---

**Article:**
Ozdemir, A., Barron, A.B., Philippides, A. et al. (Submitted: 2021) EchoVPR : echo state networks for visual place recognition. arXiv. (Submitted)

---

# EchoVPR: Echo State Networks for Visual Place Recognition

Anıl Özdemir[1], Andrew B. Barron[2], Andrew Philippides[3],
Michael Mangan[1,*], Eleni Vasilaki[1,*], and Luca Manneschi[1,*]

***Abstract*— Recognising previously visited locations is an important, but unsolved, task in autonomous navigation. Current visual place recognition (VPR) benchmarks typically challenge models to recover the position of a query image (or images) from sequential datasets that include both spatial and temporal components. Recently, Echo State Network (ESN) varieties have proven particularly powerful at solving machine learning tasks that require spatio-temporal modelling. These networks are simple, yet powerful neural architectures that—exhibiting memory over multiple time-scales and non-linear high-dimensional representations—can discover temporal relations in the data while still maintaining linearity in the learning. In this paper, we present a series of ESNs and analyse their applicability to the VPR problem. We report that the addition of ESNs to pre-processed convolutional neural networks led to a dramatic boost in performance in comparison to non-recurrent networks in four standard benchmarks (GardensPoint, SPEDTest, ESSEX3IN1, Nordland) demonstrating that ESNs are able to capture the temporal structure inherent in VPR problems. Moreover, we show that ESNs can outperform class-leading VPR models which also exploit the sequential dynamics of the data. Finally, our results demonstrate that ESNs also improve generalisation abilities, robustness, and accuracy further supporting their suitability to VPR applications.**

## I. Introduction

Visual Place Recognition (VPR) challenges algorithms to recognise previously visited places despite changes in appearance caused by illuminance, viewpoint, and weather conditions [1] (see Fig. 2 for example images). Unlike in many machine learning domains, typical VPR benchmark require learning of position from images gathered during one route traversal, when compared with data during another route traversal, meaning that there are very few examples to learn from (typically only the images within a few metres of the correct location) making the task even more challenging. One approach is to recognise places based on matching single views using image processing methods to remove the variance between datasets. For instance, models have been developed that use different image descriptors to obtain meaningful image representations that are robust to visual change (e.g. AMOSNet [2], DenseVLAD [3], and NetVLAD [4]). While matching single images is successful in many benchmarks, it can suffer from the effects of aliasing, individual image corruption, or sampling mismatches between datasets (e.g. it is challenging to ensure that images sampled along the same route precisely overlap).

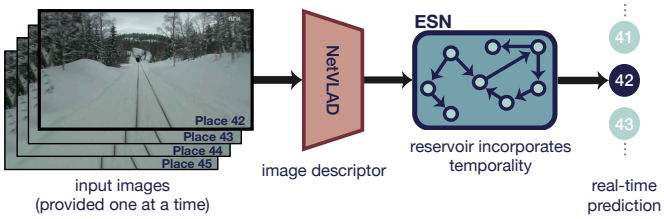


Fig. 1: **An illustration of EchoVPR framework.** Echo State Networks (ESN) incorporate temporality while still maintaining real-time prediction capability, which is a key feature for a robotic system in real-world applications. Given an input image at a time (from snowy Nordland [11] in this example), an image descriptor (class-leading NetVLAD [4]) provides a meaningful representation to the ESN to update the *fixed* reservoir.

One way to improve performance is to exploit the temporal relationships inherent in images sampled along routes (see models by [5], [6], [7], [8], [9], [10]). Milford and Wyeth [5] were the first to demonstrate improved VPR performance through matches sequences of images using a global search to overcome individual image mismatches. These methods often have an explicit encoding of speed to limit the image search space and/or store a stack of images to allow comparison of image sequences: both of which are undesirable for autonomous robots that may have limited memory and external sensing capabilities.

Echo State Networks (ESN) [15] are a class of recurrent neural networks, ideally suited to addressing VPR problems without the need for additional support cues or input data caching, see Fig. 1 ESNs are a subset of reservoir computing models in which the reservoir neurons possess fixed, random and recurrent interconnections that sustain recent memories, i.e. *echoes* [16] with the practical benefit that only the output layer weights require training. ESNs thus act as a temporal kernel [17] over a variety of time-scales, creating a form of working memory dispensing of the need for input caching. They are therefore well-suited to temporal problems such as VPR and have excelled when applied to problems that involve sequential data including dynamical system predictions [18], [19], robotic motion and navigation tasks [20], [21], [22].

In this paper, we will therefore apply ESNs to VPR to see

This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/P006094/1, EP/S030964/1, EP/S009647/1 and EP/V006339/1, and by Templeton World Charity Foundation Grant № 0539.

[1]Department of Computer Science, The University of Sheffield, UK m.mangan@sheffield.ac.uk
[2]Department of Biological Sciences, Macquarie University, Australia
[3]School of Engineering and Informatics, University of Sussex, UK
*Joint last authors

Fig. 2: **Example dataset images.** Reference (top) and query (bottom) images from four VPR benchmarking datasets, from left to right; **GardensPoint** [12] and **ESSEX3IN1** [13]: different viewpoint and illuminance conditions, **SPEDTest** [14] and **Nordland** [11]: fixed viewpoint but different season and weather conditions.

if these temporal networks can take advantage of the inherent structure of visual input, focusing in particular on two recent advances in ESNs. First, the application of neuron-specific learnable thresholds of reservoir activity results in an improved capacity and performance in comparison to traditional ESNs. Second, layering ESNs in a hierarchical framework facilitates learning of cues from different time-scales concurrently [23], [24], [25]. Such hierarchical ESNs invoking multiple and diverse time-scales to enrich the dynamics of the system have achieved class-leading performance in the permuted-sequential MNIST task [25]. The best operational regime of such systems occurs when the first reservoir of neurons (the ones closer to the input signal) have faster time-scales in comparison to the 'deeper' ones. In this way, the first reservoirs can quickly adapt to changes in the external signal (i.e input) while deeper ESNs can maintain longer memory and react more slowly. We hypothesise that these advances can help in addressing complex VPR problems on real-world image datasets which require a large memory capacity (often containing a lot of redundant information between subsequent images) and have long and short time dependencies.

For recent reviews of the state-of-the-art in visual place recognition, refer to [1], [26], [27], and for overviews of most prominent benchmarking datasets, model results, and recommended protocols, see [28], [29].

The remainder of the paper is organised as follows: Section II summarises the VPR problem formulation and presents four varieties of ESNs (standard and hierarchical, with/without SpaRCe) that will be evaluated. Section IV, compares the performance of these ESNs combined with a NetVLAD [4] image descriptor against state-of-the-art single-view matching models (AMOSNet [2] and DenseVLAD [3]) in three benchmark datasets (GardensPoint, SPEDTest, ESSEX3IN1). We then compare the (best) ESN approach to the current best sequence matching models (FlyNet+RNN & FlyNet+CANN [10]) in the highly challenging Nordland dataset. Section V places these results in the context of current methods and offers an outlook for future work as well as potential bio-inspired extensions.

## II. Methods

### A. Problem Formulation

VPR algorithms are provided with a sequence of places (in form of images) sampled along a route, then they are asked to correctly match (within an acceptable threshold) the places by the image key-frames along the same route at a different time, see Fig. 2.[1] The input data is composed of videos where the network has to correctly infer the location, i.e. the image key-frame that is processed at the considered time. In all the tasks there are at least two sequences of images, one used as a training set (i.e. reference) and the other used as a test set (i.e. query), acquired by visiting the same locations and following the same path twice. Even though there is a one-to-one mapping between training and test samples, the latter is acquired by visiting the locations at different times, leading to differences in visual appearances, such as seasonal or illuminance as well as viewpoint changes. Often times, perfect matching is not possible, hence, there can be a tolerance term that allow a close match to be accepted. A match is considered successful, if $||\text{reference} - \text{query}|| \leq$ tolerance.

In our specific implementation, we consider supervised learning with the ESNs as a predictor, hence, forming a classification problem. The number of read-out nodes is equal to the number of places, and therefore, specific to the given dataset. The read-out nodes (the final and the only learnable layer) output a probability distribution, $\mathcal{P}_{\text{query}}$, for each given query image. The prediction (i.e. key-frame of the query) is the number of the read-out node, i.e. $\arg\max \mathcal{P}_{\text{query}}$.

### B. Standard ESN

An ESN is a reservoir of recurrently connected nodes, whose temporal dynamics $\mathbf{x}(t)$ evolves following [15]:

$$\mathbf{x}(t+\delta t) = (1-\alpha)\mathbf{x}(t) + \alpha f\left(\mathbf{h}(t)\right), \tag{1}$$

$$\mathbf{h}(t) = \gamma \mathbf{W}_{\text{in}}\mathbf{s}(t) + \rho \mathbf{W}\mathbf{x}(t), \tag{2}$$

[1]The VPR challenge and recent models were summarised in VPR-Bench [29].

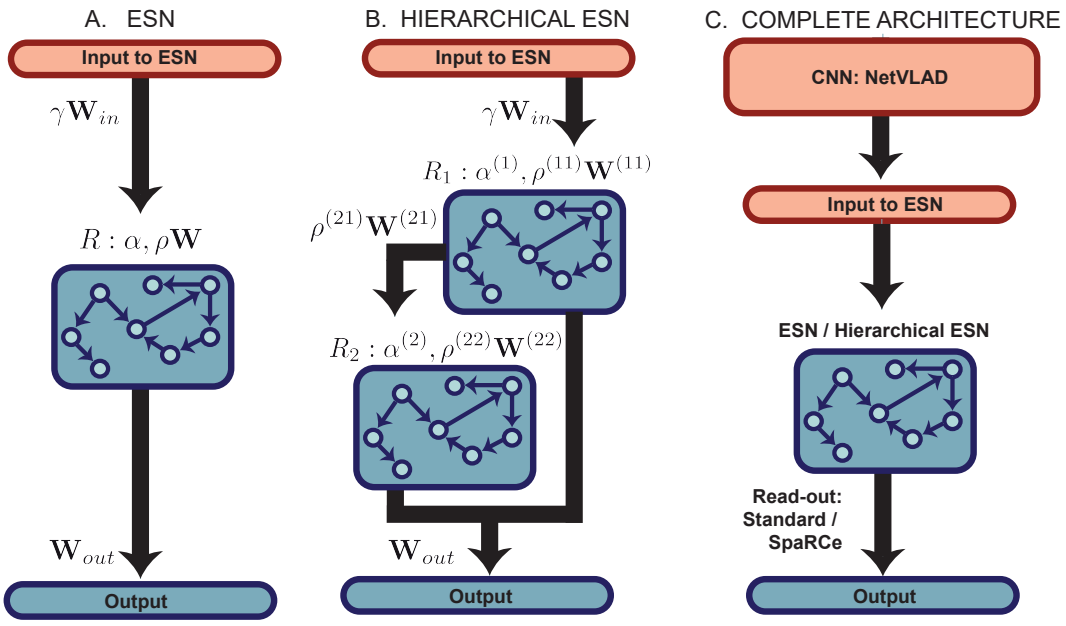


Fig. 3: **Scheme of the ESN models and the overall network architecture. A**: ESN protocol. The input is fed to an ESN and the training process occurs on the read-out $\mathbf{W}_{\text{out}}$ from the network representation. When the SpaRCe algorithm is adopted, additional thresholds $\theta$ are initialised and adapted through the gradient. **B**: Hierarchical ESN. The input is first processed by the first reservoir ($R_1$), which is then connected to a second ESN ($R_2$, tuned with different values of the hyper-parameters to exhibit diverse dynamical properties) unidirectionally. As in **A**, learning occurs on the output weights $\mathbf{W}_{\text{out}}$ defined from the representation of both reservoirs and on the thresholds $\theta$ when SpaRCe is adopted. **C**: Scheme of the overall model, composed of a pre-processing module (red boxes) and a reservoir model (blue boxes). In the pre-processing, an image is fed through a CNN (i.e. NetVLAD [4]), and through a hidden layer (the input to the ESN), pre-trained to reduce the dimensionality of NetVLAD output (4096 to 500) and to be fed into the reservoir system. The reservoir model can then be a single or hierarchical ESN with or without the SpaRCe model. Input images are perceived sequentially as a video, and the network has to correctly classify the location of the current image

where $\alpha$ is the leakage term and defines the rate of integration of information, $f$ is a non-linear activation function (usually $\tanh$), $\mathbf{s}(t)$ is the input signal, $\mathbf{W}_{\text{in}}$ is the input connectivity matrix, which is commonly drawn from a random Gaussian distribution, and $\gamma$ is a multiplicative factor of the external signal. The recurrent connectivity $\mathbf{W}$ is a sparse, random and fixed matrix whose eigenvalues are constrained inside the unit circle of the imaginary plane, with a hyper-parameter $\rho$ (usually in the range of $[0, 1]$) set to further control the spectral radius. As depicted in Fig. 3, learning occurs on the read-out weights $\mathbf{W}_{\text{out}}$ from a representation $\mathbf{x}$ of the ESN dynamic through minimisation of a cost function:

$$E = \left[\mathbf{y} - \mathbf{y}^{\text{target}}\right]^2, \tag{3}$$

$$\mathbf{y} = \mathbf{W}_{\text{out}}\mathbf{x}. \tag{4}$$

Optimisation of $\mathbf{W}_{\text{out}}$ can be accomplished through different techniques, as ridge regression or iterative gradient descent methods [30].

### C. Hierarchical ESNs and SpaRCe

Recent works have started to analyse the benefits of reservoir computing systems composed of multiple ESNs. In these composed architectures, ESNs are connected hierarchically and are tuned differently to exhibit diverse dynamical properties. For instance, the values of the leakage term $\alpha^{(k)}$, where $k$ is the reservoir number, can vary for different networks, allowing to regulate the time-scales at which diverse reservoirs operate. As a result, the overall system can be characterised by a wider range of time constants that has richer dynamics and improved memory abilities. Following the architecture in Fig. 3(b), the equations that describe a system of hierarchically connected reservoirs can be easily defined by generalising Eqs.(1-2),

$$\mathbf{x}^{(k)}(t + \delta t) = (1 - \alpha^{(k)})\mathbf{x}^{(k)} + \alpha^{(k)} f\left(\mathbf{h}^{(k)}(t)\right), \tag{5}$$

$$\mathbf{h}^{(k)}(t) = \sum_{l}^{\text{N}_{\text{ESN}}} \rho^{(kl)} \mathbf{W}^{(kl)} \mathbf{x}^{(l)}(t), \tag{6}$$

where parameters have similar definitions to the ones in Eq. (1). In the hierarchical structure of Fig. 3(b), $\mathbf{W}^{(kl)} \neq 0$ if $k = l$ or $k = l+1$. In detail, $\mathbf{W}^{(kk)}$ indicates the recurrent connectivity of reservoir $k$ and needs to have a spectral radius smaller than one, while $\mathbf{W}^{(kl)}$, where $k = l + 1$ is the connectivity among different reservoirs and can be drawn from any desirable distribution. In this work, we focus on a hierarchical structure of two ESNs with different values for the two leakage terms.

While the exploitation of multiple ESNs can enrich the dynamics of the system by discovering temporal dependencies over multiple time-scales, the definition of sparse representations through the SpaRCe model [31] can enhance the capacity of the reservoir to learn associations by introducing specialised neurons through the definition of learnable thresholds. Considering the representation $\mathbf{x}$ from which the read-out is defined, as in Eq. (1), SpaRCe consists of the following normalisation operation:

$$x'_i = \text{sign}\left(x_i\right) \text{ReLU}\left(\left|x_i\right| - \theta_i\right) \tag{7}$$

$$\theta_i = P_n\left(\left|\mathbf{x}_i\right|\right) + \bar{\theta}_i \tag{8}$$

where $i$ is the $i$-th dimension, sign is the sign function and ReLU is the rectified linear unit. Of course, the new read-out is defined from $x'_i$, that is after the transformation given in Eq. (7) and (8), which leaves unaltered the dynamics of the system and can be easily applied to any reservoir representation. The threshold $\theta_i$ is composed of two factors: $P_n\left(\left|\mathbf{x}_i\right|\right)$, i.e. the $n$-th percentile of $\mathbf{x}_i$, which stands for the distribution of activities of dimension $i$ after the presentation of a number of samples with sufficient statistics, and a learnable part $\bar{\theta}_i$, which is adapted through gradient descent and is initialised to arbitrarily small values at the beginning of training. The percentile $n$ can be considered as an additional *interpretable* hyper-parameter that controls the sparsity level of the network at the start of the training phase.[2]

### D. Benchmarks and Pre-processing

Convolutional neural networks (CNN) are the best performing architectures for processing images and discover high-level features from visual data. However, they are static and lack temporal dynamics. In contrast, recurrent connections can be fundamental for the considered tasks where the driving signals are a succession of images acquired during the exploration of an environment. Thus, after a pre-processing module composed of NetVLAD [4], a pre-trained CNN, we adopted a system composed by one or multiple ESNs. Considering that the reservoir computing paradigm is more effective when the reservoir expands the dimensionality of its corresponding input, we first decreased the dimensionality of NetVLAD output (original dimension is $4096$) by training a feedforward network composed of one hidden layer (with $500$ nodes) on the considered classification task. This new representation is then considered as the input to the reservoir computing system, see Fig. 3(c). The reservoir is then trained to distinguish the different locations, which are processed successively in the natural order of acquisition by the overall architecture. The four reservoir computing models we study are summarised below.[3]

- **Echo State Network (`ESN`)**, where learning happens on the output weights only. The critical hyper-parameters of the system for the cases studied, and will be tuned are $\alpha, \gamma, \eta$ (leakage term, input factor, learning rate).
- **Echo State Network with SpaRCe (`ESN+SpaRCe`)**, where thresholds are applied to the reservoir following Eq. (7) and learning occurs on $\bar{\theta}$ and $\mathbf{W}_{\text{out}}$. The hyper-parameters are the same as the standard `ESN` with the addition of the starting percentile $P_n$ of Eq. (8).
- **Hierarchical ESN (`H-ESN`)**, composed by two reservoir connected unidirectionally. The read-out is defined from both reservoirs and, as for the case of a single ESN, $\mathbf{W}_{\text{out}}$ is subject to training. In this case, the number of hyper-parameters is theoretically more than doubled in comparison to a single ESN and it is practically challenging to perform an exhaustive tuning procedure of all of them. We selected the value of $\gamma$ as the optimal one found for the single `ESN` and fixed $\alpha^{(1)} \approx 1$, focusing on the tuning of $\alpha^{(22)}, \rho^{(21)}, \eta$. The constraint $\alpha^{(1)} \approx 1$ is justified by considering that the second reservoir would lose information that lives on fast time-scales if $\alpha^{(1)} \ll 1$, leading to an overall system with slow reacting dynamics. On the contrary, if $\alpha^{(1)} \approx 1$ and $\alpha^{(2)} < \alpha^{(1)}$, the first reservoir can react to rapid changes of the input and the second can maintain past temporal information, leading to a system that is robust to signals with both short and long temporal dependencies.
- **Hierarchical ESN and SpaRCe (`H-ESN+SpaRCe`)**, which is the same as a hierarchical reservoir, but with the addition of `SpaRCe`.

The total number of reservoir nodes is $\text{N} = 1000$[4] and learning of $\mathbf{W}_{\text{out}}$ and $\bar{\theta}$ is accomplished trough mini-batches and by minimisation of softmax cross-entropy loss:

$$E = \sum_{j}^{\text{N}_{\text{batch}}} \sum_{i} y_{ij}^{\text{target}} \log\left(\frac{\exp\left(y_{ij}^{\text{target}}\right)}{\sum_i \exp(y_{ij})}\right), \tag{9}$$

where $\text{N}_{\text{batch}}$ is the minibatch size, $y$ the output of the neural network, $y^{\text{target}}$ the target output, and the indexes $i$ and $j$ correspond to the sample number and to the output node considered. The models are trained for up to $50$ epochs, i.e. each training image is passed $50$ times.

Specifically for the Nordland dataset, which is more challenging than the previous benchmarks, we used the sigmoid cross-entropy loss as the error function, which led to better performance:

$$E = \sum_{j}^{\text{N}_{\text{batch}}} \sum_{i} y_{ij}^{target} \log\left(\frac{1}{1+\exp(-y_{ij})}\right), \tag{10}$$

where the terms have similar meaning to the ones of Eq. (9). The models are trained for up to a total of $50000$ iterations.

## III. Experiments

### A. Datasets and Performance Metrics

We evaluate the performance of the models proposed on four standard benchmarks: GardensPoint [12], ESSEX3IN1 [13], SPEDTest [14], and Nordland [11], using

[2]For different methodologies to estimate the percentile operation, see [31].

[3]The source-code for ESN implementations can be found in `https://github.com/anilozdemir/EchoVPR`.

[4]It is $\text{N} = 2000$ for the hierarchical models and for the Nordland.

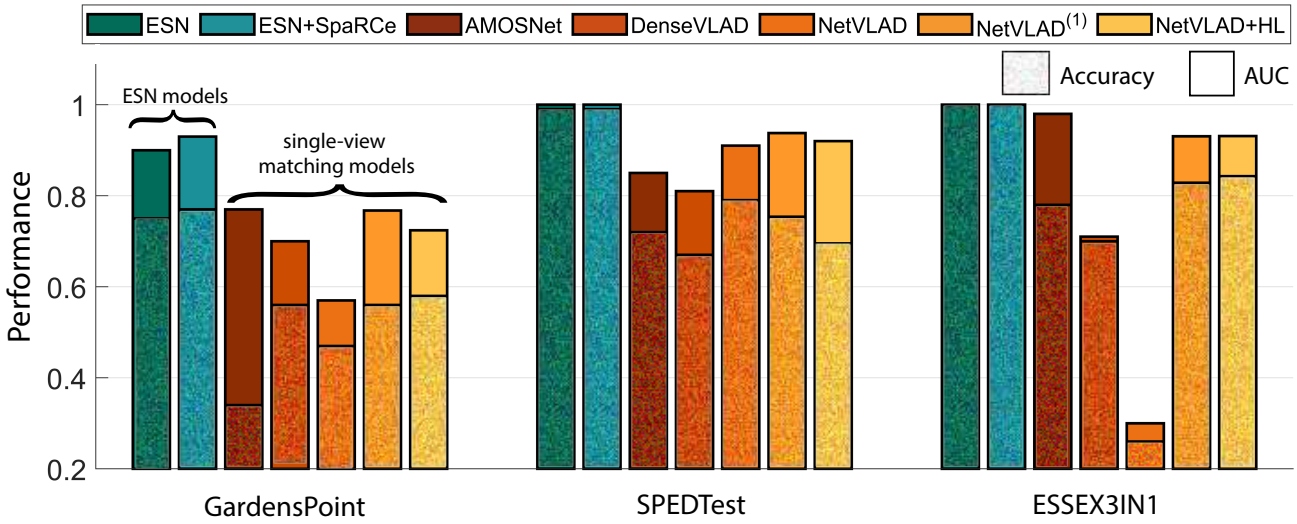


Fig. 4: **Comparison between different models.** The utilisation of reservoir computing models permits to capture of the temporal dynamics of the problem and improve the performance of CNNs. ESN and ESN with SpaRCe are shown in blue-green colours, while the performance of static neural networks is reported in red-yellow colours. The performance of AMOSNet, DenseVLAD and NetVLAD were taken from [29], where image matching was achieved by computing distances among the representation. $\mathrm{NetVLAD}^{(1)}$ and $\mathrm{NetVLAD} + \mathrm{HL}$ correspond to models in which a simple read-out or a hidden layer were trained from the representation of the convolutional network respectively. This was achieved through the minimisation of Eq. (9) on the specific task considered, similar to the approach used for ESNs. The bar plots for our method shows average performance over 20 trials.

two metrics: prediction *accuracy* and precision-recall *area-under-curve* (AUC). **GardensPoint** consists of 200 indoor, outdoor and natural environments with both viewpoint and conditional changes throughout the dataset. A tolerance of 2 is acceptable. **ESSEX3IN1** consists of 210 images taken at the university campus and surroundings, focusing on perceptual aliasing and confusing places. There is no tolerance for this dataset, hence, the prediction has to be exact. **SPEDTest** consists of 607 low-quality but high-depth images collected from CCTV cameras around the World; it includes environmental changes including variations in weather, seasonal and illumination conditions. There is no tolerance for this dataset. **Nordland** consists of 1000 images taken at train traversals in four different seasons in Norway; the viewpoint angle is fixed although there is a high weather, seasonal and illumination variability. A tolerance of 10 is acceptable—the same as the sequential models [10] we compare against (see Section IV-D for more details).

### B. Training ESNs and Hyper-parameter Tuning

The lack of a validation set for the considered tasks makes the hyper-parameters selection challenging. This difficulty is emphasized by the small number of samples in the training set (i.e. one sample per place) and by the major statistical differences between training and test data. In particular, the seasonal difference in the acquisition of reference and query data lead to the possible presence or absence of snow and shifts in colours intensities. In our preliminary experiments, different hyper-parameters would reach perfect accuracy (i.e. $100\%$) on the training set and degraded, variable performance on the test set. We believe that there is a lack of clarity in previous research works regarding the definition of a clear methodology to overcome the problem of hyper-parameter selection.

We tuned the hyper-parameters of the reservoir by using a small percentage (i.e. $10\%$) of samples of the test set as validation. In other words, while the read-out was always optimised from reference samples, hyper-parameters were optimised through grid search over the performance achieved on $10\%$ of the query data. Being aware of the limitations of this methodology, we will later show how it is possible to use the test set of one task as validation for another task with little performance lost, demonstrating how the model can achieve generalisation abilities if the hyper-parameters were selected to be robust to non-excessive statistical changes (see Section IV-C).

## IV. Results

### A. Assessing ESN Utility to Visual Place Recognition

The performance of `ESN` and `ESN+SpaRCe` were first evaluated in three datasets (GardensPoint, SPEDTest and ESSEX3IN1). Fig. 4 shows that both ESN variants outperform state-of-the-art single-view matching models (including NetVLAD with read-out and hidden layers) in all three conditions. The `ESN` achieves mean accuracy scores of $0.75$, $0.99$ and $1.0$ and mean AUC scores of $0.9$, $1.0$ and $1.0$. The addition of the `SpaRCe` layer provides additional improvement with accuracy scores of $0.77$, $0.99$ and $1.0$ and mean AUC scores of $0.93$, $1.0$ and $1.0$.

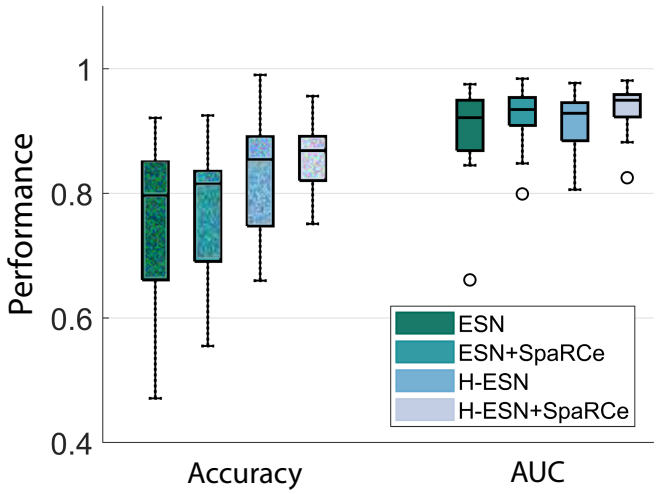


Fig. 5: **Hierarchical models improves performance.** More complex models (`H-ESN` and `H-ESN+SpaRCe`) yields to higher and more robust performance. The box plots show the results over 20 trials.

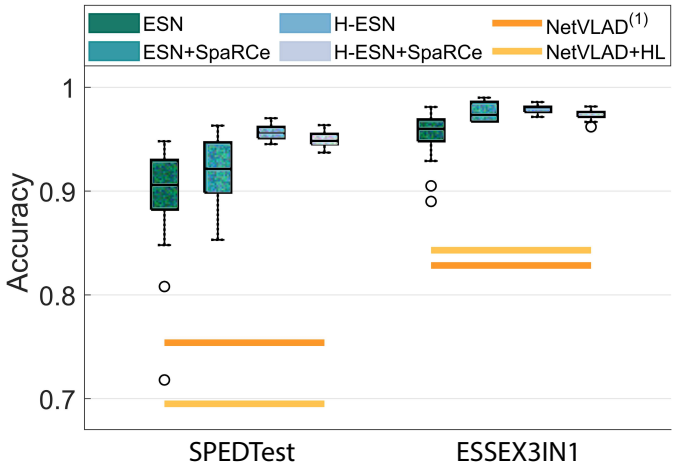


Fig. 6: **Generalisability of the hyper-parameter transferring.** The proposed models show generalisation ability by maintaining performance despite the hyper-parameters were selected using a different dataset (GardensPoint). All four variants of ESN are well above the accuracy achieved by static models (horizontal lines). The box plots represent the distribution of 20 trials.

### *B. Hierarchical Models for Performance Improvement*

We then assessed if a hierarchical ESN architecture would improve results in the challenging GardensPoint dataset. Fig. 5 shows that the introduction of hierarchical ESNs increased the median accuracy scores while decreasing their variance (`ESN` median: 0.80 and std: 0.14 vs `H-ESN+Sparce` median: 0.87 and std: 0.06; both for 20 trials). AUC scores showed little change but they were already close to the maximum possible ($> 93\%$) and thus there was little room for improvement. Considering the performance improvement consequent to the utilisation of the hierarchical model, it is evident how the GardensPoint dataset contains longer temporal dependencies among images that cannot be captured by a single ESN. This result can be intuitively understood by comparing the sequences of images between the three datasets presented in Fig. 4. After an inspection of the datasets, it is clear that data of GardensPoint are captured at a higher frame-rate in comparison to the other datasets, where images appear more static and separated in time across each other. Consequently, GardensPoint has a more complex underlying temporal structure.

### *C. Generalisability Study*

We also analysed the sensitivity of the ESN models with respect to hyper-parameter selection. Fig. 6 shows accuracy scores for hyper-parameters tuned by training the models on GardensPoint and maintaining them when training in SPEDTest and ESSEX3IN1. The reason we chose the hyper-parameters from GardensPoint is that generalisation is more likely to occur when the baseline task is more complex than the new tasks to which it is applied. Indeed, richer and more difficult datasets can lead neural networks to discover high-level features that are transferable to simpler datasets, while the contrary is difficult. Fig. 6 demonstrates how, even with sub-optimal hyper-parameters, the introduction of ESNs leads to higher performance in comparison to single-view matching models, NetVLAD and NetVLAD$^{(1)}$. Again, hierarchical ESNs provide a noticeable improvement in median accuracy and AUC scores as well as reducing variance again. Moreover, the performance remains above $90\%$ for both accuracy and AUC compared to the virtually perfect scores achieved when hyper-parameters were tuned using the same dataset (see Fig 4).

### *D. Comparing ESN with sequential VPR models*

In this section, we benchmark the performance of ESNs against state-of-the-art sequence matching VPR models. Specifically, we compare with two models recently reported to achieve great performance [10] in the challenging Nordland dataset [11]. Both models use a bio-inspired feedforward neural network (FlyNet) to encode visual information and either a recurrent neural network (RNN) or a continuous attractor network (CANN) to introduce temporality. Fig. 7 shows accuracy scores of 0.72 and 0.92 for the standard `ESN` and `ESN+SpaRCe` respectively (no accuracy scores are available for comparison). For the AUC test, `ESN` achieves scores of 0.95, with `SpaRCE` improving results to 0.98. This compares favourably to both static view matching models (e.g. NetVLAD+HL) which score 0.24, and sequential models which score 0.21 (FlyNet+RNN) and 0.91 (FlyNet+CANN).

## V. Conclusions

In this paper, we have demonstrated the viability of ESNs as a solution to the VPR problem. All the ESN variants implemented achieve higher performance than single-view matching models (AMOSNet, DenseVLAD, NetVLAD, NetVLAD+HL, NetVLAD$^{(1)}$), in three benchmarking datasets (GardensPoint, SPEDTest, ESSEX3IN1). In

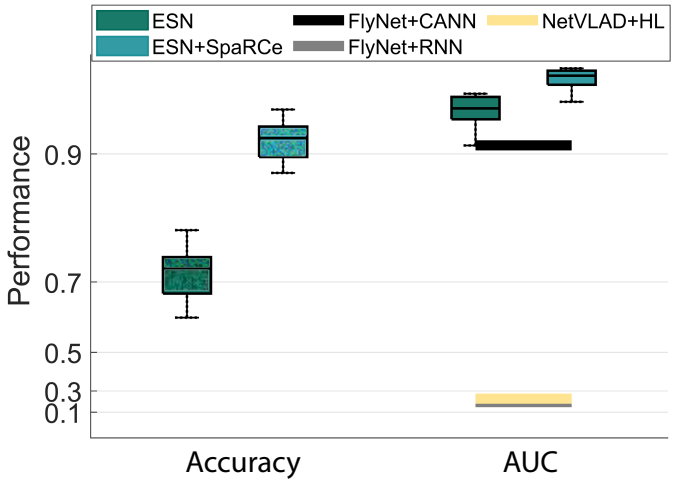


Fig. 7: **Comparison against state-of-the-art sequential models in Nordland dataset.** The `ESN` model and in particular `SpaRCe`, show class-leading performance on the Nordland dataset. The horizontal lines report the performance of FlyNet+RNN and FlyNet+CANN, taken from [10]. The box plots represent the distribution of 20 trials.

the more challenging Nordland dataset, two of our models (single reservoir ESN and SpaRCe) achieved performance above/equal to the class-leading results achieved by sequential matching models (FlyNet+RNN and FlyNet+CANN). While performance is comparable we note that FlyNet [10] have many fewer parameters. However, the ESNs do not require images to be cached during multiple comparisons and also serve to implicitly assess any velocity dependence through the temporal dynamics.

In terms of the recent ESN advances, namely hierarchical and SpaRCe, the results differ depending on the dataset. The addition of SpaRCe to the standard ESN improved performance considerably, showing how the introduction of sparse representations can efficiently help the classification process. The utilisation of hierarchical ESNs was beneficial in the GardensPoint dataset, but not for the larger and more challenging Nordland dataset. Hierarchical models have higher complexity, in terms of the number of hyper-parameters, and can overfit the training data. This is particularly an issue when considering the benchmarking VPR datasets, as there is only a 'single' sample to learn from (as opposed to standard machine learning datasets that have many samples per class, e.g. approximately 6000 samples per class for the well-known MNIST dataset). Preliminary analysis supports this hypothesis: hierarchical models achieved perfect scores on the Nordland training sets (summer) but low performance when presented with test set (winter). Such issues might be addressed by augmenting training data [32] (e.g. through artificial illuminance changes or weather effects) to supply a variety of real-world conditions.

While there are many ways to optimise the ESNs for the VPR problem, an intriguing future course of action is to take inspiration from invertebrate mini-brains that possess analogous structural motifs of both deep and shallow ESNs. A simple example is the insect mushroom body. This is considered the cognitive centre of the insect brain [33] and is necessary for learning relationships sequences and patterns in honey bees [33], [34], [35], [36]. Structurally the mushroom body is a three-layer network with a compact input layer, an expanded middle layer of inter-neurons called Kenyon cells, and a small layer of output neurons [37]. The connections between the Kenyon cells and output neurons are plastic and modified by learning [38], and there are chemical and electrical synapses between the Kenyon cells [39], [40], [41]. These features are analogous to the recurrent connections in the reservoir layer of an ESN, and it has been hypothesised [31], [25] that these recurrent connections in the Kenyon cell layer could contribute to the reverberant activity of the mushroom body that supports forms of memory [42]. Given the similar structures, insights gained from neurobiology could help shape the future ESN investigations and in turn, analysis of the optimal structure for VPR could shed light on the function of different brain areas.

In practice, it is desirable that places are recognised from a single input image allowing robotics to truly solve the kidnapped robot problem. However, in the cases where such methods fail, traversing portions of a familiar path can help to disambiguate input . ESNs provide a means to exploit such temporal dynamics using only visual data but more powerful variants require tuning of a large number of parameters which may not be possible when only a small amount of training examples are provided. Other methods [5], [10] have focused on low-parameter models but often require additional cues such as velocity to focus the image search. Ensemble methods [43], [44] that combine these features are emerging that may provide the best of both worlds.

Finally, assessment of methods on robots in the real-world is essential. This will not only challenge current approaches to be more robust but can also show some difficulties caused by the pre-collected datasets, such as continual learning or robotic safety.

## REFERENCES


[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[2] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 3223–3230.

[3] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[5] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[6] M. J. Milford, "Vision-based place recognition: how low can you go?" *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 766–789, 2013.

[7] P. Hansen and B. Browning, "Visual place recognition using HMM sequence matching," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4549–4555.

[8] E. Kagioulis, A. Philippides, P. Graham, J. C. Knight, and T. Nowotny, "Insect inspired view based navigation exploiting temporal information," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2020, pp. 204–216.

[9] L. Zhu, M. Mangan, and B. Webb, "Spatio-temporal memory for navigation in a mushroom body model," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2020, pp. 415–426.

[10] M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, and M. Milford, "A hybrid compact neural architecture for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 993–1000, 2020.

[11] N. Sünderhauf, P. Neubert, and P. Protzel, "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*. Citeseer, 2013, p. 2013.

[12] A. Glover, "Day and night, left and right," 2014. [Online]. Available: https://doi.org/10.5281/zenodo.4590133

[13] M. Zaffar, S. Ehsan, M. Milford, and K. D. McDonald-Maier, "Memorable maps: A framework for re-defining places in visual place recognition," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[14] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, "Learning context flexible attention model for long-term visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4015–4022, 2018.

[15] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of Echo State Networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[16] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[17] M. Hermans and B. Schrauwen, "Recurrent kernel machines: Computing with infinite echo state networks," *Neural Computation*, vol. 24, no. 1, pp. 104–133, 2012.

[18] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust Echo State Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 787–799, 2012.

[19] A. Deihimi and H. Showkati, "Application of Echo State Networks in short-term electric load forecasting," *Energy*, vol. 39, no. 1, pp. 327–340, 2012.

[20] P. G. Plöger, A. Arghir, T. Günther, and R. Hosseiny, "Echo State Networks for mobile robot modeling and control," in *Robot Soccer World Cup*. Springer, 2003, pp. 157–168.

[21] K. Ishu, T. van Der Zant, V. Becanovic, and P. Ploger, "Identification of motion with Echo State Network," in *MTS/IEEE Techno-Ocean*, vol. 3. IEEE, 2004, pp. 1205–1210.

[22] C. Hartland and N. Bredeche, "Using Echo State Networks for robot navigation behavior acquisition," in *IEEE International Conference on Robotics and Biomimetics*. IEEE, 2007, pp. 201–206.

[23] H. Jaeger, "Discovering multiscale dynamical features with hierarchical echo state networks," Jacobs University Bremen, Tech. Rep., 2007.

[24] C. Gallicchio, A. Micheli, and L. Pedrelli, "Design of deep echo state networks," *Neural Networks*, vol. 108, pp. 33–47, 2018.

[25] L. Manneschi, M. O. Ellis, G. Gigante, A. C. Lin, P. Del Giudice, and E. Vasilaki, "Exploiting Multiple Timescales In Hierarchical Echo State Networks," *Frontiers in Applied Mathematics and Statistics*, 2021.

[26] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.

[27] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[28] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" *arXiv preprint arXiv:2103.06443*, 2021.

[29] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier, and S. Ehsan, "VPR-Bench: Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change," *International Journal of Computer Vision*, pp. 1–39, 2021.

[30] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 659–686.

[31] L. Manneschi, A. C. Lin, and E. Vasilaki, "SpaRCe: Improved Learning of Reservoir Computing Systems through Sparse Representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[33] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in Cognitive Sciences*, vol. 5, no. 2, pp. 62–71, 2001.

[34] C. Boitard, J.-M. Devaud, G. Isabel, and M. Giurfa, "Gabaergic feedback signaling into the calyces of the mushroom bodies enables olfactory reversal learning in honey bees," *Frontiers in Behavioral Neuroscience*, vol. 9, p. 198, 2015.

[35] J.-M. Devaud, T. Papouin, J. Carcaud, J.-C. Sandoz, B. Grünewald, and M. Giurfa, "Neural substrate for higher-order learning in an insect: mushroom bodies are necessary for configural discriminations," *Proceedings of the National Academy of Sciences*, vol. 112, no. 43, pp. E5854–E5862, 2015.

[36] A. J. Cope, E. Vasilaki, D. Minors, C. Sabo, J. A. Marshall, and A. B. Barron, "Abstract concept learning in a simple neural network inspired by the insect brain," *PLoS Computational Biology*, vol. 14, no. 9, p. e1006435, 2018.

[37] S. E. Fahrbach, "Structure of the mushroom bodies of the insect brain," *Annual Review of Entomology*, vol. 51, pp. 209–232, 2006.

[38] B. Gerber, H. Tanimoto, and M. Heisenberg, "An engram found? evaluating the evidence from fruit flies," *Current Opinion in Neurobiology*, vol. 14, no. 6, pp. 737–744, 2004.

[39] Z. Zheng, J. S. Lauritzen, E. Perlman, C. G. Robinson, M. Nichols, D. Milkie, O. Torrens, J. Price, C. B. Fisher, N. Sharifi, *et al.*, "A complete electron microscopy volume of the brain of adult drosophila melanogaster," *Cell*, vol. 174, no. 3, pp. 730–743, 2018.

[40] S.-y. Takemura, Y. Aso, T. Hige, A. Wong, Z. Lu, C. S. Xu, P. K. Rivlin, H. Hess, T. Zhao, T. Parag, *et al.*, "A connectome of a learning and memory center in the adult drosophila brain," *eLife*, vol. 6, p. e26975, 2017.

[41] Q. Liu, X. Yang, J. Tian, Z. Gao, M. Wang, Y. Li, and A. Guo, "Gap junction networks in mushroom bodies participate in visual learning and memory in drosophila," *eLife*, vol. 5, p. e13238, 2016.

[42] P. Cognigni, J. Felsenberg, and S. Waddell, "Do the right thing: neural network mechanisms of memory formation, expression and update in drosophila," *Current Opinion in Neurobiology*, vol. 49, pp. 51–58, 2018.

[43] S. Hausler, A. Jacobson, and M. Milford, "Multi-process fusion: Visual place recognition using multiple image processing methods," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1924–1931, 2019.

[44] T. Fischer and M. Milford, "Event-based visual place recognition with ensembles of temporal windows," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6924–6931, 2020.