Visual Place Recognition Using Brain Inspired Neural Architectures

Ryan Yeh
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York 14623-5603
Email: rxy8548@rit.edu

Udit Sharma
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York 14623-5603
Email: us2848@rit.edu

Divyansh Gupta
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York 14623-5603
Email: dg9679@rit.edu

***Abstract*—Using the paper *A Hybrid Compact Neural Architecture for Visual Place Recognition* [1], the authors replicated the main neural architecture *FlyNet*, a neural network inspired by the brain of a fruit fly, as well as two of the tests within the aforementioned paper. These two tests comprise of testing the accuracy of *FlyNet* by manipulating various hyperparameters and testing the performance of *FlyNet* in a hybrid model with a Continuous Attractor Neural Network (CANN). These tests were a success, as the behavior presented by the network matched that of the original paper. Through these experiments, a updated version of *FlyNet* was developed which shows improved performance from the original paper. This was compared to the performance of other network types such as Convolutional Neural Networks (CNNs) and Fully Connected (FC) networks with and without dropout.**

1. Introduction

Visual place recognition (VPR) is one of the fundamental elements of many computer vision and robotics applications. It is the process of matching a view of a place with a different view of the same place taken at a different time. It has applications in autonomous driving and long-term mobile robots. The basis of the project discussed throughout this paper, *A Hybrid Compact Neural Architecture for Visual Place Recognition* [1], developed a neural network architecture which performs state-of-the-art VPR. This architecture is a hybrid model of *FlyNet*, a network architecture based on the brains of fruit flies, and a CANN.

The current state-of-the-art neural network models, namely Multi-Process Fusion [2], LoST-X [3] are all outperformed by the hybrid *FlyNet* and CANN model. This model similarly outperforms the algorithm based model SeqSlam [4]. These current state-of-the-art models all perform VPR slower, less accurately, and with a higher computational footprint than the *FlyNet* and CANN hybrid model.

Chancán et al [1] tested the aforementioned current state-of-the-art VPR systems against various *FlyNet* hybrid models. Figure 1 contains the results of testing each network using a subset of the Oxford RobotCar dataset [5], specifically VPR matching between overcast afternoon photographs and nighttime photographs.

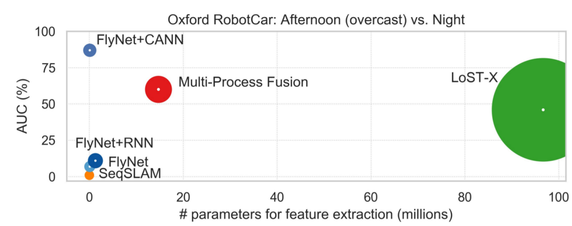

Figure 1. VPR AUC performance versus network size for the Oxford RobotCar dataset [1]

The computational footprint, the relative size of the network, of *FlyNet* is much smaller than any other current state-of-the-art network. *FlyNet* contains three network layers [1] while the network with the next smallest computational footprint is Multi-Process Fusion with thirteen network layers [2]. This is compared to LoST-X which contains two-hundred thirteen network layers within its image processing configuration [3].

The paper is organized as follows. After the introduction, Section 2 explains the techniques and technologies used. Section 3 describes the method developed to achieve original functionality and perform testing. Section 4 presents and discusses experimental results from the project. Finally, Section 5 contains concluding remarks.

2. Background

This section will discuss any technical background required in order to understand the project. This includes a discussion of VPR, temporal modelling, the origins of the brain-inspired *FlyNet* algorithm, and the basics of CANNs.

2.1. Visual Place Recognition

VPR is a very important element of many computer vision and robotics applications. It is the process of matching one view of a specific location with another. This difference in view can be a difference in weather, lighting, or perspective conditions. One of the major hurdles of VPR is performing it quickly, efficiently, and accurately. One major application of VPR is within the control systems of automated cars. Quick, efficient, and accurate VPR is required in this field to ensure the safety of both the passengers within and objects around the car.

There are multiple different methods to achieve functional VPR, with one of the most popular being Convolutional Neural Network (CNN) based solutions. These CNN models, while relatively effective, are limited by their large computational footprint sizes and training requirements. Using more biologically-inspired networks, such as *FlyNet*, help alleviate these issues [1].

2.2. Time-Based Models

Exploring and using temporal relationships has been an effective measure to increase the performance of VPR solutions, making it easier to maintain accuracy in both navigation and localization. This is achieved traditionally through recurrent neural networks (RNNs) such as long short-term memory (LSTM) networks [1]. Further development of this field has resulted in convergent evolution with biological systems, as artificial neural networks have begun resembling naturally occurring networks. This has led to further research into the brains of various species such as rats [6] and insects [7] in a attempt to optimize these systems, increase their effectiveness, and reduce their computational footprints.

2.3. Fruit Fly Network Algorithm

FlyNet is inspired by the olfactory neural circuit of a fruit fly from the Drosophila family. Research indicates the this neural circuit identifies odors by assigning neural activity patterns to similar input odors. In order to identify odors, the neural circuit first normalizes the input signal. This removes any bias induced from an odors concentration. This first layer is then connected to the second layer through a sparsely populated, randomly connected, binary matrix. This layer contains a winner-take-all (WTA) circuit of five percent. This means that only the strongest five percent of neurons are used across the layer. These five percent of neurons are used to generate the specific binary identifier of the inputted odor [7].

2.4. Continuous Attractor Neural Networks

A CANN is a type of neural network that is comprised of a weighted array of neurons with weighted connections. The neurons sum the activities from other neurons based on its weighted connections. This type of network is not feed-forward, but rather recurrent resulting the convergence over time to certain states rather than an external output [6]. CANN weights are not updated during runtime, but are exclusively assigned during network creation [1].

For the CANN described throughout this paper's experimentation, Chancán et al [1] developed a model based on the contents of [8], which prioritizes output filtering to improve VPR results. The implemented CANN uses excitatory and inhibitory connections, coupled with a global inhibitor, to allow for temporal relationships between images to be utilized. The structure of this CANN can be found in Figure 3.

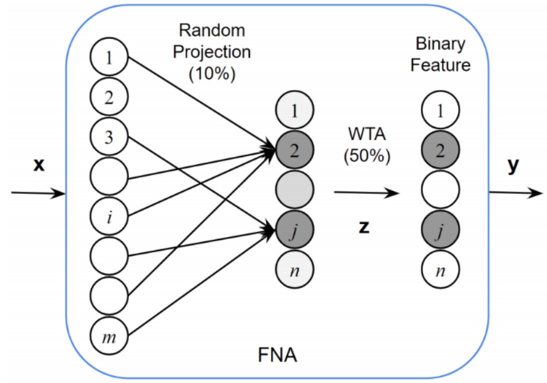

Figure 2. *FlyNet* neural network architecture [1]

3. Methodology

This section will describe the technical approach performed throughout the project. This includes development of the *FlyNet* algorithm, datasets used, and tests performed.

3.1. FlyNet

Chancán et al [1] developed the network *FlyNet* based on the brain of a fruit fly, specifically from the Drosophila family. To simplify the neural network while still achieving all required functionality, only the Drosophila's olfactory neural circuit was used as a basis. As shown in Figure 2 *FlyNet* comprises of a two-layer neural network whose input layer is randomly connected to a hidden WTA layer with a random sampling matrix layer. The input layer comprises of each pixel and its value of the input image [1]. The WTA layer creates a binary feature vector from the output of the random projection matrix layer. This process is accomplished by simply setting the top k% scores to 1 and the rest to 0.

After development of *FlyNet*, Chancán et al [1] tested the network using both the Oxford Robot Car dataset [5] and the Nordland Railway dataset [9]. Hybrid networks comprising of *FlyNet* feeding results into different VPR-related models. These models included SeqSlam [4], a recurrent neural network (RNN), and a CANN with each improving upon the VPR performance of a pure *FlyNet* system. Throughout testing, it was found that the hybrid network comprising of *FlyNet* feeding into a CANN provided state-of-the-art VPR performance. Figure 3 displays the network architecture of the *FlyNet* and CANN hybrid network.

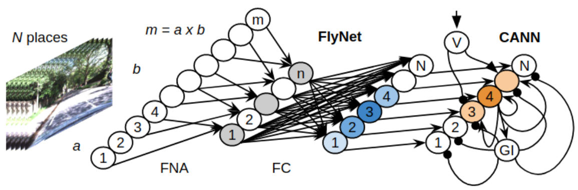

Figure 3. *FlyNet* and CANN hybrid neural network architecture [1]

Figure 4. Example of the differences between seasons at the same location within the Nordland Railcar dataset [9]

3.2. Datasets and Image Preprocessing

Throughout the original paper by Chancán et al [1], two main datasets were used for testing. These datasets are the Oxford Robot Car dataset [5] and the Nordland Railway dataset [9]. These datasets were both developed primarily for the purpose of VPR research.

The Oxford Robot Car data is a set of images from a camera attached to the roof a car traversing a consistent route through Oxford, United Kingdom. This dataset contains over one hundred versions of the route images captured over a period of over a year. These different versions contain differences in lighting, weather, and temporary objects (e.g. pedestrians). Chancán et al [1] used a subset comprising of overcast, autumn images for training and day/night images for testing. The viewpoint changes between each image used was considered moderate by the testers.

The Nordland Railcar data is a set of images from a camera attached to the front of a train in northern Norway. It captures the same train route throughout four different seasons. The main differences between each of the versions of the train route images are the seasons, with the main purpose of the dataset being to test VPR generalization rather than visual appearance. Chancán et al [1] used a subset comprising of summer images for training and the rest of the seasons for testing. The viewpoint changes between each image was considered small by the testers. Figure 4 demonstrates the differences between each set of images with the dataset, showing the same location in summer, fall, and winter.

Due to both the size of the images from the two datasets used throughout testing and the nature of the input layer of *FlyNet*, preprocessing of the images were needed. This preprocessing normalized the sizes of the images (originally 1920 x 1080 for Nordland and 1280 x 960 for Oxford), reduced the computational footprint of *FlyNet*, and decreased the training time of the network. Each image was converted into single-channel gray-scale images with pixel values normalized between zero and one. Each image was also downscaled to a size of 32 x 64.

3.3. Testing

Three tests in total were performed on the *FlyNet* code. Two of these tests were replicated from the original paper by Chancán et al [1] and one test was originally designed by the authors of this paper. These tests were performed to verify the functionality of *FlyNet* and to observe the behavior of various network hyperparameters.

The first test, based on work performed in [1], was testing the accuracy of *FlyNet* when trained through different amounts of epochs. The network was trained on a range of up to three-hundred epochs on the previously described Nordland Railcar subset. Summer season images were used for training and winter/autumn images were used for testing. The accuracy was measured with the results for each season being recorded separately.

The second test was also based on work performed in [1]. The experiment performed consisted of testing the accuracy of the *FlyNet* and CANN hybrid model. As this hybrid model was the greatest performing model described within the original paper, a test including it was performed. The test consisted of testing the AUC of the hybrid model against other *FlyNet* hybrid models. This was performed using the same Nordland Railcar data subset used throughout the first test. Performances of models outside of *FlyNet* alone and the *FlyNet*+CANN model were sourced from the baseline paper.

The third test, a test designed originally by the authors of the paper, consisted of adjusting the random projection ratio from the input layer of *FlyNet* to the hidden WTA layer. The network was tested on a random projection ratio range of one to one-hundred percent with increments of ten percent. This was performed using the same Nordland Railcar data subset used throughout the two previous tests. The accuracy for each season was recorded separately for each random projection configuration.

After testing was performed, the authors explored original methods to improve the performance of *FlyNet*. These changes included further hyperparameter tuning and architectural changes. The performance of this network was then compared to the performances of similar networks, such as Multi-Layered Perceptrons (MLPs) and convolutional neural networks (CNNs).

4. Results and Analysis

The section describes the results of the tests described in Section 3.3. The two tests based on the original paper [1] will have their original results included.

4.1. Hyperparameter Tuning

Figure 5 contains results of the epoch manipulation testing. This includes both the results from [1] and the results from the testing performed for this paper. The performance and behavior shown in both graphs within Figure 5 are very similar. This helps verify the functionality of the *FlyNet* code. The recreated version outperforms the original, but

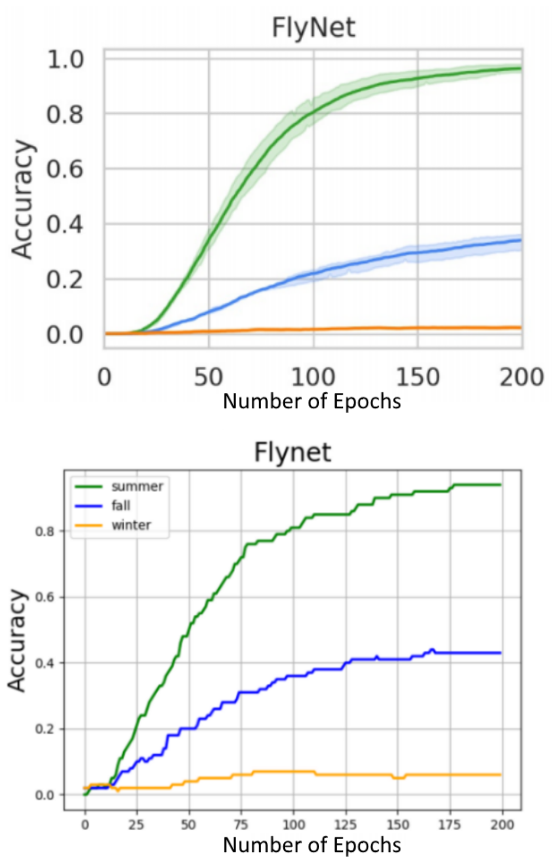

Figure 5. *FlyNet* accuracy versus number of epochs. The original results from [1] (top) and this paper's results (bottom)

this can be contributed to differences in implementation or the inherent randomness of the results and does not necessarily mean that the two are functionally different. Overall two results are similar enough to conclude that *FlyNet* was successfully implemented and the test was successfully recreated from the original paper [1].

It was found through the testing demonstrated in Figure 5, that the accuracy of *FlyNet* plateaus after about 175 epochs. To avoid overfitting and decrease training times, the lowest possible number of epochs which provide the greatest performance should be selected. Based on the results from Figure 5, this ideal training amount would be about 180 epochs.

4.2. *FlyNet* and CANN Hybrid Model Testing

Figure 6 contains the results of *FlyNet* and CANN hybrid network testing based on a Nordland Railcar subset. This figure comprises of both the original results from [1] and the recreated results by the authors of this paper.

While a CANN was successfully developed and it successfully increased the VPR accuracy of *FlyNet*, the CANN was not able to reach the level of accuracy achieved in

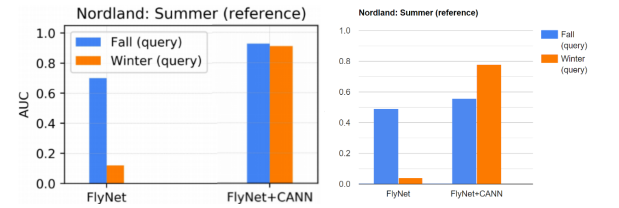

Figure 6. Original *FlyNet*+CANN hybrid model results from [1] (left), recreated results (right).

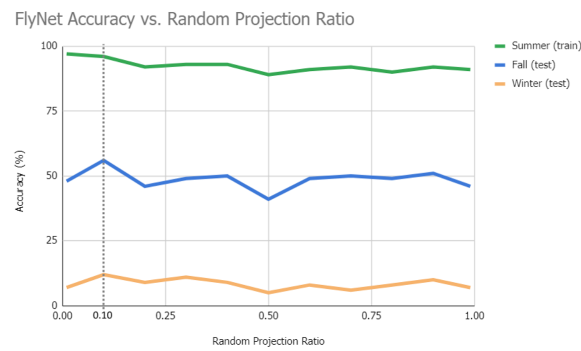

Figure 7. Random projection manipulation results on *FlyNet*

the original paper [1]. There are many reasons as to why this occurred with the main issue being CANN initialization weights. The original paper [1] did not provide any information on the appropriate weights to assign each neuron in CANN. This meant that to recreate the network perfectly, a large amount of testing would have to be carried out. Due to time constraints, this was not possible.

The final results demonstrated in Figure 6 was deemed acceptable as it increased the performance of *FlyNet* especially for the winter test dataset. The main purpose of the hybrid model was to increase network performance. This was achieved, just not to the effect within the original paper [1].

4.3. Random Projection Manipulation Testing

Figure 7 contains the results of random projection manipulation testing of *FlyNet*. Through testing it was found that a ten percent random projection ratio provided the greatest performance. While lower percentages outperformed it with training data, ten percent random projection had the highest accuracy for both testing datasets. This was deemed as more important than training accuracy as it is the more difficult operation to perform and is the entire purpose of VPR.

The ideal random projection ratio being ten percent matches both the original *FlyNet* in [1] and the fruit fly brain algorithm originally explored in [7]. Through both testing and evolution, ten percent was found to be the most effective random projection ratio. This helps validate the efficacy of using brain inspired networks to solve problems and the effectiveness of evolution as the fruit fly naturally evolved to use these ideal conditions.

4.4. Improved *FlyNet* Architecture

Although the *FlyNet* implementation in the original paper claimed to achieve 96% accuracy on the Nordland summer images dataset, the same was verified by our implementation of the *FlyNet* algorithm achieving similar results. We introduced a new FC layer of 1000 neurons in between the FNA and FC layer of the original Flynet network to improve the accuracy.

Surprisingly, the new architecture improved the accuracy by 2% resulting in 98% accuracy on the Nordland summer images dataset. Also improving the test dataset accuracy by a significant amount. These results show that by smartly changing the architecture and tuning the hyper-parameters, we can improve the state-of-the-art results of an algorithm and encouraged us to try new hybrid *FlyNet* architecture in future for more accurate VPR.

We also compared the single-frame *FlyNet* network performance with several neural networks including the MLPs (with and without Dropout) and CNN's. The MLP network was tested for Dropout values of (0.25, 0.50, 0.75), with the pre-trained VGG16 network being utilized for the CNN. The results obtained from this testing were not at par with the *FlyNet* network. Even after using only 64k parameters in the *FlyNet* network, better training and test accuracies were measured. This is compared to MLP and CNN networks having more than 200K parameters.

5. Conclusion

The purpose of the project is to recreate the results *Hybrid Compact Neural Architecture for Visual Place Recognition* [1] and perform three tests upon it. Two of these tests must be from the aforementioned paper and one test must be original. The *FlyNet* neural network was successfully replicated, further improved by a replicated CANN. Two tests from the paper, specifically hyperparameter tuning and hybrid network testing, were successfully replicated with results closely matching the results in the original paper. The original test, random projection ratio manipulation, was successfully performed on *FlyNet* with the resulting behavior being observed. Additionally, improvements were made to the original *FlyNet* architecture resulting in improved network performance. Overall, all objectives of the project were achieved, as demonstrated by the results discussed in this paper.

5.1. Future work

In the original paper, Chancán et al [1] demonstrate the power of using biologically inspired neural networks in VPR related applications. Future work could include further refinement of *FlyNet* itself, or the development of another neural network inspired by another animal's brain. This new network could use a more complex neural circuit to achieve even greater results than what was achieved in the original paper.

More *FlyNet*-inspired hybrid networks could be explored in the future, as the amount of hybrid networks experimented on in the original paper is limited to three different networks. As performance was consistently gained through network hybridization, further research of this kind could yield great results.

Future work related directly to this paper, rather than the original [1], would include further refinements to the developed CANN. Since the developed CANN did not provide as high of performance gains as the original paper [1], it must be possible to improve the developed CANN implementation. This would be the main subject of further work related exclusively to this paper.

Acknowledgments

The authors would like to thank Dr. Merkel for his assistance throughout the project and the course.

References

[1] 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, April 2020.

[2] S. Hausler, A. Jacobson, and M. Milford, "Multi-process fusion: Visual place recognition using multiple image processing methods," *CoRR*, vol. abs/1903.03305, 2019. [Online]. Available: http://arxiv.org/abs/1903.03305

[3] S. Garg, N. Sünderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," *CoRR*, vol. abs/1804.05526, 2018. [Online]. Available: http://arxiv.org/abs/1804.05526

[4] 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*, 2012, pp. 1643–1649.

[5] 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. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[6] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired slam system," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, 2008.

[7] C. Pehlevan, A. Genkin, and D. B. Chklovskii, "A clustering neural network model of insect olfaction," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 593–600.

[8] P. Miller, "Dynamical systems, attractors, and neural circuits," *F1000Research*, vol. 5, p. 992, 05 2016.

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