Divide&Classify: Fine-Grained Classification for City-Wide Visual Place Recognition (Supplementary Material)

Gabriele Trivigno*[1] Gabriele Berton*[1] Juan Aragon[1] Barbara Caputo[1] Carlo Masone[1]
[1] Politecnico di Torino
{gabriele.trivigno, gabriele.berton, barbara.caputo, carlo.masone}@polito.it

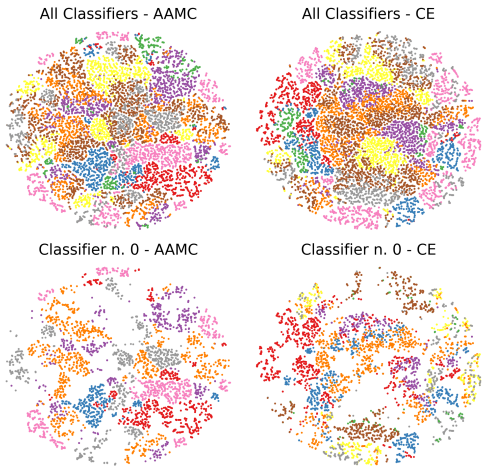

Figure 1: **t-SNE analysis** of embeddings in a 100m x 100m square. Each color codifies a different 20m cell.

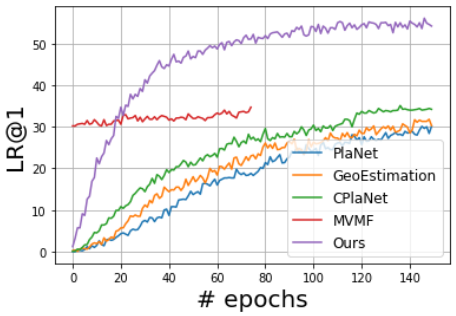

Figure 2: **Behaviour of LR@1 during training** for each of the methods. Note that MVMF [7] starts with a high LR because it uses the weights of a trained PlaNet model.

1. Experiments

In this Supplementary Material we report details that could not fit in the main paper. In Sec. 1.1 we provide further ablations to better understand how our proposed method functions.

In Sec. 2 we provide a thorough discussion into how we adapted the partitioning scheme of previous works, that originally targeted planet-scale localization, for the proposed task of city-wide localization.

1.1. Further Ablations

Embedding learnt with our AMCC vs Cross Entropy. The first row in Fig. 1 reports the t-SNE of all embeddings in a 100m square, with a model trained either with AAMC or a fully connected layer with cross-entropy loss; each color codifies a different 20m cell. Even though some structures are visible, there is an amount of overlap which is understandable given that adjacent cells at such fine resolution can present high appearance similarities. The second row shows why in D&C each classifier is able to learn a meaningful distribution: inside each group, thanks to the non-adjacency of cells, classes are well defined. In particular, the two plots show how the AAMC yields a better-structured embedding space thanks to the concept of large margin.

Behaviour of LR during training with different methods. In Fig. 2 we analyze how the LR@1 changes after each epoch for different methods (given the huge size of the dataset we define an epoch as 2k iterations). We find that most previous works, namely PlaNet [13], CPlaNet [12] and Hierarchical Geolocation Estimation [10] present very mild improvements on LR within the first few epochs w.r.t. our D&C, which on the other hand grows very steeply right from the beginning. MvMF initializes its mixture assignment weights from a pretrained PlaNet model, and it terminates the training after less than 100 epochs.

Behaviour of classification accuracy during training using different N. To better understand how the value of N affects training stability, we built a plot using $N = 2$ and $N = 3$ and showing the accuracy on the train set at the end of each epoch. The plot (Fig. 3) shows that in the first epochs of training the accuracy forms waves with a period length of size $|G| = N \times N$, where $|G|$ represents the number of groups and the number of classifiers. This is due to

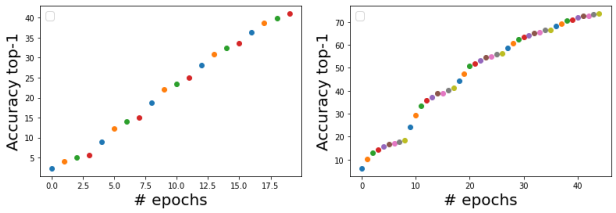

Figure 3: **Evolution of classification accuracy during training with different values of N.** We can see that in the first epochs of training, the accuracy on the train set presents waves with period length of size $|G|$. Each color represents a different classifier being trained at the given epoch for a total of $|G|$ colors.

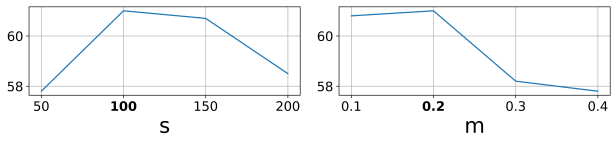

Figure 4: **Ablation on *s* (left) and *m* (right).** Best values are highlighted in bold.

the fact that each classifier is trained once every $|G|$ epochs, meaning that at the $|G|th$ epoch the model will for the first time reuse a classifier that has been previously trained, resulting in a steep increase in accuracy every $|G|$ epochs.

Ablation on AAMC hyperparameters.

Our AAMC classifier shares the 2 hyperparameter of the ArcFace formulation. Namely, s determines the radius of the hypersphere onto which prototypes are projected, and m is the enforced margin (in cosine space) between different prototypes.

Qualitative results. In Figs. 6 and 7 we show some qualitative results of challenging queries and the retrieved Top-3 candidates by some retrieval-only methods (namely CosPlace and NetVLAD) and by some classification-retrieval pipelines (using respectively our D&C and CPlanet as classification modules).

Approximate Nearest Neighbor Search. In Fig. 5 we report the results with the best combinations of methods / hyperparameters for our experiments with Approximate Nearest Neighbor search algorithms. The plot shows only the best performing configurations. Among other ANNs that we tried are standard Product Quantization [5], Inverted File Indexes (these two methods can be combined in the IVFPQ), and Inverted File MultiIndex [2]. We didn't report these results as they performed poorly w.r.t. their counterparts in the plot.
For Table 1 of the main paper we chose two configuration from this pareto-optimal curve, one being optimized for performances and one for speed. For performances, we picked the configuration that grants at least 10x speed, with the maximum performances, and this turned out to be IVFPQ(128,50). For speed, we selected the methods that provided a speedup of at least 100x. This resulted in choosing IVFPQ(128,2) and HNSW(512).

Method	C-Pitts30k (30k images)		C-Tokyo 24/7 (76k images)	
	LR@1	Inf. time	LR@1	Inf. time
Classification				
PlaNet [12]	31.5	12 ms	19.5	12 ms
HGE [10]	33.6	15 ms	22.0	15 ms
CPlaNet [12]	33.0	17 ms	21.5	17 ms
MvMF [4]	31.5	12 ms	19.9	12 ms
D&C (ours)	40.5	12 ms	33.7	12 ms
Retrieval		(kNN time)		(kNN time)
NetVLAD [1]	86.1	58 ms	62.2	130 ms
CRN [6]	86.3	58 ms	62.8	130 ms
SARE [8]	87.2	58 ms	74.8	130 ms
SFRS [3]	88.7	58 ms	78.5	130 ms
GeM [11]	77.9	16 ms	46.4	25 ms
CosPlace	88.5	16 ms	82.8	25 ms
Mixed pipeline				
D&C(ours) + CosPlace	81.9	1 ms	74.9	3.5 ms

Table 1: **Comparison of LR@1** of different methods for *Pitts30k* and *Tokyo24/7* using *EfficientNet-B0* as backbone

1.2. Experiments on small datasets

In the main paper we discussed how classification methods are outperformed by retrieval approaches for small datasets due to the lack of enough positives during training. On the other hand, the inference time gap between both procedures loses relevance when dealing with smaller datasets. In Tab. 1 it is presented a quantitative analysis on how the proposed methods behave on datasets that are 1000x smaller than SF-XL, covering geographical areas less than $3km^2$ and having half the density of SF-XL.

2. Baselines Implementation Details

Although previous works use different partitioning methods of the dataset in classes, we carefully tuned the partitioning hyperparameters to ensure fair comparisons among different methods. While some methods split the geographical area according to the density of the training points [13, 10] others fix the dimension of the cells into a predefined value and merge them until the number of geographical regions satisfies the desired condition [12].

The optimal number of classes generated with each method is shown in Tab. 2, and in the next paragraphs we detail how we empirically found such values for each partitioning method.

Partitions of HGE, PlaNet and MVMF. The three methods of PlaNet [13], Hierarchical Geolocation Estimation (HGE) [10] and MVMF [4] all use the same partitions, with

Partition method	SF-XL	C-Pitts30k	C-Tokyo 24/7
PlaNet / MVMF	65k	486	1840
HGE	19k / 35k / 65k	158 / 272 / 486	508 / 961 / 1840
CPlaNet	54k	369	1236
Ours	114k	687	2492

Table 2: **Number of classes in different datasets** using different partitioning methods.

HGE Num. Classes			
coarse	medium	fine	LR@1
65.3k	119k	200k	19.0
35.0k	65.3k	119.0k	21.2
18.5k	35.0k	65.3k	27.0
9.4k	18.5k	35.0k	25.3
3.8k	9.4k	18.5k	19.2
1.8k	3.8k	9.4k	10.6

Table 3: **Results with different partitions** using HGE on SF-XL.

the only difference that HGE also uses two coarser splits (*medium* and *coarse*) besides the regular partition (*fine*) used by the other two. The partitions are built using Google S2 Sphere library, and take as input two parameters, namely τ_{min} and τ_{max}, which define the minimum and maximum number of images within each cell. We empirically search for the best values for the parameters on the San Francisco eXtra Large (SF-XL) dataset, and we report the results in Tab. 3. We choose the partitions that lead to the best LR@1 using HGE, and, following their implementation, we use the finer HGE partition also as training set for PlaNet and MVMF. In practice, this leads to a value of $\tau_{min} = 100$ and $\tau_{max} = 2500$, as shown in Tab. 4 (where we also report the value of τ for other partitions. Note that we use proportions between different partitions size according to [10].

We tuned cells density on SF-XL since it is the most representative dataset for the studied setting. Remembering that these partitioning schemes are based on keeping cell density constant, to extend the comparison to the other adopted datasets (C-Pitts30k, C-Tokyo24/7), we scaled τ_{min} and τ_{max} according to the relative density of the other datasets with respect to SF-XL. In our method, instead, the partitioning only depends on the desired granularity of localization, so we kept the same $20m$ cells across all datasets.

Partitions of CPlaNet.

Regarding CPlaNet's [12] partitions, we carefully followed the authors' implementation: we created five *geoclass sets* for each of the experiments, where *geoclass set*$_1$ and *geoclass set*$_2$ evaluate the proximity distance using only the geographical and visual properties of the images respectively, while the remaining *geoclass sets* were generated by considering the distance as a stochastic linear combination

hyperparams	fine	HGE-medium	HGE-coarse
τ_{min}	100	100	100
τ_{max}	2500	5000	10000

Table 4: **Chosen hyperparameters** for previous methods partitioning. Note that Planet, HGE-fine and MvMF use the same partitioning.

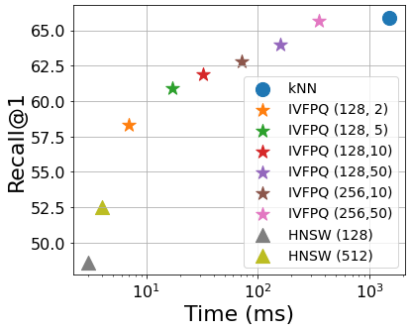

Figure 5: **Comparisons of best-performing Approximate Nearest Neighbor search algorithms.** We show only the pareto-optimal results, which are computed with an Inverted File Index with Product Quantization (IVFPQ) [5] and Hierarchical Navigable Small Worlds (HNSW) [9]. The parameters in parenthesis for IVFPQ indicate the number of subquantizers and the *nprobe*, *i.e.* the number of Voronoi cells to be searched (out of 1000). The parameters in parenthesis for HNSW indicates the number of connections each vertex has within the HNSW graph.

of these two modalities. We refer the reader to their paper for more details about how each *geoclass set* is formed. In their method, an additional hyperparameter is the number of classes in each *geoclass set* (*i.e.* their partition algorithm stopping condition). Finally, at inference time, the granularity considered for prediction is given by the intersections of the 5 *geoclass sets*. In Tab. 5 we report results using different values for each and using the same parameters α and β, which define the differences between the 5 *geoclass sets*. The table also reports in the first column the number of distinct cells obtained by the intersection of the different partitions. Also in this case we choose from the table the split which gave the best results for LR@1.

To export these hyperparameters to the other datasets, we kept the same average size of the cells in each *geoclass set*.

References

[1] 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. 2

Cells per geoclass						
# classes	gcs 1	gcs 2	gcs 3	gcs 4	gcs 5	LR@1
58233	30k	30k	39k	36k	33k	27.6
54144	20k	20k	26k	24k	22k	27.7
47412	10k	10k	13k	12k	11k	25.7

Table 5: **CPlaNet preliminary results** on SF-XL.

[2] Artem Babenko and Victor S. Lempitsky. The inverted multi-index. In *CVPR*, pages 3069–3076. IEEE Computer Society, 2012. 2

[3] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 369–386, Cham, 2020. Springer International Publishing. 2

[4] Mike Izbicki, Evangelos Papalexakis, and Vassilis Tsotras. Exploiting the earth's spherical geometry to geolocate images. In Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, volume 11907 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2019. 2

[5] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011. 2, 3

[6] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3251–3260, 2017. 2

[7] Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, S. Papadopoulos, and Y. Kompatsiaris. Leveraging efficientnet and contrastive learning for accurate global-scale location estimation. *ACM International Conference on Multimedia Retrieval*, 2021. 1

[8] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization. In *IEEE International Conference on Computer Vision*, 2019. 2

[9] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:824–836, 2020. 3

[10] Eric Müller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, volume 11216 of *Lecture Notes in Computer Science*, pages 575–592. Springer, 2018. 1, 2, 3

[11] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[12] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *ECCV*, 2018. 1, 2, 3

[13] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet - photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, 2016. 1, 2

Figure 6: **Qualitative results** using different pipelines on challenging queries.

Figure 7: **Qualitative results** using different pipelines on challenging queries.