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

Nikolaos-Antonios Ypsilantis\textsuperscript{1} \hspace{1cm} Kaifeng Chen\textsuperscript{2} \hspace{1cm} Bingyi Cao\textsuperscript{2} \hspace{1cm} Mário Lipovský\textsuperscript{2}

Pelin Dogan-Schönberger\textsuperscript{2} \hspace{1cm} Grzegorz Makosa\textsuperscript{2} \hspace{1cm} Boris Bluntschli\textsuperscript{2} \hspace{1cm} Mojtaba Seyedhosseini\textsuperscript{2}

Mojtaba Seyedhosseini\textsuperscript{2} \hspace{1cm} Ondřej Chum\textsuperscript{1} \hspace{1cm} André Araujo\textsuperscript{2}

\textsuperscript{1}VRG, FEE, Czech Technical University in Prague \hspace{1cm} \textsuperscript{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/

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 Challenge1, 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].

#### 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× the number of images and 15× the number of classes compared to previ-

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Year</th>
<th># Images</th>
<th># Domains</th>
<th># Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTRE [63]</td>
<td>2015</td>
<td>28k</td>
<td>3</td>
<td>250</td>
</tr>
<tr>
<td>GPR1200 [53]</td>
<td>2021</td>
<td>12k</td>
<td>6</td>
<td>1.2k</td>
</tr>
<tr>
<td>MRT [14]</td>
<td>2022</td>
<td>267k</td>
<td>6</td>
<td>234k</td>
</tr>
<tr>
<td>UnED (ours)</td>
<td>2023</td>
<td>4.1M</td>
<td>8</td>
<td>349k</td>
</tr>
</tbody>
</table>

1https://www.kaggle.com/competitions/google-universal-image-embedding
In our experience, this setting is more common in practice. Additionally, we consider the more challenging evaluation

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

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

Table 3. In such cases, the query itself is always excluded 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 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 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 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 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 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 specific statistics for the training spli

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 us since we consider supervision to be available at the different domains in our task.

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.

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.
which is equivalent to the nearest neighbour accuracy. For natural world domain

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.

### 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 the merged index set 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:

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

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:

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

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
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 $\ell_2$ 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 $\ell_2$ normalized and then projected to 64-D using a (trainable) linear layer. The 64-D embedding is $\ell_2$ 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 $\ell_2$ 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 optimization 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...
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 constitutes 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 (R@1 is at test time is used, so the corresponding specialist of that domain is utilized.

### 4.2. Experimental results and discussion

The experimental results are summarized in Table 4, where we report performance of each baseline 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.

### Table 4: Model evaluation on UnED test set, all results for 64-D (unless stated otherwise) $\ell_2$ 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, UJCCR: Universal Joint Classifier Round Robin sampling, USCCR: 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.
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.
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 pretrained 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Team</th>
<th>Techniques</th>
<th>mMP@5 (private split)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 1</td>
<td>cuilab.ai [57]</td>
<td>CLIP ViT/H/14 → Multi-domain data + multi-stage finetuning</td>
<td>72.8</td>
</tr>
<tr>
<td>Top 2</td>
<td>Xiao [30]</td>
<td>CLIP ViT/H/14 → Multi-domain data + stratified learning rates</td>
<td>70.9</td>
</tr>
<tr>
<td>Top 3</td>
<td>- [15]</td>
<td>CLIP ViT/H/14 → Multi-domain data + drop rate ensembling</td>
<td>69.2</td>
</tr>
<tr>
<td>Top 4</td>
<td>Ivan &amp; Simjeg &amp; CLIP-Art [20]</td>
<td>CLIP ViT/H/14 + CLIP ViT/L/14 → Multi-domain data + model size ensembling</td>
<td>68.8</td>
</tr>
<tr>
<td>Top 5</td>
<td>NS embedding [46]</td>
<td>CLIP ViT/H/14 → Multi-domain data + adding image heuristics (height, width)</td>
<td>68.8</td>
</tr>
<tr>
<td>Top 6</td>
<td>iRonCLIP [27]</td>
<td>CLIP ViT/H/14 → Multi-domain data + multi-stage finetuning + Test time augmentation (TTA)</td>
<td>68.5</td>
</tr>
<tr>
<td>CLIP ViT/H/14</td>
<td></td>
<td>Reference model (1024-D)</td>
<td>62.1</td>
</tr>
</tbody>
</table>

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.
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 fine-tuned [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, 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 proposed 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 OPVV) and the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/173/OHK3/3T/13.
References


