Employing foundational models for zero shot image retrieval: study on MSLS London subset

Marko Haralović[1,2], **Onat Akca**[1], **Rushat Gabhane**[1]

[1]University of Twente, The Netherlands
[2]University of Zagreb, Croatia

1 Introduction

Image retrieval is a computer vision task of retrieving relevant images from a base of visual images for a given query image, a task relevant for localization, autonomous driving, and reconstruction. For that purpose, we are using the Mapillary Street Level Sequences (MSLS), specifically the London imagery subset, which consists of 1000 gallery and 500 query images. Our work focuses on non-gradient-based approaches to retrieve the highest possible number of relevant images for each query image. For that purpose, we explore both standard descriptor-based approaches, where we extract visual descriptors (ORB, SURF, and SIFT specifically), generate feature clusters based on that vision data, and use them to generate histograms that serve as image representations for retrieval. Such an approach was standard and commonly used in the pre–deep learning era. Nowadays, more examples have shown that using pretrained vision foundation models to generate features provides competitive performance out of the box, without any fine-tuning. For that purpose, we have decided to use DINOv3 models, Perception Encoder models, CLIP models, and StreetCLIP, out of which the latter has been fine-tuned out of domain compared to the MSLS dataset on street recognition data.

In this work, we explore the performance of histogram-based bag-of-words approaches compared to linear probing of vision backbones on the MSLS London subset. In addition, we perform ablations of the best-performing bag-of-words approach with respect to its hyperparameters. We also explore how different pooling strategies for DINOv3 and CLIP features perform on our dataset. Lastly, we investigate whether using multiple different vision backbones provides a performance advantage.

Finally, we present our centroid-based retrieval logic, where we assume the training data consists of image and GPS pairs, meaning that gallery images have geolocation and query images do not. In this setting, we cluster images based on GPS data and compute mean features for each centroid, using both centroid similarity and image-to-image similarity for retrieval.

In the end, we compare all our results and provide conclusions.

2 Related work

The task of visual place recognition (VPR) involves geolocalization of images given a reference database of geo-tagged images. A commonly reported baseline is image retrieval using a visual bag-of-words approach, where local visual descriptors are treated as visual words [9]. For such bag-of-words methods, SIFT [6], SURF [8], and ORB [7] features are often used as useful baselines. However, these traditional approaches lack robustness, fine-grained detail, and global context awareness, limiting their retrieval accuracy.

Compared to such low-level feature methods, deep learning approaches leverage convolutional neural networks trained in an end-to-end fashion, such as NetVLAD [10] and related architectures. More recent works have explored using pretrained models for zero-shot evaluation via linear probing or feature extraction. For instance, AnyLoc [11] uses a pretrained semantic segmentation model to generate a visual-language vocabulary based on pixel-level descriptors, while FM-Loc [12] leverages pretrained CNN embeddings. Other methods employ self-attention architectures for zero-shot place recognition [2].

Multiple Visual Foundation Models (VFMs) with semantically rich and dense features have been suggested to use in zero-shot manner on place recognition downstream task. Best performing examples include DINOv2 [14], DINOv3 [3], CLIP [13], the Perception Encoder [2], and SAM [15], among others. In the geolocation domain, StreetCLIP [4] fine-tuned CLIP for open domain street level image recognition.

Using DINOv2 as a backbone, several works have recently focused on finetuning for VPR. SALAD [16] proposed an optimal transport-based feature aggregation mechanism as a replacement for NetVLAD's soft assignment. SelaVPR [19] and CricaVPR [20] introduced trainable adapters integrated into the DINOv2 architecture. Both SelaVPR and CricaVPR incorporate GeM pooling for feature aggregation, which has been shown to outperform alternative pooling strategies such as max or mean pooling or the use of the [CLS] token.

In our work, we focus on evaluating DINOv3 [3], Perception Encoder [2], CLIP [13], and StreetCLIP [4] using zero-shot linear probing approach and we compare such approach to traditional bag-of-words approaches.

3 Methodology

3.1 BoW approach

Our baseline is BoW approach using ORB features. We compare performance of those features to SURF and SIFT features. Our method is that for each image in gallery we extract features. After going through collection of gallery images $\mathcal{G}$, we have features f_g or visual vocabulary, which are then clustered into k clusters using a clustering algorithm C. In inference, for each query image at input, we extract features and create histograms of centroids, by comparing the distances of features to k centroids. Such histogram counts are used as vectors then for retrieval, where distance metric d is used to compare similarity of gallery and query vectors, which are constructed as histogram counts for all the gallery and query images. Then for the best between ORB, SURF and SIFT features, we do an extended hyperparameter tuning, as well as ablation regarding the clustering algorithm C, distance measurement (between features and centroids) d and compare performance if we omitted *tf-idf* weighting.

3.2 Linear probing

In our linear probing setup, we start by using pretrained vision backbones from pretrained (foundational) models. We explore using various pooling mechanisms across network outputs, specifically we use mean pooling, max pooling, mean pooling without CLS token, using CLS token and GeM pooling. Formulas are:

Mean Pooling:

$$f_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

Max Pooling:

$$f_{\max} = \max_{i=1,\dots,N} x_i \tag{2}$$

Mean Pooling without CLS token:

$$f_{\text{mean-noCLS}} = \frac{1}{N-1}\sum_{i=2}^{N} x_i \tag{3}$$

CLS Token:

$$f_{\text{CLS}} = x_1 \tag{4}$$

GeM Pooling:

$$f_{\text{GeM}} = \left(\frac{1}{N}\sum_{i=1}^{N} x_i^p\right)^{1/p} \tag{5}$$

where N is the number of tokens, x_i represents the i-th token embedding, and p is the GeM pooling parameter (we use $p = 3$ for our experiments).

Our setup is simple: we freeze the model M, extract features for gallery and query images, apply pooling p, and perform image-to-image similarity based retrieval using cosine distance $\cos_d$. Such setup enabled us to quickly iterate and evaluate number of models. We used our approach across model sizes, which can be seen in the following tables:

Table 1: DINOv3 Models

Model	Arch.	Dim	Params
vits16	ViT-S/16	384	22M
vits16+	ViT-S/16+	384	28M
vitb16	ViT-B/16	768	86M
vitl16	ViT-L/16	1024	304M
vith16+	ViT-H/16+	1280	632M
convnext-T	CNX-T	768	28M
convnext-S	CNX-S	768	50M
convnext-B	CNX-B	1024	89M
convnext-L	CNX-L	1536	198M

Table 2: Perception Models

Model	Arch.	Dim	Params
PE-T16-384	ViT-T/16	512	23M
PE-S16-384	ViT-S/16	512	44M
PE-B16-224	ViT-B/16	1024	149M
PE-L14-336	ViT-L/14	1024	304M
PE-G14-448	ViT-G/14	1280	1.1B

Table 3: CLIP Models

Model	Arch.	Dim	Params
StreetCLIP	ViT-L/14	1024	304M
vit-large-p14	ViT-L/14	1024	304M
vit-base-p16	ViT-B/16	768	86M
vit-base-p32	ViT-B/32	768	88M

Given the number of models we used, our pooling comparison tables will be presented in 6, and in 4 we will present table of average percentage gains of using a pooling p.

3.3 Centroid-based Retrieval

All the previous approaches used only visual data for image retrieval. Since our images are GPS-annotated, we believe it is reasonable to assume that, in production, one would have an annotated database of (I_i, gps_i) pairs for $i \in \{1, \dots, |\mathcal{D}|\}$, and that GPS data could be used for retrieval. We further assume that query images are not GPS-annotated. For instance, when building a real-world application, this dataset of London can be used to create a retrieval system where the locations of images are only implicitly known. Our goal is to investigate whether having a priori knowledge of locations can be effectively leveraged when designing such a system.

We therefore, similarly to [17, 18], have decided to use a centroid-based approach, where we have two modes of operation.

General approach: Extract features for gallery images to construct f_g, then perform K-means clustering based on GPS data (assuming pairs (f_{g_i}, gps_i) for $i \in \mathcal{G}$). We then create k centroids, group f_g into clusters, and apply average pooling to obtain centroid representations. For a query image, we compare it to centroids c_i and get the most similar centroid, then perform retrieval using:

Approach A: Retrieve images firstly among the features in the matched centroid cluster, then if the number of images to retrieve is higher than the number of images in target cluster C_t, retrieve images in other clusters.

Approach B: Use the weighted similarity formula:

$$s(q, I_i) = \alpha \cdot \text{sim}(q, c_j) + (1 - \alpha) \cdot \text{sim}(q, I_i) \quad (6)$$

where q is the query image, I_i is a gallery image, c_j is the matched centroid, and $alpha \in [0, 1]$ is a weighting parameter.

3.4 Using Multiple Vision Backbones Output Concatenation

In the last part of our work, we are asking whether a pair of vision backbones extract visually different semantic data and encode them in features. If such an assumption holds, meaning that we have model m_1 and model m_2, and if features of m_1 differ from m_2 in such a way to implicitly encode higher order representations or semantically richer information, whether coarse or fine-grained, then it would be useful to concatenate outputs from two models to get the final f_{concat} vector which we think could be a better representation of images in the dataset. Now, for such matter we decided to use two of the best performing backbones from two different model families. We also choose best pooling for individual backbone and normalize features before concatenation (as normalization after concatenation could hinder performance as features from one model could be of lower dimensionality than of the other model). We then perform visual only (Section 3.2) and GPS aided (Section 3.3) approach using concatenated feature representation.

4 Results

Based on official [1] work, we record following metrics: recall@1,recall@5,recall@10 and recall@20, as well as MAP@1,MAP@5,MAP@10 and MAP@20. Our primary metric of comparison is Recall@1.

BoW approach Results for BoW approach using ORB, SIFT and SURF are summarized in Table 4

SIFT BoW approach ablation. We performed hyperparameter tuning and ablation studies on SIFT features, varying the number of features (n), cluster size (k), assignment strategy, distance metrics, and TF-IDF weighting, with results shown in Table 5.

Table 4: Bag-of-Words Approach Results

Method	R@1	R@5	R@10	R@20
ORB	1.63	3.71	5.35	9.21
SURF + TF-IDF	2.90	8.17	11.52	18.39
SIFT + TF-IDF	**8.80**	**17.42**	**22.25**	**28.83**
mAP				
ORB	1.63	0.64	0.51	0.53
SURF + TF-IDF	2.90	1.43	1.23	1.33
SIFT + TF-IDF	**8.80**	**4.62**	**3.98**	**4.05**

Table 5: SIFT Ablation Study Results

Configuration	R@1	R@5	R@10	R@20
Grid Search (n features, k clusters)				
n=50, k=500	4.8	14.2	21.0	33.6
n=50, k=1000	6.6	17.4	25.8	33.6
n=50, k=2000	4.8	18.0	28.2	36.6
n=100, k=500	7.2	19.6	29.2	40.8
n=100, k=1000	10.6	21.4	29.2	39.4
n=100, k=2000	11.2	24.6	32.0	42.0
n=500, k=500	8.6	24.4	31.6	41.8
n=500, k=1000	11.4	27.0	34.4	46.8
n=500, k=2000	12.2	29.0	41.0	48.2
Ablations (baseline: n=500, k=2000, hard, L2, TF-IDF)				
Soft assignment	**13.2**	**31.2**	**40.2**	**49.2**
L1 distance	0.0	2.2	11.0	21.2
Cosine distance	12.2	29.0	41.0	48.2
Chi-square distance	0.2	4.2	12.8	22.4
Without TF-IDF	12.0	30.4	38.0	47.0
mAP				
n=50, k=500	4.8	2.4	2.3	2.8
n=50, k=1000	6.6	3.1	3.0	3.2
n=50, k=2000	4.8	2.8	2.6	3.0
n=100, k=500	7.2	4.5	4.0	4.3
n=100, k=1000	10.6	5.5	5.1	5.2
n=100, k=2000	11.2	5.8	5.2	5.6
n=500, k=500	8.6	4.7	4.7	5.2
n=500, k=1000	11.4	5.9	5.9	6.4
n=500, k=2000	12.2	6.7	6.6	7.2
Ablations (baseline: n=500, k=2000, hard, L2, TF-IDF)				
Soft assignment	**13.2**	**7.3**	**7.3**	**8.0**
L1 distance	0.0	0.2	0.5	0.8
Cosine distance	12.2	6.7	6.6	7.2
Chi-square distance	0.2	0.4	0.8	1.1
Without TF-IDF	12.0	6.5	6.4	6.9

Linear probing: pooling ablation Baseline approach here is using the CLS token. For Perception Encoder network we simply use that one, as the vision encoder outputs it. For DINOv3 and CLIP models, we have a $[1, n, d]$ output where n is the number of tokens and d is the feature dimension, and we ablate pooling methods regarding image-to-image retrieval performance. In Section 6 we provide ablations for all our models; here we give table overviews of gains from using certain pooling methods across DINOv3 and CLIP models.

For per-model tables please refer to Section 6. Here we present Table 6 for performance gains across both DINOv3 and CLIP models:

We observe GeM pooling performs best across models, with gain of 5.13% in Recall@1.

Table 6: Average Percentage Point Gains over CLS Token

Pooling	R@1	R@5	R@10	R@20
CLS	0.00	0.00	0.00	0.00
Mean	+3.13	+5.31	+5.50	+5.64
Mean-noCLS	+3.10	+5.19	+5.51	+5.51
Max	+4.20	+6.64	+6.64	+6.24
GeM	**+5.13**	**+7.51**	**+7.93**	**+7.47**
mAP				
CLS	0.00	0.00	0.00	0.00
Mean	+3.13	**+2.30**	**+2.16**	**+2.35**
Mean-noCLS	+3.10	+2.26	+2.14	+2.31
Max	+4.20	+2.00	+1.69	+1.56
GeM	**+5.13**	+3.50	+3.31	+3.36

Linear probing: overall performance Here we are comparing all or DINOv3, CLIP and Perception Encoder backbones. Results are presented in Table 7. In 6 we provide evaluation of models for mAP metrics.

Table 7: Linear Probing Results - Recall (Best Pooling per Model)

Model	R@1	R@5	R@10	R@20
Perception Encoder				
PE-T16-384	24.4	45.4	58.6	71.0
PE-S16-384	24.8	43.6	57.2	70.0
PE-B16-224	28.8	57.8	69.2	79.6
PE-L14-336	31.0	57.2	65.6	76.2
PE-G14-448	26.2	50.2	61.4	72.2
CLIP				
CLIP-B/32 (GeM)	38.8	61.4	72.8	82.2
CLIP-B/16 (Mean)	31.0	51.8	64.0	76.8
CLIP-L/14 (Mean-noCLS)	32.6	55.2	67.0	80.4
StreetCLIP (GeM)	32.6	55.8	66.0	74.6
DINOv3 ViT				
ViT-S/16 (CLS)	38.0	57.0	65.6	77.6
ViT-S/16+ (CLS)	33.6	52.8	64.4	73.6
ViT-B/16 (GeM)	41.4	63.6	73.8	**85.4**
ViT-L/16 (Max)	40.4	59.0	69.8	80.2
ViT-H/16+ (GeM)	44.0	64.8	73.4	82.4
DINOv3 ConvNeXt				
CNX-T (Max)	26.6	45.6	55.0	67.2
CNX-S (GeM)	20.6	41.0	52.8	60.8
CNX-B (GeM)	24.6	37.2	45.4	55.6
CNX-L (Max)	24.2	47.4	55.4	64.2
Multi-Backbone				
DINOv3-H + CLIP-B/32	**44.8**	**66.6**	**76.0**	85.0

We can observe that individually, the best performing model is DINOv3-H/16+, with performance of 44.0 Recall@1. We used that model and the best performing model from CLIP and PE models, which was CLIP-B/32 with 38.8 Recall@1, and used it in a multi-backbone setup. Such setup performed best across all but one metric (Recall@20), but the gains were not significant (only 0.8 percentage points better recall@1 performance than DINOv3-H/16+ alone), suggesting semantic similarity in features extracted from both backbones. It is possible that both are too coarse-grained to capture specifics of some locations needed in these image-to-image retrieval tasks.

Centroid based retrieval We are using our centroid-based approach using only DINOv3-H/16+ (Table Table 8) and both DINOv3-H/16+ and CLIP-B/32 features (Table Table 9). We motivate our approach by comparing our results to results obtained from a simple baseline: assuming our query image is GPS annotated and we perform a within 100m radius search for relevant images. That is not an approach to take, it just serves as a an observation we had, since London dataset is dense, and there is number of location groups, so such filtering does make sense.

Table 8: GPS-Aided Retrieval - DINOv3-H/16+ (GeM)

Approach	**R@1**	**R@5**	**R@10**	**R@20**
Baseline (No GPS)	44.0	64.8	73.4	**82.4**
Approach A (Two-Stage)	43.2	61.6	70.0	77.2
Approach B (Weighted, α=0.5)	**44.4**	**65.6**	**75.6**	81.8
Goal (GPS-Filtered, r=100)	63.2	87.8	97.2	100.0
mAP				
Baseline (No GPS)	44.0	31.7	31.2	33.1
Approach A (Two-Stage)	43.2	32.1	32.8	**35.8**
Approach B (Weighted, α=0.5)	**44.4**	**32.6**	**33.0**	35.5
Goal (GPS-Filtered, r=100)	63.2	49.1	51.4	57.3

Table 9: GPS-Aided Retrieval - DINOv3-H/16+ + CLIP-B/32

Approach	**R@1**	**R@5**	**R@10**	**R@20**
Baseline (No GPS)	44.8	66.6	76.0	**85.0**
Approach A (Two-Stage)	44.4	62.0	70.4	77.0
Approach B (Weighted, α=0.6)	**46.4**	**66.6**	**76.2**	81.8
Goal (GPS-Filtered, r=100)	63.2	89.8	97.2	100.0
mAP				
Baseline (No GPS)	44.8	32.7	32.2	34.0
Approach A (Two-Stage)	44.4	33.5	33.8	36.7
Approach B (Weighted, α=0.6)	**46.4**	**34.4**	**34.6**	**37.0**
Goal (GPS-Filtered, r=100)	63.2	50.3	52.0	58.1

We observe marginal gains for the single backbone approach (+0.4 percentage points for Recall@1) and somewhat larger gains for the combined backbone approach (+1.6 percentage points for Recall@1). We can see that the GPS-filtered baseline performs significantly better, with a large gap, which we take as motivation for our GPS-aided approach. Furthermore, linearly weighting similarity to both centroid and image (Approach B) proved superior to using only centroids (Approach A), due to fine-grained details lost when averaging features within a centroid that query images are compared against.

5 Conclusion

We performed image retrieval on the London subset of the Mapillary Street Level Sequences dataset. For our baseline approach, we used BoW with ORB features (1.63% Recall@1), extended by SURF (2.90% Recall@1) and SIFT (8.80% Recall@1). Ablation on the SIFT BoW approach proved that using soft assignment, L2 distance, and TF-IDF weighting gave the best performance (13.2% Recall@1).

Comparing visual-features-only approaches with DINOv3, CLIP, StreetCLIP, and Perception Encoder, the best performance was 44.0% Recall@1 when using DINOv3-H/16+. Using both DINOv3-H/16+ and CLIP-B/32 scored 44.8%. Our conclusion is that both CLIP and DINOv3 provide semantically rich but coarse features, possibly encoded in similar visual cues, and using both does not significantly improve performance. Further study could investigate using PCA on concatenated features.

Our ablation study on pooling strategies proved that best performance is gained when using GeM pooling, with an average 5.13 percentage point Recall@1 gain over baseline performance of just using the CLS token.

In the second part of our work, our GPS-aided approach gave 0.4 percentage point gain in Recall@1 for the DINOv3-H/16+ approach and 1.6 percentage points for the DINOv3-H/16+ and CLIP-B/32 approach, with best Recall@1 of 46.4%. A limitation in this approach is the same as before: the centroid that we extract is a coarse-grained, general visual encoding of a location. The idea is to use that information to navigate decisions in GPS space for actual nearest neighbors of query images. The improvement of 1.6 percentage points proved the value in this approach, and comparison to real GPS-based filtering when GPS data of query images is known validated the idea.

To conclude, linear evaluation of pretrained vision backbones serves as a strong method for non-gradient-based approaches, which does not require further fine-tuning of such models. Combinations of two models give marginal improvement due to the coarse granularity of encoded features, while best performance is achieved using the centroid-based approach with two strong backbones.

References

[1] F. Warburg, M. N. Jensen, *et al.* Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] D. Bolya, P.-Y. Huang, P. Sun, J. H. Cho, A. Madotto, C. Wei, T. Ma, J. Zhi, J. Rajasegaran, H. Rasheed, J. Wang, M. Monteiro, H. Xu, S. Dong, N. Ravi, D. Li, P. Dollar, and C. Feichtenhofer. Perception Encoder: The Best Visual Embeddings Are Not at the Output of the Network. *arXiv preprint arXiv:2504.13181*, 2025.

[3] O. Simeoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, and L. Wehrstedt. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[4] L. Haas, S. Alberti, and M. Skreta. Learning Generalized Zero-Shot Learners for Open-Domain Image Geolocalization. *arXiv preprint arXiv:2302.00275*, 2023.

[5] E. Karami, S. Prasad, and M. Shehata. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *arXiv preprint arXiv:1710.02726*, 2017.

[6] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[8] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[9] R. Mur-Artal and J. D. Tardos. Fast Relocalisation and Loop Closing in Keyframe-Based SLAM. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 846–853. IEEE, 2014.

[10] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[11] Q. Fang, L. Chen, Z. Wang, H. Wang, X. Zhang, Z. Cao, J. Zhao, J. Liu, and H. Lu. AnyLoc: Towards Universal Visual Place Recognition. *arXiv preprint arXiv:2410.19341*, 2024. Available at https://arxiv.org/abs/2410.19341.

[12] R. Mirjalili, M. Krawez, and W. Burgard. FM-Loc: Using Foundation Models for Improved Vision-Based Localization. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1381–1387. IEEE, 2023.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models from Natural Language Supervision. *arXiv preprint arXiv:2103.00020*, 2021. Available at https://arxiv.org/abs/2103.00020.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*, 2023. Available at https://arxiv.org/abs/2304.07193.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, and P. Dollar. Segment Anything. *arXiv preprint arXiv:2304.02643*, 2023. Available at https://arxiv.org/abs/2304.02643.

[16] S. Izquierdo and J. Civera. Optimal Transport Aggregation for Visual Place Recognition (SALAD). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. Available at https://arxiv.org/abs/2311.15937.

[17] W. Wieczorek, T. Trzcinski, and P. Spurek. On the Unreasonable Effectiveness of Centroids in Image Retrieval. *arXiv preprint arXiv:2104.13643*, 2021.

[18] J. Park and S. Hwang. Efficient Image Retrieval Using Hierarchical K-Means Clustering. *Sensors*, 24(8):2401, 2024.

[19] F. Lu, X. Lan, L. Zhang, and C. Yuan. Towards Seamless Adaptation of Pre-trained Models for Visual Place Recognition (SelaVPR). *arXiv preprint arXiv:2402.14505*, 2024. Available at https://arxiv.org/abs/2402.14505.

[20] F. Lu, X. Lan, L. Zhang, D. Jiang, Y. Wang, and C. Yuan. CricaVPR: Cross-image Correlation-aware Representation Learning for Visual Place Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

6 Appendix

Here we report pooling method comparison across our DinoV3 models. We are looking at gains for all the models on average.

Table 10: DINOv3 Models - Percentage Point Gains over CLS

Pooling	**R@1**	**R@5**	**R@10**	**R@20**
CLS	0.00	0.00	0.00	0.00
Mean	+1.56	+4.50	+6.00	+6.44
Mean-noCLS	+1.52	+4.36	+6.04	+6.28
Max	**+4.82**	**+7.70**	+8.18	+7.24
GeM	+4.26	+7.52	**+8.68**	**+8.16**
		mAP		
CLS	0.00	0.00	0.00	0.00
Mean	+1.56	+1.09	+0.97	+0.97
Mean-noCLS	+1.52	+1.02	+0.94	+0.92
Max	**+4.82**	+2.49	+2.30	+2.37
GeM	+4.26	**+2.69**	**+2.54**	**+2.56**

Here we report pooling method comparison across our CLIP models. We are looking at gains for all the models on average.

Table 11: CLIP Models - Percentage Point Gains over CLS

Pooling	**R@1**	**R@5**	**R@10**	**R@20**
CLS	0.00	0.00	0.00	0.00
Mean	+7.05	+7.35	+4.25	+3.65
Mean-noCLS	+7.05	+7.25	+4.20	+3.60
Max	+2.65	+4.00	+2.80	+3.75
GeM	**+7.30**	**+7.50**	**+6.05**	**+5.75**
		mAP		
CLS	0.00	0.00	0.00	0.00
Mean	+7.05	+5.35	+5.13	**+5.80**
Mean-noCLS	+7.05	+5.36	+5.14	**+5.80**
Max	+2.65	+0.76	+0.18	-0.48
GeM	**+7.30**	**+5.54**	**+5.22**	+5.34

We wanted to explore how our primary metric of Recall@1 changes across models and pooling strategies per model and on average. We find that overall GeM pooling yields best performance results.

Table 12: All Models - Recall@1 by Pooling Strategy

Model	**CLS**	**Mean**	**WoCLS**	**Max**	**GeM**
CLIP					
StreetCLIP	27.0	30.0	30.0	25.2	**32.6**
CLIP-base-patch16	26.6	31.0	31.0	28.0	28.4
CLIP-base-patch32	25.2	35.6	35.4	32.6	**38.8**
CLIP-large-patch14	22.0	32.4	**32.6**	25.6	30.2
DINOv3					
convnext-base	20.2	18.6	18.6	21.8	24.6
convnext-large	17.6	17.8	17.8	24.2	21.4
convnext-small	20.6	17.4	17.2	21.2	20.6
convnext-tiny	19.2	20.2	20.6	26.6	24.2
vitb16	37.2	37.6	37.6	41.2	41.4
vith16plus	27.4	**43.8**	**43.8**	**41.6**	**44.0**
vitl16	13.6	17.2	17.2	16.2	17.4
vits16	38.0	32.4	32.4	38.0	36.0
vits16plus	33.6	34.0	33.8	36.4	33.6
Overall Average	25.7	28.3	28.3	29.1	**30.2**

This is complementary table to Table 7. We present performance of each model using pooling strategy with highest mAP for each. We can observe that mAP is generally the highest when using multi-backbone approach with DINOv3-H/16+ and CLIP-B/32 vision backbones. DINOv3 model family (ViT, non distilled) performs superior to CLIP and Perception Encoder models, with DINOv3-H/16+ being the best performing individual model.

Table 13: Linear Probing Results - mAP (Best Pooling per Model)

Model	**mAP@1**	**mAP@5**	**mAP@10**	**mAP@20**
Perception Encoder				
PE-T16-384	24.4	13.7	13.3	14.3
PE-S16-384	24.8	15.6	14.3	15.1
PE-B16-224	28.8	20.8	20.2	21.4
PE-L14-336	31.0	20.9	19.7	20.7
PE-G14-448	26.2	17.5	16.9	18.3
CLIP				
CLIP-B/32 (GeM)	38.8	24.6	23.3	24.2
CLIP-B/16 (Mean)	31.0	19.7	19.1	20.9
CLIP-L/14 (Mean-noCLS)	32.6	22.5	22.2	24.8
StreetCLIP (GeM)	32.6	21.3	20.8	22.1
DINOv3 ViT				
ViT-S/16 (CLS)	38.0	25.9	25.3	26.9
ViT-S/16+ (CLS)	33.6	24.4	24.4	26.4
ViT-B/16 (GeM)	41.4	28.4	28.4	30.8
ViT-L/16 (Max)	40.4	28.3	28.1	30.3
ViT-H/16+ (GeM)	44.0	31.7	31.2	33.1
DINOv3 ConvNeXt				
CNX-T (Max)	26.6	16.9	16.6	17.8
CNX-S (GeM)	20.6	12.9	13.1	14.5
CNX-B (GeM)	24.6	15.2	15.0	15.8
CNX-L (Max)	24.2	15.0	14.1	15.6
Multi-Backbone				
DINOv3-H + CLIP-B/32	**44.8**	**32.7**	**32.2**	**34.0**

Our motivation for Section 3.3 lies in the fact that the MSLS dataset is dense: for the top 10% most crowded locations, the 20 nearest neighbours are within 3 meters, while for half of the data, the 20 nearest neighbours are within 80 meters. An overview is provided in Table Table 14.

Table 14: Distance measurements of nearest neighbours to our query images (in meters).

Neighbor (k)	**P10**	**P25**	**P50**	**P75**	**P90**
1-NN	1.16	2.56	4.78	7.76	11.68
5-NN	6.30	9.93	16.62	25.64	40.60
8-NN	11.40	15.86	27.77	43.51	63.57
10-NN	14.47	19.52	35.18	53.64	82.86
20-NN	33.84	45.01	79.51	120.76	321.50

To disclose, as requested, AI (ChatGPT and Claude) was only used for grammar check and formatting, as well as latex table generation. All code and article content are the product of authors.