Using Knowledge Graphs to harvest datasets for efficient CLIP model training

Simon Ging*, Sebastian Walter, Jelena Bratulić, Johannes Dienert, Hannah Bast, and Thomas Brox

University of Freiburg, Germany
{gings,swalter,bratulic,dienertj,bast,brox}@cs.uni-freiburg.de
Code and data: https://entity-net.github.io

Abstract. Training high-quality CLIP models typically requires enormous datasets, which limits the development of domain-specific models – especially in areas that even the largest CLIP models do not cover well – and drives up training costs. This poses challenges for scientific research that needs fine-grained control over the training procedure of CLIP models. In this work, we show that by employing smart web search strategies enhanced with knowledge graphs, a robust CLIP model can be trained from scratch with considerably less data. Specifically, we demonstrate that an expert foundation model for living organisms can be built using just 10M images. Moreover, we introduce EntityNet, a dataset comprising 33M images paired with 46M text descriptions, which enables the training of a generic CLIP model in significantly reduced time.

1 Introduction

Contrastive Language-Image Pretraining (CLIP) [1] has become a cornerstone for training Vision-Language Models (VLMs). CLIP models learn high-quality visual embeddings and establish a link to the semantic level of brief text descriptions by training on pairs of images and their corresponding text descriptions collected from the web. The features and the link between images and text have been used directly for, e.g., zero-shot classification or text-to-image retrieval, and enable dialogues with visual input, such as in the LLaVA family of models [2–4]. The link can also be exploited in the opposite direction to enable text-conditional image generation, e.g., Stable Diffusion [5].

Training state-of-the-art CLIP models is computationally expensive. The original CLIP model has seen 12.8B image-text pairs, and later works have scaled this further [6, 7]. This need for scale has limited most of the research to fine-tuning, which comes with reduced architectural flexibility and control over the data selection. It is particularly problematic for analytic research that demands full control over training to find causes of emergent behavior.

The effort to collect vast datasets is also a key bottleneck for building foundation models for expert domains. Although CLIP models are supposed to be

* Corresponding author. Email: gings@cs.uni-freiburg.de

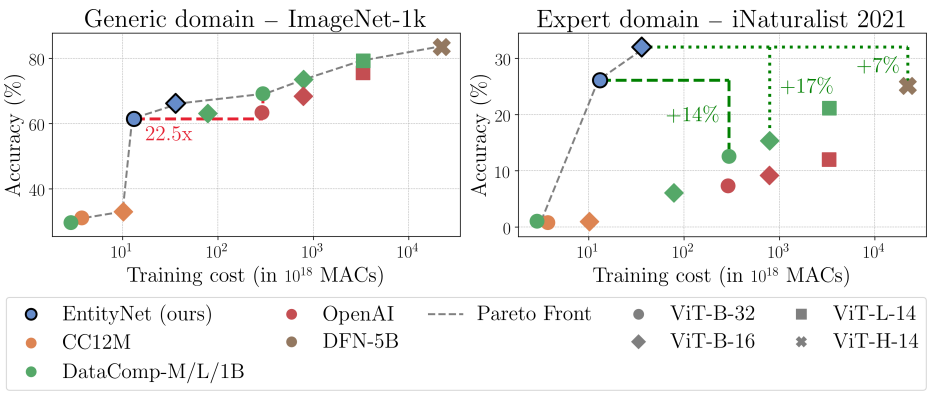

Fig. 1: We demonstrate how to harvest datasets for training CLIP models with an improved quality-cost trade-off, for a generic (left) or an expert domain (right).

generic and cover most of the world, they are not good enough for use in specific expert domains such as medicine or biology. Building foundation models for expert domains requires an efficient data collection process, taking into account the availability of fewer data samples in these domains.

Our goal is to tackle these challenges from the dataset side while keeping the CLIP algorithm fixed. This strategy is backed by recent literature. For example, Li et al. [8] explored CLIP "along three dimensions: data, architecture, and training strategies" and they stress the "significance of high-quality training data". For Large Language Models (LLMs), data curation was shown to reduce training time and model size, achieved through heavily filtered publicly available web data and synthetic data [9]. With the dataset creation process, we aim (1) for improved performance in the expert domain of living organisms, in order to demonstrate the creation of expert foundation models; and (2) we aim for a good trade-off between training efficiency and model performance on the broad domain of the visual world, in order to enable compute-efficient from-scratch analysis of fully functional CLIP models.

We built a dataset we call *EntityNet*, where we leveraged knowledge graphs and targeted web image search. Specifically, from the knowledge graphs Wikidata and WordNet, we collected 135k entities (e.g. *eagle*) as well as their aliases and descriptions. We extracted entity attributes from Wikidata related to color, partonomy, behavior, and other aspects, and used them to guide an LLM in generating entity-attribute queries for image search. For example, from the entity *plastic* and the attribute *small* we generated the query *small plastic item*.

The resulting EntityNet consists of 33M images paired with 45M alt texts and 613k text labels from the knowledge graphs. The dataset is partitioned into a subset of 10M images of living organisms, capturing high-quality visual and semantic information about the taxonomy of animals, plants, and funghi, as well as a subset of 23M images covering a wide range of categories, such as tools, geographical features, materials, and buildings. Notably, from this process we

obtain not only alt texts, but also a link back to the knowledge graph information that was used to create the search query for a given image. We show that this information can be used during training to achieve better performance than by training on alt texts alone. The method of creating our dataset is largely generic and can be applied to other knowledge graphs.

Training on this dataset, we obtain a foundation model that is both specialized on the target expert domain and is also able to understand the overall visual world. In our domain-specific evaluations on iNaturalist and RareSpecies, the model demonstrates robust generalization and clearly surpasses CLIP models trained on much more data (Figure 1). On ImageNet, we demonstrate our dataset to be highly compute efficient and to achieve a performance comparable to models trained 20x longer (Figure 1).

- We propose a method to automatically create a vision-language dataset based on a given knowledge graph and an image search engine.
- We apply this method to create the *EntityNet* dataset, consisting of 33M images paired with 45M alt texts and supplementary text information from the knowledge graphs.
- We train an expert CLIP model for living organisms on a single 8xL40S machine from scratch in 55 hours. This *EntityNet-CLIP* is highly specialized in the target expert domain of living organisms, *and* comparably strong on ImageNet.
- We evaluate our model and a suite of other CLIP models for object classification, image retrieval, and domain shift robustness. In the expert domain of animals and plants, our model achieves higher performance than models with orders of magnitude more parameters or training data. It is also much stronger than CLIP models that specialize only for this domain. In the generic domain, our model performs remarkably well given the low amount of compute required to train it.

2 Related work

Datasets. Many recent studies have investigated methods for building large-scale datasets for multimodal training. Radford et al. [1] trained the original CLIP model on a private dataset of 400M images using image-text pairs with text derived from Wikipedia and WordNet terms. Schuhmann et al. [10] further built the publicly available LAION-400M dataset by filtering HTML data from Common Crawl [11] based on the similarity estimated by the CLIP model. In a follow-up work, they [12] scaled their approach one order of magnitude with the multilingual LAION-5B dataset. Xu et al. [13] sought to replicate the original CLIP's data curation approach. Gadre et al. [6] proposed DataComp, a filtering challenge containing up to 13B image-text pairs from CommonCrawl, and a baseline DataComp-1B dataset. It contains 1.4B pairs filtered with a combination of CLIP score and clustering to match ImageNet [14] training examples.

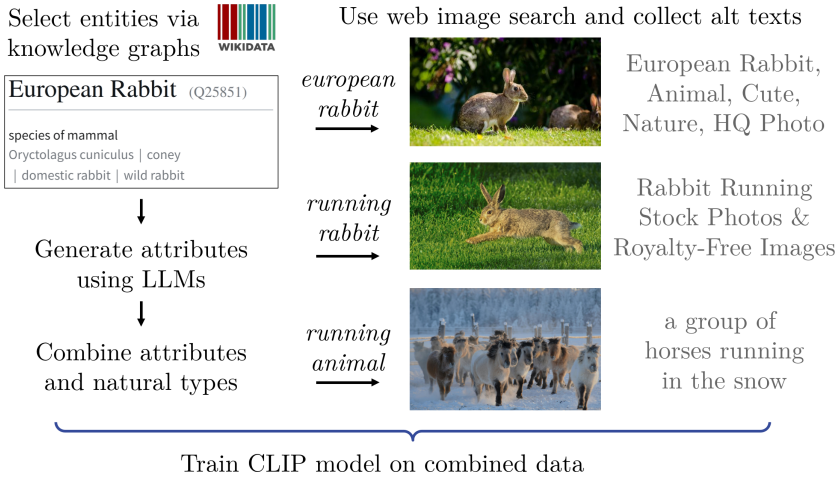

Fig. 2: We create a dataset for vision-language pretraining: First, we extract entities from knowledge graphs, then generate attributes and natural types for them. We search for different combinations of entities, attributes, and types in image search engines, and collect alt texts for each image. Finally, we train our model on the combined data.

Fang et al. [7] trained a Data Filtering Network on 357M human-verified image-text pairs, which they used to filter 42B candidates into the DFN-5B dataset and use that dataset to train the current top model of the OpenCLIP leaderboard [15]. These large datasets have largely supplanted smaller ones like ConceptualCaptions12M (CC12M) [16], relying on unimodal heuristics, and Yahoo Flickr Creative Commons 15M (YFCC15M), a derived subset of 15M image-text pairs from Flickr [17]. While many prior works have focused on scaling up multimodal datasets and models, we aim to improve research on high-quality CLIP models when data and compute efficiency is essential, such as setting up a CLIP model for an expert domain or for scientific analysis of CLIP training.

Stevens et al. [18] curated the TreeOfLife-10M dataset from biological sources [19–21] to train BioCLIP, a model for organismal biology. They evaluated it on RareSpecies, a benchmark of 400 species not seen during training. While BioCLIP leverages domain-specific biological knowledge, we propose a dataset construction method that generalizes to arbitrary domains using knowledge graphs.

Training algorithms. Several works have investigated algorithmic improvements to CLIP. Li et al. [22] simply train and fine-tune with different image resolutions, while Li et al. [23] suggests masking parts of the image to reduce computation. Zhai et al. [24] propose a sigmoid loss which reduces the computational load especially in big distributed settings. They follow up [25] by extending the training objective using multiple previously developed techniques, including captioning-based pretraining [26], self-distillation [27] and online data curation [28] into a unified training strategy. Vasu et al. [29] improve learning

efficiency with synthetic captions created by an image captioning model and an ensemble of CLIP teachers to train their model. Chen et al. [30] evaluate vision encoder choices and design a hybrid architecture that improves over vanilla vision transformers (ViT) based CLIP models. These algorithmic improvements are orthogonal to our contribution. In this work, we fix the algorithm and architecture to enable a fair comparison with ViT-based CLIP baselines.

Li et al. [8] analyze scaling effects across data, architecture, and training strategies, showing that huge models require larger datasets, and data quality plays a crucial role. They create improved datasets by filtering the 3.4B WebLI dataset [31] with CLIP, while we pursue a different dataset collection process.

3 Dataset creation

Our dataset creation process consists of four steps: entity extraction, attribute generation, query building, and image search. This process is generally applicable to all visual domains covered by the underlying knowledge graph. We construct a dataset covering most visual entities in our world, additionally focusing on animals and plants, referred to as the *organism* subset. See Figure 2 for the dataset creation process and Table 1 for examples of entities and attributes.

3.1 Entity extraction

A high-quality list of visual entities forms the basis for our dataset, built from the Wikidata knowledge graph [32] and utilizing the hierarchical structure provided by the *subclass of* relation within Wikidata. For example, the entity *dog* is a subclass of the *pet* entity, which in turn is a subclass of *domesticated animal*. This hierarchy enables easy collection of entities related to a super-entity. First, we manually build a list of 21 super-entities that cover most physical and visual entities in Wikidata. For the *organism* subset, the super-entities are just *animal* and *plant*. Examples of *non-organism* super-entities include *food*, *building*, or *physical tool*, with all super-entities listed in the supplementary material. Next, we extract all entities from Wikidata linked to at least one of the super-entities through the *subclass of* relation. For animals and plants, Wikidata also models their biological taxon hierarchy via the *parent taxon* relation. Because the taxon hierarchy substantially increases the coverage of our *organism* subset, we use it together with the regular *subclass* hierarchy to extract entities. We exclude named entities (e.g., specific persons), as Wikidata models these via the *instance of* relation; we focus solely on the *subclass of* and *parent taxon* relations. For every entity, we also download its name, description, aliases, and its number of Wikimedia sitelinks [1] as additional information. Finally, we apply two filtering steps: First, we remove all entities with a sitelink count below a predefined threshold, eliminating very rare or low-quality entities unlikely to produce strong search results. We then use a LLM to filter out any remaining non-visual entities.

[1] The number of Wikimedia sitelinks is a commonly used and high-quality proxy for the popularity of an entity [33].

Table 1: **Top:** Examples of entities and additional information as extracted from the Wikidata knowledge graph. **Bottom:** Examples of attributes and corresponding search queries for different entities as generated by the LLM.

Entity	Description	Sitelinks	Aliases
tiger	species of big cat	216	tigress, tigers, Panthera tigris
chest	box-shaped type of furniture	51	coffer, kist
muscle car	type of high-performance car	30	high performance car
Entity	**Category**	**Attribute**	**Search query**
rock	Pattern and texture	porous	porous rock
wolf	Environment	snow	wolf in the snow
residence	Parts	arches	arches in residence architecture
garlic	Shape and size	big	big garlic bulb
farm	Other	tourist	tourists visiting a farm
boot	Color	multicolor	multicolored boots

For our expert domain, the *organism* subset, we also add all nouns from WordNet [34] that are a subclass of the *living thing* node, excluding humans, named entities and entities that cannot be seen with the bare eye, e.g., microorganisms. Finally, we employ heuristic methods to detect and remove potentially offensive entities via a profanity filter.

3.2 Attribute generation

Besides searching for the entities directly, we also aim to search for variations of them in different contexts, by combining them with attributes. We manually define 6 visual attribute categories we want to search for: *Color*, *Pattern and texture*, *Parts*, *Shape and size*, *Environment*, and *Other*. We extract potential attributes for each entity from the Wikidata knowledge graph and prompt an LLM[2] with this entity and attribute information to generate a list of visual attributes. We first considered generating attributes without categories, however, the results lacked diversity, and adding categories improves the variety of attributes. For each attribute we also generate a search query combining the attribute itself with the corresponding entity. We generate between 1 and 10 attributes per category and generate them for the most popular entities only, as image search engines fail to respect attributes in search queries for rare entities, where they often even struggle to return good results for the entity alone.

3.3 Query building

For the entities themselves, we use their names and aliases as search queries. We search entity-attribute combinations using the search queries generated by the LLM. We then create additional queries based on the attributes: First, we

[2] We use three LLMs and merge their generated attributes: Qwen2.5 7B [35], OpenAI GPT-4o, and OpenAI GPT-4o mini (both accessed via API at platform.openai.com)

Table 2: Details of our EntityNet dataset. We show the number of unique elements for each column, e.g. the number of images after deduplication or all unique entity aliases in the respective sets.

Query set	Images	Queries	Entities	Aliases	Attributes	Alt texts	Example query
World entity	23M	158k	74k	101k	-	23M	ship
+ attribute	19M	139k	6k	16k	20k	16M	small handbag
Living entity	9M	72k	63k	51k	-	8M	kohlrabi
+ attribute	9M	53k	1k	3k	5k	6M	tropical plant
All	33M	416k	135k	149k	23k	45M	-

determine the entity's natural type – the super-entity a human would most likely associate with it, e.g., *bird* for *eagle*, or *clothing* for *hat*. It is neither too general nor too specific, and can typically help disambiguate entities sharing the same name. We use an LLM to select the most fitting super-entity from an entity's super-class hierarchy as its natural type and generate a brief description explaining why this type is appropriate. The description is used during training as a potential text label. We then replace entity mentions in the attribute search queries with their natural types. For example, the attribute query *eagle in its nest* may turn into *bird in its nest*, or *black BMW M4* into *black car*.

3.4 Image search and filtering

We execute our search queries using the image search APIs of Bing and Google. Initial tests on the *organism* subset revealed Bing's search results to be of much higher quality at a lower cost, so we rely solely on the Bing API for all other queries. The image search APIs also provide the URL for the website hosting the image, which we use to collect alt texts from the HTML image tag. After downloading images and alt texts, we perform the following postprocessing steps.

Similar to Changpinyo et al. [16], we apply *relaxed filtering heuristics*. We do not use multimodal filtering, but rely on search engines to provide image-text correspondences. We remove JSON-like and too long text. We also remove images with an aspect ratio of more than 4 or covering less than 4096 pixels.

We deduplicate all downloaded images using the Self-Supervised Descriptor for Image Copy Detection method (SSCD) [36]. For duplicates, we retain the largest image and collect all unique alt texts and related entities from the duplicates. Deduplication increases the dataset diversity per sample, since the domain coverage stays the same, while the number of samples decreases. We also remove images that appear in any evaluation dataset using the same SSCD method.

Our final dataset comprises approximately 33M images and 45M alt texts, obtained from 416k queries. This amounts to 79 images per query and 1.4 alt texts per image on average. The total cost for all image search API calls was around 10,000$. See Table 2 for an overview over our dataset.

4 Experimental setup

4.1 Training

We trained all models with the standard CLIP loss [1], using a batch size of 8,192 for pretraining and 32,768 during finetuning, along with random resized crop augmentation. We sampled text labels from both the image alt texts and the knowledge graph. For each image, 50% of the time, we chose a random alt text, and 50% of the time, we chose randomly between search queries, aliases, or descriptions of the corresponding entity. We trained all models for 18 epochs. Training on 33M images takes ∼55 hours on 8 L40s GPUs (48GB VRAM per GPU). Our training code is based on OpenCLIP [15]. Further training details and text sampling examples are in the supplementary material.

4.2 Evaluated models

On our EntityNet dataset, we trained ViT CLIP models of size B-32 and B-16 from random initialization. For a comparison with a similarly sized dataset, we also trained models on CC12M by downloading all available URLs, and then detecting and removing duplicates relative to the evaluation datasets using the same procedure as detailed in Section 3.4, obtaining 9.3M images. We finetuned B-32 and B-16 CLIP models trained on DataComp-1B on our dataset to compare finetuning and pretraining performance. We also evaluate the original *OpenAI CLIP* [1], models pretrained on *DataComp-M/L/1B*, *CommonPool-M/L* [6], and DFN-5B [7], as well as the biological domain expert model BioCLIP [18], a ViT-B-16 CLIP model finetuned from OpenAI-CLIP on the TreeOfLife-10M dataset.

4.3 Object classification evaluation

To test the VLMs on object classification, we use the same procedure as CLIP [1], see the supplementary material for a detailed description. We evaluate all models on just encoding the class name, and on using the average embedding of the 80 context prompts that the CLIP authors used for ImageNet, and report the higher top-1 accuracy. For zero-shot object classification, we require models not to have been trained on the training set of the benchmark, to test "generalization to unseen datasets" [1]

Benchmarks in the generic domain. We evaluate on ImageNet [14], a popular image classification benchmark [37]. We use the ILSVRC2012 validation set, which contains 50,000 images from 1,000 classes. The classes include simple objects, such as *park bench*, as well as more fine-grained labels like 23 types of terrier dogs, e.g., *Staffordshire Bull Terrier*. We further evaluate the robustness under distribution shifts on ImageNet-A [38], ImageNet-R [39], ImageNet-Sketch [40], ImageNet-V2 [41], and ObjectNet [42] as proposed by Taori et al. [43]. ImageNet-A contains 7500 samples of 200 ImageNet classes. The samples were adversarially filtered to make ResNet-50s misclassify them, providing a more challenging test. ImageNet-R contains 30000 renditions, such

as paintings or embroidery, of 200 ImageNet object classes. ImageNet-Sketch contains 50000 sketches, covering 200 ImageNet classes. ImageNet-V2 replicates the original ImageNet generation process, providing an additional 10000 test images. The object-centered ObjectNet contains 18574 images from 113 ImageNet classes with control over background, rotation and viewpoint.

Benchmarks in the expert domain. We evaluate our models on iNaturalist 2021 [20], a fine-grained species classification benchmark that contains 100k images in the validation set of 10k different species. Similar to Parashar et al. [44], we report the best results after testing on both the English common name and the Latin taxon name. We further test on the Caltech-UCSD Birds (CUB) [45] dataset, which contains 5,794 images of birds in the original author's test set, each annotated as one of 200 fine-grained bird species, e.g., *grasshopper sparrow*. Additionally, we evaluate on the Rare Species benchmark proposed by Stevens et al. [18], comprising 400 species with 30 images each, specifically tailored to assess generalization to unseen taxa. To comply with the benchmark requirements of not seeing the testing 400 species during training, we exclude all entities and queries from our dataset that appear in RareSpecies, using substring matching. As class names, we evaluate all text types proposed by Stevens et al. [18]: combinations of the Latin taxonomy and the English common name. Same as in Section 4.3 we evaluate on both the CLIP ImageNet prompt and no prompt, and report the better of both accuracies.

4.4 Retrieval evaluation

We evaluate the COCO Karpathy test split [46], a subset of 5000 samples from the MS-COCO [47] dataset paired with 5 texts each. We also evaluate the 1000 samples in the Karpathy test split of Flickr30k [48] annotated with 5 texts per image, as well as on the 3600 image-text pairs in XM3600 [49]. We report the average of image-to-text and text-to-image recall@1 over all datasets.

5 Results

We evaluate CLIP **pretrained from scratch** on our EntityNet dataset and CLIP models trained on other datasets. In Figures 1 and 3 we contrast the effectiveness of models with their training cost. We show the results in detail in Table 3. In the generic domain, our models surpass others trained on similarly sized datasets while achieving comparable performance on object classification with models trained 20x longer. On image-text retrieval, our model performs similarly to models trained on the same amount of compute. While our pipeline creates a dataset efficient for understanding objects and their properties, understanding complex scenes still requires learning mainly from the alt texts more than from objects and attributes. In the expert domain, we outperform even the largest CLIP models on the challenging iNaturalist 2021 dataset, which requires classifying images among 10k fine-grained species. Our model also excels on CUB by distinguishing 200 bird species better than all other CLIP models of

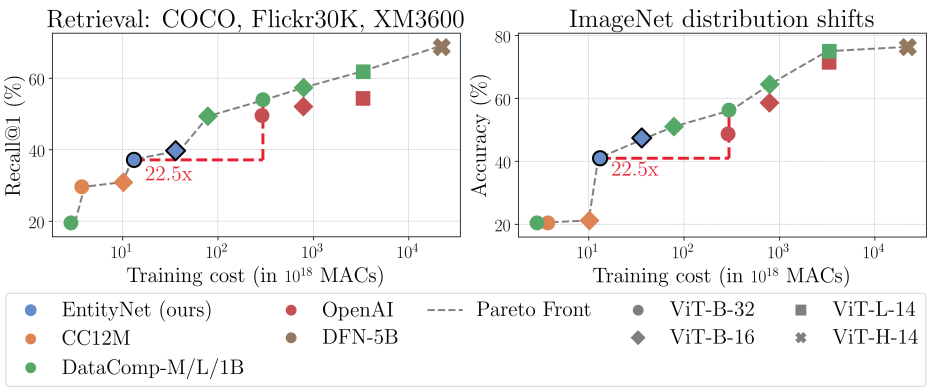

Fig. 3: Results on image retrieval and distribution shift robustness on ImageNet.

the same size. Further, when compared to the expert model *BioCLIP*, explicitly trained for organismal biology at a similar training cost, our model demonstrates superior performance. On the Rare Species benchmark, our model outperforms *BioCLIP* on unseen species, showing the effectiveness of our dataset collection method over a manually designed living organism training set.

We investigate improving existing CLIP models via **finetuning** in Table 4. The results show that our dataset can be leveraged to create expert CLIP models that outperform both the base model and our model pretrained from scratch on the expert domain. This improvement comes at the cost of trading off some capabilities in the other domains. When finetuning only on the expert domain, we trade off more capabilities, yet obtain even stronger experts.

We further validate and verify our design choices through a **component analysis** in Table 5. Training separately on the generic and the domain expert part of our dataset reveals that, while the best generic model emerges from training on everything, a slightly better expert model is the result of training only on the expert domain (first table segment). However, generalization to unseen species slightly benefits when training on the full dataset, showing that our generic domain data can enhance generalization capabilities within the expert domain. We also observe that generating and downloading attribute queries contributes to improved performance of the pretrained model.

In the second segment of the table, we evaluate the mixture of alt text and knowledge graph labels used during training. Notably, both training only on alt texts or only on knowledge graph labels mostly performs worse than our 50-50 mix. The exception is image-text retrieval, where training fully on alt text performs slightly better. Potentially, the knowledge graph labels are less useful for learning the matching between longer text queries and images, and more useful for learning fine-grained object classification.

Finally, we reduce the scale of our dataset by powers of two. While the model performance expectedly drops with reduced dataset size, the efficiency of

Table 3: Results for training CLIP B-32 and B-16 on our EntityNet dataset from scratch. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations). We only compare zero-shot results and mark results as "–" if the model does not fulfill the zero-shot requirements.

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image-Net	Retrie-val	Distr. shift	iNat. 2021	CUB	Rare Species
B-32	CC12M	3.7	9.3	28.6	25.6	18.3	0.7	9.2	–
B-32	CommonPool-M	2.9	128.0	27.2	20.2	19.8	0.8	10.1	–
B-32	DataComp-M	2.9	14.0	29.7	19.5	20.5	1.0	16.8	–
B-32	OpenAI	288.6	400.0	63.4	49.6	48.7	7.4	51.8	–
B-32	DataComp-1B	295.4	1400.0	**69.2**	**54.0**	**56.3**	12.6	73.8	–
B-32	EntityNet (ours)	13.1	32.7	61.5	37.2	41.0	**26.1**	**79.5**	**42.7**
B-16	BioCLIP	61.3	10.4	18.6	0.8	15.4	–	78.1	38.1
B-16	CommonPool-L	78.2	1280.0	57.8	45.6	47.0	4.1	35.1	–
B-16	DataComp-L	78.2	140.0	63.1	49.4	51.1	6.1	48.1	–
B-16	DataComp-1B	791.4	1400.0	**73.5**	**57.4**	**64.4**	15.3	79.0	–
B-16	OpenAI	784.6	400.0	68.3	52.1	58.6	9.2	56.1	–
B-16	EntityNet (ours)	36.0	32.7	66.2	39.8	47.4	**32.0**	**85.3**	**47.1**
L-14	OpenAI	3328.4	400.0	75.5	54.3	71.4	12.0	62.9	–
L-14	DataComp-1B	3338.6	1400.0	79.2	61.8	74.9	21.1	85.5	–
L-14	DFN-2B	3338.6	2000.0	81.4	64.2	74.8	21.6	86.5	–
H-14	DFN-5B	22164.0	5000.0	**83.4**	**68.7**	**76.3**	25.1	**88.1**	–

our dataset per datapoint stays high, with the model still reaching 33% accuracy on ImageNet with only 4M images.

6 Limitations

The proposed data harvesting approach assumes that there is a knowledge graph for the target domain and that there is a searchable database with noisy pairing of images and text. However, knowledge graphs exist in many domains, e.g., *UniProt* [50] with 246M protein sequence records or *AgriKG* [51] with 150k agricultural entities. Also, if no image search engine is available for the given domain, but a large amount of image-text data exists, pairs can be found by searching for the queries via substring matching in the image-text pairs.

Another limitation is the small, but significant drop in performance on image-text retrieval and classifying ImageNet distribution shifts in the generic domain, when finetuning a large model with EntityNet. First, our dataset by design has a strong focus on the expert domain and trades off some performance in the generic domain during finetuning. Second, our search pipeline finds many clean object-centric images and annotates them with entity information, which tremendously helps understanding object semantics, but to improve efficiency on image-text retrieval in a similar way, one needs to tackle the quality of alt texts and their alignment to the images [52]. Finally, we focused on searching photos, which

Table 4: Results for finetuning the DataComp-1B CLIP model on EntityNet. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations).

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image -Net	Retrie- val	Distr. shift	iNat 2021	CUB
B-32	DataComp-1B	295.4	1400.0	69.2	**54.0**	**56.3**	12.6	73.8
B-32	EntityNet	13.3	32.7	**69.5**	50.8	53.3	29.5	83.3
B-32	Only organisms	4.2	10.2	48.2	31.2	33.3	**37.0**	**87.0**
B-16	DataComp-1B	791.4	1400.0	**73.5**	**57.4**	**64.4**	15.3	79.0
B-16	EntityNet	36.1	32.7	**73.5**	52.2	61.0	34.9	86.5
B-16	Only organisms	11.3	10.2	51.4	34.8	39.2	**42.7**	**90.3**

Table 5: Analysis of performance when varying dataset composition, text sampling and dataset size. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations).

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image- Net	Retrie- val	Distr. shift	iNat. 2021	CUB	Rare Species
B-32	Everything	13.1	32.7	**61.5**	**37.2**	**41.0**	26.1	79.5	**42.7**
B-32	No organisms	9.0	22.5	39.2	32.1	28.0	0.8	6.2	6.9
B-32	Only organisms	4.1	10.2	36.0	16.5	21.0	**28.6**	**83.2**	42.0
B-32	No attributes	8.7	21.8	54.8	28.6	33.8	25.6	79.7	39.2
B-32	50% alt text	13.1	32.7	**61.5**	37.2	**41.0**	**26.1**	**79.5**	**42.7**
B-32	100% alt text	13.1	32.7	59.1	**38.3**	38.1	22.9	78.8	39.7
B-32	0% alt text	13.1	32.7	55.7	13.5	35.5	24.2	78.1	29.7
B-32	Full size	13.1	32.7	**61.5**	**37.2**	**41.0**	**26.1**	**79.5**	**42.7**
B-32	1/2 size	6.6	16.4	54.1	30.3	33.6	20.1	74.1	36.6
B-32	1/4 size	3.3	8.2	45.2	23.5	25.7	13.2	64.9	28.0
B-32	1/8 size	1.6	4.1	33.3	16.6	17.7	7.3	47.8	19.4
B-32	1/16 size	0.8	2.0	19.9	9.4	10.0	2.8	27.4	11.8

explains slightly lower accuracy when classifying paintings and sketches – the EntityNet dataset simply contains a lower percentage of such types of images than datasets like CommonPool.

7 Conclusions

We demonstrated how to use knowledge graphs to harvest datasets that are efficient for training CLIP models. Our strategy allows to create an expert domain dataset with little manual effort, enabling the development of CLIP models that significantly outperform standard models in the expert domain. The expert domain dataset can be used for training a model from scratch or for finetuning an existing vanilla model. The substantial size and diversity of the expert domain dataset ensures that the good generalization properties of CLIP exist also in the expert domain, in contrast to training with an over-specialized expert dataset.

Furthermore, we demonstrated that the proposed harvesting strategy is also viable to create a common domain dataset, which allows us to achieve a better quality-compute trade-off than training with previous datasets. Future work can use our EntityNet dataset to train CLIP models with all emergent properties much more efficiently, thus allowing for experiments, where training can be controlled. So far, this has been possible only with models of lacking quality.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 499552394 – SFB 1597 – Project-ID 417962828 – Project-ID 539134284. The authors acknowledge support from the state of Baden-Württemberg through bwHPC.

A Additional analyses

A.1 Verification of Bing search results

In this work, we search for images that fit our queries using Bing image search. In contrast, CLIP [1] searches for the query in a large pool of image-text pair candidates created from raw HTML via substring matching. We manually evaluate the quality of the image search engine and of substring matching in Table 6, using randomly select queries of our EntityNet dataset. In total, 87% of queries were answered correctly. Incorrect results of Bing search can be grouped into *wrong similar terms* (for the japanese "dogi" uniform, the search instead returns actual dogs), and *attribute ignored* (for "B-25 Mitchell gray paint" the search returns the correct B-25 plane, but differently painted). Bing search is clearly superior to substring matching, with the latter algorithm only fulfilling 52% of queries.

A.2 Scaling behaviour of EntityNet

In Figure 4, We study the impact of removing entire entities or reducing the number of queries per entity, and compare this with randomly dropping images. Removing entities and their associated images degrades performance more than randomly removing the same number of images, indicating that entities represent important semantic concepts for model training. In other words, our results suggest that diversity (more entities with fewer images per entity) is more important than depth (fewer entities with more images each). In all cases, shrinking the dataset lowers performance, so a certain dataset size (here 33M) is indeed essential for a good generic model.

B Detailed experimental results

We show extensive results on object classification in Table 7. For a more detailed analysis of model capabilities on ImageNet, we split the classes into *living* (410 classes) and *other* (590 classes) using WordNet: Since ImageNet labels are built on WordNet nouns, we simply select all labels that are children of the *living things* node for the *living* set. On iNaturalist, in addition to the 2021 version, we also evaluate on the 2019 version which contains 3,030 images in the validation set, each annotated with one of 1,010 species. We test with the same protocol as on iNaturalist 2021, testing on both english and latin class names and reporting the best accuracy. We show additional results on retrieval in Table 8 and object classification under distribution shift in Table 9.

C Experimental details

C.1 Hyperparameter settings

We report hyperparameters for our experiments in Table 10. Similar to Li et al. [23], we reduce the context size of the text encoder down from 77 to 32 to reduce VRAM and training time requirements. For a fair comparison with other CLIP models, we report all training cost and training duration as if the training was run at a context length of 77.

C.2 Object classification evaluation

To test the VLMs on object classification, we use the same procedure as CLIP [1]: Given an image I, class names $C_1, ..., C_N$, image encoder f and text encoder g, we embed the image using the image encoder $\mathbf{v} = f(I)$. To acquire a text embedding for class C_c, the CLIP authors started by directly encoding the class names as $\mathbf{w}_c = g(C_c)$, e.g., *dog*. Alternatively, they created several prompts P using templates, e.g., *graffiti of a dog*, *a photo of the cool dog*, etc., then encoded each prompt, and computed the average embedding: $\mathbf{w}_c = \sum_{p \in P} g(p)/|P|$. They referred to this approach as using "context prompts". Finally, given the image and text embeddings, the prediction p is the class which has the highest cosine similarity to the image.

For a fair comparison between models that have been trained with different prompts, we evaluate all models on just encoding the class name, and on using the average embedding of the 80 context prompts that the CLIP authors used for ImageNet, then report the higher top-1 accuracy.

D Qualitative example of text label sampling

We show an example of our text label sampling strategy in Table 11.

E Image Search API details

Google The Google Image Search API is available via the Google Cloud Platform, and requires an existing programmable search engine to function. It returns up to 10 images per request and page with a limit of 10 pages, i.e., 100 images per query. It costs 5$ per 1,000 API calls, resulting in costs of about 500$ to download 1M images. We found the search results from the Google API to be quite different, and arguably worse, from the ones returned when using the regular Google image search. For all our API requests we set the parameter *imgColorType* to *color*, *imgType* to *photo*, *lr* to *lang_en*, and *excludeTerms* to *drawing clipart illustration cartoon vector painting*. This way we get mostly real-world images in our search results. We additionally add all aliases and the natural type of the sought entity to the *orTerms* parameter for entity and entity-attribute queries. Because the Google API returns only up to 10 images per request and page, we search for the following number of pages: 2 pages each for entity queries, 4 pages each for entity-attribute queries, and 10 pages each for all natural-type-attribute queries. We started our search with queries from the *organism* subset on both the Google and Bing APIs. We found the quality and value-for-money ratio of the Bing API to be better, and therefore switched to only using Bing for all other queries.

Bing The Bing Image Search API is available via Microsoft Azure. It returns up to 150 images per request and has no restrictions on the number of accessible pages. It costs 18$ per 1,000 API calls, resulting in costs of about 120$ to download 1M images. In our experience, the returned images closely match the ones from the regular Bing image search. For all our API requests we set the parameter *imageType* to *Photo* and *color* to *ColorOnly*. Unlike the Google API, Bing does not have a way to specify *orTerms* via a separate request parameter, so we add the natural type of the sought entity to the search query directly, e.g., we search for *jaguar animal*. The Bing API returns 150 images per request and page, we request one page for each query.

F Querying entities with SPARQL

The SPARQL query over Wikidata used to harvest all entities under a super-entity is displayed in Figure 5. It returns a list of entities from a specified target domain as defined by one or more super-entities. The super-entities can be determined manually by searching for appropriate entities on the Wikidata website. For example, if we want to build a dataset about vehicles, we can set the super-entity to vehicle (Q42889), as done in Table 12. We list the super-entities we considered for our dataset and the relevant statistics in Table 13 for included and Table 14 for excluded super-entities.

Table 6: We randomly select 100 search queries for each query set and manually check for the following errors: *Wrong:* The majority of the results do not match the query. *Too few:* Only four or less images are found. Over all 416k queries, Bing search finds ≥ 5 images in 99.8% of cases.

Query set	Bing search		Substring matching	
	Wrong	Too few	Wrong	Too few
World entity	13%	0%	15%	26%
World entity + attribute	22%	0%	3%	66%
Living entity	0%	0%	7%	19%
Living entity + attribute	18%	0%	7%	51 %
Total	13%	0%	8%	40 %

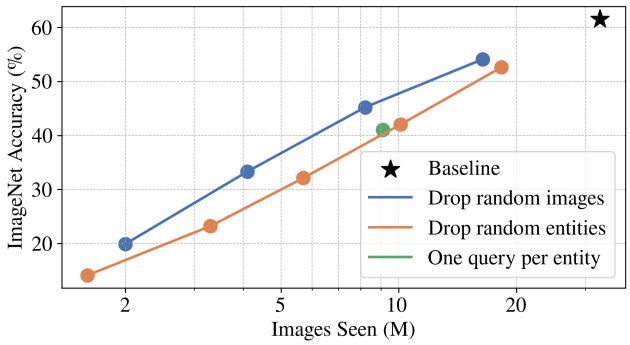

Fig. 4: Scaling entities and queries per entity.

Table 7: Detailed object classification results. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations). We only compare zero-shot results and mark results as "–" if the model does not fulfill the zero-shot requirements.

Arch.		Dataset	MACs (1e18)	Images in dataset (M)	ImageNet 1k	ImageNet living	ImageNet other	iNat. 2019	iNat. 2021	CUB	Rare spcs
				# Classes →	1,000	410	590	1,010	10k	200	400
B-32		CC12M	3.7	9.3	28.6	27.6	31.1	2.0	0.7	9.2	–
B-32		CommonPool-M	2.9	128.0	27.2	20.6	33.5	2.6	0.8	10.1	–
B-32		DataComp-M	2.9	14.0	29.7	25.5	34.5	3.0	1.0	16.8	–
B-32		OpenAI	288.6	400.0	63.4	65.5	63.1	10.9	7.4	51.8	–
B-32		DataComp-1B	295.4	1400.0	**69.2**	**71.2**	**69.1**	16.7	12.6	73.8	–
B-32		EntityNet	13.1	32.7	61.5	68.5	57.9	**38.3**	**26.1**	**79.5**	**42.7**
B-16	⋆	BioCLIP	61.3	10.4	18.6	44.3	2.6	–	–	78.1	38.1
B-16		CommonPool-L	78.2	1280.0	57.8	53.2	62.4	6.9	4.1	35.1	–
B-16		DataComp-L	78.2	140.0	63.1	61.8	65.3	9.1	6.1	48.1	–
B-16		DataComp-1B	791.4	1400.0	**73.5**	**75.9**	**73.2**	19.5	15.3	79.0	–
B-16		OpenAI	784.6	400.0	68.3	71.5	67.4	12.5	9.2	56.1	–
B-16		EntityNet	36.0	32.7	66.2	73.9	62.0	**42.2**	**32.0**	**85.3**	**47.1**
L-14		OpenAI	3328.4	400.0	75.5	78.9	74.5	15.2	12.0	62.9	–
L-14		DataComp-1B	3338.6	1400.0	79.2	82.1	78.4	23.6	21.1	85.5	–
L-14		DFN-2B	3338.6	2000.0	81.4	83.7	80.9	24.1	21.6	86.5	–
H-14		DFN-5B	22164.0	5000.0	**83.4**	**85.4**	**83.2**	31.4	25.1	**88.1**	–
B-32		DataComp-1B	295.4	1400.0	69.2	71.2	**69.1**	16.7	12.6	73.8	–
B-32	⋆	EntityNet	13.3	32.7	**69.5**	73.6	67.8	41.5	29.5	83.3	–
B-32	⋆	Only organisms	4.2	10.2	48.2	**76.3**	30.9	**49.3**	**37.0**	**87.0**	–
B-16		DataComp-1B	791.4	1400.0	**73.5**	75.9	**73.2**	19.5	15.3	79.0	–
B-16	⋆	EntityNet	36.1	32.7	**73.5**	78.1	71.5	46.7	34.9	86.5	–
B-16	⋆	Only organisms	11.3	10.2	51.4	**80.2**	33.8	**54.3**	**42.7**	**90.3**	–
B-32		EntityNet	13.1	32.7	**61.5**	**68.5**	**57.9**	38.3	26.1	79.5	**42.7**
B-32		No organisms	9.0	22.5	39.2	17.5	56.1	1.7	0.8	6.2	6.9
B-32		Only organisms	4.1	10.2	36.0	**68.5**	15.4	**41.4**	**28.6**	**83.2**	42.0
B-32		No attributes	8.7	21.8	54.8	63.0	50.4	36.4	25.6	79.7	39.2
B-32		50% alt text	13.1	32.7	**61.5**	**68.5**	**57.9**	**38.3**	**26.1**	**79.5**	**42.7**
B-32		100% alt text	13.1	32.7	59.1	66.5	55.4	36.4	22.9	78.8	39.7
B-32		0% alt text	13.1	32.7	55.7	64.4	51.5	35.4	24.2	78.1	29.7
B-32		Full size	13.1	32.7	**61.5**	**68.5**	**57.9**	**38.3**	**26.1**	**79.5**	**42.7**
B-32		1/2 size	6.6	16.4	54.1	61.5	50.5	30.7	20.1	74.1	36.6
B-32		1/4 size	3.3	8.2	45.2	52.4	41.8	23.4	13.2	64.9	28.0
B-32		1/8 size	1.6	4.1	33.3	39.6	30.5	14.5	7.3	47.8	19.4
B-32		1/16 size	0.8	2.0	19.9	25.0	18.1	7.6	2.8	27.4	11.8
B-32		Batch size 2k	13.1	32.7	59.7	65.4	57.2	31.5	21.0	74.1	38.7
B-32		Batch size 4k	13.1	32.7	60.9	67.0	**58.0**	36.2	24.1	77.6	41.0
B-32		Batch size 8k	13.1	32.7	**61.5**	**68.5**	57.9	38.3	26.1	79.5	**42.7**
B-32		Batch size 16k	13.2	32.7	60.5	68.2	56.6	38.8	**26.9**	81.2	42.1
B-32		Batch size 32k	13.3	32.7	58.6	67.0	54.2	**39.5**	26.3	**81.3**	41.7

Table 8: Detailed retrieval results. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations).

Arch.		Dataset	MACs (1e18)	Images in dataset (M)	Average	COCO I2T	COCO T2I	Flickr30K I2T	Flickr30K T2I	XM3600 I2T	XM3600 T2I
B-32		CC12M	3.7	9.3	25.6	22.4	15.2	37.2	27.1	26.1	25.5
B-32		CommonPool-M	2.9	128.0	20.2	18.3	11.2	29.9	18.9	23.6	19.6
B-32		DataComp-M	2.9	14.0	19.5	17.1	11.0	26.0	18.0	23.6	21.5
B-32		OpenAI	288.6	400.0	49.6	50.1	30.5	77.5	58.8	43.4	37.2
B-32		DataComp-1B	295.4	1400.0	**54.0**	**53.5**	**37.1**	**78.8**	**61.1**	**48.3**	**45.3**
B-32		EntityNet	13.1	32.7	37.2	32.8	22.2	52.3	37.8	40.6	37.3
B-16	⋆	BioCLIP	61.3	10.4	0.8	0.4	0.2	0.9	0.6	1.8	1.1
B-16		CommonPool-L	78.2	1280.0	45.6	44.4	28.8	68.3	51.0	42.1	39.2
B-16		DataComp-L	78.2	140.0	49.4	48.7	32.2	73.5	55.1	44.7	42.1
B-16		DataComp-1B	791.4	1400.0	**57.4**	**57.5**	**40.2**	**84.9**	**67.3**	**47.9**	**46.5**
B-16		OpenAI	784.6	400.0	52.1	52.5	33.1	81.9	62.0	43.8	39.3
B-16		EntityNet	36.0	32.7	39.8	36.0	25.2	57.1	43.3	39.9	37.1
L-14		OpenAI	3328.4	400.0	54.3	56.3	36.5	85.1	65.2	44.5	38.4
L-14		DataComp-1B	3338.6	1400.0	61.8	63.2	45.8	89.5	73.4	50.4	48.6
L-14		DFN-2B	3338.6	2000.0	64.2	65.7	48.6	89.6	75.3	53.6	52.7
H-14		DFN-5B	22164.0	5000.0	**68.7**	**72.3**	**53.9**	**93.0**	**80.2**	**57.6**	**55.4**
B-32		DataComp-1B	295.4	1400.0	**54.0**	**53.5**	**37.1**	**78.8**	**61.1**	**48.3**	45.3
B-32	⋆	EntityNet	13.3	32.7	50.8	48.1	34.4	72.1	57.1	47.8	**45.6**
B-32	⋆	Only organisms	4.2	10.2	31.2	28.0	19.7	44.8	33.7	30.9	29.9
B-16		DataComp-1B	791.4	1400.0	**57.4**	**57.5**	**40.2**	**84.9**	**67.3**	**47.9**	**46.5**
B-16	⋆	EntityNet	36.1	32.7	52.2	50.4	36.4	75.6	59.4	47.3	44.3
B-16	⋆	Only organisms	11.3	10.2	34.8	31.8	22.8	51.7	37.1	33.5	32.0
B-32		EntityNet	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		No organisms	9.0	22.5	32.1	28.6	18.7	46.3	33.6	33.2	32.3
B-32		Only organisms	4.1	10.2	16.5	12.8	9.8	23.4	15.3	19.0	18.6
B-32		No attributes	8.7	21.8	28.6	23.6	16.1	41.3	27.3	32.3	30.8
B-32		50% alt text	13.1	32.7	37.2	32.8	22.2	52.3	37.8	**40.6**	37.3
B-32		100% alt text	13.1	32.7	**38.3**	**35.2**	**23.2**	**53.4**	**39.8**	40.5	**38.0**
B-32		0% alt text	13.1	32.7	13.5	8.7	6.1	19.2	11.7	18.3	17.0
B-32		Full size	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		1/2 size	6.6	16.4	30.3	27.0	17.6	42.7	29.7	33.3	31.6
B-32		1/4 size	3.3	8.2	23.5	19.5	13.0	31.1	22.9	27.6	27.2
B-32		1/8 size	1.6	4.1	16.6	12.8	8.7	20.7	14.2	21.9	20.9
B-32		1/16 size	0.8	2.0	9.4	7.1	5.0	10.0	7.1	13.9	13.5
B-32		Batch size 2k	13.1	32.7	35.6	30.8	20.9	50.2	37.2	38.2	36.4
B-32		Batch size 4k	13.1	32.7	36.4	31.6	22.1	51.7	37.0	39.1	37.2
B-32		Batch size 8k	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		Batch size 16k	13.2	32.7	35.8	32.2	21.8	50.5	36.9	37.2	36.1
B-32		Batch size 32k	13.3	32.7	34.5	31.0	20.8	48.9	34.5	36.2	35.3

Table 9: Detailed results on ImageNet distribution shifts. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure compute cost, we calculate training MACs (multiply–accumulate operations). *INet:* ImageNet.

Arch.		Dataset	MACs (1e18)	Images in dataset (M)	INet 1K	Ave-rage	INet V2	INet R	Obj.-Net	INet Sketch	INet A
		# Classes →			1,000	–	1000	200	133	1000	200
B-32		CC12M	3.7	9.3	28.6	18.3	24.2	34.5	12.1	16.0	4.7
B-32		CommonPool-M	2.9	128.0	27.2	19.8	22.5	33.0	20.9	18.4	4.3
B-32		DataComp-M	2.9	14.0	29.7	20.5	24.4	34.0	19.7	19.3	4.9
B-32		OpenAI	288.6	400.0	63.4	48.7	56.0	69.4	44.2	42.3	**31.5**
B-32		DataComp-1B	295.4	1400.0	**69.2**	**56.3**	**60.8**	**78.2**	**55.2**	**56.8**	30.5
B-32		EntityNet	13.1	32.7	61.5	41.0	53.6	58.8	32.6	45.0	14.9
B-16	⋆	BioCLIP	61.3	10.4	18.6	15.4	17.7	16.0	3.2	7.3	32.9
B-16		CommonPool-L	78.2	1280.0	57.8	47.0	50.0	68.4	49.1	45.9	21.7
B-16		DataComp-L	78.2	140.0	63.1	51.1	55.2	71.8	53.1	49.7	25.5
B-16		DataComp-1B	791.4	1400.0	**73.5**	**64.4**	**66.0**	**83.6**	**63.7**	**60.4**	48.4
B-16		OpenAI	784.6	400.0	68.3	58.6	61.9	77.7	55.3	48.2	**49.9**
B-16		EntityNet	36.0	32.7	66.2	47.4	59.2	64.1	40.9	48.9	23.9
L-14		OpenAI	3328.4	400.0	75.5	71.4	69.9	87.9	69.0	59.6	**70.7**
L-14		DataComp-1B	3338.6	1400.0	79.2	74.9	72.0	90.8	**74.3**	68.0	69.6
L-14		DFN-2B	3338.6	2000.0	81.4	74.8	74.6	90.0	74.1	68.3	66.8
H-14		DFN-5B	22164.0	5000.0	**83.4**	**76.3**	**77.4**	**93.0**	68.4	**72.8**	69.9
B-32		DataComp-1B	295.4	1400.0	69.2	**56.3**	60.8	**78.2**	**55.2**	**56.8**	**30.5**
B-32	⋆	EntityNet	13.3	32.7	**69.5**	53.3	**61.9**	74.2	47.9	**56.8**	25.6
B-32	⋆	Only organisms	4.2	10.2	48.2	33.3	43.1	56.7	19.1	32.1	15.8
B-16		DataComp-1B	791.4	1400.0	**73.5**	**64.4**	66.0	**83.6**	**63.7**	**60.4**	**48.4**
B-16	⋆	EntityNet	36.1	32.7	**73.5**	61.0	**66.5**	79.0	56.6	59.8	42.9
B-16	⋆	Only organisms	11.3	10.2	51.4	39.2	46.1	62.1	25.9	35.1	26.8
B-32		EntityNet	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		No organism	9.0	22.5	39.2	28.0	33.6	37.7	29.1	32.0	7.5
B-32		Only organisms	4.1	10.2	36.0	21.0	31.5	39.6	8.1	17.6	8.1
B-32		No attributes	8.7	21.8	54.8	33.8	47.9	49.2	24.4	36.9	10.5
B-32		50% alt text	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		100% alt text	13.1	32.7	59.1	38.1	51.1	55.6	30.1	40.8	13.1
B-32		0% alt text	13.1	32.7	55.7	35.5	48.0	53.0	27.1	36.8	12.4
B-32		Full size	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		1/2 size	6.6	16.4	54.1	33.6	47.2	51.0	24.4	36.5	8.9
B-32		1/4 size	3.3	8.2	45.2	25.7	39.1	39.9	18.1	25.6	5.9
B-32		1/8 size	1.6	4.1	33.3	17.7	28.5	28.7	12.0	15.3	4.2
B-32		1/16 size	0.8	2.0	19.9	10.0	17.2	17.6	6.8	5.7	2.8
B-32		Batch size 2k	13.1	32.7	59.7	40.4	52.5	59.4	31.3	44.5	14.4
B-32		Batch size 4k	13.1	32.7	60.9	**41.0**	53.0	**60.0**	32.1	44.9	**14.9**
B-32		Batch size 8k	13.1	32.7	**61.5**	**41.0**	**53.6**	58.8	**32.6**	**45.0**	**14.9**
B-32		Batch size 16k	13.2	32.7	60.5	39.7	53.2	58.0	30.6	42.9	13.8
B-32		Batch size 32k	13.3	32.7	58.6	37.2	51.3	54.6	27.9	40.6	11.6

Table 10: Hyperparameters used for pretraining and finetuning, unless otherwise stated. For all experiments we use the AdamW [53] optimizer with $\epsilon = 1e-8$, $\beta_1 = 0.9$, $\beta 2 = 0.98$.

Dataset	Model	Batch Size	Learning Rate	Weight Decay	Epochs	Warmup epochs
CC12M	ViT-B/32	8k	5e-4	0.2	18	2
CC12M	ViT-B/16	8k	5e-4	0.2	18	2
Ours, pretraining	ViT-B/32	8k	5e-4	0.2	18	2
Ours, pretraining	ViT-B/16	8k	5e-4	0.2	18	2
Ours, finetuning	ViT-B/32	32k	5e-5	0.2	18	2
Ours, finetuning	ViT-B/16	32k	5e-5	0.2	18	2

Table 11: Example of our text label sampling strategy for an image returned from the entity query of zipper. Probability mass is split 50/50 between image alt texts and texts from the knowledge graph. Between alt texts, we chose uniformly. Between knowledge graph texts we chose the search query 25% of the time, a description 10% of the time (uniformly between descriptions), and an alias otherwise (uniformly between all aliases).

Text	Chance	Source
Zipper PNG	25%	Alt text
yellow zipper PNG image	25%	Alt text
zipper	12.5%	Search query
zip	5.5%	Alias
dingy	5.5%	Alias
clasp locker	5.5%	Alias
fly	5.5%	Alias
zip fastener	5.5%	Alias
device for fastening the edges of an opening of fabric or other flexible material	2.5%	Description
A device used for fastening, typically made of physical material.	2.5%	Description

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT
  ?ent
  ?label
  ?desc
  ?links
  (GROUP_CONCAT(DISTINCT ?alias; SEPARATOR=";;;") AS ?aliases)
WHERE {
  VALUES ?typ { wd:Q42889 }
  ?ent wdt:P279* ?typ .
  ?ent rdfs:label ?label .
  FILTER(LANG(?label) = "en")
  ?ent ^schema:about/wikibase:sitelinks ?links .
  FILTER(?links >= 5)
  OPTIONAL {
    ?ent schema:description ?desc .
    FILTER(LANG(?desc) = "en")
  }
  OPTIONAL {
    ?ent skos:altLabel ?alias .
    FILTER(LANG(?alias) = "en")
  }
}
GROUP BY ?ent ?label ?desc ?links
ORDER BY DESC(?links)
```

Fig. 5: Generic SPARQL query for extracting entities from Wikidata that are related to a given set of super-entities. The super-entities are manually set within the *VALUES ?typ { ... }* clause. In this example it is the motor car entity wd:Q42889. A minimum number of sitelinks can also be specified to filter out unpopular entities, here it is set to 5.

Table 12: Vehicle entities and accompanying information as extracted from the Wikidata knowledge graph. Showing the first 5 and last 5 out of 17,015 entities. Note that we only collect entities with sitelinks ≥ 5. The corresponding SPARQL query is shown in Figure 5.

Identifier	Name	Description	Sitelinks	Aliases
Q1420	motor car	motorized road vehicle designed to carry one to eight people rather than primarily goods	237	auto, motor vehicle, motor cars, motorcar, cars, car, automobiles, automobile, autocar
Q11442	bicycle	pedal-driven two-wheel vehicle	203	bike, Bicycles, cycle, pushbike, pedal cycle, pedal bike
Q197	airplane	powered fixed-wing aircraft	196	airplane, aeroplane, plane, powered fixed-wing aircraft, planes, plane, aeroplane, fixed-wing powered aircraft, fixed-wing airplane, aeroplanes, fixed-wing aeroplane, airplanes
Q870	train	form of rail transport consisting of a series of connected vehicles	193	rail-train, trains, railway train, railtrain, rail train, railroad train
Q11446	ship	large buoyant watercraft	178	marine vessel, vessel, water vessel, ships
Q813876	Bedford JJL	motor vehicle	5	
Q7077241	Odakyu 20000 series RSE	Japanese electric multiple unit trainset	5	RSE, Romancecar RSE, Resort Super Express, Odakyu Romancecar RSE, 20000 series
Q812263	Bavarian Pt 2/3	class of 97 German 2-4-0T locomotives	5	ÖBB 770, DR Class 70.0, DRG Class 70.0
Q9177196	Bombardier CRJ1000	regional jet airliner	5	CRJ1000
Q812260	Bavarian PtL 2/2	class of 6+29+13 German 0-4-0T locomotives	5	DB Class 98.3, DRG Class 98.3, ÖBB 688

Table 13: The super-entities for building our EntityNet dataset to describe the visual world. The *aliases* column refers to the set of all aliases collected from the entities. The numbers in this table are slightly higher than the ones we report in the main paper, because they refer to the raw counts of entities and aliases before profanity filtering and the removal of entities that return no results in the image search.

Super-entity	Description	Examples	Entities	Aliases
product	Anything that can be offered to a market	banh mi, navigation system	63,676	144,715
substance	Any composed matter whose origin is either biological, chemical, or mineral	solid lubricant, Chinese tea	34,259	111,383
physical tool	Physical item that can be used to achieve a goal	Patient lift, police transport	32,727	71,227
animal	Kingdom of multicellular eukaryotic organisms	saw-scaled viper, Sporathraupis cyanocephala	28,000	76,408
plant	Living thing in the kingdom of photosynthetic eukaryotes	Whitebark Pine, Eucalyptus coccifera	28,000	55,925
material	Substance that can occur in different amounts, all with some similar [mixture of some] characteristics, and with which objects can be made	dietary proteins, stone slab tomb	18,021	40,822
vehicle	Mobile machine used for transport, whether it has an engine or not, including wheeled and tracked vehicles, air-, water-, and space-craft	shipwrecks (objects), Evergreen A-class container ship	17,015	37,849
geographical feature	Components of planets that can be geographically located	hydrothermal Vents, grooves	8,683	19,030
food	Any substance consumed to provide nutritional support for the body; form of energy stored in chemical	coffee milk, tikka	8,464	15,332
architectural structure	Human-designed and -made structure	rock temples, summerhouse	4,507	10,354
anatomical structure	Entity with a single connected inherent 3d shape that's created by coordinated expression of the organism's own dna	bronchi, maxillary wisdom tooth	4,394	9,999
facility	Place, equipment, or service to support a specific function	public toilet, automobile servicing shop	2,767	6,740
physical activity	Human physical activity consisting of voluntary bodily movement by skeletal muscles	American rules football, archery	2,228	4,422
clothing	Covering worn on the body	blucher shoe, G-suit	1,929	4,313
building	Structure, typically with a roof and walls, standing more or less permanently in one place	shoestore, family restaurant	1,655	3,964
musical instrument	Device created or adapted to make musical sounds	electroencephalophone, Chinese flutes	1,450	3,493
organ	Collection of tissues with similar functions	nasal bone, cranial nerves	1,155	2,450
furniture	Movable objects used to equip households, offices, or shops for purposes such as storage, seating, sleeping	faldstool, airline seat	388	933
body of water	Any significant accumulation of water, generally on a planet's surface	dammed lake, deep-sea hydrothermal vent	379	792
weather	State of the atmosphere	cold snap, tropical cyclone	151	304
precipitation	Liquid or solid water that falls to the ground	hail, thunderstorm	43	72
Total	Before deduplication		259,891	620,527
Total	After deduplication		146,985	368,062
Total	After deduplication, without animals and plants		90,985	235,795

Table 14: We consider these super-entities either non-visual, irrelevant, or too specific and do not select related entities when building our dataset.

Super-entity	Description
abstract entity	entity that does not have a physical existence, including abstract objects and properties
astronomical object	physical body of astronomically-significant size, mass, or role, naturally occurring in a universe
city	large human settlement
concept	semantic unit understood in different ways, e.g. as mental representation, ability or abstract object (philosophy)
continent	large landmass identified by convention
country	distinct territorial body or political entity
historical event	particular incident in history that brings about a historical change
history	past events and their tracks or records
imaginary character	character known only from narrations (fictional or in a factual manner) without a proof of existence; includes fictional, mythical, legendary or religious characters and similar
language	particular system of communication, often named for the region or peoples that use it
language	structured system of communication
medical procedure	process of medicine done to heal; course of action intended to achieve a result in the delivery of healthcare
organization	social entity established to meet needs or pursue goals
planet	celestial body directly orbiting a star or stellar remnant
religion	social-cultural system
representation	entity or process that portrays something else, usually in a simplified or approximated manner
role	social role with a set of powers and responsibilities within an organization
science	systematic endeavor that builds and organizes knowledge, and the set of knowledge produced by this system
social system	patterned series of interrelationships existing between individuals, groups, and institutions
speciality	field limited to a specific area of knowledge; specialization in an occupation or branch of learning; a specific use
star	astronomical object consisting of a luminous spheroid of plasma held together by its own gravity
temporal entity	thing that can be contained within a period of time, or change in state (e.g. events, periods, acts)
work of art	aesthetic item or artistic creation; object whose value is its beauty only, not practical usefulness
written work	any work expressed in writing, such as inscriptions, manuscripts, documents or maps

References

1. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/radford21a.html
2. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 34892–34916. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf
3. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved Baselines with Visual Instruction Tuning (2023)
4. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: LLaVA-NeXT: Improved reasoning, OCR, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
5. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023), https://arxiv.org/abs/2307.01952
6. Gadre, S.Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., Orgad, E., Entezari, R., Daras, G., Pratt, S., Ramanujan, V., Bitton, Y., Marathe, K., Mussmann, S., Vencu, R., Cherti, M., Krishna, R., Koh, P.W.W., Saukh, O., Ratner, A.J., Song, S., Hajishirzi, H., Farhadi, A., Beaumont, R., Oh, S., Dimakis, A., Jitsev, J., Carmon, Y., Shankar, V., Schmidt, L.: DataComp: In search of the next generation of multimodal datasets. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 27092–27112. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/56332d41d55ad7ad8024aac625881be7-Paper-Datasets_and_Benchmarks.pdf
7. Fang, A., Jose, A.M., Jain, A., Schmidt, L., Toshev, A.T., Shankar, V.: Data Filtering Networks. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=KAk6ngZ09F
8. Li, Z., Xie, C., Cubuk, E.D.: Scaling (Down) CLIP: A Comprehensive Analysis of Data, Archite cture, and Training Strategies. arXiv preprint arXiv:2404.08197 (2024)
9. Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A.A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., et al.: Phi-3 technical report: A highly capable language model locally on your phone (2024), https://arxiv.org/abs/2404.14219
10. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: LAION-400M: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
11. Rana, A.: Common crawl – building an open web-scale crawl using hadoop (2010), https://www.slideshare.net/hadoopusergroup/common-crawlpresentation
12. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: An open large-scale dataset for training next generation image-text models. In: Koyejo,

S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 25278–25294. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/a1859debfb3b59d094f3504d5ebb6c25-Paper-Datasets_and_Benchmarks.pdf
13. Xu, H., Xie, S., Tan, X., Huang, P.Y., Howes, R., Sharma, V., Li, S.W., Ghosh, G., Zettlemoyer, L., Feichtenhofer, C.: Demystifying CLIP data. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=5BCFlnfE1g
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
15. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: OpenCLIP (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773
16. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (Jun 2021). https://doi.org/10.1109/cvpr46437.2021.00356, http://dx.doi.org/10.1109/CVPR46437.2021.00356
17. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: YFCC100M: the new data in multimedia research. Commun. ACM **59**(2), 64–73 (Jan 2016). https://doi.org/10.1145/2812802, https://doi.org/10.1145/2812802
18. Stevens, S., Wu, J., Thompson, M.J., Campolongo, E.G., Song, C.H., Carlyn, D.E., Dong, L., Dahdul, W.M., Stewart, C., Berger-Wolf, T., Chao, W.L., Su, Y.: BioCLIP: A Vision Foundation Model for the Tree of Life. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19412–19424 (June 2024)
19. Encyclopedia of Life (July 2018), http://eol.org
20. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The iNaturalist Species Classification and Detection Dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
21. Gharaee, Z., Gong, Z., Pellegrino, N., Zarubiieva, I., Haurum, J.B., Lowe, S.C., McKeown, J.T.A., Ho, C.Y., McLeod, J., Wei, Y.C., Agda, J., Ratnasingham, S., Steinke, D., Chang, A.X., Taylor, G.W., Fieguth, P.: A step towards worldwide biodiversity assessment: The BIOSCAN-1M insect dataset. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 43593–43619. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/87dbbdc3a685a97ad28489a1d57c45c1-Paper-Datasets_and_Benchmarks.pdf
22. Li, R., Kim, D., Bhanu, B., Kuo, W.: RECLIP: Resource-efficient CLIP by Training with Small Images (2023)
23. Li, X., Wang, Z., Xie, C.: CLIPA-v2: Scaling CLIP Training with 81.1% Zero-shot ImageNet Accuracy within a $10,000 Budget; An Extra$4,000 Unlocks 81.8% Accuracy. arXiv preprint arXiv:2306.15658 (2023)
24. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid Loss for Language Image Pre-Training. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE (Oct 2023). https://doi.org/10.1109/iccv51070.2023.01100, http://dx.doi.org/10.1109/ICCV51070.2023.01100

25. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
26. Wan, B., Tschannen, M., Xian, Y., Pavetic, F., Alabdulmohsin, I.M., Wang, X., Susano Pinto, A., Steiner, A., Beyer, L., Zhai, X.: Locca: Visual pretraining with location-aware captioners. Advances in Neural Information Processing Systems **37**, 116355–116387 (2024)
27. Naeem, M.F., Xian, Y., Zhai, X., Hoyer, L., Van Gool, L., Tombari, F.: Silc: Improving vision language pretraining with self-distillation. In: European Conference on Computer Vision. pp. 38–55. Springer (2024)
28. Udandarao, V., Parthasarathy, N., Naeem, M.F., Evans, T., Albanie, S., Tombari, F., Xian, Y., Tonioni, A., Hénaff, O.J.: Active data curation effectively distills large-scale multimodal models. arXiv preprint arXiv:2411.18674 (2024)
29. Vasu, P.K.A., Pouransari, H., Faghri, F., Vemulapalli, R., Tuzel, O.: MobileCLIP: Fast Image-Text Models through Multi-Modal Reinforced Training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2024)
30. Chen, J., Yu, Q., Shen, X., Yuille, A., Chen, L.C.: Design scalable vision models in the vision-language era. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
31. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A.V., Bradbury, J., Kuo, W., Seyedhosseini, M., Jia, C., Ayan, B.K., Ruiz, C.R., Steiner, A.P., Angelova, A., Zhai, X., Houlsby, N., Soricut, R.: PaLI: A Jointly-Scaled Multilingual Language-Image Model. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=mWVoBz4W0u
32. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. Communications of the ACM **57**, 78–85 (2014), http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext
33. Peshterliev, S., Dupuy, C., Kiss, I.: Self-attention gazetteer embeddings for named-entity recognition. arXiv preprint arXiv:2004.04060 (2020)
34. Fellbaum, C.: WordNet: An Electronic Lexical Database. Language, Speech and Communication, Mit Press (1998), http://books.google.at/books?id=Rehu8OOzMIMC
35. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
36. Pizzi, E., Roy, S.D., Ravindra, S.N., Goyal, P., Douze, M.: A Self-Supervised Descriptor for Image Copy Detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (Jun 2022). https://doi.org/10.1109/cvpr52688.2022.01413, http://dx.doi.org/10.1109/CVPR52688.2022.01413
37. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (2015). https://doi.org/10.1007/s11263-015-0816-y

38. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15262–15271 (2021)
39. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8340–8349 (2021)
40. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. Advances in neural information processing systems **32** (2019)
41. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: International conference on machine learning. pp. 5389–5400. PMLR (2019)
42. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. Advances in neural information processing systems **32** (2019)
43. Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., Schmidt, L.: Measuring robustness to natural distribution shifts in image classification. Advances in Neural Information Processing Systems **33**, 18583–18599 (2020)
44. Parashar, S., Lin, Z., Li, Y., Kong, S.: Prompting scientific names for zero-shot species recognition. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023), https://openreview.net/forum?id=OgK0kMz5Va
45. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
46. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3128–3137 (2015)
47. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. pp. 740–755. Springer (2014)
48. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the association for computational linguistics **2**, 67–78 (2014)
49. Thapliyal, A.V., Pont-Tuset, J., Chen, X., Soricut, R.: Crossmodal-3600: A massively multilingual multimodal evaluation dataset. arXiv preprint arXiv:2205.12522 (2022)
50. Consortium, T.U.: Uniprot: the universal protein knowledgebase in 2025. Nucleic Acids Research **53**(D1), D609–D617 (11 2024). https://doi.org/10.1093/nar/gkae1010, https://doi.org/10.1093/nar/gkae1010
51. Chen, Y., Kuang, J., Cheng, D., Zheng, J., Gao, M., Zhou, A.: Agrikg: An agricultural knowledge graph and its applications. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) Database Systems for Advanced Applications. pp. 533–537. Springer International Publishing, Cham (2019)
52. Xu, H., Huang, P.Y., Tan, X.E., Yeh, C.F., Kahn, J., Jou, C., Ghosh, G., Levy, O., Zettlemoyer, L., tau Yih, W., Li, S.W., Xie, S., Feichtenhofer, C.: Altogether: Image captioning via re-aligning alt-text. In: Conference on Empirical Methods in Natural Language Processing (2024)

53. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7