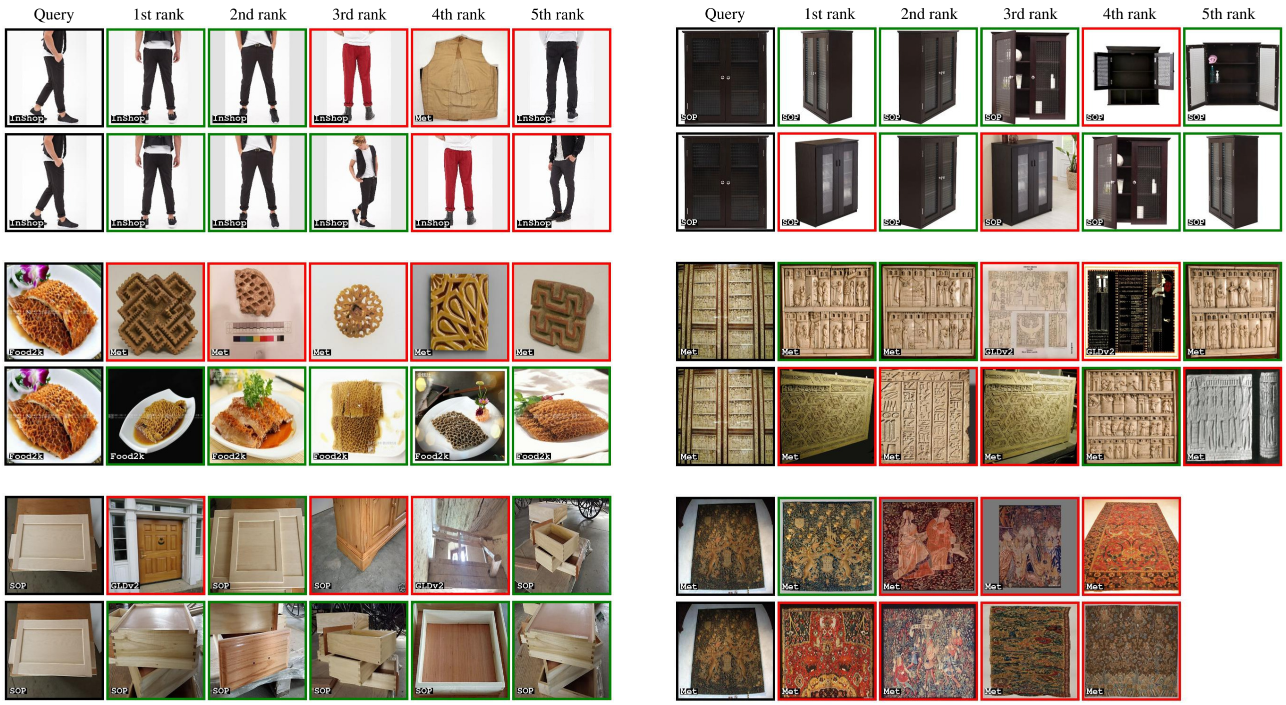

Figure 3: **Retrieval results for example queries.** Each column contains 3 queries, for each query the top row corresponds to nearest neighbors from the Specialist CLIP+Oracle embedding, bottom row from the universal model CLIP+USCSS. A correct neighbor (coming from the same class) is indicated by green border; otherwise by red. The number of neighbors shown equals the number of index images that come from the same class as the query; only those are taken into account for the computation of the mMP@5 metric. The domain that the image comes from is shown on each image. On the left column, examples of cross-domain failure for the Specialist CLIP+Oracle model are shown; on the right examples of failure for the universal model are shown, indicating degraded fine-grained capabilities.

while iNat domain benefits more from IN pretraining. (iv) For a given sampling scheme (RR), the different classifiers (Joint - J and Separate - S) produce different results across domains. For example, the SOP and InShop domains benefit the most when trained using the joint classifier. On the other hand, the Met domain benefits by the use of a separate classifier; however this specific domain suffers from a very low performance compared to the specialist. Preliminary experiments on pairs of datasets revealed that the combination of GLDv2 and Met makes the training difficult for the Met domain. (v) Regarding sampling strategies, sampling based on dataset size performs the worst on average, while the RR methods that balance the domains improve average performance consistently. Sampling according to the number of steps that the specialist maximizes its performance (SS) performs the best on average, however it produces the highest standard deviation across seeds, as the number of steps each specialist reaches maximum performance at is also dependent on the seed itself.

Qualitative examples for the comparison of retrieval between the CLIP+USCSS model and the Specialist CLIP+Oracle are shown in Figure 3. We observe the cross-domain failure for Specialist CLIP+Oracle model, as shown in some examples in the left column. This can be attributed to the universal model seeing all of the domains at train time, while the specialist model fails to handle images that are out of its train domain distribution. Also, the universal model shows degraded instance-level/fine-grained discrimination capabilities, as shown in the examples in the right column. This can be attributed to the universal model having to utilize the same capacity to learn all the domains that a specialist utilizes for one.

5. The Universal Embedding Challenge

Motivated by the strong need of having a single universal embedding for various industrial applications, complementary to the proposed Universal Embedding Dataset, we conducted the Google Universal Image Embedding Challenge in Kaggle. This competition stimulated research interests in developing ideas and methods in training universal image representations, which we introduce in detail in this section.

Challenge dataset. For the evaluation dataset for the challenge, instead of using the one presented in this paper, we introduced a separate benchmark composed of a query set with $5k$ images and an index set with $200k$ images. This dataset covered 11 image domains, including: apparel & accessories, packaged goods, landmarks, furniture & home decor, storefronts, dishes, artwork, toys, memes, illustrations

Model	Team	Techniques	mMP@5 (private split)
Top 1	cuilab.ai [57]	CLIP ViT-H/14 $\rightarrow$ Multi-domain data + multi-stage finetuning	72.8
Top 2	Xiao [30]	CLIP ViT-H/14 $\rightarrow$ Multi-domain data + stratified learning rates	70.9
Top 3	- - - [15]	CLIP ViT-H/14 $\rightarrow$ Multi-domain data + dropout rate ensembling	69.2
Top 4	Ivan & Simjeg & CLIP-Art [20]	CLIP ViT-H/14 + CLIP ViT-L/14 $\rightarrow$ Multi-domain data + model size ensembling	68.8
Top 5	NS embedding [46]	CLIP ViT-H/14 $\rightarrow$ Multi-domain data + adding image heuristics (height, width)	68.8
Top 6	IRonCLIP [27]	CLIP ViT-H/14 $\rightarrow$ Multi-domain data + multi-stage finetuning + Test time augmentation (TTA)	68.5
CLIP ViT-H/14	-	Reference model **(1024-D)**	62.1

Table 5: Summary of techniques used in the top 6 winning solutions in the Google Universal Image Embedding Challenge and their mMP@5 scores on the private split in the challenge. We also include the score of the pre-trained model (CLIP ViT-H/14) in the last row, as a reference.

and cars – which are considered of significant importance for industrial applications. These query and index images were collected and annotated with fine-grained and instance-level labels by human annotators. The distribution of the domains of query images was disclosed to participants in the challenge, who could tune their methods with this information. The queries were split into two subsets: 30% were used for scoring models while the competition was running, which gave feedback to participants on the quality of their submissions. The rest 70% queries were kept separate and only used for the final scoring, once the competition closed for new submissions. Given that this dataset is collected for industrial applications, it is not possible to release the raw images to participants.

In terms of training data, a crucial obstacle in industrial applications is on how to select the most relevant images for training. To create a similar setup, in this challenge, we didn't provide any specific guidance on the training datasets to use: selecting the right datasets was one of the challenges that participants had to face.

We would like to highlight important differences between the public dataset proposed in this paper and the dataset used in the challenge. Firstly, in the challenge, we avoid using any publicly available datasets to prevent overfitting. Secondly, the challenge dataset is collected for industrial applications, and one of the goals is to verify that techniques that work on this dataset can also be applicable to the public datasets proposed in this paper. Thirdly, in many industrial applications, the training dataset is ambiguous and only the distribution of evaluation is known. This challenge is to mimic this setting to encourage novel ideas under these scenarios.

Challenge setup. Similar to the setup discussed in this paper, the challenge focused on image retrieval task using 64-D image features, with the mMP@5 metric defined in Equation 2. The model evaluations were conducted through a scoring notebook hosted on our servers which ran on GPU P100 chips with 16G memory. During submission, participants were asked to upload their models trained with either PyTorch [47] or Tensorflow [13]. Based on the uploaded model, the scoring notebook would run feature extraction and metric computation on the evaluation dataset. We also set a loose runtime limit of 9 hours for the scoring notebook to provide flexibility of the model size and use this as an incentive for researchers to explore different model architectures.

5.1. Challenge results and findings

We ran the Universal Embedding Challenge for three months. At the end, this challenge attracted around $1k$ teams with in total $21k$ model submissions. We summarize the techniques used in the top 6 solutions in Table 5. All teams used a pre-trained CLIP model [51] as the backbone. To meet the feature dimension requirement of the challenge, shallow projection layers were added on top of the pre-trained model to produce 64-D embeddings. The models were then finetuned on multi-domain datasets using standard augmentation techniques and supervised by classification tasks with ArcFace loss [23]. This training recipe as a result showed better performance than directly applying PCA to the output features from the pre-trained model, as presented by a detailed study from the Top 4 team [20]. In the following, we highlight several key findings and discuss them in detail.

Improved pre-training via image-text foundation models. All the winning models first initialized from and then finetuned the image backbone (ViT-H/14 or ViT-L/14 [24]) of the CLIP model pre-trained on the LAION2B dataset [55]. These were some of the first explorations leveraging image-text foundation models as a central building block for complex image retrieval models (previous work [66, 31] had started investigating this direction). The success of adopting such foundation models here indicates that rich detailed information contained in text can be beneficial to vision-only models that need to be sensitive to fine-grained details. In contrast, most previous embedding models had been pre-trained on datasets with coarse categories, such as ImageNet [22]. Moreover, these foundation models are usually trained with large-scale datasets which can naturally contain many domains – this is another reason that makes these models suitable for generic representation learning. Our exploration of CLIP models presented in Section 4 is largely motivated by this observation.

Improved fine-tuning. In Table 5, we present the evaluations of the finetuned models provided by the top teams and the CLIP ViT-H/14 model in the last row, as a reference. By comparing the mMP@5 scores, we show that this pre-trained model, though having a larger feature dimension of 1024, obtains inferior performance by up to 10.7%, which demonstrates the effectiveness of finetuning. For the finetuning

	Food2k		CARS196		SOP		InShop		iNat		Met		GLDv2		Rp2k		Mean	
Model	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1	mMP@5	R@1
Top 1	41.2	53.0	96.2	97.3	54.7	70.7	52.7	78.5	**68.7**	**74.9**	**45.0**	**55.1**	**42.3**	**53.7**	83.3	91.4	60.5	71.8
Top 2	**46.4**	**57.3**	97.1	97.7	**65.1**	**79.9**	**66.4**	**87.1**	61.2	68.2	39.7	48.8	37.6	49.5	**87.1**	**93.4**	**62.6**	**72.7**
Top 3	38.2	49.9	93.7	95.2	53.7	70.0	50.2	75.5	61.5	68.1	18.6	23.8	39.8	52.4	76.7	87.8	54.0	65.3
Top 4	37.3	48.8	92.2	95.6	49.6	67.1	46.5	72.6	63.0	69.5	40.1	51.4	35.3	48.4	78.9	89.4	55.4	67.9
Top 5	41.5	54.3	96.1	97.2	54.9	71.7	51.1	76.5	67.5	73.8	45.5	56.2	34.7	47.4	73.4	87.1	58.1	70.5
Top 6	38.8	49.8	**97.5**	**97.8**	58.8	74.7	60.2	82.8	57.3	64.4	25.9	31.5	35.4	46.4	83.3	91.7	57.1	67.4

Table 6: Retrieval metrics on the test set of the Universal Embedding Dataset for the top 6 solutions of the Google Universal Image Embedding Challenge. Note that in obtaining these metrics, we followed the same steps from the challenge submissions to preprocess the test images, which are not necessarily the same as the ones used for the models in this paper.

techniques used in the top solutions, we found that treating the backbone and the shallow projection and classifier layers separately, by either making finetuning multi-stage or using different learning rates, is very necessary. For instance in the Top 2 solution [30], the learning rates for the 64-D projection layer and the classifiers are set to be 10^3 times larger than that of the backbone, and with this stratified learning rate setup, the final mMP@5 score is significantly increased to 70.9 compared to 62.1 of the pre-trained model. This training strategy also makes intuitive sense. Firstly, the extra projection layer and classifier layers are randomly initialized during finetuning, thus requiring either higher learning rate or a "warm up" stage in order to properly train the weights. Secondly, given that the initial backbone weights are already well-trained, using large learning rates might cause undesired overfitting and destroy its generic knowledge. This aligns with recent trend that using frozen or slightly finetuned [43, 26, 14] pre-trained models can improve the performance for different tasks. Our experiments presented in this paper also adopt the same strategy. Furthermore, model soup [65] and model ensembling are also experimented by several teams. In particular, ensembling features trained with various dropout rates or with different backbone sizes proved to help improve the models' performance.

Training set selection. To properly finetune the pre-trained model towards the challenge's evaluation, selecting the right set of training data is crucial. We notice that participants explored a variety of datasets in different ways, and the procedure for data selection is designed to match the distribution of the query set. Given multiple datasets containing images from different domains, we observe the Top 1 team [57] used greedy search algorithm for datasets and only keep the ones that help the performance. The Top 2 team [30] conducted very detailed investigation on data balancing and mixing, and they found that filtering out rare classes and capping the maximum number of images per class were helpful. The Top 6 team [27] leveraged the pre-trained CLIP model to generate labels for noisy datasets.

5.2. Top challenge solutions on our benchmark

To better understand the performance of the challenge winning solutions on the benchmarks proposed in this paper, in Table 6 we present the evaluation of the top 6 solutions on the proposed test sets. These models on average outperform the baseline presented in this paper since they are 1) much larger in size, 2) initialized from a model pre-trained on larger-scale datasets, and 3) trained with more involved finetuning procedures. We note that the second place solution in fact achieves the highest average performance in our benchmarks. In domains such as CARS196, Met, GLDv2 and Rp2k, the challenge solutions outperform our specialist models and the universal models. However, in domains such as Food2k, SOP, and Inshop, our trained models perform better. This is because these domains are not the majority of the challenge evaluation dataset and therefore the submitted models were either not explicitly trained on the related datasets or not optimized toward these domains. By comparing the numbers in Tables 5 and 6, we also conclude that the evaluation between the challenge dataset and the dataset proprosed in this paper are correlated given that the top 2 winners in the challenge also have the highest metrics in our benchmarks.

6. Conclusions

In this work, the novel large-scale UnED dataset for training and evaluating Universal Embedding models was introduced in order to stimulate research in the area. A comprehensive benchmarking was performed and reference models for future comparison were implemented and discussed. The universal training baselines introduced revealed some of the challenges as well as the benefits of learning the representation for multiple domains simultaneously. Finally, a public challenge on universal embeddings was conducted, techniques exploited by the top ranked teams were discussed and the winning methods were evaluated on the proposed benchmark. The proposed splits will be released on the project website, while the implemented baseline methods, as well as the evaluation scripts, will be released under two frameworks (Scenic [21] and PyTorch [47]). We expect that the metric learning field will significantly progress by focusing on learning universal image representations.

7. Acknowledgements

This research was supported by Research Center for Informatics (project CZ.02.1.01/0.0/0.0/16 019/0000765 funded by OP VVV) and the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/173/OHK3/3T/13.

References

[1] AliProducts Challenge: Large-scale Product Recognition. https://tianchi.aliyun.com/competition/entrance/231780/introduction?lang=en-us, 2020. 3

[2] DeepFashion2 Challenge. https://sites.google.com/corp/view/cvcreative2020/deepfashion2, 2020. 3

[3] eBay eProduct Visual Search Challenge. https://eval.ai/web/challenges/challenge-page/888/leaderboard/2352, 2021. 3

[4] Google Landmark Retrieval Challenge 2021. https://www.kaggle.com/c/landmark-retrieval-2021, 2021. 3

[5] Amazon app camera search. *Available online:* https://www.amazon.com/b?ie=UTF8&node=17387598011, 2023. 3

[6] Art recognition. *Available online:* https://art-recognition.com/, 2023. 3

[7] Bixby vision. *Available online:* https://www.samsung.com/global/galaxy/apps/bixby/vision/, 2023. 1

[8] Car recognition. *Available online:* https://play.google.com/store/apps/details?id=com.asdev.carmake, 2023. 3

[9] Google Lens. *Available online:* https://lens.google/, 2023. 1

[10] Landmark detection. *Available online:* https://cloud.google.com/vision/docs/detecting-landmarks, 2023. 3

[11] Picturethis - plant identifier. *Available online:* https://apps.apple.com/us/app/picturethis-plant-identifier/id1252497129, 2023. 3

[12] Snapcalorie. *Available online:* https://www.snapcalorie.com/, 2023. 3

[13] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. 8

[14] J. Almazan, B. Ko, G. Gu, D. Larlus, and Y. Kalantidis. Granularity-aware Adaptation for Image Retrieval over Multiple Tasks. In *Proc. ECCV*, 2022. 2, 3, 9

[15] N. Aoki and Y. Namba. 3rd Place Solution for Google Universal Image Embedding. *arXiv:2210.09296*, 2022. 8

[16] M. Berman, H. Jégou, A. Vedaldi, I. Kokkinos, and M. Douze. MultiGrain: a Unified Image Embedding for Classes and Instances. *arXiv:1902.05509*, 2019. 3, 5

[17] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 677–694. Springer, 2020. 4

[18] B. Cao, A. Araujo, and J. Sim. Unifying Deep Local and Global Features for Image Search. In *Proc. ECCV*, 2020. 2

[19] W. Chen, Liu Y, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew. Deep Learning for Instance Retrieval: A Survey. *TPAMI*, 2022. 1, 2

[20] M. Conde, I. Aerlic, and S. Jégou. General Image Descriptors for Open World Image Retrieval using ViT CLIP. *arXiv:2210.11141*, 2022. 8

[21] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. Scenic: A jax library for computer vision research and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21393–21398, 2022. 6, 9

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009. 8

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proc. CVPR*, 2019. 8

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5, 8

[25] Yang Feng, Futang Peng, Xu Zhang, Wei Zhu, Shanfeng Zhang, Howard Zhou, Zhen Li, Tom Duerig, Shih-Fu Chang, and Jiebo Luo. Unifying Specialist Image Embedding into Universal Image Embedding. *arXiv:2003.03701*, 2020. 3, 6

[26] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. Scaling Open-Vocabulary Image Segmentation With Image-Level Labels. In *Proc. ECCV*, 2021. 9

[27] S. Gkelios, A. Kastellos, and S. Chatzichristofis. 6th Place Solution to Google Universal Image Embedding. *arXiv:2210.09377*, 2022. 8, 9

[28] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep Image Retrieval: Learning Global Representations for Image Search. In *Proc. ECCV*, 2016. 5

[29] J. Hu, J. Lu, and Y.-P. Tan. Deep Transfer Metric Learning. In *Proc. CVPR*, 2015. 3

[30] X. Huang and Q. Li. 2nd Place Solution to Google Universal Image Embedding. *arXiv:2210.08735*, 2022. 8, 9

[31] S. Ibrahimi, A. Sors, R. Sampaio de Rezende, and S. Clinchant. Learning with Label Noise for Image Retrieval by Selecting Interactions. In *Proc. WACV*, 2022. 8

[32] H. Jégou and O. Chum. Negative Evidences and Co-Occurences in Image Retrieval: The Benefit of PCA and Whitening. In *Proc. ECCV*, 2012. 5

[33] I. Kemelmacher-Shlizerman, S. Seitz, D. Miller, and E. Brossard. The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. In *Proc. CVPR*, 2016. 2

[34] M. Kim, A. K. Jain, and X. Liu. AdaFace: Quality Adaptive Margin for Face Recognition. In *Proc. CVPR*, 2022. 2

[35] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *Proc. ICCV Workshops*, 2013. 2, 3

[36] S. Lee, H. Seong, S. Lee, and E. Kim. Correlation Verification for Image Retrieval. In *Proc. CVPR*, 2022. 2

[37] E. Levi, T. Xiao, X. Wang, and T. Darrell. Rethinking Preventing Class-Collapsing in Metric Learning with Margin-based Losses. In *Proc. ICCV*, 2021. 2

[38] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep Filter Pairing Neural Network for Person Re-Identification. In *Proc. CVPR*, 2014. 2

[39] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. *arXiv preprint arXiv:2111.12993*, 2021. 5

[40] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[41] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, and Jordan Cheney. Iarpa Janus Benchmark–C: Face Dataset and Protocol. *ICB*, 2018. 2

[42] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. Large scale visual food recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[43] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple Open-Vocabulary Object Detection. In *Proc. ECCV*, 2022. 9

[44] K. Musgrave, S. Belongie, and Ser-Nam Lim. A Metric Learning Reality Check. In *Proc. ECCV*, 2020. 2, 4

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5, 6

[46] N. Ota, S. Yokoi, and S. Yamaoka. 5th Place Solution to Kaggle Google Universal Image Embedding Competition. *arXiv:2210.09495*, 2022. 8

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. 8, 9

[48] Jingtian Peng, Chang Xiao, and Yifan Li. Rp2k: A large-scale retail product dataset forfine-grained image classification. *arXiv preprint arXiv:2006.12634*, 2020. 3, 4

[49] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-Grained Visual Categorization via Multi-stage Metric Learning. In *Proc. CVPR*, 2015. 1

[50] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In *Proc. CVPR*, 2018. 2, 4

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5, 8

[52] K. Roth, O. Vinyals, and Z. Akata. Integrating Language Guidance into Vision-based Deep Metric Learning. In *Proc. CVPR*, 2022. 2

[53] K. Schall, K. U. Barthel, N. Hezel, and K. Jung. GPR1200: A Benchmark for General-Purpose Content-Based Image Retrieval. In *Proc. International Conference on Multimedia Modeling*, 2022. 2

[54] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proc. CVPR*, 2015. 1

[55] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Proc. NeurIPS*, 2022. 8

[56] J. Seidenschwarz, I. Elezi, and L. Leal-Taixe. Learning Intra-Batch Connections for Deep Metric Learning. In *Proc. ICML*, 2021. 2

[57] S. Shao and Q. Cui. 1st Place Solution in Google Universal Images Embedding. *arXiv:2210.08473*, 2022. 8, 9

[58] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *Proc. CVPR*, 2016. 1, 2, 3, 4

[59] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 5

[60] F. Tan, J. Yuan, and V. Ordonez. Instance-level Image Retrieval using Reranking Transformers. In *Proc. ICCV*, 2021. 2

[61] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 3, 4

[62] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. *Technical Report CNS-TR-2011-001, California Institute of Technology*, 2011. 2

[63] S. Wang and S. Jiang. INSTRE: A New Benchmark for Instance-Level Object Retrieval and Recognition. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2015. 2

[64] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. CVPR*, 2020. 2, 3, 4

[65] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos,

Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proc. ICML*, 2022. 9

[66] C. Wu and S. Maji. How well does CLIP understand texture? In *Proc. ECCV Workshops*, 2022. 8

[67] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang. DOLG: Single-Stage Image Retrieval with Deep Orthogonal Fusion of Local and Global Features. In *Proc. ICCV*, 2021. 2

[68] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 3, 4

[69] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*, 2018. 5

[70] A. Zhai, H.-Y. Wu, E. Tzeng, D. H. Park, and C. Rosenberg. Learning a Unified Embedding for Visual Search at Pinterest. *Proc. SIGKDD*, 2019. 1

[71] M. Zhang, G. Song, Y. Liu, and H. Li. Towards Robust Face Recognition with Comprehensive Search. In *Proc. ECCV*, 2022. 2

[72] P. Zhang, H. Dou1, Y. Yu, and X. Li. Adaptive Cross-Domain Learning for Generalizable Person Re-Identification. In *Proc. ECCV*, 2022. 2

[73] Y. Zhang, Z. Yin, J. Shao, and Z. Liu. Benchmarking Omni-Vision Representation through the Lens of Visual Realms. In *Proc. ECCV*, 2022. 3

[74] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable Person Re-identification: A Benchmark. In *Proc. ICCV*, 2015. 2

[75] K. Zhu, H. Guo, T. Yan, Y. Zhu, J. Wang, and M. Tang. PASS: Part-Aware Self-Supervised Pre-Training for Person Re-Identification. In *Proc. ECCV*, 2022. 2