Megaloc: One retrieval to place them all

MegaLoc: One Retrieval to Place Them All

Gabriele Berton
Polytechnic of Turin
bertongabri@gmail.com

Carlo Masone
Polytechnic of Turin, Focoos AI

Abstract

Retrieving images from the same location as a given query is an important component of multiple computer vision tasks, like Visual Place Recognition, Landmark Retrieval, Visual Localization, 3D reconstruction, and SLAM. However, existing solutions are built to specifically work for one of these tasks, and are known to fail when the requirements slightly change or when they meet out-of-distribution data. In this paper we combine a variety of existing methods, training techniques, and datasets to train a retrieval model, called MegaLoc, that is performant on multiple tasks. We find that MegaLoc (1) achieves state of the art on a large number of Visual Place Recognition datasets, (2) impressive results on common Landmark Retrieval datasets, and (3) sets a new state of the art for Visual Localization on the LaMAR datasets, where we only changed the retrieval method to LaMAR's official localization pipeline. The code for MegaLoc is available at https://github.com/gmberton/MegaLoc

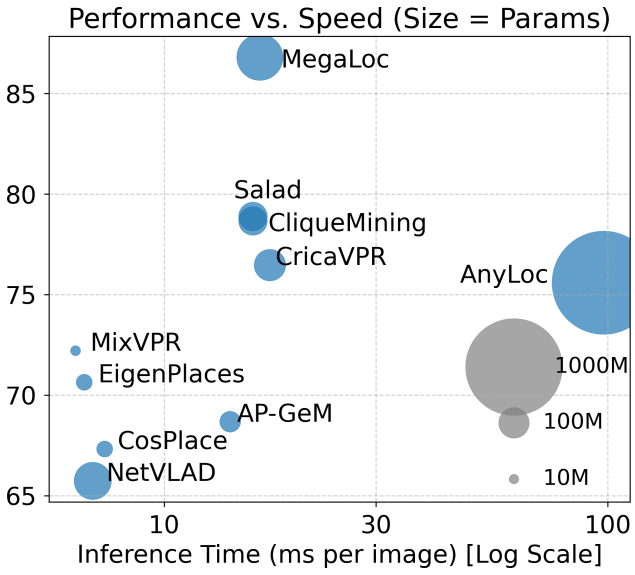

Figure 1. Performance comparison of various Visual Place Recognition models. The y axis shows the results averaged across all evaluation datasets, the x axis shows the inference time, and the size of the circle represents the number of parameters per model.

1. Introduction

This paper tackles the task of retrieving images from a large database that represent the same place as a given query image. But what does it mean for two images to be "from the same place"? Depending on who you ask, you'll get different answers:

1. Landmark Retrieval (**LR**) folks will tell you that two photos are from the same place if they depict the same landmark, regardless of how close to each other the two photos were taken [40];
2. Visual Place Recognition (**VPR**) people set a camera pose distance of 25 meters to define if two images are positives (*i.e.* from the same place) [4];
3. Visual Localization (**VL**) / 3D Vision researchers will tell you that two images need to have their pose as close as possible to be considered the same place.

Even though image retrieval is a core component in all three tasks, their different definitions and requirement has inevitably led to the development of ad-hoc image retrieval solutions for each of them. As these three tasks continued to diverge, over the years papers have avoided showing results of their methods on more than one of these tasks: VPR papers don't show results on LR, and LR papers don't show results on VPR. In the meantime, 3D vision pipelines like COLMAP [30], Hierarchical Localization [28] and GLOMAP [22] keep using outdated retrieval methods, like RootSIFT with bag-of-words [3, 10, 32] and NetVLAD [4]. In this paper we aim to put an end to this, by training a single model that achieves SOTA (or almost) on all of these tasks, showcasing robustness across diverse domains. To train this model we do not propose any "technical novelty", but we use all the lessons learned from all these three task, putting together a combination of good samplers, datasets, and general training techniques.

"*Why does it matter?*", you may ask. Imagine you are doing 3D reconstruction, where image retrieval is a funda-

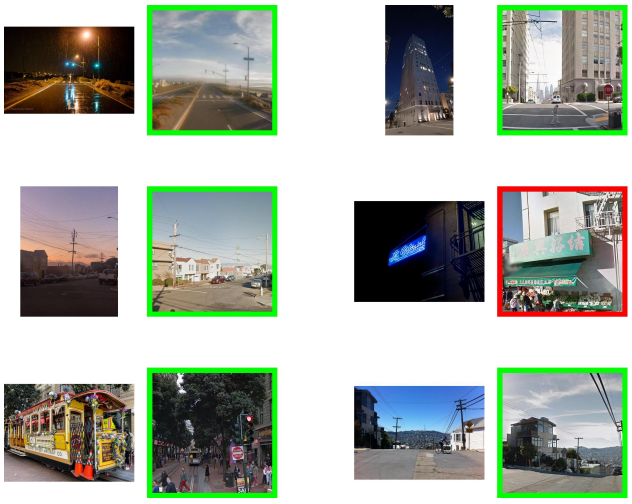

Figure 2. Qualitative examples of predictions by MegaLoc. Each pair of images represents a query and its top-1 prediction from the SF-XL dataset, searched across the 2.8M database spanning 150 km^2 across San Francisco. Predictions in green are correct, red are wrong.

mental component, on a collection of diverse scenes (*e.g.* to create datasets like MegaDepth [18], MegaScenes [37], or for the evergreen Image Matching Challenge [6]). In some cases there would be small scenes (*e.g.* reconstruction of a fountain), requiring a retrieval model that is able to retrieve nearby images (few meters away), which is something VPR models excel at, but LR models underperform (see [8] Tab. 14). In other cases however, the scene might be large (*e.g.* a big landmark like a church), with images hundreds of meters away: while LR models are designed for this, VPR models achieve poor results in this situations (see Sec. 3.2.3). Given these considerations, we note how neither VPR nor LR provide models for the diverse cases of 3D reconstructions, creating a gap in literature that is filled by MegaLoc. As another example where a model like MegaLoc is necessary, one can think of Visual Place Recognition (which is also the first step for Visual Localization), where models are evaluated by using a 25 meters threshold (and queries in popular datasets always have at least one positive within 25 meters). However, in the real world the nearest image to a given query might be 100 meters away, and while ideally we would still want to retrieve it, a VPR model is unlikely to work in such case, as it has been trained to ignore anything further away from the camera.

In this paper we demonstrate that, by leveraging a diverse set of data sources and best practices from LR, VPR and VL, we obtain a single image retrieval model that works well across all these tasks.

2. Method

The core idea of this paper is to fuse data from multiple datasets, and train a single model. We use five datasets containing both outdoor and indoor images (thorough description below) and catering to different image localization tasks: GSV-Cities [1], Mapillary Street-Level Sequences (MSLS) [39], MegaScenes [37], ScanNet [13] and San Francisco eXtra Large (SF-XL) [7]. At each training iteration, we extract six sub-batches of data, one for each dataset (except SF-XL, from which two sub-batches are sampled) and use a multi-similarity loss [38] computed over each sub-batch. Each sub-batch is made of 128 images, containing 4 images (called quadruplets) from 32 different places/classes. Given that these datasets have diverse format, they require different sampling techniques. In the following paragraphs we explain how data is sampled from each dataset.

San Francisco eXtra Large (SF-XL) is a dataset of 41M images with GPS and orientation from 12 different years, densely covering the entire city of San Francisco across time. To select ideal quadruplets for training, we use the sampling technique presented in EigenPlaces [9]. This method assures that each class contains images that represent a given place from diverse perspectives, while ensuring that no visual overlap exists between two different places. EigenPlaces provides two sub-batches, one made of frontal-facing images (*i.e.* with the camera facing straight along the street) and one of lateral-facing images.

Google Street View Cities (GSV-Cities) is a dataset of 530k images split into 62k places/classes from 40 cities, where each class contains at least 4 images with same orientation and is at least 100 meters from any other class. Given that GSV-Cities is already split into non-overlapping classes, it is not strictly necessary to apply a particular sampling technique. We therefore directly feed the GSV-Cities dataset to the multi-similarity loss, as in the original GSV-Cities paper [1].

Mapillary Street-Level Sequences (MSLS) is a dataset of 1.6M images split in contiguous sequences, across 30 different cities over 9 years. To ideally sample data from the MSLS dataset, we use the mining technique described in the CliqueMining paper [33]. This method ensures that the places selected for each batch depict visually similar (but geographically different) places (*i.e.* hard negatives), so that the loss can be as high as possible and effectively teach the model to disambiguate between similar-looking places.

MegaScenes is a collection of 100k 3D structure-from-motion reconstructions, composed of 2M images from

Wikimedia Commons. Simply using each reconstruction as a class, and sampling random images from such class, could lead to images that do not have any visual overlap, *e.g*. two images could show opposites facades of a building, therefore having no visual overlap while belonging to the same 3D reconstruction. Therefore we make sure that when we sample a set of four images from a given reconstruction, each of these four images should have visual overlap with each other (we define visual overlap as having at least 1% of 3D points in common in the 3D reconstruction).

ScanNet is a dataset of 2.5M views from 1500 scans from 707 indoor places. To train on ScanNet we use each scene as a class, and select quadruplets so that each pair of images within a quadruplet has visual overlap (*i.e*. less than 10 meters and 30° apart); simultaneously we ensure that no two images from different quadruplets has visual overlap.

3. Experiments

3.1. Implementation details

During training, images are resized to 224×224, while for inference we resize them to 322×322, following [16]. We use RandAugment [11] for data augmentation, as in [1], and AdamW [19] as optimizer. Training is performed for 40k iterations. The loss is simply computed as $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4 + \mathcal{L}_5 + \mathcal{L}_6$, where each $\mathcal{L}_n$ is the multi-similarity loss computed on one of the sub-batches.

The architecture consists of a DINO-v2-base backbone [21] followed by a SALAD [16] aggregation layer, which has shown state-of-the-art performances over multiple VPR datasets [16, 33]. The SALAD layer is computed with 64 clusters, 256 channels per cluster, a global token of 256 and an MLP dimension of 512. The SALAD layer is followed by a linear projection (from a dimension of 16640 to 8448) and an L2 normalization. During training, most of the model is frozen, except for the 4 last transformer layers, the SALAD layer and the final linear layer.

Memory-efficient GPU training is achieved using PyTorch [23], where instead of summing all the losses and computing a single backward pass, we compute backward passes independently on each dataset: the backward pass computes and accumulates the gradient while freeing memory (see Algorithm 1). This simple technique reduces the VRAM for training MegaLoc from 360GB to 60GB (a 6x reduction because there are 6 losses) with **no increase in latency**, unlike in standard gradient accumulation where a batch is split into micro batches: this allows MegaLoc to be trained on a single A100 GPU.

Algorithm 1 Memory-Efficient GPU Training

Require: Model $Model$, Optim. Opt, Datasets $\{D_i\}_{i=1}^{N}$

Initialize $Model$, Opt
for each training iteration **do**
  for each dataset D_i in $\{D_1, ..., D_N\}$ **do**
    $B_i \leftarrow \text{LoadBatch}(D_i)$
    $Y_i \leftarrow Model(B_i)$
    $L_i \leftarrow \text{Loss}(Y_i, \text{target}_i)$
    $L_i.backward()$
  end for
  $Opt.step()$
  $Opt.zero\_grad()$
end for

3.2. Results

We perform experiments on three different types of tasks:

- Visual Place Recognition, where the task is to retrieve images that are within 25 meters from the query (Sec. 3.2.1);
- Visual Localization, where retrieval is part of a bigger pipeline that aims at finding the precise pose of the query given a set of posed images (Sec. 3.2.2);
- Landmark Retrieval, *i.e*. retrieving images that depict the same landmark as the query (Sec. 3.2.3).

3.2.1. Visual Place Recognition

We run experiments on a comprehensive set of Visual Place Recognition datasets. These datasets contain a large variety of domains, including: outdoor, indoor, street-view, hand-held camera, car-mounted camera, night, occlusions, long-term changes, grayscale. Results are shown in Tab. 1. While other high-performing VPR models (like SALAD and CliqueMining) achieve very good results (*i.e*. comparable to MegaLoc) on most datasets, MegaLoc vastly outperforms every other model on Baidu, which is an indoor-only dataset.

3.2.2. Visual Localization

Image retrieval is a core tool to solve 3D vision tasks, in pipelines like visual localization (*e.g*. Hierarchical Localization [28] and InLoc [35]) and 3D reconstructions (*e.g*. COLMAP [30, 31] and GLOMAP [22]). To understand if our method can help this use case, we compute results on both query sets (phone and HoloLens) of the three datasets of LaMAR [29], which comprise various challenges, including plenty of visual aliasing from both indoor and outdoor imagery. To do this, we relied on the official LaMAR codebase[1] by simply replacing the retrieval method. We computed these experiments on the validation set, as the labels for the test set are not publicly available. Note that we only use LaMAR (or the landmark retrieval datasets) as a test set, and using it as validation would take unfeasibly

[1] https://github.com/microsoft/lamar-benchmark

Method	Desc. Dim.	Baidu [34]		Eynsham [8, 12]		MSLS val [39]		Pitts250k [4, 14]		Pitts30k [4, 14]		SF-XL v1 [7]		SF-XL v2 [7]		SF-XL night [5]		SF-XL occlusion [5]		Tokyo 24/7 [36]	
		R1	R10	R1	R10	R1	R10	R1	R10	R1	R10	R1	R10	R1	R10	R1	R10	R1	R10	R1	R10
NetVLAD [4]	4096	69.0	95.0	77.7	90.5	54.5	70.4	85.9	95.0	85.0	94.4	40.1	57.7	76.9	91.1	6.7	14.2	9.2	22.4	69.8	82.9
AP-GeM [27]	2048	59.8	90.8	68.3	84.0	56.0	72.9	80.0	93.5	80.7	94.1	37.9	54.1	66.4	84.6	7.5	16.7	5.3	14.5	57.5	77.5
CosPlace [7]	2048	52.0	80.4	90.0	94.9	85.0	92.6	92.3	98.4	90.9	96.7	76.6	85.5	88.8	96.8	23.6	32.8	30.3	44.7	87.3	95.6
MixVPR [2]	4096	71.9	94.7	89.6	94.4	83.2	91.9	94.3	98.9	91.6	96.4	72.5	80.9	88.6	95.0	19.5	30.5	30.3	38.2	87.0	94.0
EigenPlaces [9]	2048	69.1	91.9	90.7	95.4	85.9	93.1	94.1	98.7	92.5	97.6	84.0	90.7	90.8	96.7	23.6	34.5	32.9	52.6	93.0	97.5
AnyLoc [17]	49152	75.6	95.2	85.0	94.1	58.7	74.5	89.4	98.0	86.3	96.7	-	-	-	-	-	-	-	-	87.6	97.5
Salad [16]	8448	72.7	93.6	91.6	95.9	88.2	95.0	95.0	99.2	92.3	97.4	88.7	94.4	94.6	98.2	46.1	62.4	50.0	68.4	94.6	98.1
CricaVPR [20]	10752	65.6	93.2	88.0	94.3	76.7	87.2	92.6	98.3	90.0	96.7	62.6	78.9	86.3	96.0	25.8	40.6	27.6	47.4	82.9	93.7
CliqueMining [33]	8448	72.9	92.7	91.9	96.2	**91.6**	**95.9**	95.3	99.2	92.6	97.8	85.5	92.6	94.5	98.3	46.1	60.9	44.7	64.5	**96.8**	97.8
MegaLoc (Ours)	8448	**87.7**	**98.0**	**92.6**	**96.8**	91.0	95.8	**96.4**	**99.3**	**94.1**	**98.2**	**95.3**	**98.0**	**94.8**	**98.5**	**52.8**	**73.8**	**51.3**	**75.0**	96.5	**99.4**

Table 1. **Recall@1 and Recall@10 on multiple VPR datasets.** Best overall results on each dataset are in **bold**, second best results underlined. Results marked with a "-" did not fit in 480GB of RAM (2.8M features of 49k dimensions require 560GB for a float32-based kNN).

Method	CAB (Phone)		HGE (Phone)		LIN (Phone)		CAB (HoloLens)		HGE (HoloLens)		LIN (HoloLens)	
	(1, 0.1)	(5, 1.0)	(1, 0.1)	(5, 1.0)	(1, 0.1)	(5, 1.0)	(1, 0.1)	(5, 1.0)	(1, 0.1)	(5, 1.0)	(1, 0.1)	(5, 1.0)
NetVLAD	43.4	54.0	54.8	80.0	74.4	87.8	63.1	81.4	57.9	71.6	76.1	83.0
AP-GeM	39.4	52.0	58.0	81.3	69.1	82.0	62.9	82.5	65.6	76.6	80.7	91.1
Fusion (NetVLAD+AP-GeM)	41.4	53.8	56.3	82.4	76.0	89.4	63.2	83.1	63.1	75.1	78.5	87.0
CosPlace	29.0	37.4	54.4	81.3	63.3	75.7	56.4	77.8	55.6	69.8	80.6	91.4
MixVPR	40.9	50.8	59.2	83.8	77.5	89.8	65.2	84.7	63.3	74.7	83.6	92.2
EigenPlaces	32.3	44.7	56.3	81.3	70.2	82.6	63.9	81.8	60.2	72.5	84.8	93.1
AnyLoc	**48.0**	59.8	58.8	83.0	77.2	92.4	69.7	88.5	70.1	81.0	81.4	90.4
Salad	44.2	55.6	65.3	92.2	81.7	94.0	71.5	90.7	75.3	85.2	91.3	**99.4**
CricaVPR	40.4	52.0	63.7	89.3	80.7	93.1	73.9	90.7	72.5	81.6	89.1	98.4
CliqueMining	44.2	55.6	66.0	91.4	80.5	93.1	74.2	90.9	**77.3**	**86.3**	92.0	98.8
MegaLoc (Ours)	47.0	**60.4**	**67.2**	**92.9**	**83.3**	**94.9**	**77.4**	**93.4**	72.9	83.5	**92.2**	99.0

Table 2. **Results on LaMAR's datasets**, computed on each of the three locations, for both types of queries (HoloLens and Phone), which include both indoor and outdoor. For each location we report the recall at (1°, 10cm) and (5°, 1m), following the LaMAR paper [29].

Method	R-Oxford			R-Paris		
	E	M	H	E	M	H
NetVLAD	24.1	16.1	4.7	61.2	46.3	22.0
AP-GeM	49.6	37.6	19.3	82.5	69.5	45.5
CosPlace	32.1	23.4	10.3	57.6	45.0	22.3
MixVPR	38.2	28.4	10.8	61.9	48.3	25.0
EigenPlaces	29.4	22.9	11.8	60.9	47.3	23.6
AnyLoc	64.2	45.5	18.9	82.8	68.5	48.8
Salad	55.2	42.3	21.4	76.6	66.2	44.8
CricaVPR	57.0	39.2	15.3	80.0	68.9	48.9
CliqueMining	52.2	41.0	22.1	71.8	60.5	41.2
MegaLoc (Ours)	**91.0**	**79.0**	**62.1**	**95.3**	**89.6**	**77.1**

Table 3. **Results on Landmark Retrieval datasets**, respectively Revisited Paris 6k [25, 26] and Revisited Oxford 5k [24, 26].

long (multiple days). Results are reported in Tab. 2, and show that while some other models achieve good results on some datasets, MegaLoc is the only one that consistently achieves high results.

3.2.3. Landmark Retrieval

For the task of Landmark Retrieval we compute results on the most used datasets in literature, namely (the revisited versions of [26]) Oxford5k [24] and Paris6k [25]. To do this we relied on the official codebase for the datasets[2], by simply swapping the retrieval method. Results, reported in Tab. 3, show a large gap between MegaLoc and previous VPR models on this task, which can be simply explained by the fact that previous models were only optimized for the standard VPR metric of retrieving images within 25 meters from the query.

3.2.4. Failure Cases

We identified a series of 4 main categories of "failure cases" that prevent the results from reaching 100% recalls, and we present them in Fig. 3. We note however that, from a practical perspective, the only real failure cases are depicted in the second category/column of Fig. 3: furthermore, in most similar cases SOTA models (*i.e.* not only MegaLoc, but also other recent ones) can actually retrieve precise predictions, meaning that these failure cases can be likely solvable by some simple post-processing techniques (*e.g.* re-ranking with image matchers, or majority voting). Finally, another failure case that we noted, is when database images do not cover properly the search area: this is very common in the Mapillary (MSLS) dataset, where database images only show one direction (*e.g.* photos along a road taken from north to south), while the queries are photos facing the other direction. We note however, that in the real world this can be easily solved by collecting database images in multiple directions, which is also common in most test datasets,

[2] https://github.com/filipradenovic/revisitop

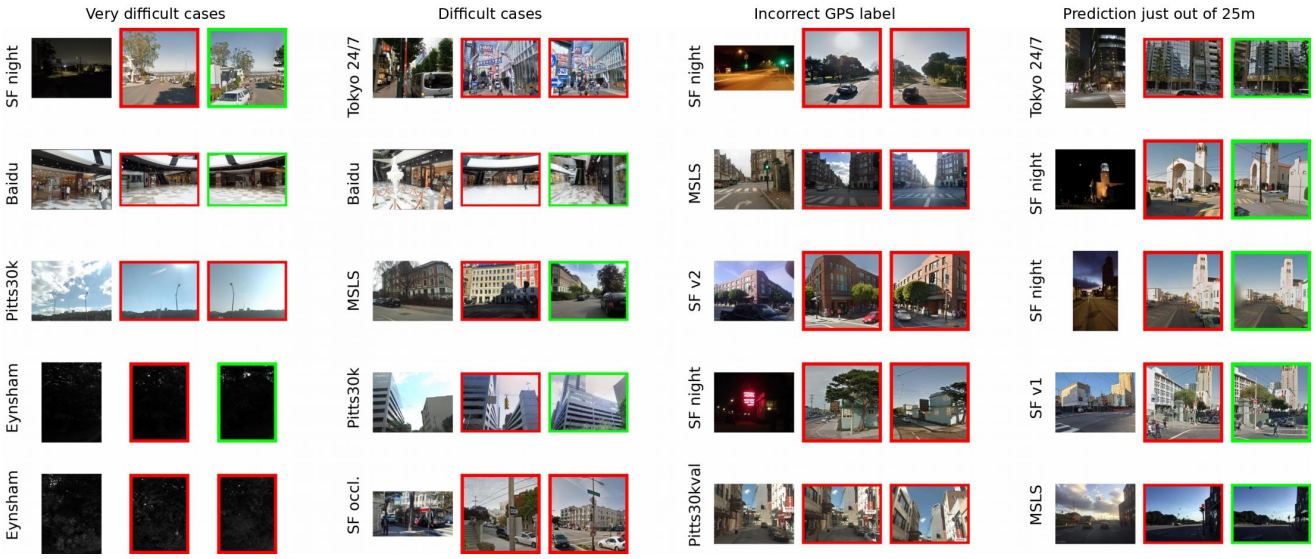

Figure 3. **Failure cases, grouped in 4 categories.** Each one of the 4 column represent a category of failure cases: for each category we show 5 examples, made of 3 images, namely the query and its top-2 predictions with MegaLoc, which can be in red or green depending if the prediction is correct (*i.e.* within 25 meters). The 4 categories that we identified are (1) *very difficult cases*, which are unlikely to be solved any time soon; (2) *difficult cases*, which can probably be solved by slightly better models than the current ones or simple post-processing; (3) *incorrect GPS labels*, which, surprisingly, exist also in Mapillary and Google StreetView data; (4) *predictions just out of the 25m threshold*, which despite being considered negatives in VPR, are actually useful predictions for real-world applications.

like Eynsham, Pitts30k, Tokyo 24/7 and SF-XL.

4. Conclusion and limitations

So, is image retrieval for localization solved? Well, almost. While some datasets still show some room for improvement, we note that this is often due to either arguably unsolvable failure cases, wrong labels, and very few cases that can be solved by better models. We emphasize however that this has been the case for some time, as previous DINO-v2-based models, like SALAD and CliqueMining, show very high results on classic VPR datasets. What is still missing from literature is models like MegaLoc that achieve good results in a variety of diverse tasks and domains.

Should you always use MegaLoc? Well, almost, except for at least 3 use-cases. MegaLoc has shown great results on a variety of related tasks, and, unlike other VPR models, achieves good results on landmark retrieval, which make it a great option also for retrieval for 3D reconstruction tasks, besides standard VPR and visual localization tasks. However, experiments show that MegaLoc is outperformed by CliqueMining in MSLS, which is a dataset made of (almost entirely) forward facing images (*i.e.* photos where the camera is facing the same direction of the street, instead of facing sideways towards the side of the street). Another use case where MegaLoc is likely to be suboptimal is in very unusual natural environments, like forests or caves, where instead AnyLoc has been shown to work well [17]. A third and final use case where other models might be preferred to MegaLoc is for embedded systems, where one might opt for more lightweight models, like the ResNet-18 [15] versions of CosPlace [7], which has 11M parameters instead of MegaLoc's 228M.

Acknowledgements Carlo Masone was supported by FAIR - Future Artificial Intelligence Research which received funding from the European Union Next-GenerationEU (Piano nazionale di ripresa e resilienza (PNRR) – missione 4 componente 2, investimento 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.
We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources.

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 2, 3

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 4

[3] Relja Arandjelovic. Three things everyone should know to improve object retrieval. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2911–2918, USA, 2012. IEEE Computer Society. 1

[4] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018. 1, 4

[5] Giovanni Barbarani, Mohamad Mostafa, Hajali Bayramov, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Are local features all you need for cross-domain visual place recognition? In *CVPRW*, pages 6155–6165, 2023. 4

[6] Fabio Bellavia, Jiri Matas, Dmytro Mishkin, Luca Morelli, Fabio Remondino, Weiwei Sun, Amy Tabb, Eduard Trulls, Kwang Moo Yi, Sohier Dane, and Ashley Chow. Image matching challenge 2024 - hexathlon. https://kaggle.com/competitions/image-matching-challenge-2024, 2024. Kaggle. 2

[7] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4868–4878, 2022. 2, 4, 5

[8] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark, 2023. 2, 4

[9] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11080–11090, 2023. 2, 4

[10] Gabriela Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision*, 2004. 1

[11] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, pages 18613–18624. Curran Associates, Inc., 2020. 3

[12] M. Cummins and P. Newman. Highly scalable appearance-only slam - FAB-MAP 2.0. In *Robotics: Science and Systems*, 2009. 4

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[14] Petr Gronát, Guillaume Obozinski, Josef Sivic, and Tomá Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 907–914, 2013. 4

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[16] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 3, 4

[17] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *arXiv*, 2023. 4, 5

[18] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 2

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 3

[20] Feng Lu, Xiangyuan Lan, Lijun Zhang, Dongmei Jiang, Yaowei Wang, and Chun 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)*, 2024. 4

[21] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 3

[22] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 3

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3

[24] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2007. 4

[25] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 4

[26] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 4

[27] Jérôme Revaud, Jon Almazán, R. S. Rezende, and César Roberto de Souza. Learning with average precision:

Training image retrieval with a listwise loss. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5106–5115, 2019. 4

[28] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 3

[29] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*, 2022. 3, 4

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3

[31] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 3

[32] Johannes L. Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Computer Vision – ACCV 2016*, pages 321–337, Cham, 2017. Springer International Publishing. 1

[33] Javier Civera Sergio Izquierdo. Close, but not there: Boosting geographic distance sensitivity in visual place recognition. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 4

[34] Xun Sun, Yuanfan Xie, Peiwen Luo, and Liang Wang. A dataset for benchmarking image-based localization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5641–5649, 2017. 4

[35] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[36] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2): 257–271, 2018. 4

[37] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *ECCV*, 2024. 2

[38] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 2

[39] Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2623–2632, 2020. 2, 4

[40] Tobias Weyand, A. Araújo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2572–2581, 2020. 1