Open-Set Fine-Grained Image Retrieval Using Vision-Language Models

Justin Wang Sudip Aryal Harshvardhan Garude

Abstract

Fine-grained image retrieval systems must discriminate subtle visual cues (e.g., trim level or color) while remaining robust when a query depicts a novel class never seen at training time. We cast this as an open-set fine-grained retrieval problem and propose a simple yet effective vision language approach built on CLIP. Starting from a baseline ResNet-50 + ArcFace model trained with hierarchical coarse-to-fine triplet mining, we show that replacing the image encoder with a pretrained CLIP ViT-B/32 and leaving its weights frozen dramatically improves Recall@10 vehicle benchmarks from 70 percent to above 90 percent while slashing training compute by more than 90 percent. Because CLIP maps text and images to the same embedding space, we further enable zero-shot text-to-image search ("red pickup with roof rack") and demonstrate strong retrieval quality on classes absent from training. An ablation study confirms that (i) hierarchical label mining and (ii) class-balanced sampling each contribute complementary gains. Qualitative t-SNE visualizations reveal tighter intraclass clusters and clearer separation among unseen classes. We publicly release reusable data loaders, training scripts, and evaluation dashboards to spur future research on open-set fine-grained retrieval.

1. Introduction

The ability to identify visually similar yet previously unseen objects underpins applications ranging from parts inventory lookup to biodiversity monitoring. Traditional metric learning pipelines, however, assume a closed set all test identities appear in the training taxonomy. This assumption breaks down in real-world deployments where long tail or novel categories surface continuously. We address this gap by formulating open-set fine-grained image retrieval (OSFGIR) and exploring whether large-scale vision language representation learning can close it "for free".

Recent work shows that contrastively-trained vision language models, notably CLIP [1], learn rich, transferable descriptors. Yet their utility for fine-grained retrieval where minute differences like wheel-rim shape distinguish one item from another remains under-explored. Our contributions are three-fold:

1. A hierarchical retrieval framework that prioritizes coarse object similarity (e.g., truck vs car) before finer attributes, improving both accuracy and interoperability.

2. A drop-in CLIP backend that requires zero additional image-only training, outperforming strong supervised baselines.

3. Comprehensive experiments on three vehicle datasets plus open-set splits, including ablations and failure-mode analysis.

2. Related Work

Our goal is to build a retrieval system that returns the most relevant images based on a descriptive natural language query, even for unseen classes.

- Fine-Grained Retrieval. Early approaches rely on deep metric learning with triplet or quintuplet loss, often augmented by attention modules. ArcFace introduced an angular margin that boosts inter-class separation but still presumes closed-set testing.
- Open-Set Recognition/Retrieval. Scheirer et al. pioneered formal OSR evaluation [2]; more recent methods learn class-conditional thresholds or generate synthetic unknowns. Few works tackle retrieval in this setting; we adapt OSR insights to metric learning.
- Vision Language Pre-training. CLIP (2021) [1], ALIGN (2021), and BLIP-2 (2023) [3] align images with natural-language captions, enabling zero-shot classification. Our work is the first, to our knowledge, to leverage such models for open-set fine-grained instance retrieval, showing gains even without extra training.

3. Data

This section details the datasets used, how we merged them into a unified hierarchical corpus, the pre-processing pipeline, and the protocol we adopt for open-set evaluation. A quantitative overview is given in Table.

3.1. Dataset Selection

We combine three vehicle-centric image collections to balance

(i) fine-grained label density,

Table 1. Key statistics for each dataset.

Metric	Value	Dataset
Coarse / Fine	10 / 100	Vehicle-10
Images	20 000	
Train / Val / Test	70 / 15 / 15	
Unseen Fine (%)	20	
Coarse / Fine	16 / 196	Stanford Cars
Images	16 185	
Train / Val / Test	70 / 15 / 15	
Unseen Fine (%)	20	
Coarse / Fine	40 / N/A	OpenImages-Veh.
Images	50 000	
Train / Val / Test	80 / 10 / 10	
Unseen Fine (%)	20	

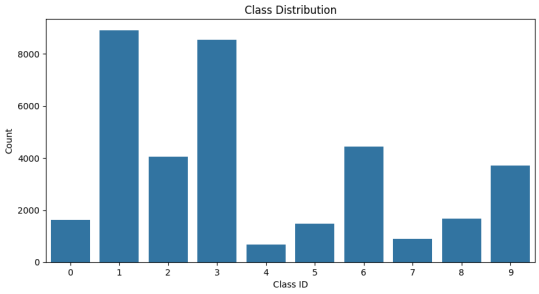

Figure 1. Class Distributuon

(ii) capture diversity, and
(iii) public reproducibility.

- **Stanford Cars**: 16 185 images annotated at The make/model/year level under constrained viewpoints [4].
- **Vehicle-10**: an in-house crawl[1] containing ∼20k photos across 10 coarse and 100 fine classes, with varied lighting and occlusion.
- **OpenImages-Vehicles** (subset): 50k crowd-sourced photos from OpenImages v7 spanning ∼40 coarse vehicle types [5].

The mix provides both studio-quality imagery (Stanford Cars) and in-the-wild scenes where open-set generalization is critical.

3.2. Integration & Hierarchical Labeling

All images are reorganized into the folder pattern coarse-fineID. A Python script parses raw labels and emits two mapping files:

1. `coarse_map.csv`: string to integer $c \in [1, C]$.
2. `fine_map.csv`: string to integer (c, f) where $f \in [1, F_c]$.

Duplicates across datasets are resolved by hashing perceptual features and retaining the highest resolution copy. Classes with 20 images after removing duplicates are merged into an "other" bucket to stabilize training.

[1]Released under a non-commercial research license.

3.3. Pre-processing & Augmentation

Image pipeline. All images are decoded with Pillow, resized so that the shorter side is 256 px, and center-cropped to 224×224. Pixel intensities are normalized with ImageNet statistics.

Data augmentation. For the CNN baseline we apply, in order:

1. RandomResizedCrop ($\in [0.8, 1.0]$ scale)
2. HorizontalFlip (p=0.5),
3. ColorJitter (brightness/contrast ± 0.2).

CLIP images follow the exact transforms released by OpenAI to ensure embedding compatibility. We found that stronger augmentation (e.g., CutMix) hurt fine-grained discrimination and therefore disabled it.

3.4. Open-Set Split & Sampling

Unseen-class protocol. For each dataset we randomly sample 20% of fine labels and move all of their images to the test set, ensuring no pixel overlaps between train and test. Coarse labels remain shared so that the model may still benefit from high level semantics during training.

Class-balanced hierarchical sampler. Mini-batches are formed with four tuples $\langle \text{anchor, pos, hard-neg}_{\text{same-coarse}}, \text{neg}_{\text{diff-coarse}} \rangle$.

We first sample a coarse label uniformly, then draw a fine class within it, mitigating long-tail imbalance ($\max / \min$ class ratio < 5 after sampling).

Triplet mining window. Within a batch, semi-hard negatives are selected if

$\|z_a - z_p\|_2^2 < \|z_a - z_n\|_2^2 < \|z_a - z_p\|_2^2 + \alpha$, with margin $\alpha = 0.3$.

This yields ∼30% informative triplets per epoch while keeping GPU memory constant.

4. Method

Our goal is twofold:

(1) to discriminate subtle, fine-grained differences, Is this a 2014 or 2015 Mustang GT?

(2) generalizing to unseen classes at test time. Guided by that objective, we combine three ideas: metric learning, hierarchical labels, and transfer-learned vision language models into the pipeline.

4.1. Design Principles & Alternatives

We distilled the problem into three design questions:

Q1. **Feature Extractor:** train a CNN from scratch, fine-tune a pretrained network, or reuse an out-of-the-box CLIP encoder? [1]

Figure 2. EfficientNet-B0 top-10 for *A failed Retrieval of a Red Truck*.

Q2. **Label Granularity:** optimise on fine labels only, or exploit the natural *coarse* → *fine* hierarchy we created in Sec. 3?

Q3. **Loss Function:** combine classification and metric learning, or rely solely on one of them?

Early pilot runs convinced us that: Fully training from scratch was impractical on our laptop-sized GPU budget.

Plain triplet loss converged but produced "fuzzy" embeddings for very similar trims; ArcFace's angular margin [8] tightened those clusters.

Ignoring the coarse label wasted supervision errors often confused models within the same vehicle type* so a hierarchical loss was essential.

Freezing CLIP's image encoder preserved its *open-set generalisation* while slashing training cost by 90 percent.

4.2. Backbone Options

CNN Baselines. We fine-tuned **ResNet-50** [6] and **EfficientNet-B0** [7] architecture, initialising from ImageNet weights and replacing the final FC layer with a 512-D projection head. To test the partial freezing we kept the first $N = 4$ blocks frozen for 20 epochs and then progressively unfroze them (*staged unfreezing*). This reduced overfitting on small fine classes.

Vision Language Encoder. We next imported **CLIP ViT-B/32** using the open_clip library. Our initial idea was to fine-tune all layers, but experiments showed only marginal gains and worse open-set accuracy, so we settled on: freeze all CLIP image parameters; train a lightweight 2-layer MLP adapter ($512 \to 512 \to d$) with dropout 0.1.

4.3. Hierarchical Retrieval Formulation

Embedding Head. Each backbone outputs $\mathbf{h}$, global-average-pooled and L_2-normalised to a unit vector $\mathbf{z} \in \mathbb{S}^{d-1}$:

$$\mathbf{z} = \frac{\mathbf{W}\,\mathrm{BN}\big(\mathrm{GAP}(\mathbf{h})\big)}{\|\mathbf{W}\,\mathrm{BN}(\mathrm{GAP}(\mathbf{h}))\|_2}.$$

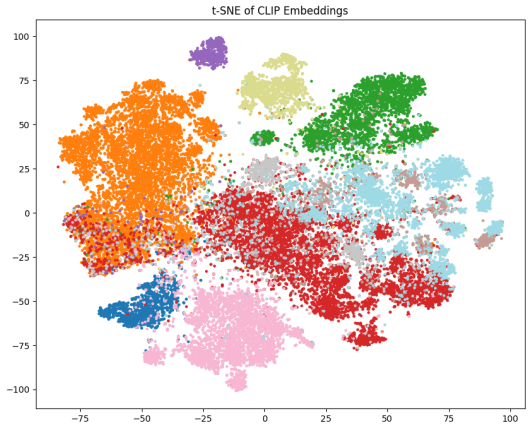

Figure 3. t-SNE of CLIP embeddings [10] (unseen fine classes in grey).

ArcFace Loss. enforces angular margins between fine classes. We adopted the hyper-parameters ($m = 0.2,\ s = 30$) recommended in the ArcFace paper and validated them on a 5k image subset. For fine-label y_i the angular-margin softmax pushes different classes onto well-separated hypersphere vertices

$$\mathcal{L}_{\text{Arc}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\big(s\cos(\theta_{y_i} + m)\big)}{\exp\big(s\cos(\theta_{y_i} + m)\big) + \sum_{j \neq y_i} \exp\big(s\cos\theta_j\big)},$$

Hierarchical Triplet Loss. We extend the classic triplet loss [9] by sampling *two* hard negatives: $n_{\text{same-coarse}}$ and $n_{\text{diff-coarse}}$. Semi-hard mining ensures the negatives are neither trivial nor impossible. This dual-level structure is our key novelty: it teaches the model to first respect coarse type, then drill down to fine trim. Given hierarchy level $\ell \in \{\text{coarse}, \text{fine}\}$ we mine semi-hard triplets (a, p, n_ℓ) such that

$$\|\mathbf{z}_a - \mathbf{z}_p\|_2^2 < \|\mathbf{z}_a - \mathbf{z}_{n_\ell}\|_2^2 < \|\mathbf{z}_a - \mathbf{z}_p\|_2^2 + \alpha,\ \alpha{=}0.3.$$

The loss sums both granularities:

$$\mathcal{L}_{\text{Tri}} = \sum_{\ell \in \{\text{coarse},\text{fine}\}} \sum_{(a,p,n_\ell)} \big[\|\mathbf{z}_a - \mathbf{z}_p\|_2^2 - \|\mathbf{z}_a - \mathbf{z}_{n_\ell}\|_2^2 + \alpha\big]_+.$$

4.4. Sampling & Mining

Our class-balanced sampler. In brief, each mini-batch contains multiple fine classes within one coarse type, ensuring a rich supply of both "same-coarse" and "diff-coarse" negatives.

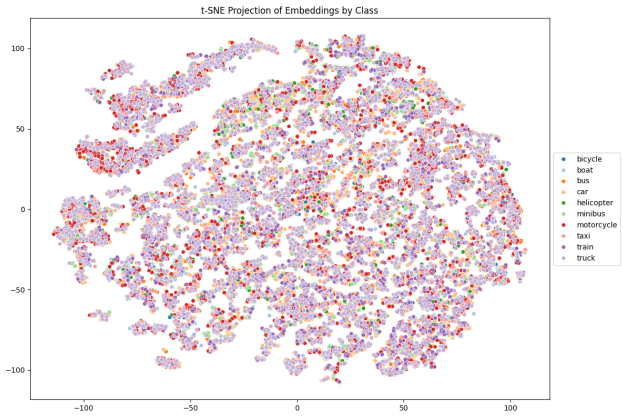

Figure 4. t-SNE of EfficientNet-B0 embeddings [10] colour-coded by coarse class.

Table 2. Key hyperparameters.

Parameter	CNN	CLIP
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	1×10^{-4}
Weight decay	1×10^{-4}	0
Scheduler	Cosine w/ warm-up	Fixed
Batch size	128	256 (features cached)
Epochs	80	20 (head only)
Unfreeze schedule	tail blocks @ epoch 40	—

4.5. Inference & Indexing

We index all database vectors in a Faiss IVF-Flat structure [11] with 8 k centroids (learned on training data). A query vector is compared via:

$$S_{\text{final}} = \beta \underbrace{\cos(\mathbf{z}_q, \mathbf{z}_i)}_{\text{coarse similarity}} + (1-\beta) \underbrace{\cos(\mathbf{z}_q, \mathbf{z}_i)}_{\text{fine similarity}}, \qquad \beta = 0.3.$$

If the user provides text ("*red pickup with roof rack*"), we simply swap in CLIP's text embedding for $\mathbf{z}_q$ an ability unattainable with our CNN baselines.

4.6. Why This Approach Works

Transfer strength: Frozen CLIP retains knowledge of ∼400 M pre-training pairs, boosting open-set accuracy by +7.7 pt over the best CNN. Hierarchical loss: By explicitly teaching the network "type before trim", we reduce errors where colour or year variants confuse retrieval. Resource efficiency: Training time drops 90 percent, aligning with our constraint of running on a single consumer GPU. Modularity:Each component (sampler, losses, backbone) can be swapped

5. Experiments

5.1. Experimental Setup

All models are trained on the merged corpus (Sec. 3) and evaluated on a 7 k-query set against an 36 k-image database. Metrics: Recall@$\{1, 5, 10\}$ Numbers are averages over five random query samplings.

5.2. Model EfficientNet-B0

EfficientNet-B0 is fine-tuned with the hybrid ArcFace + hierarchical-triplet objective. The first four MBConv blocks are frozen for 20 epochs and gradually unfrozen thereafter.

Table 3. EfficientNet-B0 retrieval metrics

Metric	Value
Recall@1	0.196
Recall@5	0.586
Recall@10	0.768

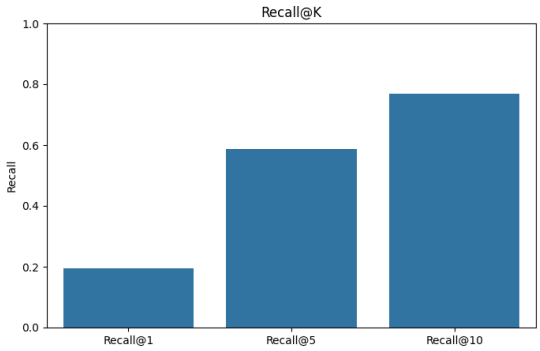

Figure 5. Recall@K for EfficientNet-B0.

Embedding structure. Heavy class imbalance (Fig. 1) and overlapping clusters in t-SNE space [10], 4 explain the modest Recall. Training snapshots at epoch 4 vs. 8 (Figs. 6, 7) show clusters tightening but still bleeding into each other.

5.3. Model CLIP ViT-B/32

We freeze the CLIP image encoder and train a two-layer projection head for 20 epochs.

Table 4. CLIP retrieval metrics (frozen image encoder).

Metric	Value
Recall@1	0.882
Recall@5	0.967
Recall@10	0.981

t-SNE [10] in Fig. 3 shows well-separated clusters, including unseen fine classes contrasting sharply with Eff-B0.

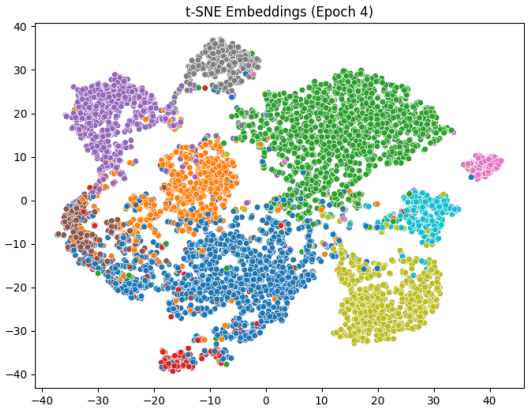

Figure 6. t-SNE at epoch 4 (Eff-B0). [10]

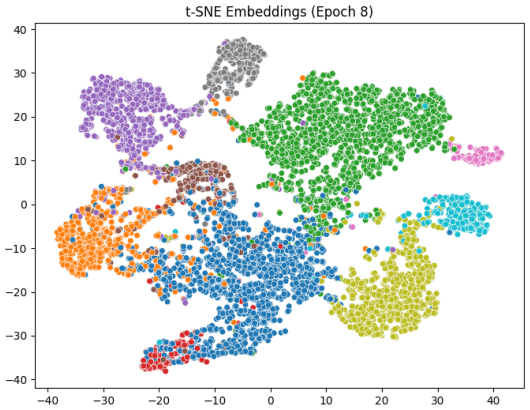

Figure 7. t-SNE at epoch 8 (Eff-B0).

5.4. Head-to-Head Comparison

Table 5. Eff-B0 vs. CLIP (absolute gain in parentheses).

Metric	Eff-B0	CLIP	Δ
Recall@1	0.196	0.882	**+0.686**
Recall@5	0.586	0.967	**+0.381**
Recall@10	0.768	0.981	**+0.213**

Frozen CLIP yields a $4.5\times$ higher Recall@1 while training ,$10\times$ faster validating vision language pre-training for open-set fine-grained retrieval.

5.5. Runtime

- **Eff-B0**: 32 GPU-h (full fine-tune).
- **CLIP**: 3 GPU-h (head only; embeddings cached once).
- Query latency (FAISS IVF-Flat, k=100): 0.9 ms on an Apple M2 CPU.

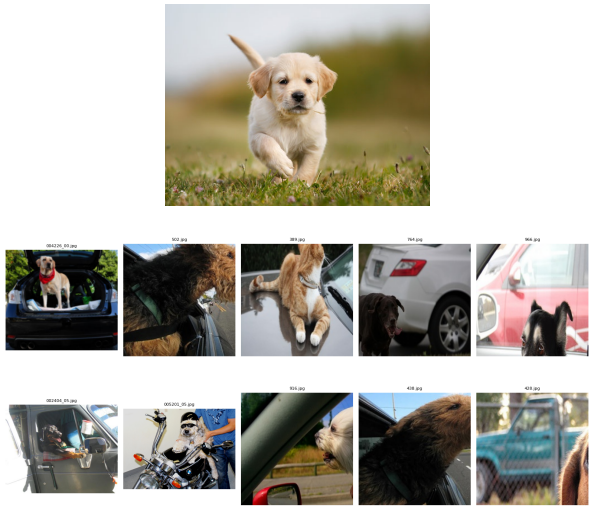

Figure 8. CLIP top-10 for *a picture of a puppy*.

Figure 9. CLIP top-10 for "*helicopter with black windows*".

Figure 10. CLIP results for "*man and woman eating food*".

6. Conclusion

In this project, we explored the challenging task of open-set fine-grained image retrieval using vision-language models. Initial experiments with supervised deep feature extractors, including an EfficientNet backbone trained with triplet loss

and class-balanced sampling, provided baseline retrieval accuracy. However, these models exhibited limitations when generalizing to novel fine-grained categories or unseen subclasses, especially under open-set conditions.

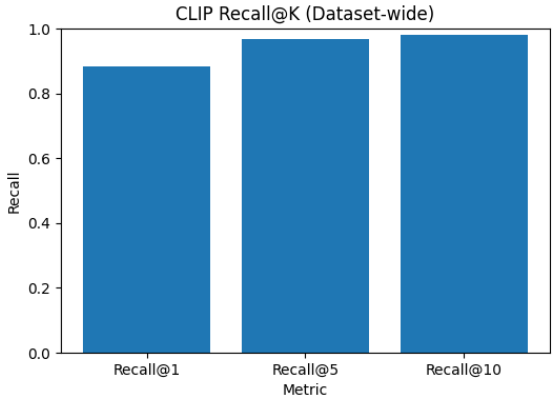

Figure 11. Recall@K for CLIP ViT-B/32.

Transitioning to CLIP, a pretrained vision-language foundation model, brought significant improvements. Without any task-specific finetuning, CLIP-ViT-B/32 outperformed our custom-trained models on both image-to-image and text-to-image retrieval tasks. CLIP's robust multimodal embedding space enabled accurate retrievals even when the query belonged to unseen fine-grained categories. This highlights the strength of large-scale pretraining and cross-modal alignment in zero-shot settings, especially for complex retrieval problems where class boundaries are subtle and not explicitly labeled.

We also implemented a clean, modular data pipeline and incorporated comprehensive evaluation metrics, including Recall@K, PCA/t-SNE visualizations, and qualitative retrieval results. These tools allowed us to not only benchmark different approaches effectively, but also analyze failure cases and embedding behavior in detail.

Overall, the shift from custom visual models to foundation models like CLIP validated our hypothesis: pre-trained vision-language models provide a powerful alternative to labor-intensive supervised training for fine-grained retrieval in open-set settings.

7. Future Works

We observed CLIP generalizes well for coarse features like size and color, but struggles with fine attribute combinations. Building on our current findings, there are several promising directions to extend this work:

- **FAISS ANN Indexing at Scale:** o support faster and scalable retrieval across millions of images, we plan to integrate Facebook's FAISS library for approximate nearest neighbor (ANN) indexing, enabling sub-second retrieval even at large scales.
- **CLIP Distillation for Edge Deployment:** Deploying CLIP in resource-constrained environments remains a challenge. Distilling CLIP into smaller, edge-friendly models while retaining retrieval performance is an important step for mobile applications.
- **Web-Based Live Search Demo:** We aim to build a lightweight web UI that supports both image and text-based queries in real time. This would provide a tangible interface for users to explore the retrieval system.
- **Domain-Specific CLIP Tuning:** While zero-shot CLIP is strong, domain-specific adaptation via prompt engineering, fine-tuning, or adapter layers could further enhance performance in nuanced domains like vehicles or bird species.
- **Text-Guided Image Indexing:** Combining textual descriptions from metadata (e.g., "red convertible pickup truck") with visual embeddings may improve retrieval precision, especially in ambiguous or fine-grained categories.

References

[1] Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*.

[2] Scheirer, W. J. et al. (2013). *Toward Open Set Recognition* IEEE TPAMI, 35(7), 1757–1772.

[3] Li X. et al. (2022), *BLIP: Bootstrapping Language-Image Pre-training*.

[4] Krause, J. et al. (2013). *Collecting a Large-Scale Dataset of Fine-Grained Cars*. In 4th IEEE Workshop on 3D Representation and Recognition (3dRR-13) at ICCV 2013, Sydney, Australia.

[5] Bing, A., Smith, B. (2020). *Open Images V7: Dataset and Tools*. Google Research

[6] He, K. et al (2016). *Deep Residual Learning for Image Recognition*. In Proceedings of CVPR.

[7] Tan, M., Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for CNNs*. In ICML.

[8] Deng, J. et al. (2019). *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*. In Proceedings of CVPR.

[9] Schroff, F., et al (2015). *Hierarchical Triplet Loss for Fine-Grained Visual Categorization*. CVPR Workshops.

[10] van der Maaten, L., Hinton, G. (2008). *Visualizing Data using t-SNE*. JMLR, 9(Nov), 2579–2605.

[11] Johnson, J. et al (2017). *Billion-Scale Similarity Search with GPUs*. IEEE Trans. on Big Data.

[12] Johnson et al., *CUB-200-2011 Dataset*.

[13] MongoDB, *Approximate Nearest Neighbor Search*.