A Compact Neural Architecture for Visual Place Recognition*

Marvin Chancán[1,3], Luis Hernandez-Nunez[2,3], Ajay Narendra[4],
Andrew B. Barron[4], Michael Milford[1]

***Abstract*— State-of-the-art algorithms for visual place recognition can be broadly split into two categories: computationally expensive deep-learning/image retrieval based techniques with minimal biological plausibility, and computationally cheap, biologically inspired models that yield poor performance in real-world environments. In this paper we present a new compact and high-performing system that bridges this divide for the first time. Our approach comprises two key components: FlyNet, a compact, sparse two-layer neural network inspired by fruit fly brain architectures, and a one-dimensional continuous attractor neural network (CANN). Our FlyNet+CANN network combines the compact pattern recognition capabilities of the FlyNet model with the powerful temporal filtering capabilities of an equally compact CANN, replicating entirely in a neural network implementation the functionality that yields high performance in algorithmic localization approaches like SeqSLAM. We evaluate our approach and compare it to three state-of-the-art methods on two benchmark real-world datasets with small viewpoint changes and extreme appearance variations including different times of day (afternoon to night) where it achieves an AUC performance of 87%, compared to 60% for Multi-Process Fusion, 46% for LoST-X and 1% for SeqSLAM, while being 6.5, 310, and 1.5 times faster respectively.**

I. INTRODUCTION

Performing visual localization reliably is a challenge for any robotic system operating autonomously over long time periods in real-world environments, due to viewpoint changes, perceptual aliasing (multiple places may look similar), and appearance variations over time (e.g. day/night cycles, weather/seasonal conditions) [1]. Convolutional neural networks (CNN) [2], heavily used in computer vision for feature extraction and image classification tasks, have been applied to the field of visual place recognition (VPR) with great success [3]–[5], typically only used in real-time [6]–[8] with dedicated hardware (GPUs). In addition, vanilla CNN pre-trained on benchmark datasets such as ImageNet [9] generally neglect any temporal information between images.

To address these shortcomings, researchers have introduced recurrent models [10]–[12], but this has increased the complexity of the training process of CNN models, further limiting their deployability in a range of real-world applications, especially in scenarios with limited computational resources and/or training data.

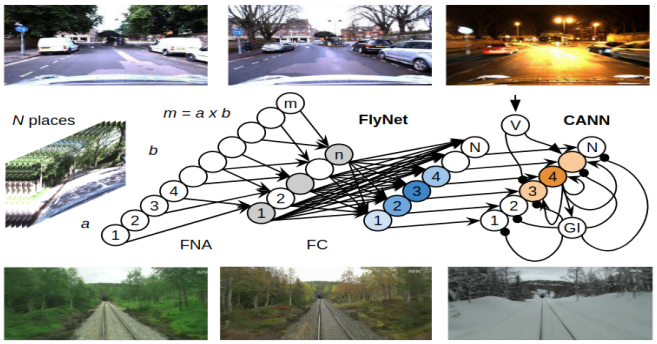

Fig. 1. **Our FlyNet+CANN architecture**. The FlyNet neural network comprises a hidden-layer inspired by the *Drosophila melanogaster* olfactory system—FlyNet algorithm (FNA), and a fully-connected (FC) output layer. We leverage this compact model by combining it with a continuous attractor neural network (CANN) to perform appearance-invariant visual place recognition. Experiments using two real-world datasets, Oxford RobotCar (top) and Nordland (bottom), show that our FlyNet+CANN network achieves competitive state-of-the-art results with a fraction of the computational footprint of conventional approaches (see Fig. 2).

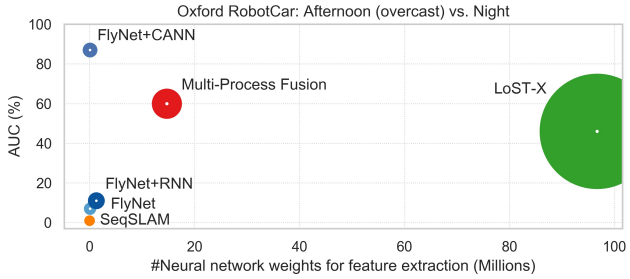

Fig. 2. **Oxford RobotCar AUC performance vs. Model Size**. Footprint comparison for the most challenging appearance change (day vs. night).

In this work, we introduce FlyNet, a compact network architecture inspired by the fruit fly olfactory neural circuit—that is known to assign similar neural activity patterns to similar odors [13]. Furthermore, FlyNet addresses the issue of temporal relationships between images from previously visited locations since it is coupled with a temporal encoding model such as a recurrent neural network (RNN) or a continuous attractor neural network (CANN) (see Fig. 1). We show that our resulting network FlyNet+CANN achieves competitive performance on two benchmark robotic datasets but with far less parameters (see Fig. 2), minimal training time, storage and computation footprint than conventional CNN-based approaches. In Fig. 2 the area of the circle is proportional to the number of layers per model. The number of layers of the pre-trained networks used in MPF (`VGG-16`) and LoST-X (`RefineNet`) for feature extraction purposes are 13 and 213 respectively, while for FlyNet+CANN is 2.

* This work was supported by the Peruvian Ministry of Education to M.C. and by an ARC Future Fellow FT140101229 to M.M.

[1] School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD 4000, Australia

[2] Center for Brain Science & Department of Physics, Harvard University, Cambridge, MA 02138, USA

[3] School of Mechatronics Engineering, Universidad Nacional de Ingeniería, Lima, Rímac 15333, Peru. mchancanl@uni.pe

[4] Department of Biological Sciences, Macquarie University, Sydney, NSW 2109, Australia

This paper is structured as follows. Section II provides a brief overview of related work on VPR and the biological inspiration for the FlyNet architecture; Section III describes the FlyNet architecture in detail; Section IV and Section V present the experiments and evaluations respectively, where our results are compared against three existing state-of-the-art techniques on two real-world benchmark datasets; and Section VI provides discussion around our results as well as future work.

II. Related Work

This section outlines some key biological background for navigation in the insect brain, reviews the usage of deep learning-based approaches for localization and discusses recent developments in temporal filtering of sequential data to further improve performance.

A. Navigation in Biological Brains

Our understanding of how animals navigate using vision has been used as inspiration for designing effective VPR algorithms. RatSLAM [14] is one example of this, using a model based on the rodent brain to perform visual SLAM over large environments for the first time [15]. Likewise researchers have developed a range of robotic navigation models based on other animals including insects [16]–[18].

Insects such as ants, bees and wasps regularly return to specific locations that are of significance to them such as the nest, large food sources and the routes leading to them. It is typical for bees to range more than 6 km from the hive on a foraging trip [19]. Furthermore, these insects use environmental cues, as well as visual and odor information for effective navigation [20], [21]. They can effectively encode multiple places in their memory and use visual features to identify and navigate between them [22], [23].

Flies also exhibit a similar ability to navigate [24], [25]. In fact, their brains share the same general structure [20], [26], with the central complex being closely associated with navigation, orientation and spatial learning [27], [28]. VPR is, however, most likely mediated by processing within the "mushroom bodies", a separate pair of structures within the brain. These regions are involved in classification, learning, and recognition of both olfactory and visual information in bees and ants [27]. They receive densely coded and highly processed input from the sensory lobes, which then connects sparsely to the very large numbers of intrinsic neurons within mushroom bodies. Their structure has been likened to a perceptron, and is considered optimal for learning and correctly classifying complex input [29].

These impressive capabilities, achieved with relatively small brains, make them attractive models for roboticists. For FlyNet, we take inspiration from the olfactory neural circuits found in the fruit fly to design our network architecture. Our focus here is primarily on taking high level inspiration from the size and structure of what is known of the fly brain and investigating the extent to which it can be used for VPR and localization, much as in the early RatSLAM work [14].

B. Convolutional Neural Networks

Over recent years CNN have been applied to a range of recognition problems with great success, including the VPR problem addressed here. These models can handle many challenging environments with both visual appearance and viewpoint changes, as well as large scale VPR problems [3], [30]–[33]. Despite their success, these approaches typically rely on the usage of CNN models that are pre-trained on various computer vision datasets [34] using millions of images [5]–[7], [30]. CNN models trained in an end-to-end fashion specifically for the VPR tasks have also recently been proposed [4], [30]. However, they are still initialized using pre-trained network architectures, i.e., `AlexNet` [35], `VGGNet` [52], `ResNet` [53], with slight modifications to the model architecture to perform VPR. All these systems share common undesirable characteristics with respect to their widespread deployability on robots including large network sizes, significant computational and storage requirements, and onerous training requirements.

C. Modeling Temporal Relationships

To access and exploit the power of temporal information in many applications, researchers have developed a range of approaches including Recurrent Neural Networks (RNN) [10], [12] such as Long Short-Term Memory (LSTM) [11]. These temporal-based approaches have been applied specifically to navigation [36] and spatial localization [37] in artificial agents. In a nice closure back to the inspiring biology, these approaches led to the emergence of grid-like representations in networks trained to perform path integration, resembling neural representations found in grid cells and other entorhinal cortex cell types [38] in mammalian brains [39]. One of the older approaches to filtering temporal information in a neural network incorporated continuous attractor neural networks (CANNs), with pre-assigned weight structures set up to model the activity dynamics of place and grid cells found in the rat mammalian brain [14]. Other non-neural techniques have been developed including SeqSLAM [40], which matches sequences of pre-processed frames from a video feed to provide an estimate of place, with a range of subsequent follow on system development [41]–[43].

The work to date has captured many key aspects of the VPR problem, investigating complex but powerful deep learning-based approaches, biologically-inspired approaches that work in simulation or in small laboratory mazes, or larger mammalian-brain based models with competitive real-world robotics performance. In this work we attempt to merge the desirable properties of several of these techniques, by developing a novel bio-inspired neural network architecture for VPR that Through being loosely based on insect brains, this architecture is extremely compact but also achieves competitive localization performance using the filtering capabilities of a continuous attractor neural network. We also show how our compact core Flynet architecture can easily be adapted to other filtering techniques including RNNs and SeqSLAM.

III. Methods

In this Section, we present the FlyNet core algorithm inspired by the *fly algorithm* introduced in [13], and its derived neural network architectures.

A. Fruit Fly Olfactory Neural Circuit

The fly olfactory system solves an essential computing problem, approximate similarity search, by assigning similar neural activity patterns to similar odors in the fly olfactory neural circuit. The neural strategy that this neural circuit performs is a three-step procedure as the input odor information goes through a three-layer neural circuit [13]. First, the firing rates across the first layer are centered to the same mean for all odors (removing the concentration dependence). Second, the third layer is connected by a sparse, binary random matrix to the second layer (each output neuron receives and sums about 10% of the firing rates in the second layer). Third, only the highest-firing 5% neurons across the third layer create a specific binary tag to the input odor (winner-take-all).

B. Proposed FlyNet Algorithm

In this work, we leverage the *fly algorithm*, previously described, from a computer vision perspective to propose the Algorithm 1 - FlyNet Algorithm (FNA). Fig. 3 show the FNA mapping, when the input and output dimensions are $m = 20$, $n = 6$ respectively. The FNA then computes n random projections defined by $\mathbf{W}$ (the scheme only shows the projections to y_2 and y_j). The WTA step then creates a binary output tag, which is a compact representation of the input feature vector.

Algorithm 1 FlyNet Algorithm (FNA)

Input: $\mathbf{x} \in \mathbb{R}^m$

Output: $\mathbf{y} \in \mathbb{R}^n$, $n < m$

1: Initialize $\mathbf{W} \in \mathbb{R}^{n \times m}$: A binary, sparse random connection matrix between the input $\mathbf{x}$ and the output $\mathbf{y}$.

2: Compute the output $\mathbf{y} = \mathbf{W}\mathbf{x}$: Each output y_j receives and sums the values from approximately 10% of the randomly selected input units x_i.

3: Winner-take-all (WTA): Set the top 50% output units y_i to 1, and the remaining units to 0.

We utilize the proposed FNA as our core feature encoder to develop the compact neural network architectures described in the next subsection. We also perform an image preprocessing step before applying Algorithm 1. The details of this step are outlined in the following Section IV-C.

C. FlyNet Architectures

We implement three neural network architectures based on the FNA: a single-frame based system (FlyNet), and two multi-frame based (FlyNet+RNN, FlyNet+CANN) VPR systems with temporal filtering capabilities. For comparison purposes, we also deploy a FlyNet+SeqSLAM system, described in more detail in Section IV.

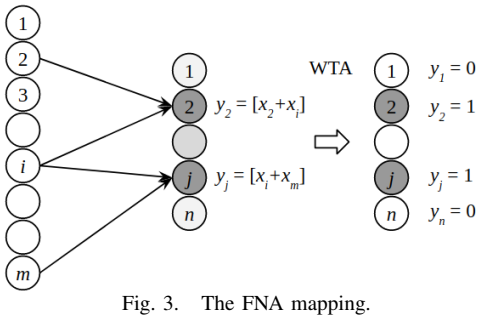

Fig. 3. The FNA mapping.

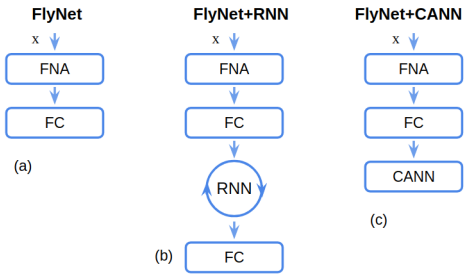

Fig. 4. **FlyNet baselines**: (a) single- and (b, c) muti-frame networks.

1) FlyNet: Our custom two-layer neural network architecture, FlyNet (see Fig. 4 left), has an input image dimension of 32×64 ($m = 2048$) in gray-scale, and incorporates the FNA as a hidden-layer with output dimension $n = 64$. Then, the FNA output feeds into a fully-connected (FC) layer of 1000 units with a soft-max activation function in order to compute a particular class score for each input image to the network. In other words, the FNA maps the input images into a compact feature vector representation to perform VPR.

FlyNet vs. Fully-Connected Neural Network. A generalized version of our proposed FlyNet architecture was also considered during our experiments, that is, one where the FNA hidden-layer would be replaced by a FC layer. The main difference between FlyNet and a two-layer neural network is that FlyNet does not require trainable parameters and encodes the input vector using a sparse, binary matrix. In contrast, a conventional two-layer neural network would require $(m \times n + n)$ trainable parameters, and hence was left as an avenue for future work.

2) FlyNet+SeqSLAM: We conducted additional experiments incorporating the algorithmic technique SeqSLAM [40] (described in Section II-C) on top of our single-frame FlyNet network in order to obtain a multi-frame system as a comparison method along with our temporal filtering based architectures (FlyNet+RNN and FlyNet+CANN) described in the following Subsections.

3) FlyNet+RNN: The first temporal filtering enhancement involved incorporating a vanilla RNN on top of the FlyNet architecture. We also investigated the usage of more sophisticated types of recurrent layers such as a Gated Recurrent Unit (GRU) and LSTM, however they showed no significant performance improvements despite having far more parameters. Fig. 4 (middle) illustrates one of our temporal filtering based neural architectures, FlyNet+RNN.

4) FlyNet+CANN: We implemented a variation of the CANN model introduced in [15], motivated by its suitability as a compact neural network-based way to implement the filtering capabilities of SeqSLAM [40]. As described in Section II-C, a CANN is a type of recurrent network that utilizes pre-assigned weight values within its configuration. The Fig. 1 (middle) shows our detailed FlyNet+CANN implementation, see also Fig. 4 (right). It can also be seen in Fig. 1 (middle) that, in contrast to an RNN, a unit in a CANN can excite or inhibit itself and units nearby using excitatory (arrows) or inhibitory (rounds) connections respectively. For this implementation, activity shifts in the network representing movement through the environment were implemented through a direct shift and copy action, although this could be implemented with more biologically faithful details such as velocity units and asymmetric connections, as in prior CANN research [44].

IV. Experiments

To provide a comprehensive evaluation of the capabilities of our proposed systems, we conducted extensive evaluations on two of the most widespread benchmarks used in VPR, the Nordland [45] and Oxford Robotcar [46] datasets. We also conduct comparison to three state-of-the-art techniques. In this section we describe our network configurations, dataset preparation, any relevant pre-processing and the comparison systems we implemented[1].

A. FlyNet Baselines

We test our four FlyNet baselines (described in Section III-C) in order to compare them and further evaluate our best performing network against the current state-of-the-art VPR methods. We define the same number of units for the corresponding layers between our models. For instance, the baseline architectures for both FlyNet and FlyNet+RNN use an FNA layer with 64 units, and corresponding FC layers with 1000 units. The number of recurrent units in the RNN-based model was 512. The CANN-based model uses 1002 units (see Table I). The Adam optimizer [47] was chosen to train our baselines, with a learning rate set to 0.001 for all our experiments.

TABLE I
FlyNet baseline architectures

Architecture	# layers	# params	# neurons
FlyNet	2	64k	1064
FlyNet+RNN	4	1.3M	2576
FlyNet+CANN	3	72k	2066

B. Real-World Datasets

The experiments are performed on two benchmark datasets widely used in the VPR literature: Nordland and Oxford RobotCar.

[1]Code available at `https://github.com/mchancan/flynet`

1) Nordland: The Nordland dataset, introduced in [45], comprises four single traverses of a train journey, in northern Norway, with extreme seasonal changes (spring, summer, fall, and winter). The dataset is primarily used to evaluate visual appearance change, as instantiated through its four season coverage. In our experiments we used the four traverses to perform VPR at 1fps as in [45]. We use the summer subset for training our models, and the reminder for testing.

2) Oxford RobotCar: The Oxford RobotCar dataset [46] provides several traverses with different lighting (e.g. day, night) and weather (e.g. direct sun, overcast) changes through a car ride in Oxford city—that implicitly contains various challenges of pose and occlusions such as pedestrians, vehicles, and bicycles. In our evaluations we used the same subsets as in [33], which include: two traverses with the most challenging illumination conditions, referred to here as day (overcast summer) and night (autumn) for testing purposes, and overcast (autumn) for training. The variable distance between frames ranged from 0 to 5 meters.

TABLE II
Datasets and tested traverses: Reference-Query

Dataset	Appearance Changes	Viewpoint Changes
Nordland	Summer-Fall	Small
Nordland	Summer-Winter	Small
Oxford RobotCar	Overcast-Day	Moderate
Oxford RobotCar	Overcast-Night	Moderate

C. Image Preprocessing

The image preprocessing procedure we use to evaluate our FlyNet baselines comprises two steps: converting the images into a single channel (grayscale) with normalized pixel values between [0, 1], and resizing to 32×64 pixels. The dataset length we use in all the traverses is 1000 images (places). On the other hand, the three state-of-the-art VPR methods we compare with, were provided the original image resolutions, 1920×1080 for Nordland, and 1280×960 for Oxford RobotCar.

D. Comparative Analysis

Our best performing baseline, FlyNet+CANN, is compared to three state-of-the-art multi-frame VPR systems: the algorithmic technique SeqSLAM [40] (without FlyNet attached), and two deep learning based methods such as LoST-X [33] and the recent work Multi-Process Fusion (MPF) [56].

1) SeqSLAM: SeqSLAM shows state-of-the-art results on VPR with challenging appearance variations. We used the MATLAB implementation of SeqSLAM presented in [45], with a sequence length of $ds = 20$ frames, and threshold of 1 in order to have consistent and comparable results. The remaining SeqSLAM parameters maintained its default values.

2) LoST-X: The multi-frame LoST-X pipeline [33] use visual semantics to perform VPR with opposite viewpoints across day and night cycles. This method uses the RefineNet architecture [55] (a `ResNet-101` [53] based model) as a feature extractor, pre-trained on the Cityscapes [54] for high-resolution semantic segmentation.

Fig. 5. **Top**: Sample images (reference) of the Nordland summer traversal. **Bottom**: Corresponding frames retrieved using our FlyNet+CANN network from the winter traversal (query).

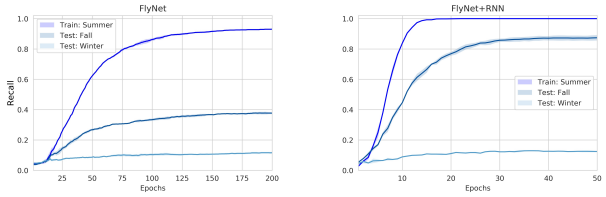

Fig. 6. Recall performance during training and testing procedures for our FlyNet (left) and FlyNet+RNN (right) networks on the Nordland dataset.

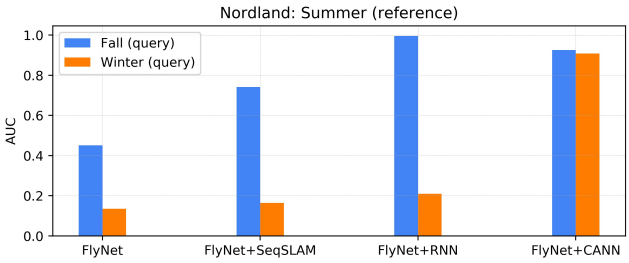

Fig. 7. FlyNet baselines AUC performance comparison on Nordland.

3) Multi-Process Fusion (MPF): MPF [56] is also a multi-frame, state-of-the-art recent VPR technique. We use the `VGG-16` network [52] trained on Places365 [57] as the feature extractor to feed the MPF algorithm.

E. Evaluation Metrics

We evaluate the quantitative results of our best performing network against three multi-frame state-of-the-art methods (SeqSLAM, LoST-X, and MPF) by using both the precision-recall (PR) curves and area under the curve (AUC) metrics. The tolerance used to classify a query place as a correct match was being within 20 frames of the ground truth location in the Nordland dataset, and up to 50 meters (10 frames) away from the ground truth location in the Oxford RobotCar dataset, as per previous research [33], [56], [58].

V. Results

We trained our four FlyNet baselines in an end-to-end manner from scratch. On the Nordland dataset, we used the Summer traversal (reference) for training and the Fall/Winter traverses (query) for testing. On the Oxford RobotCar dataset, the Overcast traversal (reference) was chosen for training, and the Day/Night traverses (query) for testing. Similarly, the three state-of-the-art methods were evaluated using the same reference/query traverses, respectively.

A. FlyNet Baselines Evaluations

Figs. 5 and 8 show the qualitative results for our best performing baseline (FlyNet+CANN) on both benchmark

Fig. 8. **Top**: Sample images (reference) of the Oxford RobotCar overcast traversal. **Bottom**: Corresponding frames retrieved using our FlyNet+CANN network from the night traversal (query).

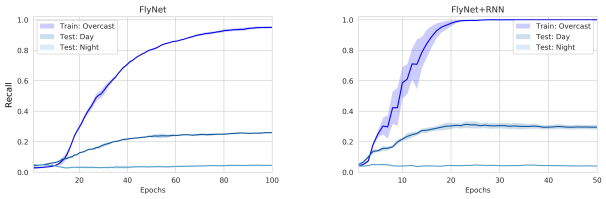

Fig. 9. Recall performance during training and testing procedures for our FlyNet (left) and FlyNet+RNN (right) networks on the Oxford dataset.

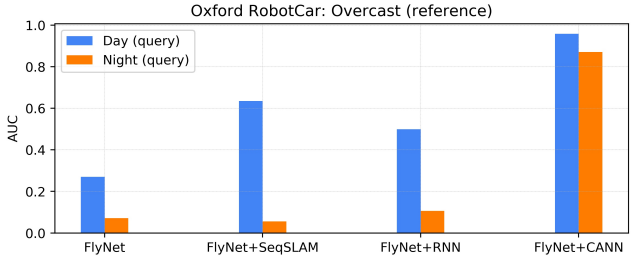

Fig. 10. FlyNet baselines AUC performance comparison on Oxford.

datasets. It can be seen in Fig. 5 that the system is able to correctly match places under significant seasonally-driven appearance changes (summer vs winter) on the Nordland dataset. Similarly, Fig. 8 shows again a correctly retrieved image sequence under drastic illumination changes (overcast vs night), even when facing occlusions such as vehicles.

The single-frame based model, FlyNet, and its recurrent version, FlyNet+RNN, struggled in these conditions. FlyNet alone did not perform well; as shown in Fig. 9 (left) the best test recall on the Nordland dataset was below 40% (Summer vs Fall), and on the Oxford RobotCar the best test recall was around 20% (Overcast vs Day), see Fig. 9 (left). When integrating FlyNet with a RNN performance improved in some cases, see Fig. 9 (right), but on the other Nordland traverses (spring, winter) as well as on the Oxford RobotCar traverses (day, night) the RNN did not improve the performance of the single-frame model.

Additionally, we compared our four FlyNet baselines' performance by calculating the area under the curve (AUC) metric when trained on Summer and tested on Fall/Winter seasonal changes for the Nordland dataset (see Fig. 7) and when trained on Overcast and tested on Day/Night time changes for the Oxford RobotCar dataset (see Fig. 10). It can be seen that the FlyNet+CANN network outperformed the other baselines, and this is the network we choose to compare with the state-of-the-art methods in the next subsection.

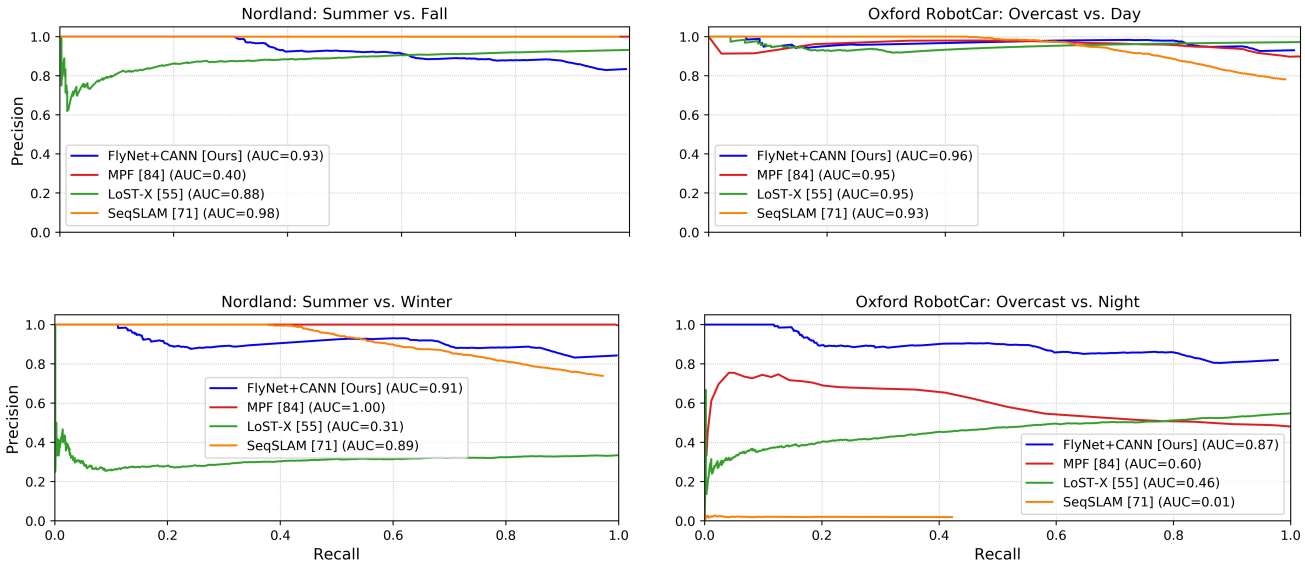

Fig. 11. PR curves of FlyNet+CANN vs. SeqSLAM, LoST-X and MPF on 1000-places of the Nordland (left) and Oxford RobotoCar dataset (right).

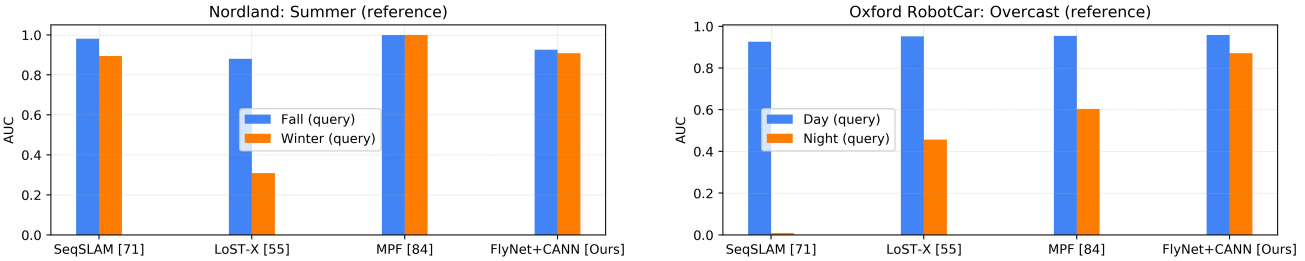

Fig. 12. AUC performance of FlyNet+CANN vs. SeqSLAM, LoST-X, and MPF on the Nordland (left) and Oxford RobotCar (right) dataset.

B. State-of-the-Art Analysis

Figs. 11-12 shown the quantitative results of our FlyNet+CANN network in comparison with three state-of-the-art, multi-frame VPR methods: SeqSLAM, LoST-X, and MPF.

Figs. 11 (left) show the PR curves on the Nordland dataset. It can be seen that the MPF is the best performing VPR system which is able to recall almost all places at 100% precision on both Fall and Winter testing traverses, and also achieve higher AUC values (see Fig. 12 (left)). On the other hand, the deep learning-based method, LoST-X, is not able to recall a single match at 100% precision on both testing traverses. In contrast, our FlyNet+CANN network achieves state-of-the-art performance comparable with SeqSLAM and MPF for all of the testing traverses, as can be seen in Fig. 12 (left).

Similarly, the PR performance on the Oxford RobotCar dataset is shown in Fig. 11 (right). Also notable in this case is that the FlyNet+CANN baseline again achieves state-of-the-art results that are now comparable with SeqSLAM, LoST-X, and MPF approaches. The FlyNet+CANN model consistently maintains its performance even under the extreme appearance change faced in that experiment caused by overcast-night cycle, as seen in Fig. 11 (right-bottom), which is represented in Fig. 12 (right), where the FlyNet+CANN network show higher AUC values than the remaining state-of-the-art techniques.

C. Computational performance

The computational cost required by our best performing VPR network (FlyNet+CANN) is compared with the three state-of-the-art methods (SeqSLAM, LoST-X, and MPF) in terms of the running time for (1) feature extraction by the neural networks models, (2) feature matching between the reference and query traverses, and the (3) average processing time to compare and match a single query image to the 1000 database images (reference), that can be calculated as (Feat. Ext. + Feat. Match.)/1000. Table III shows that our FlyNet+CANN network is 6.5, 310, and 1.5 times faster than MPF, LoST-X, and MPF respectively in terms of the average processing time (m: minutes, s: seconds).

TABLE III
PROCESSING TIME COMPARISON ON THE NORDLAND DATASET

VPR system	Feat. Ext.	Feat. Match.	Average
FlyNet+CANN	35s	25s	0.06s (16.66 Hz)
MPF	1.9m	4.6m	0.39s (2.56 Hz)
LoST-X	110m	200m	18.6s (0.05 Hz)
SeqSLAM	50s	40s	0.09s (11.11 Hz)

The tested VPR systems were processed using their default parameters and recommended configurations. We performed all our tests on an Ubuntu 16.04 LTS computer with 2× GeForce GTX1080Ti GPU. The SeqSLAM algorithm used CPU processing with MATLAB. Both LoST-X and MPF used 1× GPU for their RefineNet and VGG-16 pre-trained

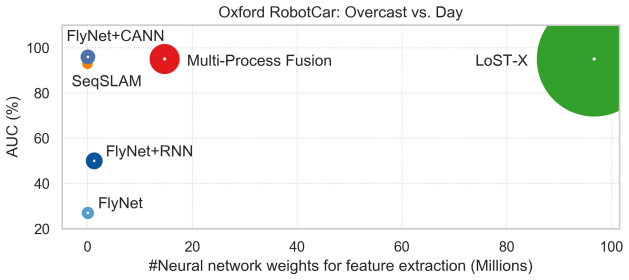

Fig. 13. **Oxford RobotCar AUC performance vs. Model Size**. Similar to Fig. 1; it compares small appearance changes (overcast vs. day).

networks, respectively. Similarly, our FlyNet+CANN network used 1× GPU. However, the higher time required by the LoST-X framework was due to the intermediate checking and processing steps as it used the CPU.

Fig. 13 shows a similar comparison presented in Fig. 2 but with moderated appearance changes (overcast vs. day) on the Oxford RobotCar dataset. In both figures, again, the area of the circle is proportional to the number of layers per model, except for the SeqSLAM system which performs an algorithmic matching procedure. We can see that state-of-the-art systems such as MPF, LoST-X and SeqSLAM achieve better results than in Fig. 2 in terms of AUC metrics: 95%, 95% and 93% respectively, while FlyNet+CANN also present competitive performance with 96%, as explained in Subsection V-B,C and shown in Fig. 11 (top right) and Fig. 12 (right).

VI. Conclusion

In this paper, we presented a novel bio-inspired visual place recognition model based by the part on the fruit fly brain and combined with a compact continuous attractor neural network. Our proposed system was able to achieve competitive performance compared to benchmark systems that have much larger network storage and compute footprints. It was also, to the best of our knowledge, the furthest in capability an insect-based place recognition system has been pushed with respect to demonstrating real-world appearance-invariant without resorting to full deep learning architectures.

There are a number of promising avenues for future research. The first is the untapped source of architecture inspiration that can be drawn from further study of insect brains. Insects face tremendous pressure to minimize neural costs for metabolic reasons [48], [49]; they have to have the most efficient brains possible [50]. For example, there is a small recurrent pathway in the honey bee mushroom body called the protocerebral calycal tract. It is implicated in sharpening representation in the mushroom bodies as well as performing complex classification tasks [51], both capabilities that might enhance the versatility and utility of the system described here.

Future research bridging the divide between well-characterized insect neural circuits [59], [60] as well as recent deep neural network architectures and computational models of network dynamics related to spatial memory and navigation [61], [62] are likely to yield further performance and capability improvements, and may also shed new light of the functional purposes of these neural systems.

Acknowledgment

The authors would like to thanks Jake Bruce currently at Google DeepMind for helpful discussions about the potential ways to implement the FlyNet multi-frame baselines which enabled to develop the FlyNet+RNN network, and also thanks to Stephen Hausler for helping to configure his recent work Multi-Process Fusion (MPF) to perform our state-of-the-art comparison.

References

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," in *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton. "Deep Learning." *Nature*, vol. 521, pp. 436–444, Mar. 2015.

[3] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional Neural Network-based Place Recognition," in *Australasian Conference on Robotics and Automation (ACRA)*, 2014.

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

[5] 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 (ICRA)*, vol. 3, no. 4, pp. 4015–4022, Oct 2018.

[6] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," *Proceedings of Robotics: Science and Systems XIV*, 2015.

[7] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the Performance of ConvNet Features for Place Recognition," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 4297–4304.

[8] Z. Xin, X. Cui, J. Zhang, Y. Yang, and Y. Wang, "Real-time visual place recognition based on analyzing distribution of multi-scale CNN landmarks," *Journal of Intelligent & Robotic Systems*, 2018.

[9] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 248–255.

[10] M. I. Jordan. "Attractor dynamics and parallelism in a connectionist sequential machine." In Artificial neural networks, J. Diederich (Ed.). IEEE Press, 1990, Piscataway, NJ, USA, pp. 112–127.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Compututation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[12] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," in *International 1989 Joint Conference on Neural Networks*, 1989, pp. 365–372 vol.2.

[13] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, 2017.

[14] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 403–408, 2004.

[15] M. J. Milford and G. F. Wyeth, "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System," in *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.

[16] A. Cope, C. Sabo, E. Yavuz, K. Gurney, J. Marshall, T. Nowotny, and E. Vasilaki, "The green brain project – Developing a neuromimetic robotic honeybee," in *Biomimetic and Biohybrid Systems*, N. F. Lepora, A. Mura, H. G. Krapp, P. F. M. J. Verschure, and T. J. Prescott, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 362–363.

[17] B. Webb, "Using robots to model animals: a cricket test," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 117–134, 1995.

[18] J. Dupeyroux, J. R. Serres, and S. Viollet, "Antbot: A six-legged walking robot able to home like desert ants in outdoor environments," *Science Robotics*, vol. 4, Feb. 2019.

[19] M. Beekman and F. Ratnieks, "Long-range foraging by the honey-bee, apis mellifera l." *Functional Ecology*, vol. 14, no. 4, pp. 490–496, 2000.

[20] A. B. Barron and J. A. Plath, "The evolution of honey bee dance communication: a mechanistic perspective," *Journal of Experimental Biology*, vol. 220, no. 23, pp. 4339–4346, 2017.

[21] A. Si, M. V. Srinivasan, and S. Zhang, "Honeybee navigation: properties of the visually driven odometer," *Journal of Experimental Biology*, vol. 206, no. 8, pp. 1265–1273, 2003

[22] A. Narendra, S. Gourmaud, and J. Zeil, "Mapping the navigational knowledge of individually foraging ants, *myrmecia croslandi*," *Proceedings. Biological sciences / The Royal Society*, vol. 280, no. 1765, 2013.

[23] J. Degen, A. Kirbach, L. Reiter, K. Lehmann, P. Norton, M. Storms, M. Koblofsky, S. Winter, P. Georgieva, H. Nguyen, H. Chamkhi, U. Greggers, and R. Menzel, "Exploratory behaviour of honeybees during orientation flights," *Animal Behaviour*, vol. 102, pp. 45–57, 2015.

[24] T. Warren, Y. Giraldo, and M. Dickinson, "Celestial navigation in drosophila," *The Journal of experimental biology*, vol. 222, 2019.

[25] T. Ofstad, C. Zuker, and M. Reiser, "Visual place learning in drosophila melanogaster," *Nature*, vol. 474, pp. 204–209, 2011.

[26] J. Plath and A. Barron, "Current progress in understanding the functions of the insect central complex," *Current Opinion in Insect Science*, vol. 12, pp. 11–18, 2015.

[27] J. Plath, B. Entler, N. Kirkerud, U. Schlegel, C. Galizia, and A. Barron, "Different roles for honey bee mushroom bodies and central complex in visual learning of colored lights in an aversive conditioning assay," *Frontiers in Behavioral Neuroscience*, vol. 11, 2017.

[28] K. Pfeiffer and U. Homberg, "Organization and functional roles of the central complex in the insect brain," *Annual Review of Entomology*, vol. 59, pp. 165–184, 2014. 2013.

[29] R. Huerta, "Learning pattern recognition and decision making in the insect brain," *AIP Conference Proceedings*, vol. 1510, 2013, pp. 101–119.

[30] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

[31] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-Scale Image Retrieval with Attentive Deep Local Features," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3476–3485.

[32] A. Torii, R. Arandjelovix0107, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 Place Recognition by View Synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 257–271, 2015.

[33] S. Garg, N. Süenderhauf, and M. Milford, "LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics," *Proceedings of Robotics: Science and Systems XIV*, 2018.

[34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[36] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran, (2018). "Vector-based navigation using grid-like representations in artificial agents". *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[37] C. J. Cueva, and X.-X. Wei. "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization." ArXiv abs/1803.07770 (2018).

[38] E. I. Moser, E. Kropff, and M.-B. Moser, "Place Cells, Grid Cells, and the Brains Spatial Representation System," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 69–89, 2008.

[39] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the cognitive map," *Nature Reviews Neuroscience*, vol. 7, pp. 663–678, 2006.

[40] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[41] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI Press, 2014, pp. 2564–2570.

[42] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[43] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 353–361.

[44] P. Stratton, M. Milford, G. Wyeth, and J. Wiles, "Using strategic movement to calibrate a neural compass: A spiking network for tracking head direction in rats and robots," *PLOS ONE*, vol. 6, no. 10, pp. 1–15, 2011.

[45] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[46] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, abs/1412.6980, 2014.

[48] S. Laughlin, "Energy as a constraint on the coding and processing of sensory information," *Current Opinion in Neurobiology*, vol. 11, no. 4, pp. 475–480, 2001.

[49] S. Laughlin, R. De Ruyter Van Steveninck, and J. Anderson, "The metabolic cost of neural information," *Nature Neuroscience*, vol. 1, no. 1, pp. 36–41, 1998.

[50] L. Chittka and J. Niven, "Are bigger brains better?" *Current Biology*, vol. 19, no. 21, pp. R995–R1008, 2009.

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

[52] K. Simonyan, and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *CoRR* abs/1409.1556, 2015.

[53] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770–778.

[54] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[55] G. Lin, et al. "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation." *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[56] S. Hausler, A. Jacobson and M. Milford, "Multi-Process Fusion: Visual Place Recognition Using Multiple Image Processing Methods," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1924–1931, April 2019.

[57] Zhou, Bolei et al. "Places: A 10 Million Image Database for Scene Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018): 1452–1464.

[58] J. Mao, X. Hu, X. He, L. Zhang, L. Wu and M. J. Milford, "Learning to Fuse Multiscale Features for Visual Place Recognition," in *IEEE Access*, vol. 7, pp. 5723-5735, 2019.

[59] L. Hernandez-Nunez, J. Belina, M. Klein, G. Si, L. Claus, J.R. Carlson, A.D. Samuel, "Reverse-correlation analysis of navigation dynamics in Drosophila larva using optogenetics." *eLife*, 4 (2015), p. e06225.

[60] M. E. Berck, A. Khandelwal, L. Claus, L. Hernandez-Nunez, G. Si, C.J. Tabone, F. Li, J. W. Truman, R. D. Fetter, M. Louis, A. Samuel, A. Cardona, "The wiring diagram of a glomerular olfactory system." *eLife* 5 (2016), p. e14859.

[61] M. G. Campbell, S. A. Ocko, C. S. Mallory, I. I. C. Low, S. Ganguli, L. M. Giocomo, "Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation." *Nature Neuroscience* (2018) 21:10961106.

[62] L. M. Giocomo, "Spatial Representation: Maps of Fragmented Space," *Current Biology*, Volume 25, Issue 9, 2015, pp. R362–R363.