frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 29 January 2025
DOI 10.3389/fnbot.2024.1490267

Check for updates

OPEN ACCESS

EDITED BY
Long Jin,
Lanzhou University, China

REVIEWED BY
Elishai Ezra Tsur,
Open University of Israel, Israel
Fabio Schittler Neves,
Fraunhofer Institute for Industrial Mathematics (ITWM), Germany

*CORRESPONDENCE
Ugur Akcal
✉ makcal2@illinois.edu
Ivan Georgiev Raikov
✉ iraikov@stanford.edu
Girish Chowdhary
✉ girishc@illinois.edu

RECEIVED 02 September 2024
ACCEPTED 26 December 2024
PUBLISHED 29 January 2025

CITATION
Akcal U, Raikov IG, Gribkova ED, Choudhuri A, Kim SH, Gazzola M, Gillette R, Soltesz I and Chowdhary G (2025) LoCS-Net: Localizing convolutional spiking neural network for fast visual place recognition.
*Front. Neurorobot.* 18:1490267.
doi: 10.3389/fnbot.2024.1490267

COPYRIGHT
© 2025 Akcal, Raikov, Gribkova, Choudhuri, Kim, Gazzola, Gillette, Soltesz and Chowdhary. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# LoCS-Net: Localizing convolutional spiking neural network for fast visual place recognition

Ugur Akcal[1,2,3]*, Ivan Georgiev Raikov[4]*, Ekaterina Dmitrievna Gribkova[3,5], Anwesa Choudhuri[3,6], Seung Hyun Kim[7], Mattia Gazzola[7], Rhanor Gillette[5,8], Ivan Soltesz[4] and Girish Chowdhary[2,3,9]*

[1]The Grainger College of Engineering, Department of Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, IL, United States, [2]The Grainger College of Engineering, Siebel School of Computing and Data Science, University of Illinois Urbana-Champaign, Urbana, IL, United States, [3]Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL, United States, [4]Department of Neurosurgery, Stanford University, Stanford, CA, United States, [5]Neuroscience Program, Center for Artificial Intelligence Innovation, University of Illinois Urbana-Champaign, Urbana, IL, United States, [6]The Grainger College of Engineering, Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL, United States, [7]The Grainger College of Engineering, Mechanical Science and Engineering, University of Illinois Urbana-Champaign, Urbana, IL, United States, [8]Department of Molecular and Integrative Physiology, University of Illinois Urbana-Champaign, Urbana, IL, United States, [9]The Grainger College of Engineering, College of Agriculture and Consumer Economics, Department of Agricultural and Biological Engineering, University of Illinois Urbana-Champaign, Urbana, IL, United States

Visual place recognition (VPR) is the ability to recognize locations in a physical environment based only on visual inputs. It is a challenging task due to perceptual aliasing, viewpoint and appearance variations and complexity of dynamic scenes. Despite promising demonstrations, many state-of-the-art (SOTA) VPR approaches based on artificial neural networks (ANNs) suffer from computational inefficiency. However, spiking neural networks (SNNs) implemented on neuromorphic hardware are reported to have remarkable potential for more efficient solutions computationally. Still, training SOTA SNNs for VPR is often intractable on large and diverse datasets, and they typically demonstrate poor real-time operation performance. To address these shortcomings, we developed an end-to-end convolutional SNN model for VPR that leverages backpropagation for tractable training. Rate-based approximations of leaky integrate-and-fire (LIF) neurons are employed during training, which are then replaced with spiking LIF neurons during inference. The proposed method significantly outperforms existing SOTA SNNs on challenging datasets like Nordland and Oxford RobotCar, achieving 78.6% precision at 100% recall on the Nordland dataset (compared to 73.0% from the current SOTA) and 45.7% on the Oxford RobotCar dataset (compared to 20.2% from the current SOTA). Our approach offers a simpler training pipeline while yielding significant improvements in both training and inference times compared to SOTA SNNs for VPR. Hardware-in-the-loop tests using Intel's neuromorphic USB form factor, Kapoho Bay, show that our on-chip spiking models for VPR trained via the ANN-to-SNN conversion strategy continue to outperform their SNN counterparts, despite a slight but noticeable decrease in performance when transitioning from off-chip to on-chip, while offering significant energy efficiency. The results highlight the outstanding rapid prototyping and real-world deployment capabilities of this approach, showing it to be a substantial step toward more prevalent SNN-based real-world robotics solutions.

KEYWORDS

**spiking neural networks, robotics, visual place recognition, localization, supervised learning, convolutional networks**

# 1 Introduction

Visual place recognition (VPR) refers to the capability of identifying locations within a physical environment solely through visual inputs. It is essential for autonomous navigation of mobile robots, indoor assistive navigation aid, augmented reality, and geolocalization (Lanham, 2018; Reinhardt, 2019; Weyand et al., 2016; Seo et al., 2018; Li et al., 2018; Shan et al., 2015). These applications generally involve complex dynamic scenes, perceptual aliasing, viewpoint and appearance variation, which render VPR extremely challenging.

VPR has been approached via deep learning techniques (Radenović et al., 2018; Chen et al., 2017; Sünderhauf et al., 2015) and through various supervised and self-supervised feature descriptor representations (DeTone et al., 2018; He et al., 2018; McManus et al., 2014). Despite their promise, many of these methods face significant practical challenges (Lynen et al., 2015, 2020). For example, they often rely on large, deep networks with time-consuming training processes and dense feature extraction, ultimately making them computationally expensive, memory-intensive, and energy-demanding. Such limitations significantly reduce the ability for real-world deployment of conventional artificial neural networks (ANNs) on robotic platforms with limited on-board resources (Doan et al., 2019). Spiking neural networks (SNNs) offer an alternative with their remarkable potential for computationally efficient operation when they are implemented on neuromorphic hardware (Davies et al., 2021). However, previous work on SNN models for VPR has suffered from scalability problems that impede their application to data with a large number of locations. In addition, the majority of the aforementioned methods formulate VPR as an image retrieval task (Garg et al., 2021), the solution of which aims for the correct association of given query images with a set of reference images. Such formulation requires the employment of a confusion matrix (a.k.a. distance matrix) (Garg et al., 2022) populated with similarity scores based on the distances between model-specific feature descriptors. A commonly-used similarity metric is the cosine similarity (Naseer et al., 2018), which is reported to be computationally expensive when evaluating high-dimensional feature vectors (Zhang et al., 2021).

These drawbacks have motivated our approach to VPR, described in this paper, in which an SNN model is implemented using an ANN-to-SNN conversion method to enable backpropagation-based training, resulting in fast training and inference times. We employ a smooth rate-based approximation (Hunsberger and Eliasmith, 2015) of the leaky integrate-and-fire (LIF) neurons (Burkitt, 2006) during the training. Once the training session is completed the rate-based units are substituted with the spiking LIF neurons and the resulting spiking network is used for inference.

We formulate VPR as a classification task, where the SNN model predicts place labels that uniquely correspond to the locations in a discretized navigation domain. We evaluate our method with the challenging real-world benchmark datasets Nordland (Olid et al., 2018) and Oxford RobotCar (Maddern et al., 2017, 2020). Our model, the Localizing Convolutional Spiking Neural Network (LoCS-Net), outperforms other SOTA SNN-based VPR methods on both the Nordland (Olid et al., 2018) and the Oxford RobotCar dataset (Maddern et al., 2017, 2020) in terms of precison at 100% recall (P@100%R).

The main contributions of this work are as follows. (a) To the best of our knowledge, LoCS-Net is the first SNN that is trained to perform the VPR task by means of ANN-to-SNN conversion and backpropagation. (b) LoCS-Net is an end-to-end SNN solution. Therefore, LoCS-Net does not require further processing of its outputs for recognizing places. In that sense, LoCS-Net saves all the computation resources that traditional VPR algorithms would typically expend on feature encoding, descriptor matching, computing similarity scores, and storing a distance matrix. (c) We demonstrate that our proposed SNN model yields the fastest training time, the second fastest inference time, and the best VPR performance in P@100%R among its SNN counterparts. This poses LoCS-Net as a significant step toward deployment of SNN-based VPR systems on robotics platforms for real-time localization. (d) We report the challenges we experienced when deploying LoCS-Net on the neuromorphic Loihi chips in detail. We strongly believe that our in-depth discussion on hardware deployment will be useful for the SNN-VPR community.

# 2 Related work

Task-specific feature descriptors are the very core of traditional VPR systems, which can be grouped into two categories: (1) Local descriptors, (2) Global descriptors. Local descriptors may scan the given images in patches of arbitrary size and stride. These patches are then compared to their immediate neighborhood to determine the distinguishing patterns (Loncomilla et al., 2016). In general, previous VPR work utilizing local descriptors (Johns and Yang, 2011; Kim et al., 2015; Zemene et al., 2018) employs sparse filters that extract so-called key-points (Mikolajczyk and Schmid, 2002; Matas et al., 2004). These key-points can be marked by the descriptions generated through the application of methods including SIFT (Lowe, 1999), RootSIFT (Arandjelović and Zisserman, 2012), SURF (Bay et al., 2006), and BRIEF (Calonder et al., 2011). In this way, the combination of heuristics-based detectors and local descriptors can be used for: (A) Representing images, (B) Comparing two images with respect to their descriptors to determine how similar they are. In addition, local features can be combined with other embeddings (Tsintotas et al., 2022) while leveraging their robustness against the variations in the robot's pose. However, local descriptors can be computationally heavier and more sensitive to illumination changes (Masone and Caputo, 2021). Global descriptors (Oliva and Torralba, 2006; Torralba et al., 2008), on the other hand, do not require a detection phase and directly encode the holistic properties of the input images. Although this might save the global descriptor-based VPR methods (Liu and Zhang, 2012; Schönberger et al., 2018; Revaud et al., 2019; Yin et al., 2019) some compute time, they are more vulnerable to robot pose changes than their local descriptor-based counterparts while being inept at capturing geometric structures (Dube et al., 2020). Yet, global descriptors are reported to be more effective in the case of varying lighting conditions (Lowry et al., 2015). Furthermore,

there are hybrid approaches (Siméoni et al., 2019; Cao et al., 2020; Hausler et al., 2021), which combine the strengths of both approaches.

Deep learning has made key contributions to recent work on VPR. An influential deep-learning-based approach is NetVLAD (Arandjelovic et al., 2016), which is a supervised method for place recognition, based on the Vector of Locally Aggregated Descriptors (VLAD), a technique to construct global image feature representations from local feature descriptors. NetVLAD uses a pre-trained feature extraction network, such as AlexNet (Krizhevsky et al., 2017), to extract the local features, and a loss function that aims to minimize the distance between a baseline input and the most similar image (the positive example), while maximizing the distance between baseline input and the most dissimilar image (the negative example). This loss function is also known as the triplet loss function. Several authors have extended NetVLAD in different directions, and NetVLAD-based methods still perform very competitively (Hausler et al., 2021; Yu et al., 2020).

SNNs have been of interest for various robotics tasks, including not only VPR, but also object detection (Kim et al., 2020), regression (Gehrig et al., 2020), and control of aerial platforms (Vitale et al., 2021) due to their significant potential for computational efficiency (Zhu et al., 2020). Published VPR methods based on SNNs are relatively recent, compared to other robotics research areas. Among them, Hussaini et al. (2022) is reported to be the first high-performance SNN for VPR. There, the authors propose a feed-forward SNN, where the output neuron activations are filtered through a custom softmax layer. Follow-up work by the same authors (Hussaini et al., 2023) introduced a framework where localized spiking neural ensembles are trained to recognize places in particular regions of the environment. They further regularize these networks by removing output from "hyper-active neurons," which exhibit intense spiking activity when provided with input from the regions outside of the ensemble's expertise. This framework yields a significant improvement over its predecessor while demonstrating either superior or competitive VPR performance compared to the traditional methods. A recent study by Hines et al. (2024) presented an SNN model composed of an ensemble of modified BliTNet (Stratton et al., 2022) modules, each tuned to specific regions within the navigation domain. During training, spike forcing is utilized to encode locations uniquely, which are later identified by monitoring the output neuron with the highest spike amplitude. The authors report remarkable improvements in both training and inference times, alongside achieving superior or comparable VPR performance compared to earlier SNN models. However, training of these SNN approaches do not scale with the increasing volume of training data. In addition, heuristics such as the assignment of neural ensembles to spatial regions, nearest neighbor search in the similarity matrix, and the regularization process further complicate the training process and the computational efficiency of the model. In contrast to these previous SNN-based approaches, we propose an end-to-end solution that is much easier to train and to deploy without requiring heuristic training.

# 3 LoCS-Net model for visual place recognition

Here, we begin with an overview of the task formulation and the architecture of LoCS-Net in Section 3.1. Section 3.2 formally poses the VPR problem as a classification task. Then, in Section 3.3, we walk through the LoCS-Net pipeline and its key design choices. Moreover, Section 3.3 provides a summary of the ANN-to-SNN conversion paradigm while elaborating on its use for the present work. We would like to refer the readers to the supplementary information and to the figshare repository of our code for further implementation details: https://figshare.com/s/c159a8680a261ced28b2.

## 3.1 Overview

Figure 1 depicts the overall architecture of LoCS-Net. The input to the model is a set of images sampled along a trajectory that traverses a bounded navigation domain. The domain is discretized by means of a uniform grid (orange lines in Figure 1) and each image is assigned an integer *place* label based on the tile traversed at the time of sampling the image. In this manner, we define the VPR task as a classification problem as discussed in Section 3.2.

Each layer in the LoCS-Net model consists of LIF neurons (Burkitt, 2006). In order to train the model, these neurons are converted to rate-based approximations of LIF units (Hunsberger and Eliasmith, 2015). Rate-based LIF approximations are continuous differentiable representations of the LIF activation function. The LIF activation function describes the time evolution of the neuron's membrane potential, and it is discontinuous: when the membrane potential reaches a threshold value, it is reset back to a pre-determined state. The rate-based approximation is a continuous function that describes the neuron's firing rate as a function of its input, enabling the use of back-propagation algorithms for training. However, this doesn't prevent the substitution of the approximate LIF neurons with the original ones for inference after the training is complete. A number of authors have reported successful applications (Rueckauer et al., 2017; Hu et al., 2021; Patel et al., 2019) of ANN-to-SNN conversion.

## 3.2 VPR as a classification task

A common practice in approaching the VPR task is to pose it as an image retrieval problem where the goal is to compute and store descriptors that would effectively encode both the set of query images and the collection of reference images to match (Lajoie and Beltrame, 2022). The encoding process is followed by an image retrieval scheme, which is based on comparing query embeddings ($z_q$) to the database of reference descriptors ($z_r$) with respect to the customized similarity metrics. Nevertheless, computation of the descriptors is numerically expensive. In contrast, we formulate the VPR task as a classification problem in order to bypass the encoding phase of the images. We designed the LoCS-Net so that it would uniquely map the given input images to the mutually

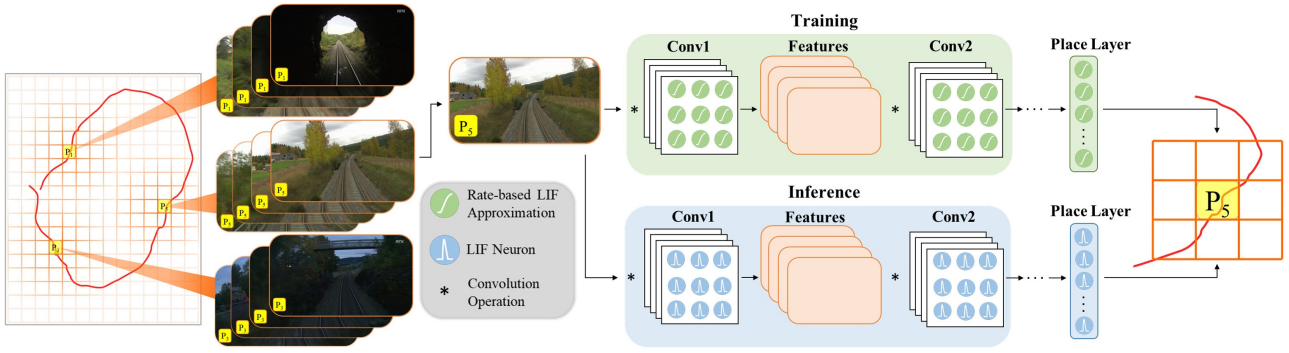


FIGURE 1
LoCS-Net VPR system: a convolutional network of rate-based LIF neurons (Hunsberger and Eliasmith, 2015) is trained over a set of annotated images sampled over a trajectory (the red curve) traversing a finite discretized (orange grid bounded by gray lines) navigation domain. The VPR task is formulated as a classification problem where each tile of the grid ($P_1, P_2, P_3, \ldots$) corresponds to a distinct location. After training, the LIF approximations are substituted with the spiking LIF neurons (Burkitt, 2006) for the inference step.

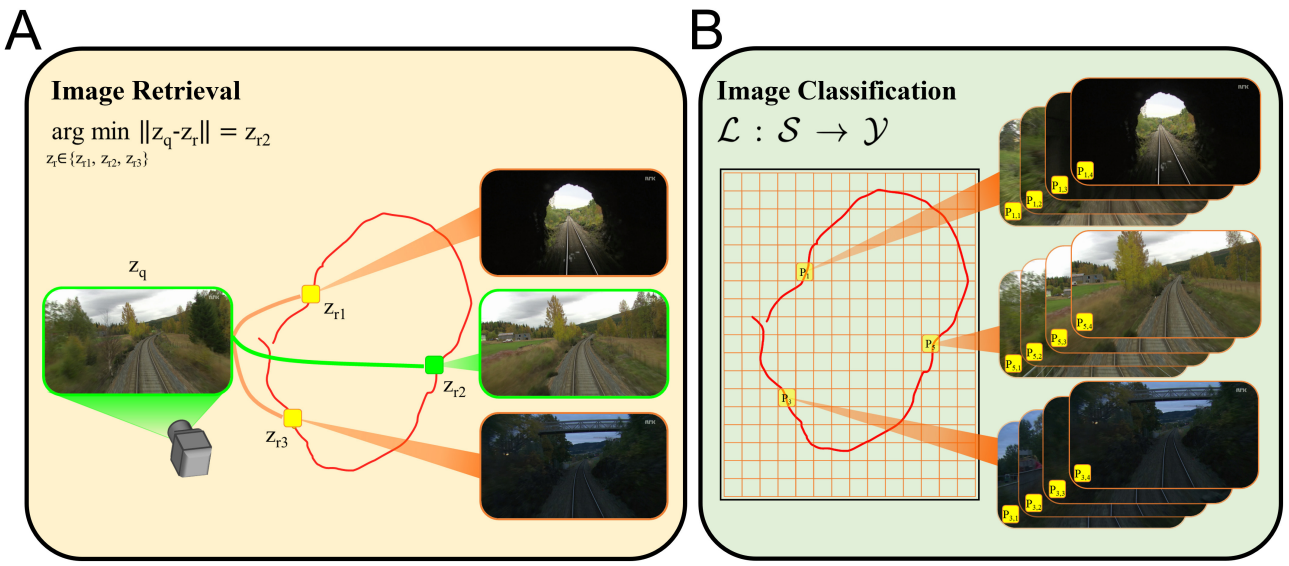


FIGURE 2
VPR can be posed as image retrieval task or image classification problem. For both formulations we consider a set of images collected over a trajectory (the red curves) traversing a finite navigation domain. **(A)** A popular VPR solution is based on generating descriptors for query ($z_q$) and reference images ($z_r$), which are then compared to each other in terms of a distance (or similarity) metric $\|\cdot\|$ in order to retrieve the reference image corresponding to the correct place. For instance, $z_{r2}$ represents the most similar reference image to the given query image. **(B)** In contrast, the image classification formulation of the VPR task requires an arbitrary discretization of the navigation domain to define the classes $P_i$ (the places where $i \in \mathcal{Y}$) that annotate the images $s \in \mathcal{S}$. Then, a classifier $\mathcal{L}$ is trained to map images $s \in \mathcal{S}$ to the correct place labels $i \in \mathcal{Y}$. The image annotation $P_{i,j}$ denotes the $j^{\text{th}}$ image associated with the class $P_i$.

exclusive classes, which are the distinct places, as discussed in Sections 3.1, 3.3.

Figure 2 illustrates how our work formulates VPR differently compared to the image retrieval VPR formulation. We first discretize the navigation domain by using a uniform rectangular grid (Figure 2B, the orange lines). Here, each tile of the grid defines a distinct place $P_i$, $i = 1, 2, 3, \cdots$. We would like to note that the navigation domain can be any physical environment with points described by spatial coordinates. Although we use a uniform rectangular grid to discretize the top-down view of the domain of interest, our approach is flexible with respect to the definition of places, and permits 3-D as well as 2-D discretization. As one of many ways to generate the training and test data, we sample images over numerous trajectories traversing the discretized navigation domain. Suppose that an image $s \in \mathcal{S}$ is sampled at the time instant when the camera is in the region represented by tile $P_5$. Then, this image would be annotated by the place label $5 \in \mathcal{Y}$. Namely, the image $s$ belongs to the class represented by the tile $P_5$. Thus, given a query image, our goal is to train a spiking neural network model that would correctly infer the associated place labels. Hence, we pose the VPR task as an image classification problem in this fashion. We

now formally describe the VPR task as a classification problem as follows.

Consider a set of images, $\mathcal{S} = \{s \in \mathbb{R}^{C\times H\times W} | \, X(s) \in \mathcal{D}\}$, where $C$ is the number of color channels, $H$ and $W$ are the height and width of the images in pixels and $\mathcal{D}$ is a pre-determined finite horizontal navigation domain. Here, $X : \mathcal{S} \to \mathcal{D}$ is a function that maps the images $s \in \mathcal{S}$ to the planar spatial coordinates $[x_s, y_s]^{\mathrm{T}} \in \mathcal{D} = \{[d_1, d_2]^{\mathrm{T}} \in \mathbb{R}^2 | \, x_{\min} \leq d_1 \leq x_{\max} \wedge y_{\min} \leq d_2 \leq y_{\max}\}$ where $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ are the bounds of $\mathcal{D}$. $X(s)$ describes the in-plane spatial state of the camera with respect to a local frame of choice when $s \in \mathcal{S}$ is sampled. The set $\mathcal{Y} = \{i \in \mathbb{N} | \, i \leq N_P\}$ contains the place labels that annotate $s \in \mathcal{S}$ where $N_P$ is the number of assumed places. Each $y \in \mathcal{Y}$ corresponds to a $P_y \subset \mathcal{D}$ such that $P_y \cap P_i \equiv \emptyset$, $y \neq i \wedge i \in \mathcal{Y}$. We formulate the VPR task as an image classification problem, where each class is assumed to be mutually exclusive. That is, each image belongs exactly to one class. Our goal is to design a mapping $\mathcal{L} : \mathcal{S} \to \mathcal{Y}$ that correctly predicts the place label $y \in \mathcal{Y}$ of any given $s \in \mathcal{S}$. One should note that the approach we describe here is different than the image retrieval formulation as we want $\mathcal{L}$ to predict the place labels instead of directly associating the input images with the reference images.

## 3.3 Localizing convolutional spiking neural network

The design of LoCS-Net is defined mainly by two ideas: (1) Discretization of the given finite navigation domain, (2) Leveraging the back-propagation algorithm by adopting the ANN-to-SNN conversion paradigm. We now walk through the details of these ideas together with the architecture of LoCS-Net and its building blocks, LIF neurons.

### 3.3.1 The LIF neuron model

Unlike standard artificial neurons, which are defined by time-independent differentiable non-linear transfer functions with continuous outputs, spiking neurons have time-dependent dynamics that aim to capture the information processing in the biological neural systems by emitting discrete pulses (Burkitt, 2006). Equation 1 describes the dynamics of an LIF neuron.

$$C_m \frac{dv(t)}{dt} = -\frac{C_m}{\tau_m}\left[v(t) - v_0\right] + I_s(t) + I_{inj}(t) \tag{1}$$

where $C_m$ is the membrane capacitance, $\tau_m$ is the passive membrane time constant, and $v_0$ is the resting potential. Above formulation considers a resetting scalar state variable, the membrane potential $v(t)$, which will be reinitialized at $v(t) = v_{reset}$ after reaching a threshold, $v(t) = v_{th}$. Whenever the re-initialization happens at time $t = t_{spike}$, the output of the LIF neuron ($o(t)$) will be an impulse signal of unity. We name this a spike event. One can express a spike event of an LIF neuron by Equation 2, which incorporates Dirac's delta function centered at the time of re-initialization.

$$o(t_{spike}) = \delta\left[v(t_{spike}) - v_{th}\right] \tag{2}$$

The right hand side of Equation 1 includes three terms: (1) An exponential decay term (a.k.a the passive membrane leak), (2) $I_s(t)$, the sum of incoming synaptic currents, which are mostly unit impulses filtered through a first order delay and/or multiplied by some scalar, and finally (3) An injection term, $I_{inj}(t)$, that describes the input currents other than synaptic currents. This can be some bias representing the background noise in the corresponding neural system, or just some external input.

Solving the sub-threshold dynamics described by Equation 1 for the firing rate $\rho[I_s(t)]$ of an LIF neuron and assuming $I_{inj}(t) = 0$ for all $t \geq 0$ yields the following.

$$T_{spike} = -\tau_m \log\left(1 - \frac{(v_{th} - v_{reset})\frac{C_m}{\tau_m}}{(v_0 - v_{reset})\frac{C_m}{\tau_m} + I_s(t)}\right) \tag{3}$$

$$\rho[I_s(t)] = \begin{cases} 0 & \text{if } I_s(t) \leq I_{th} \\ \dfrac{1}{T_{ref} + T_{spike}} & \text{if } I_s(t) > I_{th} \end{cases} ; \; I_{th} = (v_{th} - v_0)\frac{C_m}{\tau_m} \tag{4}$$

$T_{ref}$ is the refractory period, which is the time it takes a neuron to start accepting input currents after a spike event. $T_{spike}$ is the time it takes a neuron to reach $v_{th}$ from $v_{reset}$ after a spike event at some $t = t'$ given $v_{th} < I_s(t) = c \in \mathbb{R}$, $t' < t \leq t' + T_{spike}$. Equations 3, 4 describe the response curve of an LIF neuron, which has a discontinuous and unbounded derivative ($\partial\rho/\partial I_s$) at $I_s = (v_{th} - v_0)C_m/\tau_m$. However, one can modify (Equation 4) as described by Hunsberger and Eliasmith (2015) in order to obtain a smooth rate-based LIF approximation.

$$\begin{aligned} \rho'\left[I_s(t)\right] &= \left\{T_{ref} + \tau_m \log\left(1 + \frac{v_{th}}{\Theta\left[I_s(t) - v_{th}\right]}\right)\right\}^{-1}; \\ \Theta(x) &= \gamma \log\left(1 + e^{x/\gamma}\right) \end{aligned} \tag{5}$$

where $\gamma$ is the smoothing factor of choice.

### 3.3.2 ANN-to-SNN conversion

Due to the discontinuities introduced by discrete spike events, the conventional gradient-descent training techniques need to be modified for spiking neural networks. Various approximation methods have been developed to overcome these discontinuities (Neftci et al., 2019). One such method is based on the utilization of the rate-based approximations, a.k.a. the tuning curves. Given a loss function, the main idea is to build a network of differentiable rate-based approximation units and solve for the synaptic weights by using an arbitrary version of gradient descent. Once the solution is obtained, the approximation units can be substituted with LIF neurons to use the resulting spiking network during inference as shown in Figure 3. We utilized NengoDL Rasmussen (2018) to implement the aforementioned ANN to SNN conversion methodology. We employed the standard sparse categorical cross entropy as our loss function.

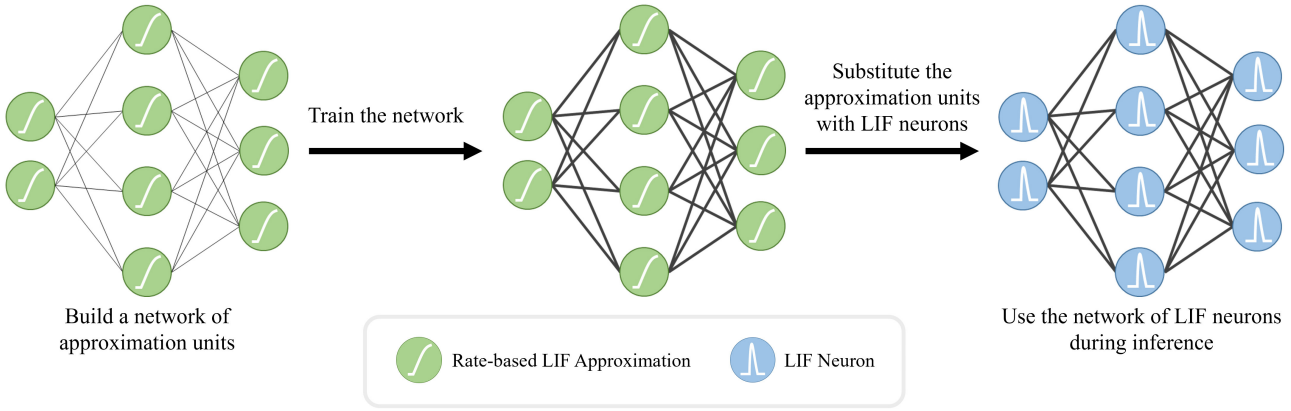


FIGURE 3
ANN-to-SNN conversion work-flow: we first employ rate-based approximations of the LIF neurons to train our network, since the discontinuous spike event outputs of the original LIF neurons prevents the training of the network through the back-propagation algorithm. After completing the training of this interim network, we substitute the LIF approximations with the original ones while keeping the network topology and the trained weights (bold black lines) the same.

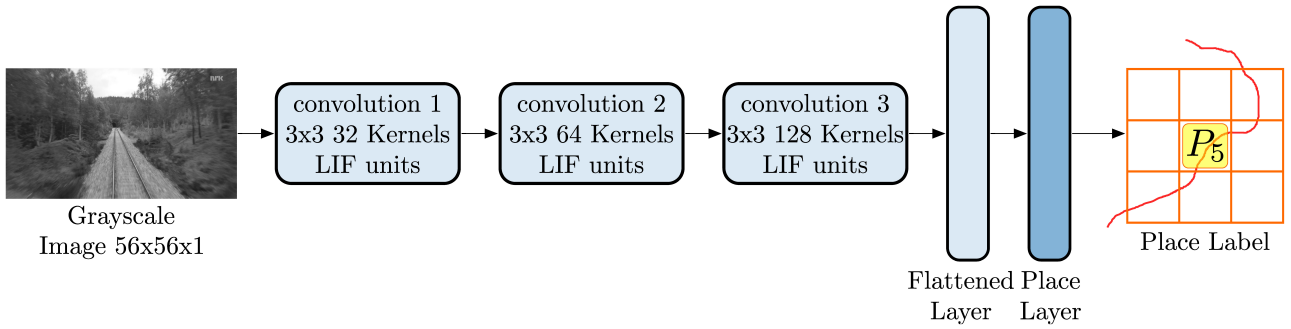


FIGURE 4
The LoCS-Net architecture consists of 3 convolutional layers followed by a fully connected output layer, known as the *place layer*. The units within this layer correspond to unique locations within the environment. LoCS-Net accepts 56 × 56 pixel grayscale images as inputs, using them to predict the associated places from which the input images were sampled.

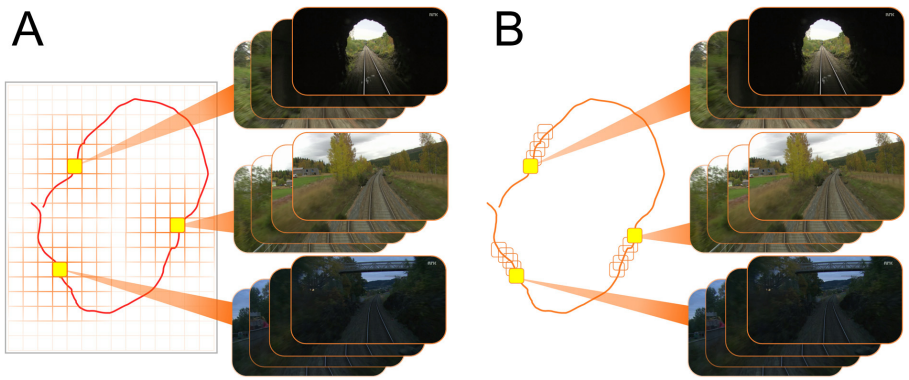


FIGURE 5
Annotating images: **(A)** top-down navigation domain is discretized by defining a grid of arbitrary resolution. Each tile of of the grid annotates the images sampled within its boundaries. **(B)** Images are sampled over a traverse at a pre-determined frequency while each image is corresponding to a unique place. For instance, if the first image is sampled at time $t = 0\,s$, then the second and the third image will be sampled at $t = T\,s$ and $t = 2T\,s$, respectively.

TABLE 1 LoCS-Net training and test data specifications.

| Specifications \ dataset | Nordland | Oxford RobotCar (ORC) |
|---|---|---|
| Train size [# of images] | 6,144 | 32,475 |
| Test size [# of images] | 3,072 | 17,055 |
| # of labels | 3,072 | 2,500 |
| # of unique labels | 3,072 | 185 |

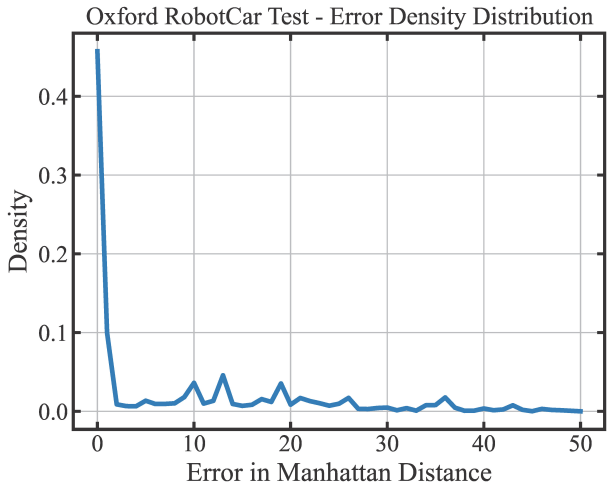


FIGURE 8
Prediction error distribution of LoCS-Net over the ORC dataset: 55.7% of the place predictions of LoCS-Net are within 1-Manhattan Distance of the ground truth labels. Approximately 40% of the LoCS-Net place predictions fall beyond 5-Manhattan Distance of the ground truth labels.

### 3.3.3 LoCS-Net architecture

As depicted in Figure 4, LoCS-Net is composed of a sequence of 3 convolutional layers followed by a fully connected output layer, also known as the "place layer," the units of which correspond to distinct places in the environment. Inputs to LoCS-Net are grayscale images of 56 × 56 pixels. The number of neurons in the place layer is set to be the number of possible places ($N_P$) as explained in Section 3.2. We considered 50 × 50 grid for Oxford RobotCar (ORC) data in our principal experiments. Note that for training, we employ the smooth rate-based approximated LIF units while maintaining the same architecture illustrated in Figure 4. We use sparse categorical cross entropy as the loss function during training. For inference, we replace the approximated LIF units of the trained network with spiking LIF neurons, keeping both the weights and the architecture unchanged. For further details of the network structure and the corresponding hyper-parameters, we refer the readers to the supplementary information and to the repository of the current work's code at https://figshare.com/s/c159a8680a261ced28b2.

## 4 Experiments

### 4.1 Datasets and evaluation metrics

We evaluate our proposed approach on the challenging Nordland (Olid et al., 2018) and ORC data (Maddern et al., 2017, 2020) following prior work (Hussaini et al., 2023). For the Nordland data experiments, we trained LoCS-Net using the spring and fall traverses and tested it with the summer traverse. For the ORC data experiments, we trained LoCS-Net on the sun (2015-08-12-15-04-18) and rain (2015-10-29-12-18-17) traverses, and tested its performance on the dusk (2014-11-21-16-07-03) traverse. We followed the Nordland data processing directions in Hussaini et al. (2023) for the same training and test data. We obtained 3,072 Nordland data (Olid et al., 2018) places, and 2,500 ORC data (Maddern et al., 2017, 2020) places (set by our grid definition) while considering the complete sun, rain, and dusk traverses used in Hussaini et al. (2023).

Although our discretization of the ORC domain yields a total of 2,500 possible places, the trajectories traversed in that dataset cover a much smaller number of labels. Some of the ORC data places are either occasionally visited or not visited at all. This is because the trajectories were generated by a vehicle traversing the road network, making it impossible to visit all parts of the spatial domain. Therefore, we filter out places that do not contain a minimum number (10) of unique training images. We also bound the number of unique instances per place from above (maximum 700) as the training of the baseline SNN models are getting infeasible due to increasing size of the data. Table 1 provides the training and the test data specifications yielded by our data pre-processing pipeline. We would like to note that LoCS-Net can still be trained and be tested on the full ORC data in a matter of minutes.

We employ standard VPR performance metrics, including the precision-recall curves, area-under-the-precision-recall curves (AUC-PR or AUC) (Cieslewski and Scaramuzza, 2017; Camara and Přeučil, 2019), and recall-at-N (R@N) curves (Perronnin et al., 2010; Uy and Lee, 2018) in order to assess the performance of our model.

### 4.2 Experimental set-up

We adopt two annotation methods as the Nordland (Olid et al., 2018) and the ORC data (Maddern et al., 2017, 2020) were structured in different ways. Figure 5A describes the labeling process of the ORC images (Maddern et al., 2017, 2020). As it is shown, we first encapsulated the top-down projection of the path within a rectangular region. Then, we discretize this region to obtain grid tiles, each of which represents a distinct place. These tiles annotate the images sampled within its boundaries.

To label the Nordland images (Olid et al., 2018) we followed the annotation method defined in Hussaini et al. (2023). As depicted in Figure 5B, we sample images over a traverse at a pre-determined frequency (every 8th image) while each image is corresponding to a unique place.

TABLE 2 VPR performance comparison in terms Precision at 100% Recall (P@100%R), area-under-the-precision-recall curves (AUC), mean inference time (MIT), mean training time (MIT), and effective energy consumed per inference.

| Method | Approach | Nordland | | | | | ORC | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P@100%R | AUC | MIT [ms] | MTT [min] | Effective energy per inference [J] | P@100%R | AUC | MIT [ms] | MTT [min] | Effective energy per inference [J] |
| LoCS-Net on GPU (ours) | SNN | **78.6%** | **0.980** | 25 | **1** | 2.545 | **45.7%** | **0.702** | 10 | **3.5** | 1.095 |
| LoCS-Net on NUC (ours) | SNN | **78.6%** | **0.980** | 796 | - | 13.183 | **45.7%** | **0.702** | 371 | - | 5.871 |
| LoCS-Net on Loihi (ours) | SNN | 71.1% | 0.761 | 288 | - | **0.060** | 41.0% | 0.653 | 147 | - | **0.032** |
| VPRTempo on GPU (Hines et al., 2024) | SNN | 73.0% | 0.975 | **8** | 15 | 0.079 | 20.2% | 0.435 | **5** | 54 | 0.053 |
| Ensemble SNNs on CPU (Hussaini et al., 2023) | SNN | 66.9% | 0.975 | 408 | 725 | 3.405 | 17.6% | 0.485 | 290 | 3,408 | 3.051 |
| WNA (Hussaini et al., 2022) | SNN | 0.3% | 0.005 | - | - | - | 4.0% | 0.042 | - | - | - |
| MixVPR (Ali-Bey et al., 2023) | ANN | **94.6%** | - | 29 | 3 | 0.907 | **87.7%** | - | 14 | **6** | 0.578 |
| Conv-AP (Ali-bey et al., 2022) | ANN | 91.3% | - | 27 | **2** | 0.847 | 84.6 | - | 18 | 8 | 0.632 |
| EigenPlaces (Berton et al., 2023) | ANN | 80.2% | - | 57 | 5 | 6.443 | 71.5% | - | 31 | 17 | 3.335 |
| CosPlace (Berton et al., 2022) | ANN | 75.3% | - | 60 | 6 | 6.842 | 71.0% | - | 35 | 17 | 3.524 |
| AP-GeM (Revaud et al., 2019) | ANN | 65.1% | - | 95 | 9 | 10.512 | 60.7% | - | 54 | 27 | 5.376 |
| NetVLAD (Arandjelovic et al., 2016) | ANN | 51.4% | - | 107 | 10 | 12.641 | 43.8% | - | 62 | 29 | 5.496 |

Bold values indicate the best performance metrics for SNN- and ANN-based approaches on individual datasets.





10.3389/fnbot.2024.1490267

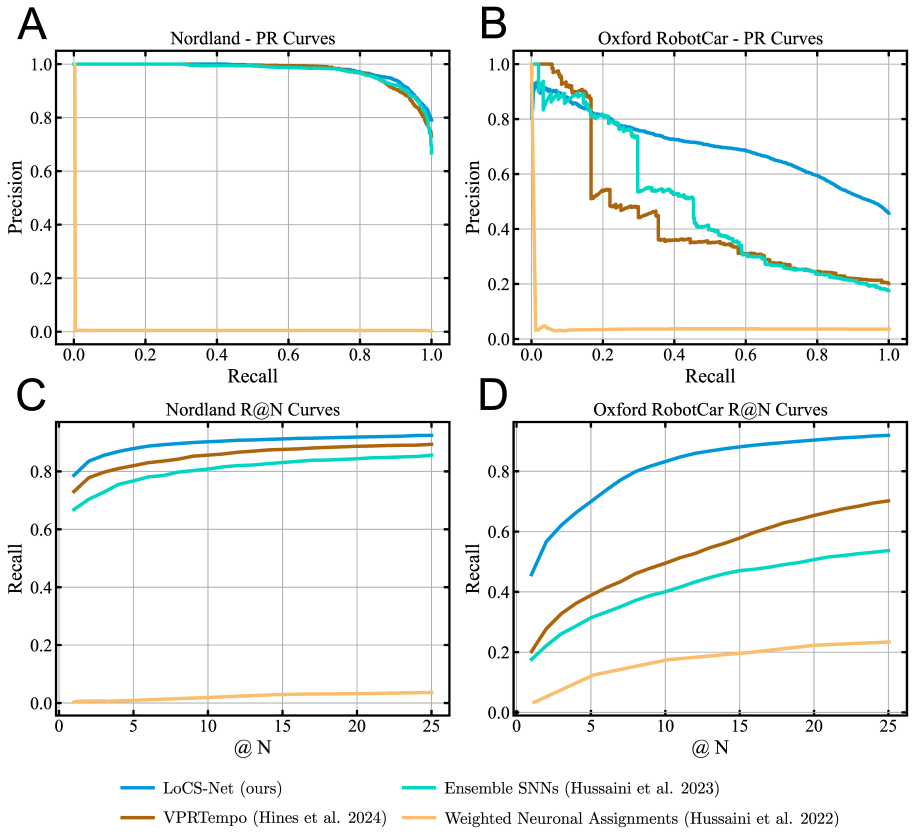


FIGURE 6
Precision-Recall and Recall @ N curves for the baseline SNN-based VPR methods and LoCS-Net: The blue, brown, cyan, and orange curves correspond to LoCS-Net, VPRTempo (Hines et al., 2024), Ensemble SNNs (Hussaini et al., 2023), and Weighted Neuronal Assignments (Hussaini et al., 2022), respectively. These figures demonstrate that LoCS-Net yields the best SNN-based VPR performance on both datasets. **(A)** PR curves obtained from the experiments on the Nordland dataset. **(B)** PR curves obtained from the experiments on the ORC datasets. **(C)** The R@N curves obtained from the experiments on the Nordland dataset. **(D)** The R@N curves obtained from the experiments on the ORC dataset.

## 4.3 Quantitative results

We conducted several performance comparisons of LoCS-Net with the current SOTA SNN methods, Ensemble SNNs (Hussaini et al., 2023), VPRTempo (Hines et al., 2024), and Weighted Assignment SNN (WNA) (Hussaini et al., 2022). In order to save computational resources, we did not train and test WNA ourselves. Instead, in Table 2, we listed the performance metrics published in Table 1 of Hussaini et al. (2023). We also included additional performance comparisons of LoCS-Net to a set of ANN-based SOTA VPR techniques such as AP-GeM (Revaud et al., 2019), NetVLAD (Arandjelovic et al., 2016), MixVPR (Ali-Bey et al., 2023), Conv-AP (Ali-bey et al., 2022), EigenPlaces (Berton et al., 2023), and CosPlace (Berton et al., 2022). We utilized the benchmark tool developed by Berton et al. (2023) in order to perform these additional comparisons.

Table 2 and Figure 6 summarize the VPR performance of LoCS-Net along with the reference methods. We observe that LoCS-Net outperformed all the SNN-based methods on both the Nordland (Olid et al., 2018) and ORC dataset (Maddern et al., 2017, 2020) by a large margin (78.6% and 45.7% respectively) in terms of P@100%R. LoCS-Net took much less time to train as reported in Table 2, which highlights LoCS-Net's compatibility for rapid prototyping and real-world deployment. Although it falls short of top-performing ANNs such as MixVPR and Conv-AP, LoCS-Net's strengths lie in energy efficiency and training time. While its GPU-based energy usage (2.545J) sits between that of ANNs like EigenPlaces (1.283J) and AP-GeM (5.376J), deploying LoCS-Net on neuromorphic hardware (Loihi) drastically reduces energy consumption, reaching just 0.032J per inference.

Moreover, Figures 6C, D present the Recall @ N curves obtained from the evaluations of the methods on the Nordland (Olid et al., 2018) and ORC datasets (Maddern et al., 2017, 2020). LoCS-Net consistently yields the best Recall @ N performance compared to SNN methods on both datasets. These results indicate good scalability of the LoCS-Net model across thousands of locations, while maintaining computationally efficient inference, as illustrated by Table 2.

We conduct a sensitivity analysis of LoCS-Net with respect to the number of neurons used for signal representation, the maximum firing rate, and the synaptic smoothing factor. We note that the nominal LoCS-Net does not include synaptic filters in order to avoid the additional complexity imposed by temporal dynamics during training, as capturing precise synaptic dynamics is not our primary objective. Figure 7 presents the P@100%R sensitivity analysis of LoCS-Net on the Nordland Figure 7A and

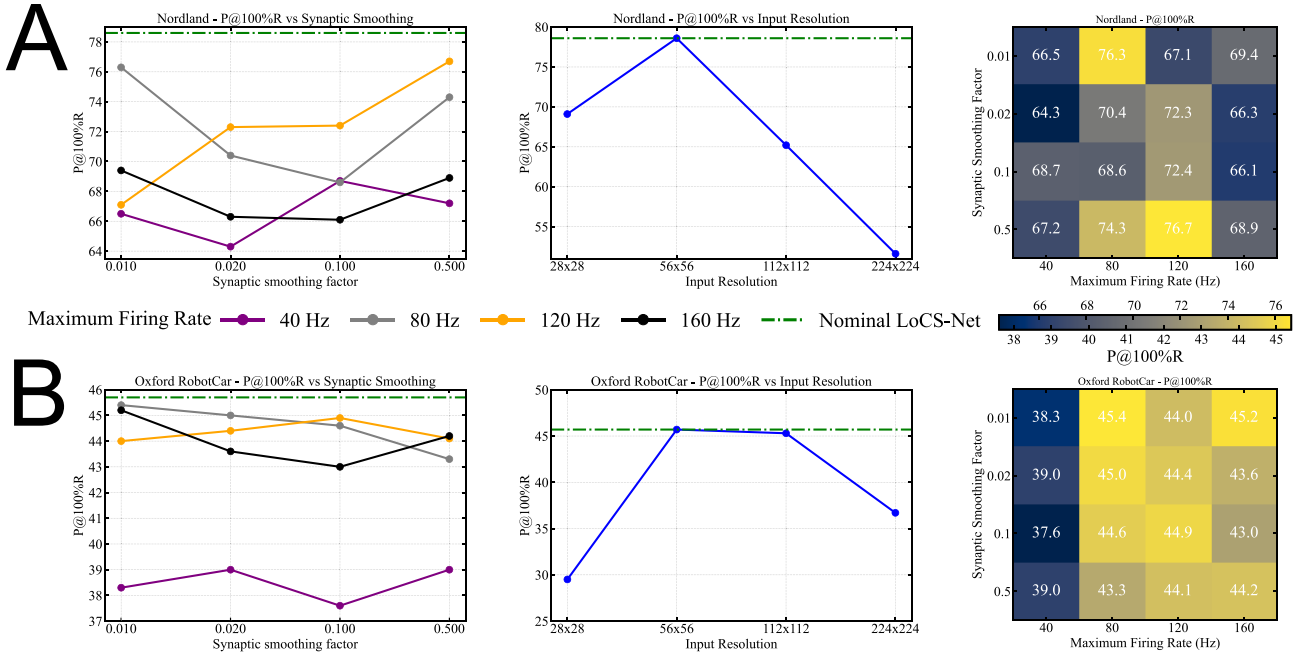


FIGURE 7
VPR performance in P@100%R sensitivity of LoCS-Net on Nordland **(A)** and ORC **(B)** datasets with respect to synaptic smoothing factor, maximum firing rate, and input layer resolution: The remaining neuronal and training parameters of the nominal LoCS-Net model (see Supplementary Tables S1, S2) are kept the same. Curves in the synaptic smoothing sensitivity plots (on the left) are color-coded, indicating model instances with different maximum firing rates with a nominal input resolution of 56 × 56. Plots in the middle depict the sensitivity with respect to only input resolution. Dashed green curves represent the performance of the nominal LoCS-Net model instance. Synaptic smoothing factor-maximum firing rate tables (on the right) illustrate the same data presented in the plots on the left. These tables contain the P@100%R values in white text while exhibiting the corresponding color codes at the same time. The P@100%R values are obtained by averaging the simulation results of five models, each initiated with a different seed for synaptic weights. The results suggest that LoCS-Net is sensitive to the synaptic smoothing factor, maximum firing rate, and input layer resolution, especially on Nordland data.

ORC Figure 7B datasets, focusing on the synaptic smoothing factor, maximum firing rate, and input layer resolution. All other neuronal and training parameters of the nominal LoCS-Net model remain unchanged. In Figures 7A, B, the synaptic smoothing sensitivity plots on the left use color-coded curves to represent different maximum firing rates for a nominal input resolution of 56 × 56. The middle plots isolate the effect of input resolution on model sensitivity, with dashed green curves showing the performance of the nominal LoCS-Net configuration. On the right, tables summarize the combined influence of the variances in synaptic smoothing factor and maximum firing rate, providing the same information as the left-hand plots. The P@100%R values of Figure 7 are computed as averages across simulations of five models initialized with different seeds per parameter set. The findings highlight that LoCS-Net is sensitive to variations in synaptic smoothing factor, maximum firing rate, and input layer resolution, with sensitivity being particularly evident on the Nordland dataset.

We further seek to understand the distribution of LoCS-Net's prediction errors on the ORC dataset. We quantify the prediction error in in terms of Manhattan Distance, as illustrated in Figure 8. 55.7% of the place predictions are within 1-Manhattan Distance of the ground truth labels. Yet, approximately 40% of the LoCS-Net place predictions fall beyond 5-Manhattan Distance of the ground truth labels. We did not perform the same analysis for Nordland data as it doesn't utilize a grid-based labeling structure as the Oxford RobotCar data.

## 4.4 Neuromorphic hardware deployment

We deployed the trained LoCS-Net on Kapoho Bay, a USB form that hosts 2 of Intel's neuromorphic Loihi chips (Davies et al., 2021). We utilized NengoLoihi (DeWolf et al., 2020) to deploy LoCS-Net on the Loihi chips. The hardware supports up to 260M trainable synaptic connections with 260k neurons; however, the network structure must be sufficiently tuned to fully utilize the hardware due to its architecture.

After the network parameters are trained, neurons and connections must be distributed between two chips, with 128 neuromorphic cores in each. Each core is designated to handle 1,024 neurons at a time, and the number of core-to-core connections is restricted to about 4,000 synapses due to the limited synapse memory. Therefore, networks with large input/output connections must be partitioned across several cores, and the biases are removed from the convolutional layers to reduce inter-core communication. These strategies have be followed to avoid under-utilization of the cores. As a result, our hardware-deployed network architecture contains fewer trainable parameters with sparse connections due to the above constraints, which may result in a slight decrease in performance. Additionally, as the Kapoho Bay is optimized for mobile deployment and energy efficiency, the device handles spike-timing with 8-bit accuracy, which defines its quantization limit. Training the simulated network without accounting for these hardware specifications might lead

to performance drop during on-chip inference. To minimize such discrepancies, the regularization parameter of the training is tuned to adjust the magnitude of the network weights. Here, we note that the hardware limitations mentioned above may be resolved in future versions of neuromorphic chips.

To examine the energy-saving benefits of neuromorphic hardware, we measured the average energy consumption per inference of LoCS-Net when deployed on Loihi, Intel NUC7i7BNH (a small-form-factor PC suitable for mobile robotics), and GPU (NVIDIA RTX 3060). We utilized pyJoules (Belgaid et al., 2019) to measure the average energy consumption on the GPU and NUC, while employing a standard off-the-shelf USB tester to observe the power drawn by Kapoho Bay. For each type of hardware hosting LoCS-Net, we first measured the idle power and then the power drawn under load while LoCS-Net was operational. We then subtracted the idle power from the load (or total) power to obtain the closest estimate of LoCS-Net's effective energy consumption, which we list in Table 2. Moreover, we report the total inference energy values in Figure 9.

## 5 Discussion

We observe a noticeable performance drop of the on-chip LoCS-Net, while achieving at least an order of magnitude improvement in energy efficiency on both datasets compared to CPU and GPU deployments. We believe that the gradient mismatch between the LIF neurons (Burkitt, 2006) and their rate approximations (Hunsberger and Eliasmith, 2015) significantly contribute to the reduced performance of LoCS-Net in this case, as also mentioned by Che et al. (2022). LoCS-Net outperforms all SNN-based methods on both datasets in terms of area-under-the-precision-recall curves, while demonstrating the second fastest inference as shown in Table 2. VPRTempo turns out to be the fastest (in terms of inference time) and the most energy-efficient simulated SNN, coming close after on-chip LoCS-Net in terms of energy consumption per inference. However, it fails to exhibit robustness against dynamic scenes, noise, variance in viewpoint, and lighting conditions in the ORC dataset.

We observe relatively poor performance of our method on the ORC dataset (Maddern et al., 2017, 2020). In addition to ANN-to-SNN conversion losses, we hypothesize that the more dynamic scene content of the ORC images (Maddern et al., 2017, 2020) and the substantial noise levels in a significant portion of the test ORC images impede better VPR performance of LoCS-Net.

As reported in Table 2, LoCS-Net consumes 0.06 J per inference when processing the Nordland images, approximately 1/40th of the energy consumed by the GPU and about 1/220th of the energy consumed by the CPU of the NUC. Similarly, Loihi chips demonstrate the greatest energy efficiency (0.032 J/inference vs. 5.871 J/inference on NUC and 1.095 J/inference on GPU) by a large margin when processing the ORC images. We consistently observe the total energy consumption of neuromorphic chips does not scale intuitively with respect to the size of the neural network in terms of the number of trainable parameters. Instead, the inference energy cost appears to be more related to the communication time between the integrated CPU and the Loihi chip. This includes the time required to generate and to send the spike signals through the input layer of LoCS-Net using the integrated CPU within the Loihi device, and to decode the output signal back to numerical data. This observation suggests that a significant restriction of energy efficient neuromorphic computation involves data conversion during encoding and decoding, which must be managed by traditional CPU architecture. The communication bottleneck also affects the total inference time. Due to the communication delay, there is a challenge in optimally including the spike-conversion stage in-between the sensing and input neurons, as well as between the output layer and the actuator. Unfortunately, this spike-conversion step scales linearly with the data size and the resolution we aim to represent. However, testing the proposed method on alternative neuromorphic hardware designs (Hazan and Ezra Tsur, 2022; Halaly and Ezra Tsur, 2023) might offer even greater power efficiency and yield faster inference times.

The overall performance of the SOTA ANN methods proved superior to that of their SNN competitors in terms of precision at 100% recall on both the Nordland and ORC datasets. As shown in Table 2, while these SOTA ANN techniques outperform their SNN competitors, they also require significantly longer time to generate descriptors (loosely corresponding to training time) and to compute reference-query matches (corresponding to inference time) compared to LoCS-Net. On-chip LoCS-Net remains the most energy-efficient VPR method, with the fastest training time by a large margin.

We must also note that the SOTA ANN approaches included in our comparison studies use pre-trained networks (e.g., ResNet50 and ResNet101 backbones), which are subsequently fine-tuned for the VPR tasks. In contrast, LoCS-Net is trained solely on data from the navigation domain of interest. In this sense, comparing our network to these SOTA ANN techniques may not be entirely fair, as they benefit from cumulative training over a much larger dataset. We would like to emphasize that our work focuses on the SNN domain, and LoCS-Net significantly advances the state of the art in SNN-based VPR techniques.

We may further analyze the performance of LoCS-Net by investigating the spiking activity in the convolutional layers of the model. Figures 10, 11 depict representative samples of Nordland and ORC training and test images. Compared to the ORC images, Nordland test and training instances are much more visually aligned. ORC test images, on the other hand, are extremely challenging due to intense variance in lighting, appearance, viewpoint, and noise. Some of these test instances are impossible to recognize by a human observer. We believe that these characteristics of the ORC data significantly contribute to LoCS-Net's reduced performance on this dataset.

We examined the activities generated by a set of randomly chosen images that were either correctly or incorrectly labeled by LoCS-Net. The images shown in this section are representative samples for both mislabeled and correctly labeled images from the Nordland and Oxford RobotCar datasets. We observed similar spiking activity patterns in all of the images we randomly picked as in those demonstrated in Figures 10, 11.

Figures 10, 11 depict the spiking unit activations of the first convolutional layer in the model, when presented images

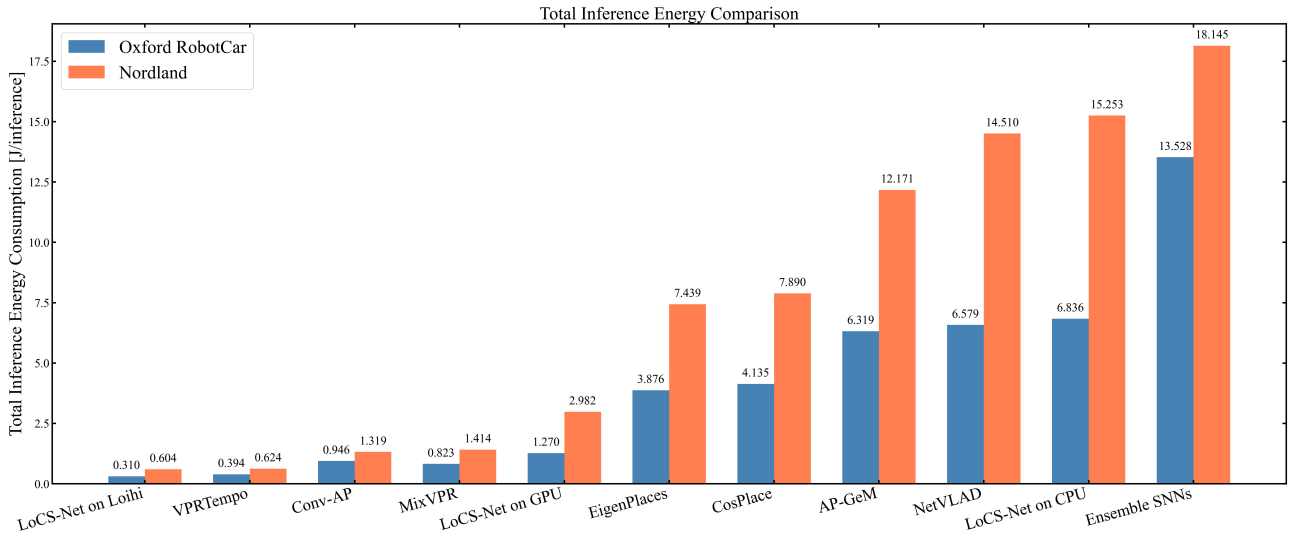


FIGURE 9
Total energy consumption (in Joules) per inference for different VPR methods: Orange bars represent energy consumption on Nordland data, while blue bars correspond to ORC data. The LoCS-Net model deployed on Loihi achieves the lowest energy consumption per inference, closely followed by its on-GPU SNN competitor, VPRTempo.

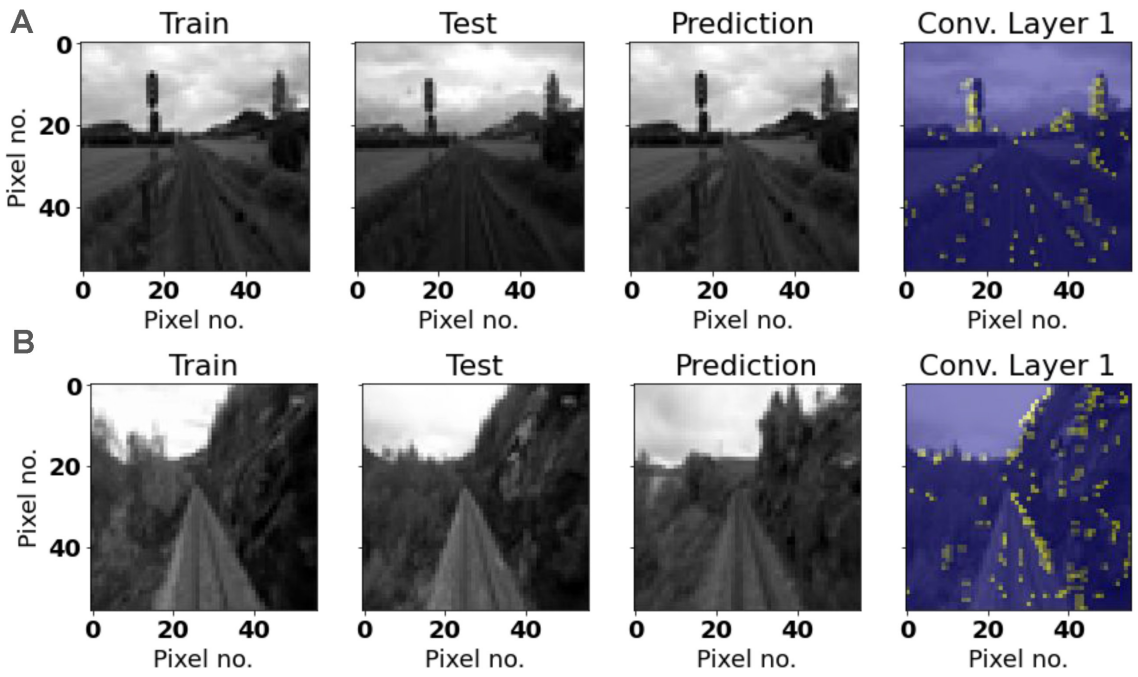


FIGURE 10
Representative samples of Nordland training and test images: Yellow regions represent the associated active neurons. **(A)** An instance of correct prediction. Representative convolutional layer activation appears to represent features in the input image. **(B)** An instance of incorrect prediction. Although the activation of the convolution layer appears to represent features of the image, the prediction made is for a similar, but incorrect class label.

from the Nordland and ORC datasets. In both figures, the correct predictions are associated with spiking activity that is clustered over large features in the input image that could potentially help to distinguish the input image from others. When we examine the spiking activities generated by the images that are mislabeled by LoCS-Net, we observe matching spiking patterns with the activities generated by the training image for the correct class. This implies that LoCS-Net struggles to distinguish images marginally different from each other.

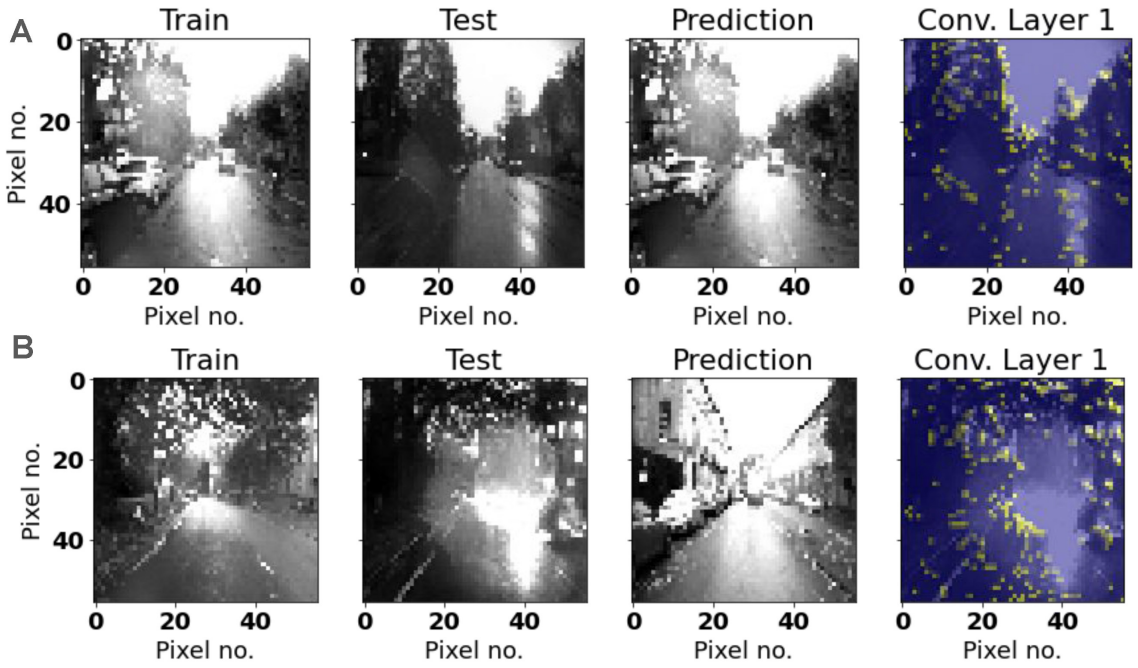


FIGURE 11
Representative samples of Oxford Robot Car training and test images: Yellow regions represent the associated active neurons. **(A)** An instance of correct prediction. Representative convolutional layer activation appears to represent features in the input image. **(B)** An instance of incorrect prediction. A possible water droplet on the lens causes significant image distortion that results in incorrect place prediction.

## 6 Conclusion

In this work, we formulate visual place recognition as a classification problem and develop LoCS-Net, a convolutional SNN to solve VPR tasks with challenging real-world datasets. Our approach leverages ANN-to-SNN conversion and back-propagation for tractable training, by using rate-based approximations of leaky integrate-and-fire (LIF) neurons. The proposed method substantially surpasses existing state-of-the-art SNNs on challenging datasets such as Nordland and ORC, achieving 78.6% precision at 100% recall on the Nordland dataset (compared to the current SOTA's 73.0%) and 45.7% on the Oxford RobotCar dataset (compared to the current SOTA's 20.2%). Our approach simplifies the training pipeline, delivering the fastest training time and the second fastest inference time among SOTA SNNs for VPR. Hardware-in-the-loop evaluations using Intel's neuromorphic USB device, Kapoho Bay, demonstrate that our on-chip spiking models for VPR-trained through the ANN-to-SNN conversion strategy-continue to outperform their SNN counterparts, despite a slight performance drop when transitioning from off-chip to on-chip, while still offering significant energy savings. These results emphasize the LoCS-Net's exceptional rapid prototyping and deployment capabilities, marking a significant advance toward more widespread adoption of SNN-based solutions in real-world robotics.

The SOTA ANN methods over-shadowed their SNN counterparts in terms of precision at 100% recall on both the Nordland and ORC datasets. However, as detailed in Table 2, these ANN techniques come with notable trade-offs, requiring longer times for descriptor generation and inference relative to LoCS-Net. Among the evaluated methods, the on-chip implementation of LoCS-Net stands out as the most energy-efficient VPR solution, achieving the shortest training time by a considerable margin.

We would like to emphasize that this manuscript proposes LoCS-Net as a environment-specific VPR solution. In that sense, providing a long-term general spatial memory as a global VPR solution is beyond the capabilities of the current work. In addition, LoCS-Net's performance is sensitive to the definition of places, which in turn may require the implementation of domain-specific discretization techniques to maximize the performance of LoCS-Net over different navigation environments. As discussed in Section 4.3, LoCS-Net doesn't perform as well over the Oxford RobotCar (Maddern et al., 2017, 2020) data as compared with the Nordland (Olid et al., 2018) dataset. This might be due to the LIF neuron approximation errors as well as the significantly varying lighting and road conditions of the ORC traverses (Maddern et al., 2017, 2020), which suggests the lack of robustness to such dynamic scenes. Nevertheless, we emprically show that LoCS-Net is much better than its SNN competitors at handling such challenges.

## Data availability statement

The computational model and data preparation scripts generated for this study can be found in the FigShare repository at https://figshare.com/s/c159a8680a261ced28b2. This includes all necessary code to reproduce the results presented in this paper.

Please direct further inquiries and questions about the model implementation to the corresponding authors.

## Author contributions

UA: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. IR: Data curation, Formal analysis, Software, Validation, Visualization, Writing – original draft, Writing – review & editing, Investigation. EG: Visualization, Writing – review & editing, Formal analysis, Validation, Data curation. AC: Formal analysis, Validation, Visualization, Writing – review & editing. SK: Data curation, Software, Validation, Writing – original draft, Writing – review & editing. MG: Resources, Supervision, Writing – review & editing. RG: Supervision, Writing – review & editing. IS: Resources, Supervision, Writing – review & editing. GC: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Office of Naval Research, United States [grant number N00014-19-1-2373].

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2024.1490267/full#supplementary-material

## References

Ali-bey, A., Chaib-draa, B., and Giguere, P. (2022). GSV-cities: toward appropriate supervised visual place recognition. *Neurocomputing* 513, 194–203. doi: 10.1016/j.neucom.2022.09.127

Ali-Bey, A., Chaib-Draa, B., and Giguere, P. (2023). "Mixvpr: feature mixing for visual place recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2998–3007. doi: 10.1109/WACV56688.2023.00301

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). "Netvlad: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5297–5307. doi: 10.1109/CVPR.2016.572

Arandjelović, R., and Zisserman, A. (2012). "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 2911–2918. doi: 10.1109/CVPR.2012.6248018

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). "SURF: speeded up robust features," in *Computer Vision – ECCV 2006. ECCV 2006* (Berlin, Heidelberg: Springer), 404–417. doi: 10.1007/11744023_32

Belgaid, M. C., Rouvoy, R., and Seinturier, L. (2019). *pyJoules: Python library that measures python code snippets* [computer software]. Available at: https://github.com/powerapi-ng/pyJoules (accessed January 6, 2025).

Berton, G., Masone, C., and Caputo, B. (2022). "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4878–4888. doi: 10.1109/CVPR52688.2022.00483

Berton, G., Trivigno, G., Caputo, B., and Masone, C. (2023). "Eigenplaces: training viewpoint robust models for visual place recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11080–11090. doi: 10.1109/ICCV51070.2023.01017

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biol. Cyber.* 95, 1–19. doi: 10.1007/s00422-006-0068-6

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., and Fua, P. (2011). Brief: computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 1281–1298. doi: 10.1109/TPAMI.2011.222

Camara, L. G., and Přeučil, L. (2019). "Spatio-semantic convnet-based visual place recognition," in *2019 European Conference on Mobile Robots (ECMR)* (IEEE), 1–8. doi: 10.1109/ECMR.2019.8870948

Cao, B., Araujo, A., and Sim, J. (2020). "Unifying deep local and global features for image search," in *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16* (Springer), 726–743. doi: 10.1007/978-3-030-58565-5_43

Che, K., Leng, L., Zhang, K., Zhang, J., Meng, Q., Cheng, J., et al. (2022). "Differentiable hierarchical and surrogate gradient search for spiking neural networks," in *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 24975–24990.

Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., et al. (2017). "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 3223–3230. doi: 10.1109/ICRA.2017.7989366

Cieslewski, T., and Scaramuzza, D. (2017). "Efficient decentralized visual place recognition from full-image descriptors," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)* (IEEE), 78–82. doi: 10.1109/MRS.2017.8250934

Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., et al. (2021). Advancing neuromorphic computing with loihi: a survey of results and outlook. *Proc. IEEE* 109, 911–934. doi: 10.1109/JPROC.2021.3067593

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). "Superpoint: self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 224–236. doi: 10.1109/CVPRW.2018.00060

DeWolf, T., Jaworski, P., and Eliasmith, C. (2020). Nengo and low-power ai hardware for robust, embedded neurorobotics. *Front. Neurorobot.* 14:568359. doi: 10.3389/fnbot.2020.568359

Doan, A.-D., Latif, Y., Chin, T.-J., Liu, Y., Do, T.-T., and Reid, I. (2019). "Scalable place recognition under appearance change for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9319–9328. doi: 10.1109/ICCV.2019.00941

Dube, R., Cramariuc, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J., et al. (2020). Segmap: segment-based mapping and localization using data-driven descriptors. *Int. J. Rob. Res.* 39, 339–355. doi: 10.1177/0278364919863090

Garg, S., Fischer, T., and Milford, M. (2021). "Where is your place, visual place recognition?" in *Proceedings of the Thirtieth International Joint Conference on Artificial*

*Intelligence (IJCAI-21)* (International Joint Conferences on Artificial Intelligence), 4416–4425. doi: 10.24963/ijcai.2021/603

Garg, S., Suenderhauf, N., and Milford, M. (2022). Semantic-geometric visual place recognition: a new perspective for reconciling opposing views. *Int. J. Rob. Res.* 41, 573–598. doi: 10.1177/0278364919839761

Gehrig, M., Shrestha, S. B., Mouritzen, D., and Scaramuzza, D. (2020). "Event-based angular velocity regression with spiking networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 4195–4202. doi: 10.1109/ICRA40945.2020.9197133

Halaly, R., and Ezra Tsur, E. (2023). Autonomous driving controllers with neuromorphic spiking neural networks. *Front. Neurorobot.* 17:1234962. doi: 10.3389/fnbot.2023.1234962

Hausler, S., Garg, S., Xu, M., Milford, M., and Fischer, T. (2021). "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*, 14141–14152. doi: 10.1109/CVPR46437.2021.01392

Hazan, A., and Ezra Tsur, E. (2022). Neuromorphic neural engineering framework-inspired online continuous learning with analog circuitry. *Appl. Sci.* 12:4528. doi: 10.3390/app12094528

He, K., Lu, Y., and Sclaroff, S. (2018). "Local descriptors optimized for average precision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 596–605. doi: 10.1109/CVPR.2018.00069

Hines, A. D., Stratton, P. G., Milford, M., and Fischer, T. (2024). "Vprtempo: a fast temporally encoded spiking neural network for visual place recognition," in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 10200–10207. doi: 10.1109/ICRA57147.2024.10610918

Hu, Y., Tang, H., and Pan, G. (2021). Spiking deep residual networks. *IEEE Trans. Neur. Netw. Learn. Syst.* 34, 5200–5205. doi: 10.1109/TNNLS.2021.3119238

Hunsberger, E., and Eliasmith, C. (2015). Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*.

Hussaini, S., Milford, M., and Fischer, T. (2022). Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robot. Autom. Lett.* 7, 4094–4101. doi: 10.1109/LRA.2022.3149030

Hussaini, S., Milford, M., and Fischer, T. (2023). "Ensembles of compact, region-specific regularized spiking neural networks for scalable place recognition," in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 4200–4207. doi: 10.1109/ICRA48891.2023.10160749

Johns, E., and Yang, G.-Z. (2011). "From images to scenes: compressing an image cluster into a single scene model for place recognition," in *2011 International Conference on Computer Vision* (IEEE), 874–881. doi: 10.1109/ICCV.2011.6126328

Kim, H. J., Dunn, E., and Frahm, J.-M. (2015). "Predicting good features for image geo-localization using per-bundle vlad," in *Proceedings of the IEEE International Conference on Computer Vision*, 1170–1178. doi: 10.1109/ICCV.2015.139

Kim, S., Park, S., Na, B., and Yoon, S. (2020). "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 11270–11277. doi: 10.1609/aaai.v34i07.6787

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Lajoie, P.-Y., and Beltrame, G. (2022). Self-supervised domain calibration and uncertainty estimation for place recognition. *IEEE Robot. Autom. Lett.* 8, 792–799. doi: 10.1109/LRA.2022.3232033

Lanham, M. (2018). *Learn ARCore-Fundamentals of Google ARCore: Learn to build augmented reality apps for Android, Unity, and the web with Google ARCore 1.0*. Birmingham: Packt Publishing Ltd.

Li, B., Munoz, J. P., Rong, X., Chen, Q., Xiao, J., Tian, Y., et al. (2018). Vision-based mobile indoor assistive navigation aid for blind people. *IEEE Trans. Mobile Comput.* 18, 702–714. doi: 10.1109/TMC.2018.2842751

Liu, Y., and Zhang, H. (2012). "Visual loop closure detection with a compact image descriptor," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), 1051–1056. doi: 10.1109/IROS.2012.6386145

Loncomilla, P., Ruiz-del Solar, J., and Martínez, L. (2016). Object recognition using local invariant features for robotic applications: a survey. *Pattern Recognit.* 60, 499–514. doi: 10.1016/j.patcog.2016.05.021

Lowe, D. G. (1999). Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (IEEE), 1150–1157. doi: 10.1109/ICCV.1999.790410

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., et al. (2015). Visual place recognition: a survey. *IEEE Trans. Robot.* 32, 1–19. doi: 10.1109/TRO.2015.2496823

Lynen, S., Sattler, T., Bosse, M., Hesch, J. A., Pollefeys, M., and Siegwart, R. (2015). "Get out of my lab: large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, 1. doi: 10.15607/RSS.2015.XI.037

Lynen, S., Zeisl, B., Aiger, D., Bosse, M., Hesch, J., Pollefeys, M., et al. (2020). Large-scale, real-time visual-inertial localization revisited. *Int. J. Rob. Res.* 39, 1061–1084. doi: 10.1177/0278364920931151

Maddern, W., Pascoe, G., Gadd, M., Barnes, D., Yeomans, B., and Newman, P. (2020). Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint arXiv:2002.10152*.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). "1 Year, 1000km: the Oxford RobotCar dataset. *Int. J. Robot. Res.* 36, 3–15. doi: 10.1177/0278364916679498

Masone, C., and Caputo, B. (2021). A survey on deep visual place recognition. *IEEE Access* 9, 19516–19547. doi: 10.1109/ACCESS.2021.3054937

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* 22, 761–767. doi: 10.1016/j.imavis.2004.02.006

McManus, C., Upcroft, B., and Newman, P. (2014). Scene signatures: localised and point-less features for localisation. *Robotics* 10, 1–9. doi: 10.15607/RSS.2014.X.023

Mikolajczyk, K., and Schmid, C. (2002). "An affine invariant interest point detector," in *Computer Vision-ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7* (Springer), 128–142. doi: 10.1007/3-540-47969-4_9

Naseer, T., Burgard, W., and Stachniss, C. (2018). Robust visual localization across seasons. *IEEE Trans. Robot.* 34, 289–302. doi: 10.1109/TRO.2017.2788045

Neftci, E. O., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Proc. Mag.* 36, 51–63. doi: 10.1109/MSP.2019.2931595

Olid, D., Fácil, J. M., and Civera, J. (2018). "Single-view place recognition under seasonal changes," in *PPNIV Workshop at IROS 2018*.

Oliva, A., and Torralba, A. (2006). Building the gist of a scene: the role of global image features in recognition. *Prog. Brain Res.* 155, 23–36. doi: 10.1016/S0079-6123(06)55002-2

Patel, D., Hazan, H., Saunders, D. J., Siegelmann, H. T., and Kozma, R. (2019). Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to atari breakout game. *Neural Netw.* 120, 108–115. doi: 10.1016/j.neunet.2019.08.009

Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010). "Large-scale image retrieval with compressed fisher vectors," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE), 3384–3391. doi: 10.1109/CVPR.2010.5540009

Radenović, F., Tolias, G., and Chum, O. (2018). Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 1655–1668. doi: 10.1109/TPAMI.2018.2846566

Rasmussen, D. (2018). NengoDL: Combining deep learning and neuromorphic modelling methods. *Neuroinformatics* 17, 611–628. doi: 10.1007/s12021-019-09424-z

Reinhardt, T. (2019). *Using Global Localization to Improve Navigation*. Available at: https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html (accessed May 15, 2023).

Revaud, J., Almazán, J., Rezende, R. S., and de Souza, C. R. (2019). "Learning with average precision: training image retrieval with a listwise loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5107–5116. doi: 10.1109/ICCV.2019.00521

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682

Schönberger, J. L., Pollefeys, M., Geiger, A., and Sattler, T. (2018). "Semantic visual localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6896–6906. doi: 10.1109/CVPR.2018.00721

Seo, P. H., Weyand, T., Sim, J., and Han, B. (2018). "Cplanet: enhancing image geolocalization by combinatorial partitioning of maps," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 536–551. doi: 10.1007/978-3-030-01249-6_33

Shan, M., Wang, F., Lin, F., Gao, Z., Tang, Y. Z., and Chen, B. M. (2015). "Google map aided visual navigation for uavs in gps-denied environment," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE), 114–119. doi: 10.1109/ROBIO.2015.7418753

Siméoni, O., Avrithis, Y., and Chum, O. (2019). "Local features and visual words emerge in activations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11651–11660. doi: 10.1109/CVPR.2019.01192

Stratton, P. G., Wabnitz, A., Essam, C., Cheung, A., and Hamilton, T. J. (2022). Making a spiking net work: robust brain-like unsupervised machine learning. *arXiv preprint arXiv:2208.01204*.

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). "On the performance of convnet features for place recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), 4297–4304. doi: 10.1109/IROS.2015.7353986

Torralba, A., Fergus, R., and Weiss, Y. (2008). "Small codes and large image databases for recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 1–8. doi: 10.1109/CVPR.2008.4587633

Tsintotas, K. A., Bampis, L., and Gasteratos, A. (2022). The revisiting problem in simultaneous localization and mapping: a survey on visual loop closure detection. *IEEE Trans. Intell. Transp. Syst.* 23, 19929–19953. doi: 10.1109/TITS.2022.3175656

Uy, M. A., and Lee, G. H. (2018). "Point-netvlad: deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4470–4479. doi: 10.1109/CVPR.2018.00470

Vitale, A., Renner, A., Nauer, C., Scaramuzza, D., and Sandamirskaya, Y. (2021). "Event-driven vision and control for uavs on a neuromorphic chip," in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 103–109. doi: 10.1109/ICRA48506.2021.9560881

Weyand, T., Kostrikov, I., and Philbin, J. (2016). "Planet-photo geolocation with convolutional neural networks," in *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14* (Springer), 37–55. doi: 10.1007/978-3-319-46484-8_3

Yin, P., Srivatsan, R. A., Chen, Y., Li, X., Zhang, H., Xu, L., et al. (2019). "Mrs-VPR: a multi-resolution sampling based global visual place recognition method," in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE), 7137–7142. doi: 10.1109/ICRA.2019.8793853

Yu, J., Zhu, C., Zhang, J., Huang, Q., and Tao, D. (2020). Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE Trans. Neur. Netw. Learn. Syst.* 31, 661–674. doi: 10.1109/TNNLS.2019.2908982

Zemene, E., Tesfaye, Y. T., Idrees, H., Prati, A., Pelillo, M., and Shah, M. (2018). Large-scale image geo-localization using dominant sets. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 148–161. doi: 10.1109/TPAMI.2017.2787132

Zhang, X., Wang, L., and Su, Y. (2021). Visual place recognition: a survey from deep learning perspective. *Pattern Recognit.* 113:107760. doi: 10.1016/j.patcog.2020.107760

Zhu, L., Mangan, M., and Webb, B. (2020). "Spatio-temporal memory for navigation in a mushroom body model," in *Biomimetic and Biohybrid Systems: 9th International Conference, Living Machines 2020, Freiburg, Germany, July 28–30, 2020, Proceedings 9* (Springer), 415–426. doi: 10.1007/978-3-030-64313-3_39