An Efficient and Scalable Collection of Fly-Inspired Voting Units for Visual Place Recognition in Changing Environments

Bruno Arcanjo, Bruno Ferrarini, Michael Milford, *Senior Member, IEEE*,
Klaus D. McDonald-Maier, *Senior Member, IEEE*, and Shoaib Ehsan

***Abstract*—State-of-the-art visual place recognition performance is currently being achieved utilizing deep learning based approaches. Despite the recent efforts in designing lightweight convolutional neural network based models, these can still be too expensive for the most hardware restricted robot applications. Low-overhead visual place recognition techniques would not only enable platforms equipped with low-end, cheap hardware but also reduce computation on more powerful systems, allowing these resources to be allocated for other navigation tasks. In this work, our goal is to provide an algorithm of extreme compactness and efficiency while achieving state-of-the-art robustness to appearance changes and small point-of-view variations. Our first contribution is DrosoNet, an exceptionally compact model inspired by the odor processing abilities of the fruit fly, Drosophila melanogaster. Our second and main contribution is a voting mechanism that leverages multiple small and efficient classifiers to achieve more robust and consistent visual place recognition compared to a single one. We use DrosoNet as the baseline classifier for the voting mechanism and evaluate our models on five benchmark datasets, assessing moderate to extreme appearance changes and small to moderate viewpoint variations. We then compare the proposed algorithms to state-of-the-art methods, both in terms ofarea under the precision-recall curve results and computational efficiency.**

***Index Terms*—Vision-based navigation, localization, bioinspired robot learning.**

I. Introduction

VISUAL place recognition (VPR) refers to the ability of a computer system to determine if it has previously visited a given place using visual information. Performing highly robust and reliable VPR is a key feature for autonomous robotic navigation as Simultaneous Localization and Mapping (SLAM) systems are dependent on loop-closures mechanisms for map correction [1]. While the VPR problem is well-defined, it remains an extremely difficult task to perform reliably as there are a range of challenges that must be dealt with. Firstly, a revisited place can look extremely different from when it was first seen and recorded due to a variety of changing conditions: seasonal changes [2], different viewpoints [3], illumination levels [4], dynamic elements [5] or any combination of these factors. It is also possible for different places to appear identical, especially within the same environment, an error known as perceptual aliasing.

Initially used for difference computer vision tasks, Convolution Neural Network (CNN) based models have been made their way into the VPR field over recent years, achieving impressive performance on a variety of datasets [6]. However, real-time visual place recognition CNN approaches often rely on powerful graphic processing units (GPUs) and large amounts of memory, making them unsuitable for extremely hardware restricted applications [7]. Mobile robotics with resource-constrained hardware are common and these systems cannot afford to run such computationally expensive algorithms [7], [8]. VPR techniques which manage to keep memory usage and computational complexity to a minimum, without compromising performance, are key to enable platforms equipped with low-end hardware. Furthermore, low-overhead VPR algorithms would also benefit systems that are able to run expensive models, freeing resources that can be allocated to other essential functionalities of a robot's navigation. This is the motivation for the recent development of several low-overhead alternatives [9]–[11] and in this work we continue to add on to this literature.

In this paper, we start by presenting a lightweight biological-inspired algorithm dubbed DrosoNet, designed after the brain of drosophila melanogaster [12] and its ability to recognize odors by encoding complex patterns in a small representation tag. DrosoNet features a low model size of 190KiBs and an inference time of around 1 ms, making it both extremely compact and computationally efficient. However, performance is compromised when compared to state-of-the-art models and while robustness to extreme appearance changes and moderate viewpoint shifts is promising, most applications require more reliable VPR.

Our solution to DrosoNet's compromised performance was to utilize multiple of these small models in conjunction,

Manuscript received September 9, 2021; accepted December 17, 2021. Date of publication January 6, 2022; date of current version January 28, 2022. This letter was recommended for publication by Associate Editor G. Costante and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council under Grants EP/R02572X/1 and EP/P017487/1, and in part by the RICE Project funded by the National Centre for Nuclear Robotics Flexible Partnership Fund. *(Corresponding author: Bruno Arcanjo.)*

Bruno Arcanjo, Bruno Ferrarini, Klaus D. McDonald-Maier, and Shoaib Ehsan are with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: bq17319@essex.ac.uk; bferra@essex.ac.uk; kdm@essex.ac.uk; sehsan@essex.ac.uk).

Michael Milford is with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD 4000, Australia (e-mail: michael.milford@qut.edu.au).

Digital Object Identifier 10.1109/LRA.2022.3140827

made possible by the low memory size algorithm features. Furthermore, there is intrinsic randomness to DrosoNet's training process, both in the model's initialization and in it's fully connected layer, which results in variation of its predictions in deployment. The key observation is that one DrosoNet might perform sub-optimally with one image while other DrosoNets actually output a correct prediction. Exploiting both of these features and the image-sequential nature of the SLAM environment, we propose a voting mechanism that takes the outputs of multiple trained DrosoNets and combines them to perform more reliable VPR. While this results in a larger and more complex algorithm, its inference time of 18 ms and memory size of 6 MBs is still substantially inferior to many state-of-the-art approaches such as NetVLAD [13]. Moreover, the developed voting system is not exclusive to DrosoNet as a baseline and can be utilized with any classifier-type algorithm that works on sequential imagery. Ideally, the baseline model should be compact, as multiple will be used, and present some degree of variation in its predictions for the voting process to take advantage of.

The remainder of this paper is structured as follows. Section II gives an overview of related work in the VPR field. DrosoNet and the proposed voting algorithm are presented in detail in Section III. The experimental setup and evaluation criteria are explained in Section IV. Results are displayed and discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

Many approaches have been explored as possible solutions to the visual place recognition problem. Handcrafted techniques such as Speeded-up Robust Features (SURF) [14] and Scale-Invariant Feature Transform (SIFT) [15] retrieve local features for image matching and have been widely used for VPR [8], [16], [17]. To complement these feature extractors, image retrieval algorithms such as Bag-of-Words (BoW) have gotten a lot of attention as the process of searching the previously built feature map for a match must be efficient to be performed in real time. The focus on robotics with constricted hardware led to more compact BoW implementations employing binary BoW representations [18]. Nevertheless, the identification and extraction of descriptive and repeatable features in an image is an incredibly complicated task. Furthermore, these models usually do not face off well against large appearance changes that are bound to occur with long term robotic operations. In contrast to local feature extractors, whole-image descriptors such as Histogram of Oriented Gradients (HOG) [19] and GIST [20] describe an entire image at once and are used for VPR in [21] and [22]. Pre-trained CNNs can be used out of the box as a whole-image descriptor. Hou *et al.* [23] used a pre-trained AlexNet [24] model as a feature extractor for loop closure detection showing that *conv3* features are more robust to appearance changes while *pool5* and *fc6* features are more robust to viewpoint changes. These findings where used to provide SeqSLAM [25] with viewpoint tolerance [26]. HybridNet, ASMOSNet [27], Region-VLAD [28] and NetVLAD [13] were proposed specifically for performing visual place recognition by utilizing CNN extracted features. Indeed, convolutional network based models have achieved impressive VPR performance results in recent years [29].

However, convolutional neural networks demand high memory and computational resources. In order to increase prediction accuracy, these networks have become deeper and more complex [11], resulting in high memory and computational power needed to run these algorithms online, making them unsuitable for platforms equipped with low-end hardware, which is often the case in mobile robotic applications [7]. Recently, the interest in low-overhead CNN based algorithms has led to the development of several compact and efficient techniques. MobileNets [11] uses depth-wise separable convolutions to decrease model size and achieve inference times of 113 ms on mobile CPUs. MobileNetsv2 [30] builds on top of the previous version by introducing a novel inverted residual and linear bottleneck layer module, decreasing inference time to 75 ms while offering better accuracy results on ImageNet classification. Developed specifically for scene recognition, BiMobileNet [31] combines the MobileNetsv2 architecture with feature fusion in the bilinear model [32] resulting in impressive accuracy across multiple datasets with small memory footprints. Another drawback of CNN models is the need for large volumes of labeled data which is particularly problematic for the VPR field, as a frame-to-frame correspondence is required. CALC [33] is proposed as an efficient unsupervised deep neural network model which was shown to perform real-time reliable VPR. While not a convolutional neural network itself, CoHOG [34] is a train-free and computationally efficient VPR algorithm when compared to CNNs. CoHOG detects entropy rich regions in an image [35] that are subsequently assigned with HOG descriptors.

Biological inspired algorithms are yet another approach to deal with the VPR problem in highly restricted platforms. Small animals are able to perform complex navigation tasks, such as localization, with neural activations [36], [37] which are simple and elegant when compared to artificial deep neural networks. Motivated by this observation, bio-models have been developed for general navigation [38] and VPR [39]. Of particular interest for this work is the research conducted to understand the fruit fly's ability to navigate [40] and process odors [39] by encoding complex patterns into compact representations, which inspired efficient lightweight algorithms for the VPR problem [9].

In this paper, we present two novel algorithms designed for hardware constrained platforms. The first model, dubbed DrosoNet, is a lightweight neural network architecture inspired by the brain of the fruit fly processing of odors. It employs 8-bit quantization [41] to achieve extremely small memory sizes. We then build on top of DrosoNet, exploiting its compact and randomness properties by utilizing multiple DrosoNets in an ensemble, whose output is combined with an underlying voting mechanism inspired by Multi-Process Fusion [42]. With our suggested hyperparameters, the ensemble features a model size of 6 MBs and inference time of 18 ms, while maintaining good VPR performance.

III. Methodology

We present two novel lightweight VPR algorithms as our contributions in this work: DrosoNet and the voting mechanism that builds on top of it. DrosoNet works by computing a low-memory representation of a given image. It is then trained as a

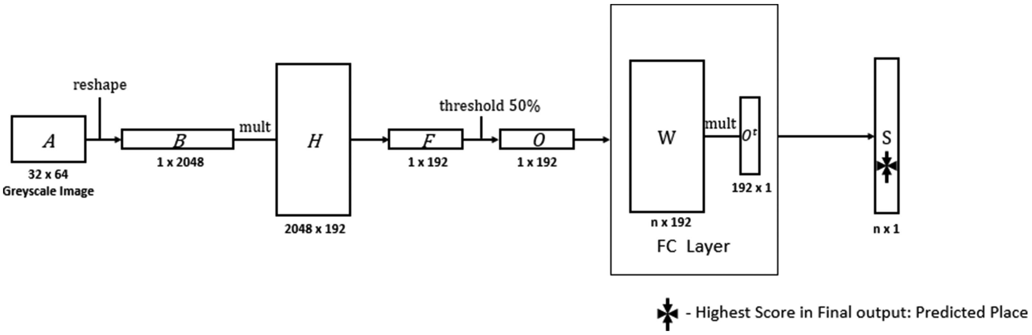


Fig. 1. DrosoNet implementation diagram. This process is repeated for each input image (each image corresponds to a place, n places). A is the input image;B is the input image reshaped to a row vector; H is a sparse binary matrix where each column has only 10% of its values set to 1, the remaining to 0; F contains the 192 activation values; O is a binary image representation resulting from the threshold of the values of F; W is the weight matrix of the fully connected classifier; S is the final output of the DrosoNet, where each value corresponds to a place score. After training takes place, DrosoNet is dynamically quantized to 8-bit precision, further reducing its memory usage.

classifier, where each place is a different class, that recognizes these small image tags and associates them with their respective place. The voting method exploits the stochastic nature of the DrosoNet training process, making use of several DrosoNet individuals to perform more accurate and consistent VPR while remaining compact relative to state-of-the-art approaches. The remaining of this section will explore both DrosoNet and the voting algorithm in depth.

A. DrosoNet

DrosoNet is a bio-inspired model that draws inspiration from the fruit fly's brain circuits. The brain of these small insects is extremely efficient at recognizing different odors, especially when considering its size. While the VPR problem deals with visual information, the algorithm attempts to use a simplified version of the information processing that the fruit fly's brain uses for odor recognition.

A Computer Science focused implementation of the odor recognising process was proposed in [39], presenting three main steps. Firstly, a sort of normalization occurs, centering the mean of the activation rates of the flies' neurons for all odors. Secondly, around 10% of the neurons that respond to an odor are evaluated and their activation rate is summed up. Finally, 5% of the summed up values are used to create a binary representation of the given odor - this compact representation is then used to compare and recognize odors.

The proposed DrosoNet algorithm makes use of the fly's schema to encode a compact image representation that is then fed to a fully connected layer for classification. The process aims at a low memory footprint by utilizing small image input sizes and other hyperparameters are chosen according to empirical data. Fig. 1 shows the operations that occur in the DrosoNet algorithm, displaying the dimensions of each matrix. The image is stretched into a row vector of size 1 × 2048. It is then multiplied by the matrix H, H is binary and sparse, with 10% of the elements of each column randomly set to 1 and the remaining being 0 on the DrosoNet instantiating. This results in 10% random pixels of the input image being taken into account when calculating the activation values, stored in F. The number of columns in H corresponds to the number of activations used, we set this value to 192. The top 50% higher values in F are then set to 1 while the lower 50% are set to 0, resulting in a binary representation for the input image, matrix O. O is then fed into a fully connected layer where the learning process takes place. The fully connected layer works as a classifier to predict the current place fromthevector O, hence including exactly one neuron per map's location. The highest value among the n elements of the output vector is regarded as the matching location. Finally, after the model is trained, we reduce the parameters' precision down to 8-bit integers in a process named dynamic quantization, further reducing the memory size of DrosoNet.

B. Voting Mechanism

Our proposed voting mechanism, illustrated in the diagram of Fig. 2, combines the output of several DrosoNets to perform more effective and consistent VPR. In practice, the random initialization of DrosoNet's H matrix as well as the stochastic nature of training the fully connected layer means that one particular DrosoNet might have a poor prediction for a given place while most other DrosoNets actually output an acceptable prediction. The voting exploits this observation and does not rely on any single DrosoNet to cast its prediction. Instead, it selects the prediction that most models agree on, following some specified ruling. The remaining of this section expands on our proposed voting mechanism.

We start by training a number of DrosoNets on the same training dataset and storing these trained models in a collection. We then run these models with the test dataset, obtaining the score vector and hence the predicted place for each DrosoNet.

We now detail how the voting model works with n trained DrosoNets to perform a prediction. The voting method takes into account the highest score given by the individual (the predicted place) as well as all the scores within a range around that predicted index. The selection of a score for a single DrosoNet, d, can be represented as

$$v_i^d = \begin{cases} s_i^d & \text{if } l^d \leq i \leq u^d \\ 0 & \text{else} \end{cases} \tag{1}$$

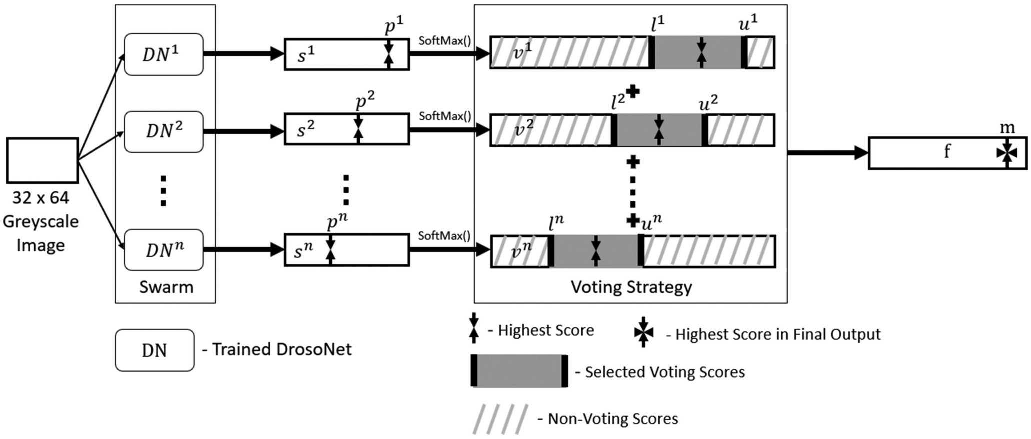


Fig. 2. Voting mechanism diagram, displaying the combination of several DrosoNets' outputs using our proposed voting method.

where s^d is the complete vector score given by the d^{ith} DrosoNet after being normalized by a soft max function pass, s_i^d denotes the i^{th} score in s^d, v_i^d is the i^{th} score that is either copied from s^d or set to 0 and stored in vector v^d. l^d and u^d denote for the lower and upper index bounds of the scores to be selected around the highest score d, for the d^{ith} DrosoNet, and are defined as

$$p^d = argmax(s^d) \tag{2}$$

$$l^d = max(0, p^d - r) \tag{3}$$

$$u^d = min(len(s^d) - 1, p^d + r) \tag{4}$$

where p^d is the index of the highest score (hence the place predicted by the d^{ith} DrosoNet), r is a hyperparameter for the chosen range of selection and $len(s)$ is the length of the score vector s^d (also corresponding to the number of places and hence is constant for all n DrosoNets). The max and min functions are used to avoid negative and out-of-bound indexing.

Using the above definitions, we construct the vector v^d for each of the n DrosoNets in the ensemble, where each score is set to either 0 or the corresponding score in s^d. We obtain a single vector score f by summing element-wise over all n vectors v^d

$$f = \sum_{d=1}^{n} v^d \tag{5}$$

Finally, the index m of the highest score in f is selected as the matching place for the input image:

$$m = argmax(f) \tag{6}$$

IV. Experiments

We ran experiments with our proposed models as well as withseveral other algorithms: CALC [33], HOG [19], CoHOG [34], FlyNet [9], HybridNet [27], ASMOSNet [27], NetVLAD [13] and GIST [20]. We use different datasets to assess the models' capacity to deal with different VPR challenges: moderate to extreme appearance changes and small to moderate point-of-view (POV) variations. We note the performance of these algorithms as well as their memory usage and time required to process a single image. The remaining of this section provides details on models' settings, datasets and evaluation metrics.

A. Model Settings

For FlyNet, we use the exact same model settings and architecture as described in [9]. We utilize the implementations provided in [43] for the HOG, CoHOG, CALC, HybridNet, ASMOSNet and NetVLAD models. For the GIST model, we utilize the implementation in [44].

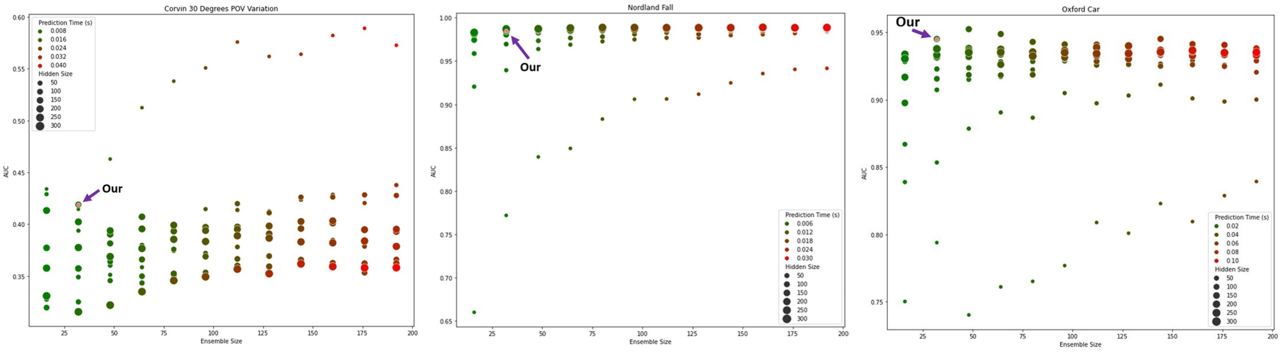
For the stand alone DrosoNet and for DrosoNet in conjunction with the voting mechanism, we conducted a series of ablation studies to find reasonable choices for the number of activations for DrosoNet and the number of models to be used in conjunction with the voting system. Fig. 3 shows the results of these studies in the Corvin (Fig. 3(a)), Nordland Fall (Fig. 3(b)) and Oxford (Fig. 3(c)) datasets. Using this information, we select the hyperparameters that achieve best average performance across all datasets. By this criteria, we select an ensemble size of 32 DrosoNet models, each with a hidden size of 192, indicated by the arrows in the figures. The 50% threshold value was selected after analysing the average performance for several threshold values across all the datasets. Also by experimentation, we select a value for the voting range r equal to 50% of the total number of places in the dataset.

B. Datasets

1) Nordland: The Nordland dataset [45] presents four distinct traversals of a train journey, one per season: Summer, Spring, Fall and Winter. This dataset is used to assess how a model deals with moderate to severe appearance changes. In our experiments, we utilize 1000 images per traversal, with models being trained on the Summer dataset and tested on the Fall and Winter sets, respectively assessing moderate and extreme appearance changes. A match is considered correct if the predicted place falls within 3 frame of the ground-truth image. Thus, for query image q and ground-truth image t, images t-1 to t+1 would be considered correct matches.

2) Gardens Point: The Gardens Point dataset is recorded in the Queensland University of Technology. We utilize two

ABLATION STUDIES



(a) (b) (c)

Fig. 3. Ablation studies to determine best hidden and ensemble size on the Corvin 30 (a), Nordland Fall (b) and Oxford Car (c) datasets.

distinct traversals of 200 images from this dataset to assess model performance on strong point of view variations. Both traversals were captured during the day, with the second being laterally shifted to the right. A match is considered correct if the predicted place falls within 5 frames of the ground-truth image. Thus, for query image q and ground-truth image t, images t-2 to t+2 would be considered correct matches.

3) Oxford RobotCar: The Oxford RobotCar [46] traversals used present challenging illumination changes. We utilize the CrossSeasons subset [47], consisting of 200 sunny query images and 200 dusk reference images. We allow for a 10 frame margin tolerance around the ground truth location [42], [48]. Thus, for query image q and ground-truth image t, images t-10 to t+10 would be considered correct matches.

4) Lagout 15 Degrees POV Variation: The Lagout 15 synthetic dataset consists of aerial footage captured at a 15° angle. It assesses model performance on moderate POV variations with 6 Degrees-Of-Freedom (DOF) movement. We train the models on the Lagout traversal at a 0° angle and test on Lagout 15. We utilize all the images in both sequences and correct matches are considered accordingly to the ground-truth provided by the dataset.

5) Corvin 30 Degrees POV and Scale Variation: The Corvin 30 synthetic dataset was recorded using flight imagery of the Corvin Castle at a 30 degrees angle. It is intended to assess model resilience on strong point-of-view and zoom variations when allowing 6 DOF movements. We train our models using the Corvin traversal at a 0 degrees angle and then test on Corvin 30, utilizing 1000 images per traversal and allowing for a 20 frame margin of error around the ground-truth location frame.

C. Evaluation Metrics

Precision-Recall (PR) curves are often utilized to evaluate the performance of VPR techniques [49]–[51] as they are preferable when dealing with imbalanced datasets. We utilize PR curves and the area under these curves (AUC) to assess the performance of the different models. Precision and recall are given by the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

where TP stands for true positive, FP stands for false positive and FN stands for false negative. In practice, we have a classification problem of two classes: correctly matched and incorrectly matched. For example, for place 500 in the Nordland Fall dataset, images 499, 500 and 501 would be considered correct matches and all the remaining ones as incorrect. For each image query, the models output a vector of scores corresponding to each image in the training set. By interpreting these scores as probabilities, one can vary the certainty threshold at which the model considers the score as the match. Different threshold values yield different recall and precision values, allowing us to plot the recall and precision at each threshold point. The area under this plot, ranging from 0 (worst) to 1 (best), is used to evaluate VPR performance.

Furthermore, we are interested in the size and complexity of each model as we are focusing on developing extremely compact algorithms. To evaluate compactness, we show the memory usages and inference times of the proposed models and common state-of-the-art algorithms.

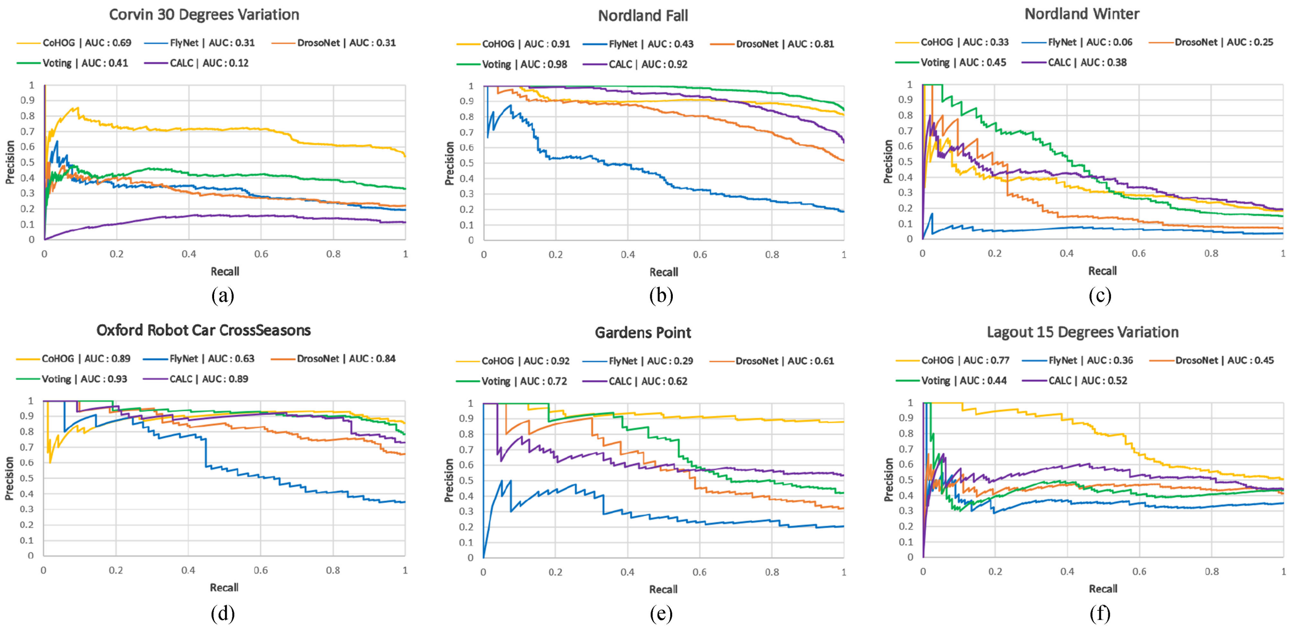
V. RESULTS AND ANALYSIS

In this section, we discuss the results obtained by experiments in both fronts: VPR performance and prediction times together with memory usage of the tested techniques. We also take a look at these metrics for popular state-of-the-art approaches and highlight the compromise between performance and compactness.

A. VPR Performance

We organize our results as follows. We provide precision-recall curves graphs in Fig. 4 for DrosoNet, our voting system and a subsection of the tested models which claim to be lightweight and efficient VPR algorithms: CoHOG, CALC and FlyNet. For comparing VPR performance across all tested techniques and datasets, we provide a bar graph with all the AUC results in Fig. 5 which also includes the performance for NetVLAD, HybridNet, AMOSNet, GIST and HOG in addition to the aforementioned models.

PR CURVES FOR DROSONET, VOTING AND LIGHTWEIGHT MODELS



(a) (b) (c)

(d) (e) (f)

Fig. 4. Precision recall curves with respective AUCs for DrosoNet, voting mechanism and models which present themselves as lightweight.

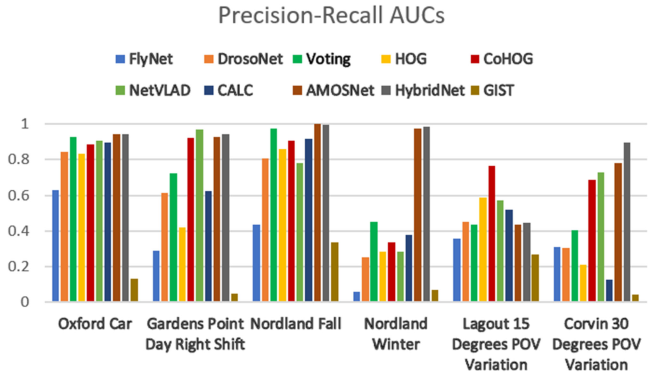


Fig. 5. Precision-recall AUC values for every tested model across all benchmark datasets.

1) Corvin 30 Degrees POV and Scale Variation: In Fig. 4(a) we observe that our voting mechanism is only outperformed by CoHOG among the lightweight models. The Corvin dataset presents extreme point-of-view variations with 6-degrees-of-freedom, making it especially challenging for non-local feaure techniques as is the case of DrosoNet and consequentially our voting mechanism. As expected, all the top performers are region-based techniques: AMOSNet, HybridNet, CoHOG and NetVLAD. Nevertheless, Fig. 5 shows that our voting mechanism outperforms CALC, FlyNet, GIST and HOG. Furthermore, when tuned specifically for this dataset, the voting ensemble is able to achieve better AUC values, as seen in Fig. 3(a).

2) Nordland Fall: From the subset of models shown in Fig. 4(b), our voting mechanism outperforms every other method. When considering all methods in Fig. 5, the voting technique is outperformed only by more computational demanding algorithms such as HybridNet and AMOSNet and only by a small difference of 0.02 AUC value.

3) Nordland Winter: For the more challenging Nordland Winter dataset, our voting mechanism is the top performer out of the lightweight methods, as seen in Fig. 4(c). Once again, the only models that outperform the ensemble method are HybridNet and AMOSNet, albeit with a much larger difference on the AUC metric.

4) Oxford RobotCar: Again a dataset with mainly appearance changes, our voting ensemble outperforms every efficient model in Fig. 4(d). The only techniques that outperform our voting system are HybridNet and AMOSNet but only by a small AUC margin (Fig. 5).

5) Gardens Point: Fig. 4(e) shows that CoHOG is the best performer out of the considered lightweight algorithms, with an AUC value of 0.92. Our voting mechanism comes in second at 0.72. As this dataset presents a lateral POV shift, we once again see in Fig. 5 that region-based techniques achieve best performance.

6) Lagout 15 Degrees POV Variation: We observe in Fig. 4(f) that Lagout 15 is the only dataset where our voting mechanism is outperformed by CALC. However, this dataset is challenging for every technique, including HybridNet and AMOSNet (Fig. 5).

B. Computational Resources

In this section, we focus on two evaluation metrics, inference times with respective frames-per-second rates and memory size, showing how the different tested models compare in relation to their computational efficiency. Table I shows the aforementioned metrics, obtained while running the different models with the Corvin 30 dataset in a Ryzen 7 4000 Series processor. Our choice to utilize a CPU rather than a GPU is motivated by the fact that

TABLE I
PREDICTION TIMES AND MEMORY USAGE COMPARISON

Model	Prediction time (ms)	FPS	Size (MBs)
HybridNet	1143.92	0.87	61.44
AMOSNet	1138.97	0.88	61.44
CoHOG	3627.18	0.28	123.01
CALC	73.62	13.58	4.26
GIST	225.04	4.44	4.53
HOG	208.71	4.79	142.88
NetVLAD	1435.11	0.70	16.38
FlyNet	1.00	1000	0.26
DrosoNet	1.00	1000	0.19
Voting	18.43	54.27	6.19

CPUs are available even in the lowest-end of hardware, when GPUs are usually not present in extremely constrained mobile robots.

DrosoNet outperforms FlyNet in most datasets because of to the increased size of its fully connected layer. Despite the increase in parameters, 8-bit quantization allows DrosoNet to remain more compact than FlyNet, as seen in Table I. The voting mechanism on multiple DrosoNets is the fastest model apart from FlyNet and DrosoNet. It shows a prediction time of 18 ms, 4 times less than the next fastest model CALC, achieving a 54 FPS rate. The models with overall best AUC results were HybridNet and AMOSNET, both having significantly larger model sizes, prediction times and consequently smaller FPS rates. For the Oxford Robot Car and Nordland Fall datasets, our voting mechanism achieves almost identical VPR performance for a fraction of the computational cost. Although CoHOG presents better VPR performance on 6 DOF viewpoint changes than our voting system, it takes 200 times the processing time. For appearance change datasets, voting performs better and faster than CoHOG. NetVLAD is not only slower when compared to the voting ensemble but also performs worse when dealing with appearance changes. However, it does achieve better results with POV variations. For CALC, HOG and GIST, our voting mechanism is faster and it performs better VPR across all tested datasets with the exception of Lagout 15.

VI. Conclusions and Future Work

In this work, two techniques are proposed to address the need for lightweight VPR algorithms for the most hardware restricted of robotic platforms. We first introduce DrosoNet, an extremely compact algorithm inspired by the brain of the fruit fly. Relative to its size, it obtains respectable results in the benchmark datasets, especially when dealing with moderate appearance changes. However, most systems require more robust VPR than what DrosoNet is able to achieve. Our solution is to employ a voting scheme across multiple DrosoNets, exploiting the low memory usage and variation of DrosoNet, utilizing multiple of these small models to achieve competitive VPR performance while remaining compact relatively to CNN based algorithms. When comparing the trade-off between size and performance, the DrosoNet based voting model stands as a compact VPR algorithm with competitive performance, suitable for hardware-restrictive robotic applications.

For further research, one should focus on how to improve the performance of standalone DrosoNet and DrosoNet coupled with voting for extreme POV variation challenges, as the models currently struggle in these settings. Furthermore, investigating how different DrosoNets, or any other baseline voting classifier, complement each other can prove beneficial to select only the most informative subset of models [52].

References

[1] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[2] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014.

[3] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 3829–3836.

[4] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3791–3798.

[5] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, 2007.

[6] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," 2014, *arXiv:1411.1509*.

[7] F. Maffra, Z. Chen, and M. Chli, "Tolerant place recognition combining 2D and 3D information for UAV navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2542–2549.

[8] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. Milford, and K. D. McDonald-Maier, "Visual place recognition for aerial robotics: Exploring accuracy-computation trade-off for local image descriptors," in *Proc. NASA/ESA Conf. Adaptive Hardware Syst.*, 2019, pp. 103–108.

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

[10] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 1 MB model size," 2016, *arXiv:1602.07360*.

[11] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[12] C. Pehlevan, A. Genkin, and D. B. Chklovskii, "A clustering neural network model of insect olfaction," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 593–600.

[13] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[16] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1100–1123, 2011.

[17] A. C. Murillo, J. J. Guerrero, and C. Sagues, "Surf features for efficient robot localization with omnidirectional images," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3901–3907.

[18] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2005, pp. 886–893.

[20] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Prog. Brain Res.*, vol. 155, pp. 23–36, 2006.

[21] C. McManus, B. Upcroft, and P. Newmann, "Scene signatures: Localised and point-less features for localisation," in *Proc. Robot.: Sci. Syst.*, Berkeley, USA, 2014, pp. 1–9.

[22] N. Sünderhauf and P. Protzel, “Brief-gist-closing the loop by simple means,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 1234–1241.

[23] Y. Hou, H. Zhang, and S. Zhou, “Convolutional neural network-based image representation for visual loop closure detection,” in *Proc. IEEE Int. Conf. Inf. Automat.*, 2015, pp. 2238–2245.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[25] M. J. Milford and G. F. Wyeth, “SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 1643–1649.

[26] D. Bai, C. Wang, B. Zhang, X. Yi, and X. Yang, “Sequence searching with CNN features for robust and fast visual place recognition,” *Comput. Graph.*, vol. 70, pp. 270–280, 2018.

[27] Z. Chen *et al.*, “Deep learning features at scale for visual place recognition,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3223–3230.

[28] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, “A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes,” *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 561–569, Apr. 2020.

[29] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. McDonald-Maier, “Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions,” 2019, *arXiv:1903.09107*.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[31] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, “An efficient and lightweight convolutional neural network for remote sensing image scene classification,” *Sensors*, vol. 20, no. 7, 2020, Art. no. 1999.

[32] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1449–1457.

[33] N. Merrill and G. Huang, “Lightweight unsupervised deep loop closure,” 2018, *arXiv:1805.07703*.

[34] M. Zaffar, S. Ehsan, M. Milford, and K. McDonald-Maier, “CoHOG: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1835–1842, Apr. 2020.

[35] M. Zaffar, S. Ehsan, M. Milford, and K. D. McDonald-Maier, “Memorable maps: A framework for re-defining places in visual place recognition,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7355–7369, Dec. 2021.

[36] A. Cope *et al.*, “The green brain project-developing a neuromimetic robotic honeybee,” in *Conference on Biomimetic and Biohybrid Systems*. Berlin, Heidelberg, Germany: Springer, 2013, pp. 362–363.

[37] A. Narendra, S. Gourmaud, and J. Zeil, “Mapping the navigational knowledge of individually foraging ants, myrmecia croslandi,” *Proc. Roy. Soc. B: Biol. Sci.*, vol. 280, no. 1765, 2013, Art. no. 20130683.

[38] M. J. Milford, G. F. Wyeth, and D. Prasser, “RatSLAM: A hippocampal model for simultaneous localization and mapping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, 2004, pp. 403–408.

[39] 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.

[40] T. A. Ofstad, C. S. Zuker, and M. B. Reiser, “Visual place learning in drosophila melanogaster,” *Nature*, vol. 474, no. 7350, pp. 204–207, 2011.

[41] Y. Xu, S. Zhang, Y. Qi, J. Guo, W. Lin, and H. Xiong, “DNQ: Dynamic network quantization,” 2018, *arXiv:1812.02375*.

[42] S. Hausler, A. Jacobson, and M. Milford, “Multi-process fusion: Visual place recognition using multiple image processing methods,” *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1924–1931, Apr. 2019.

[43] M. Zaffar *et al.*, “VPR-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change,” *Int. J. Comput. Vis.*, vol. 129, no. 7, pp. 2136–2174, Jul. 2021.

[44] “GIST MATLAB implementation,” Accessed: Nov. 6, 2021. [Online]. Available: http://people.csail.mit.edu/torralba/code/spatialenvelope/

[45] N. Sünderhauf, P. Neubert, and P. Protzel, “Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons,” in *Proc. Workshop Long-Term Autonomy, IEEE Int. Conf. Robot. Automat.*, 2013, Art. no. 2013.

[46] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The oxford robotcar dataset,” *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[47] M. M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl, “A cross-season correspondence dataset for robust semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9524–9534.

[48] S. Garg, N. Suenderhauf, and M. Milford, “Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics,” 2018, *arXiv:1804.05526*.

[49] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[50] D. M. Powers, “Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[51] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, “Exploring performance bounds of visual place recognition using extended precision,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1688–1695, Apr. 2020.

[52] M. Waheed, M. J. Milford, K. Mcdonald-Maier, and S. Ehsan, “Improving visual place recognition performance by maximising complementarity,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5976–5983, Jul. 2021.