# Supplementary for VLAD-BuFF: Burst-aware Fast Feature Aggregation for Visual Place Recognition

Ahmad Khaliq[1], Ming Xu[2], Stephen Hausler[3], Michael Milford[1], and Sourav Garg[4]

[1] Queensland University of Technology, Australia
[2] Australian National University, Australia
[3] CSIRO, Australia
[4] University of Adelaide, Australia
ahmad.khaliq@hdr.qut.edu.au, mingda.xu@anu.edu.au, stephen.hausler@csiro.au, michael.milford@qut.edu.au, sourav.garg@adelaide.edu.au

In this supplementary document, we present extended results, additional implementation and background details for the proposed method, details of different training configurations, dataset details, abbreviations, and additional qualitative analyses.

## 1 Extended Results

Here, we provide additional results for different preprocessing and training configurations of the baseline methods and our proposed method.

***Upgraded Baselines:*** In Tab. 4, we present an extended version of state-of-the-art comparisons (Tab. 1 of the main paper).

In block **A**, we include results for the authors' provided ResNet-101 based CosPlace model with 2048 dimensional descriptors, which is their best performing model as reported in their supplementary [4]. We also retrained CosPlace using its default GeM as well as our proposed VLAD-BuFF aggregation. It can be observed that our proposed aggregation on top of CosPlace's pipeline improves recall for most of the datasets.

In blocks **B** and **C**, we include the respective authors' trained models for CosPlace, EigenPlaces, and MixVPR – they all differ in terms of `dataset-loss-backbone` but provide a reference to compare against their other variants in Tab. 4.

We further include 'backbone-enhanced' versions of they key baselines. CosPlace and EigenPlaces used ResNet-101 and ResNet-50 backbones with GeM pooling and SFXL training dataset using their respective proposed loss functions. In block **D**, we include a newly trained model for EigenPlaces where we *only* replace its ResNet backbone with DINOv2. It can be observed that this replacement improves EigenPlaces' performance (compared to block **B**) but does not achieve state-of-the-art results. In block **F**, we use `GSV-MS-DINOv2` training configuration (as used by SALAD and our method). To compare different aggregation methods, we include newly trained `GSV-MS-DINOv2-GeM` and `GSV-MS-DINOv2-MixVPR` models. Table 4 shows that `GSV-MS-DINOv2-GeM` leads to an improved recall for MSLS, SPED, Tokyo247, and Baidu datasets, in comparison to `SFXL-Clas-R101/R50` based CosPlace and `SFXL-Clas-R50` based EigenPlaces. For `GSV-MS-DINOv2-MixVPR`, performance

**Table 4:** Recall@K comparison of our VLAD-BuFF against state-of-the-art VPR techniques on challenging benchmark datasets with the best recall **bolded** and second-best underlined.

| Method | | MSLS | | NordLand | | P250k-t | | SPED | | SFSM | | Tokyo247 | | StLucia | | AmsterTime | | Baidu | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Dim | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 |
| A) SFXL-Clas-R101: | | | | | | | | | | | | | | | | | | | |
| CosPlace (CP)† [4] | 2048 | 90.4 | 94.2 | 49.7 | 66.0 | 91.8 | 95.6 | 76.4 | 85.5 | 81.9 | 86.6 | 86.7 | 95.6 | 99.9 | 99.9 | 51.9 | 72.0 | 42.5 | 55.8 |
| CP w. GeM | 2048 | 86.2 | 91.9 | 47.1 | 63.1 | 92.0 | 97.6 | 76.3 | 88.6 | 80.2 | 85.9 | 89.2 | 94.9 | 99.2 | 99.9 | 51.7 | 72.1 | 41.7 | 54.2 |
| CP w. VLAD-BuFF | 2048 | 86.0 | 92.2 | 57.7 | 73.6 | 92.7 | 98.0 | 78.6 | 90.0 | 80.3 | 86.3 | 89.2 | 96.2 | 99.5 | 99.9 | 52.9 | 72.6 | 40.8 | 54.8 |
| B) SFXL-Clas-R50: | | | | | | | | | | | | | | | | | | | |
| CosPlace† [4] | 512 | 79.5 | 87.2 | 51.3 | 66.8 | 89.7 | 96.4 | 79.6 | 90.4 | 78.1 | 84.8 | 87.0 | 94.9 | 99.0 | 99.9 | 47.7 | 69.8 | 41.6 | 55.0 |
| EigenPlaces† [6] | 2048 | 87.4 | 93.0 | 64.4 | 79.6 | 92.8 | 97.4 | 82.2 | 93.7 | 84.6 | 87.9 | 87.9 | 94.9 | 99.5 | 99.9 | 49.8 | 72.4 | 61.5 | 76.4 |
| C) GSV-MS-R50: | | | | | | | | | | | | | | | | | | | |
| MixVPR† [1] | 4096 | 88.2 | 93.1 | 55.0 | 70.4 | 94.3 | 98.2 | 84.7 | 92.3 | 76.5 | 83.7 | 86.0 | 91.7 | 99.6 | 100 | 40.2 | 59.1 | 64.4 | 80.3 |
| D) SFXL-Clas-DINOv2: | | | | | | | | | | | | | | | | | | | |
| EigenPlaces | 512 | 91.1 | 95.4 | 66.8 | 82.4 | **95.8** | **99.1** | 86.2 | 95.1 | 86.5 | 89.9 | 92.7 | 97.1 | 99.7 | 99.9 | 51.6 | 72.6 | 62.7 | 77.7 |
| DINOv2-Pretrained: | | | | | | | | | | | | | | | | | | | |
| E) AnyLoc [11] | 49K | 59.2 | 76.6 | 9.6 | 16.3 | 88.1 | 95.5 | 76.6 | 90.0 | 70.4 | 79.9 | 85.7 | 94.3 | 79.0 | 90.4 | 41.3 | 61.8 | 75.2 | 87.6 |
| F) GSV-MS-DINOv2: | | | | | | | | | | | | | | | | | | | |
| GeM | 2048 | 88.9 | 94.7 | 54.2 | 72.0 | 93.4 | 97.9 | 85.2 | 92.6 | 81.0 | 85.4 | 90.5 | 95.2 | 99.8 | 100 | 45.7 | 67.7 | 63.7 | 79.5 |
| MixVPR | 4096 | 88.8 | 94.6 | 52.7 | 70.7 | 92.6 | 97.6 | 88.3 | 92.6 | 78.3 | 85.5 | 89.8 | 94.9 | 99.7 | 100 | 46.6 | 69.6 | 51.7 | 70.7 |
| SALAD† | 8448 | 92.2 | **96.2** | 76.1 | 89.2 | 95.1 | 98.5 | 92.1 | **96.2** | 85.4 | 88.5 | 95.2 | 97.1 | 99.9 | 100 | 58.7 | 79.2 | 68.5 | 81.7 |
| SALAD† + PCAW | 8192 | 90.3 | 95.3 | 74.5 | 87.5 | 94.8 | 98.3 | 91.3 | 95.6 | 83.9 | 87.4 | 93.3 | 97.5 | 99.9 | 100 | 52.2 | 71.3 | 67.2 | 80.9 |
| SALAD | 8448 | 91.9 | 95.8 | 70.8 | 85.4 | 94.9 | 98.7 | 90.1 | 95.9 | 86.0 | 88.9 | 95.2 | 97.8 | **100** | 100 | 59.8 | 79.2 | 69.1 | 82.5 |
| SALAD + PCAW | 8192 | 90.3 | 95.3 | 69.0 | 83.6 | 94.7 | 98.6 | 89.6 | 95.1 | 83.8 | 87.1 | 95.6 | 97.8 | 99.9 | 99.9 | 52.2 | 70.0 | 70.7 | 84.0 |
| NetVLAD + PCAW | 8192 | 91.8 | **96.2** | 76.2 | 89.2 | 95.7 | 98.6 | 91.3 | **96.2** | 88.2 | 90.4 | 97.1 | 98.1 | 99.9 | 100 | **61.8** | **82.1** | 76.4 | 86.7 |
| **VLAD-BuFF (Ours):** | | | | | | | | | | | | | | | | | | | |
| w. PCAW | 8192 | **92.4** | 95.8 | **78.0** | **90.4** | 95.6 | 98.7 | **92.8** | **96.2** | **88.3** | **91.0** | 96.5 | 98.1 | **100** | 100 | 61.4 | 81.9 | **77.5** | **87.9** |
| w. PrePool + PCAW | 4096 | 91.9 | 95.9 | 71.4 | 86.3 | 95.0 | 98.2 | 90.9 | 96.0 | 87.3 | 90.1 | **97.5** | **98.4** | 99.9 | 100 | 59.4 | 78.7 | 74.3 | 86.6 |

†Using the original authors' provided model.

marginally improved for MSLS, SPED, SFSM, Tokyo247, and AmsterTime but decreased for Baidu, Nordland, and Pitts250k. Nevertheless, all these aforementioned upgraded baselines still perform inferior to our proposed method VLAD-BuFF across all the datasets[5].

***SALAD Variants:*** In our main paper, we reported results for SALAD [9] model that we trained to ensure a fully fair comparison with our proposed method. In Table 4, we include results using the checkpoint provided by the authors (†). It can be observed that VLAD-BuFF still remains the state-of-the-art method across the majority of the datasets. It can be further observed that using WPCA on SALAD consistently reduces recall, which significantly limits its potential for efficient retrieval based on low dimensional global descriptors [4, 5, 18].

***Upgraded NetVLAD:*** We include an upgraded version of NetVLAD [2] (based on GSV-MS-DINOv2 training configuration) as another strong state-of-the-art baseline, which only differs from VLAD-BuFF in terms of aggregation and is referred to as NetVLAD + PCAW in Tab. 4. It can be observed that VLAD aggregation, common in both NetVLAD and VLAD-BuFF, is generally superior to all other baselines. VLAD-BuFF improves this with the proposed burstiness weighting, achieving higher recall on

[5] Note that models trained with DINOv2 as a backbone use 224×224 resolution for training and 322×322 for evaluation. The only exception is MixVPR, which requires a fixed image resolution due to its specific pooling method, thus we use 224×224 resolution for both training and evaluation of MixVPR.

the majority of datasets. Furthermore, our pre-pool projection based on PCA initialization retains superior performance with significantly reduced descriptor dimensions and aggregation compute time.

## 2 Implementation Details

Here, we provide background details for vanilla NetVLAD's soft assignment, followed by the description of parameter settings for our training configurations.

### 2.1 Vanilla NetVLAD Soft Assignment

Our proposed method VLAD-BuFF uses a burstiness weighting ($w_i$) based on feature-to-feature similarities. This is applied on top of feature-to-cluster assignment ($\alpha_{ik}$) to obtain intra-cluster burstiness weighting. Here, we provide the background details for $\alpha_{ik}$, originally proposed in NetVLAD [2]. The authors implemented it as a convolutional layer with $1 \times 1$ spatial support and $|C|$ weight filters/kernels, followed by a softmax across the spatial dimensions of the input local feature tensor. The convolutional filters are the learnable parameters of the NetVLAD aggregation layer, where each filter is initialized with the cluster center $C_k$. Furthermore, the authors used a multiplication factor for these weight filters during initialization to mimic the sparse assignment of conventional VLAD, as described in the supplementary material of NetVLAD [2] and detailed in their official implementation[6] which also notes the preference of 'vlad' over 'vladv2' for L2-normalized features.

### 2.2 Training and Evaluation Configurations

**GSV-DINOv2** For training and testing of GSV-DINOv2 pipeline, we used the official code[7] of the recent state-of-the-art method SALAD [9]. We substitute the SALAD aggregation layer with our VLAD-BuFF aggregation (or other baseline aggregation methods such as MixVPR and GeM). We exactly follow SALAD's training parameters setting, including the Multi-Similarity loss, AdamW optimizer, learning rate $6\mathrm{e}{-5}$, and batch size $60$ (each comprising 4 images per place). Each training iteration is completed in approx. 45 minutes on an NVIDIA A100, spanning 4 epochs, which saves the best 3 models according to Recall@1 on Pitts30k-val set. The number of clusters is set to 64 for both VLAD-BuFF and SALAD. While we employed image sizes of $224\times224$ and $322\times322$ for training and evaluation across most methods utilizing GSV-DINOv2 pipeline, MixVPR uniquely required $224\times224$ dimensions for both due to its strict architecture needs.

For our benchmarking, we used the evaluation codebases provided by [5, 6, 9][8] to access the official model checkpoints. For the datasets, we used dataset downloader provided by [5][9] for AmsterTime; SPED and Nordland were sourced from VPR-Bench [18]; Tokyo247 and StLucia from DVGL [5]; and Baidu from AnyLoc [11].

[6] https://github.com/Relja/netvlad/blob/master/README_more.md
[7] https://github.com/serizba/salad
[8] https://github.com/gmberton/VPR-methods-evaluation
[9] https://github.com/gmberton/VPR-datasets-downloader

**Deep Visual Geolocalization (DVGL) Benchmark** We used the official code[10] for the recent deep visual geolocalization benchmark [5] to train and test NetVLAD and VLAD-BuFF. This corresponds to the `P30K-Trip-R18` (and `P30K-Trip-DINOv2`) configuration. Here, we provide details of the key train/test parameter settings which strictly follow the official code. For training, batch size is 4 where each sample is composed of 12 images: one query (anchor), one positive and ten negatives mined through a 'partial' strategy [5]. Triplet margin loss [3] is used with Euclidean distances and 0.1 margin, and optimized using Adam with a learning rate of 0.00001. The backbone network is ResNet18 which is clipped beyond conv4 and its pretrained ImageNet weights frozen, keeping only conv4 as trainable. We neither use PCA nor any supervised linear projection (fully connected layer) *after* feature aggregation. All local features are L2-normalized before aggregation. No data augmentation is used for training. Images are resized to 640×480 for both training and evaluation. An early stopping criterion is defined to track improvement in Recall@5 over 3 epochs, with maximum training epochs set to 15. The number of clusters for NetVLAD and VLAD-BuFF are set to 64. For `P30K-Trip-DINOv2`, everything remains the same except that the backbone is used in the same way as that for GSV-DINOv2 training described earlier.

### 2.3 Burstiness Exponent $p$

For the burstiness weighting layer in VLAD-BuFF, we used $p$ as an exponent for the soft count. This was initialized as 1.0. We observed that its final trained value converged between 0.7 and 0.8 for different training configuration models. This is in line with the findings of [10] where their best performing value was 0.5 (as opposed to 1.0), which our method converges towards.

## 3 Datasets Details

**Pittsburgh:** The Pittsburgh dataset [15] consists 250k images collected using Google Street View. The dataset is divided into a number of places, with 24 images captured per place from different viewing directions (two pitch and twelve yaw directions). The reference and query images are captured at different times of the day and several years apart. In this work, we use the full dataset (Pitts250k) for testing and use its 30K image subset (Pitts30K) [2] for training and validation. The Pitts250k-test set contains 8280 and 83952 images in query and reference traverses.
**Tokyo 24/7:** The Tokyo 24/7 dataset [14] contains a large database (75984 images) collected from Google Street View and a small set of query images collected using smartphones. The query traverse (315 images) are captured at 125 different locations with different viewing directions at different times of the day (including night time).
**Mapillary Street Level Sequences (MSLS):** The MSLS dataset [16] consists of approximately 1.6 million images collected from 30 different cities across the world over seven years. The dataset is split into training, validation and test sets, with ground truth released for the training and validation sets. In this work, we evaluate our method on the

[10] https://github.com/gmberton/deep-visual-geo-localization-benchmark

validation set (740 and 18871 images for query and reference traverses), as per recent works [5, 8].

**St Lucia:** The St Lucia dataset [12] was collected by a car driving around a fixed route of a suburb (St Lucia) in Brisbane, at different times of day. We utilize the subset of this dataset used in the deep visual geolocalization benchmark [5], which uses the temporally first and last drives along this route. The query and reference traverses consist of 1464 and 1549 images respectively.

**Nordland:** The Nordland dataset [13] is recorded by a train traveling through Norway during four different seasons for 728km. We used the subset of the dataset as used in VPR-Bench [18], where the winter route is used as the database (27K images) and the summer route is the query set (2.7K images).

**Baidu:** The Baidu dataset [11] captured in a mall, includes images with varied camera poses and provides both the ground truth location and 3D pose for each image. It is suitable for 6-Degrees of Freedom (DoF) Localization and VPR testing, comprising 2292 query images and 689 reference images. The dataset is particularly challenging due to perceptual aliasing, dynamic objects (e.g., people), and semantically rich elements like billboards and signs, offering a comprehensive environment for VPR evaluation.

**AmsterTime:** The AmsterTime dataset [17] captures Amsterdam's urban/historic scenery at different times across decades, offering a unique challenge of recognizing the same location from different viewpoints and colorscale (RGB vs grey). The query and reference traverses consist of 1231 and 1231 images [6].

**San Francisco SMall:** The San Francisco Small (SF-SMall) dataset [4], a subset of a larger collection SF-XL, focuses on the San Francisco area, providing a dense urban environment for VPR evaluation. The query and reference traverse consists of 1K and 27191 images.

**SPED:** The SPED dataset [7] is renowned for its diverse environmental conditions, showcasing substantial seasonal and illumination changes, with winter scenes as queries (607 images) and summer scenes as references (607 images). This variability, including moderate viewpoint shifts, makes SPED a challenging dataset for VPR.

## 4 Abbreviations

In Tab. 5, we provide a list of abbreviations used in the main paper.

**Table 5:** List of abbreviations used in this paper.

| Abbreviation | Description |
|---|---|
| GAP | Global Average Pooling |
| GeM | Generalized Mean Pooling |
| GPO | Generalized Pooling Operator |
| MAC | Maximum Activations of Convolution |
| PCA | Principal Component Analysis |
| ViT | Vision Transformer |
| VLAD | Vector of Locally Aggregated Descriptors |
| VPR | Visual Place Recognition |

## 5 Qualitative Analysis

Here, we provide additional qualitative examples that compare the weighting patterns between vanilla NetVLAD's soft assignment and VLAD-BuFF's burstiness, see Fig. 4-Fig. 7 with detailed captions. In Fig. 6 and Fig. 7, we show soft assignment weighting for *both* NetVLAD and VLAD-BuFF to highlight the impact of burstiness weighting on learning backend local features, resulting in noticeable variations in their cluster assignment. In Fig. 7, we present a failure case of our method. We used the P30K-R18 trained models for these qualitative analyses; in the subsequent section, we describe our procedure to select the clusters for the visualization purpose.

### 5.1 Procedure

We select the triplet (query, positive and negative) using ground truth by obtaining query indices where VLAD-BuFF performed better than vanilla NetVLAD based weighting. For each image, we compute its $C \times D$ VLAD representation which is L2-normalized (first cluster-wise and then over the flattened vector [2]). For each normalized $D$-dimensional aggregated residual vector per cluster $C_k$, we compute a triplet margin $m_k$ as the difference between query-negative and query-positive Euclidean distances:

$$m_k = ||\mathbf{q}_k - \mathbf{n}_k|| - ||\mathbf{q}_k - \mathbf{p}_k|| \tag{1}$$

For a cluster, a larger margin indicates its ability to better separate a negative from the query-positive pair. We use these margins to obtain the ranking of clusters for both the soft-assignment NetVLAD baseline and VLAD-BuFF models, referred to as $r_k^{NV}$ and $r_k^{VB}$ respectively. We then compute cluster rank difference between the two models to obtain the cluster index which maximally changed the triplet margin, as below:

$$\bar{k} = \text{argmax}(r_k^{NV} - r_k^{VB}) \tag{2}$$

Intuitively, $\bar{k}$ will be selected as the cluster index which led to low margin in NV and high margin in VB, thus being crucial in match selection for the query.

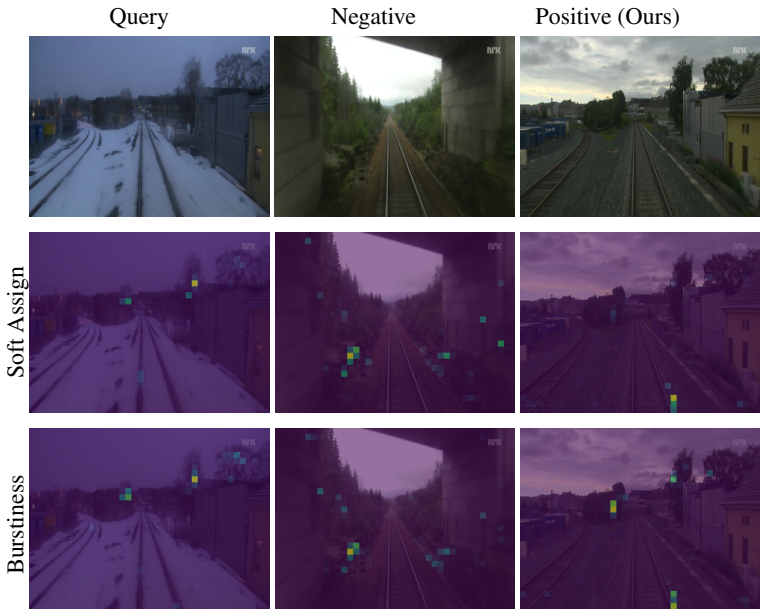


**Fig. 4:** In this example from the Nordland dataset, both the vanilla soft assignment (2nd row) and our proposed method (3rd row) select image regions lying just beneath the tree canopies (first column). In the positive image (third column), both the methods select some features on the railway tracks (highlighted in yellow). However, our method downweights these features relative to those below the tree canopies (more yellow than tracks), thus improving the query-positive matching.


**Fig. 5:** In this example from the St Lucia dataset, vanilla soft assignment results in several highly-weighted features, including those on trees' shadows (query and negative) and vehicles (positive). However, our proposed weighting (3rd row) downweights all the features found on shadows and vehicles, and instead selects a signboard with a consistent weighting pattern between the query-positive pair.

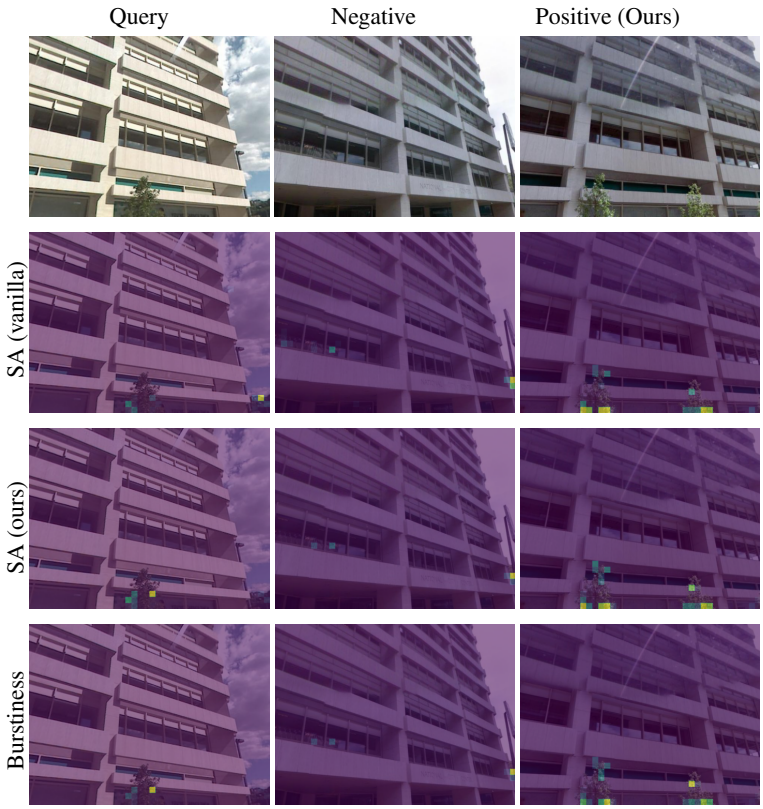


**Fig. 6:** In this example from the Pitts30k dataset, we highlight the improved behavior of local features, learnt through VLAD-BuFF with burstiness weighting. Thus, we show the soft assignment weighting using both the vanilla NetVLAD model as well as the proposed VLAD-BuFF model. The weighting patterns for the negative and the positive remain similar across the rows. However, in the query image, it can be observed that VLAD-BuFF's soft assignment selects the overlapping region between the tree and the building at the bottom center of the image, whereas NetVLAD's soft assigment selects the trees at the bottom right. The former is more consistent with the feature selection in the positive, thus improving the query-positive matching. Note that the burstiness weighting (4th row) had only a slight impact on top of the soft assignment weighting (3rd row), thus the variation in the weighting pattern between NetVLAD and VLAD-BuFF is attributed more to feature-to-centroid distances than feature-to-feature distances in this case.


**Fig. 7:** In this example from the St Lucia dataset, we show a failure case of VLAD-BuFF. Note that the middle column is the positive, correctly selected by the vanilla NetVLAD, and the third column is the negative, incorrectly selected by VLAD-BuFF. Here, the soft assignment of VLAD-BuFF model (3rd row) selects certain features on the car, consistently across the query (1st column) and the negative (3rd column), which leads to incorrect matching. On the other hand, NetVLAD's soft assignment (2nd row) selects features along the lane markings for both the query and the positive.

## References


1. Ali-bey, A., Chaib-draa, B., Giguère, P.: Mixvpr: Feature mixing for visual place recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2998–3007 (2023)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5297–5307 (2016)
3. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Bmvc. vol. 1, p. 3 (2016)
4. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4878–4888 (2022)
5. Berton, G., Mereu, R., Trivigno, G., Masone, C., Csurka, G., Sattler, T., Caputo, B.: Deep visual geo-localization benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5396–5407 (2022)
6. Berton, G., Trivigno, G., Caputo, B., Masone, C.: Eigenplaces: Training viewpoint robust models for visual place recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11080–11090 (October 2023)
7. Chen, Z., Liu, L., Sa, I., Ge, Z., Chli, M.: Learning context flexible attention model for long-term visual place recognition. IEEE Robotics and Automation Letters **3**(4), 4015–4022 (2018)
8. Hausler, S., Garg, S., Xu, M., Milford, M., Fischer, T.: Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14141–14152 (2021)
9. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition (2023)
10. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 1169–1176. IEEE (2009)
11. Keetha, N., Mishra, A., Karhade, J., Jatavallabhula, K.M., Scherer, S., Krishna, M., Garg, S.: Anyloc: Towards universal visual place recognition. IEEE Robotics and Automation Letters (2023)
12. Milford, M., Wyeth, G.: Mapping a suburb with a single camera using a biologically inspired slam system. IEEE Transactions on Robotics **24**(5), 1038–1053 (2008)
13. Sünderhauf, N., Neubert, P., Protzel, P.: Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In: Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA). p. 2013 (2013)
14. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1808–1817 (2015)
15. Torii, A., Sivic, J., Pajdla, T., Okutomi, M.: Visual place recognition with repetitive structures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 883–890 (2013)
16. Warburg, F., Hauberg, S., López-Antequera, M., Gargallo, P., Kuang, Y., Civera, J.: Mapillary street-level sequences: A dataset for lifelong place recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2626–2635 (2020)
17. Yildiz, B., Khademi, S., Siebes, R.M., van Gemert, J.: Amstertime: A visual place recognition benchmark dataset for severe domain shift. arXiv preprint arXiv:2203.16291 (2022)
18. Zaffar, M., Garg, S., Milford, M., Kooij, J., Flynn, D., McDonald-Maier, K., Ehsan, S.: Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. International Journal of Computer Vision pp. 1–39 (2021)