Bag-of-words location retrieval: including position of local features

Cas Sievers
University of Twente
PO Box 217, 7500 AE Enschede
the Netherlands
c.t.sievers@student.utwente.nl

ABSTRACT
Analyzing whether two photos depict the same scene can algorithmically be done by counting the different features of each image and comparing these totals. In doing this, however, information about where in the image each feature was found is discarded. This research investigates possible improvements to using a visual bag-of-words model in automated location retrieval. Two new models for grouping features of an image by their position are proposed and evaluated. Based on the recall rate it is shown that these models can reach a rate of 94%, compared to an 88% rate of the basic bag-of-words implementation. Both models indeed can be applied to improve the performance of bag-of-words based scene recognition. All the code used in this research is available in a public repository at https://github.com/cievers/Location-Retrieval.

Keywords
Bag-of-words, computer vision, location retrieval

1. INTRODUCTION
When describing to another person where you are, one might logically do so by describing the objects one can see. The listener can match the selection of objects against possible known locations, and with enough details, determine the location being described. This describes the bag-of-words model, which can also be used by computers to determine the location of where an image was taken [1].

Simply describing the objects in an image comes with one major issue; there is no information on the positional relation between different objects. One solution might be to evenly divide the image into cells, and pairwise compare each cell to ensure the same objects are found in the same location. Another might be to look at the distinct 'landmarks' in the skyline.

1.1 Motivation
Most research into location retrieval using computer vision focusses on the use of neural networks to represent an image. While these networks show great results, it is much more difficult to understand how they decide what is important in an image [2]. Therefore, more understanding of how to do this task algorithmically is required to fully comprehend the problem and discover new solutions.

While other positioning systems exist, such as GPS, they are not always accurate enough on smaller scales [3]. Gardening robots such as TrimBot [4] are one of the applications that need far more accurate positioning. The TB-Places data set [5] is recorded from the perspective of such a robot and depicts scenes in an outdoor environment. Combining aspects such as varying lighting conditions, a textured environment, a lack of strong geometry, and an overwhelming green colour palette makes this data set one worth investigating.

Since the TB-Places data set is recorded at a different time in an outdoor garden, it includes varying lighting situations. The chosen feature extracting algorithm should, therefore, be robust enough to detect the same features in each scenario. For this reason, the SIFT algorithm is chosen, which can also deal with small changes in viewpoint and texture [6], which are likely to occur due to wind and other outdoor circumstances.

With robust feature detection, a bag-of-words model can be effective at recognizing individual objects. It cannot, however, discern two different scenes composed of the same objects, which can be in a garden environment featuring similar plants. Some positional information of a feature is needed to achieve this.

1.2 Problem Description
Methods counteracting the downside bag-of-words approach, removing contextual data on where a word was found, are investigated.

1.3 Objectives
This research aims to answer the following main question:
Question 1: How can including some positional information of image features improve a bag-of-words model's location retrieval accuracy on the TB-places data set?

To find an answer to this question, it is divided into three different sub-questions.
Question 1.1: How accurate is a bag-of-words model for location retrieval with SIFT for feature extraction on the TB-places data set?
Question 1.2: How can implementing grid-based key point grouping into a bag-of-words model improve location retrieval accuracy on the TB-places data set?
Question 1.3: How can implementing skyline sensitive key point grouping into a bag-of-words model improve location retrieval accuracy on the TB-places data set?

These questions are answered through the implementation of different models and testing their performance on the selected data set. These performances are compared to evaluate whether bag-of-words scene recognition is a viable option and if it can be improved by adding contextual information of image features.

1.4 Background
The field of computer vision encapsulates far more methods of storing, processing, extracting and representing images than can be analyzed in this research [7], and many more applications than just location retrieval. This includes, but is not limited to, object classification, [8] human recognition [9], spatial modelling [10], and camera calibration [11].

Even within the specific application of location retrieval, there are different methods available to achieve this goal. However, since the problem consists of representing and comparing the global structure of an image, much of the work focusses on methods based on neural networks [12]–[15].

The data sets used in these approaches cover a large variety of sceneries, such as indoor environments [11], cities [12], or a little bit of everything [14], [15]. However, these data sets largely have distinct features and geometry, making one scene easily distinguishable from another. There has not been enough research specifically focused on monotone garden environments such as that by María Leyva-Vallina et al. [5], [16].

A task more frequently tackled using a bag-of-words method is that of object recognition. This works effectively with the downside of lacking positional information to be able to locate an object anywhere in an image [17]. Additionally, this also allows for efficient classification of objects within an image [8]. As described by Jia Liu [18], This would suggest that with the specificity of objects that can be detected a bag-of-words approach can also effectively be applied to image retrieval, and thus location retrieval.

While this research makes use of the SIFT algorithm, research into feature extractors is ongoing, developing extractors such as the SURF, boasting large speed improvements at the cost of precision [19], or feasible application on smaller devices such as mobile phones [20].

2. METHODS

2.1 Architecture

The process of using a bag-of-words approach for finding the location from an image and evaluating a full data set is as follows. The first step is to extract the identifying features of the images. The extracted raw feature descriptors are too specific to be compared directly and need to be standardized into a finite set of words which will form the vocabulary to which all features will be mapped. With the vocabulary, an image can be represented as a histogram through the chosen model. Comparing the histograms for all images allows the creation of a comparison matrix, describing the differences between each pair of images. Finally, using the ground truth poses and similarities the performance of the chosen model can be evaluated. This process is also summarized in figure 1 below.

2.2 Feature Extraction

Extracting the features is done using the SIFT algorithm. For each key point found SIFT then uses the 16x16 pixel area around the found key point to calculate a 128-dimensional vector describing that key point [6]. The output of the algorithm is then this list of feature descriptors for each feature. For this research, the Python OpenCV implementation of SIFT is used.

2.3 Vocabulary

Expecting to count the occurrences of an exact feature description vector is highly unlikely, due to the exceedingly large number of unique descriptors. To allow for comparing two image's representations, a fixed size vocabulary is created. A full feature descriptor will be counted as the nearest neighbour within the vocabulary.

Creating this vocabulary is done by training a k-means clustering classifier on a subset of the data set. To train the classifier in a reasonable timeframe, all training data should be in memory at once. However, the size of all feature descriptors in the data set exceeds the available memory. To counteract this, the classifier is trained on ten percent of all features. Since the images are recorded in sequences, each tenth image is selected and has its features extracted for training the classifier. This ensures a diverse selection of scenes is represented in the vocabulary.

The size of the vocabulary also affects how well an image can be represented. A fitting size of the vocabulary is determined by testing several sizes, centred around the estimated value obtained by equation 1 below.

$$vocabulary\ size = \sqrt{n/2}$$

Equation 1. Rule of thumb for estimating cluster count

Here, n is the number of features descriptors, in the data set. The best size is identified by the total squared error of the trained classifier using the elbow method.

2.4 Similarity Comparison

Once a bag-of-words vector has been calculated for a query image, it is compared to those of images in the data set. There are multiple different methods to evaluate the similarity of two vectors. This research tests and compares the performance using Euclidean distance and cosine similarity. Using Euclidean distance, the vector is normalized first to prevent skewed results

2.5 Models

2.5.1 Baseline

The baseline model only extracts features using the SIFT algorithm, with the suggested parameters as found by David Lowe [6]. This creates a list of all key points in the image. With the precomputed vocabulary, these features are directly converted into the bag-of-words representation. The created representations are compared to one another using one of the vector difference functions. The following models follow the same steps but have their own methods of creating and comparing an image representation from the extracted features.

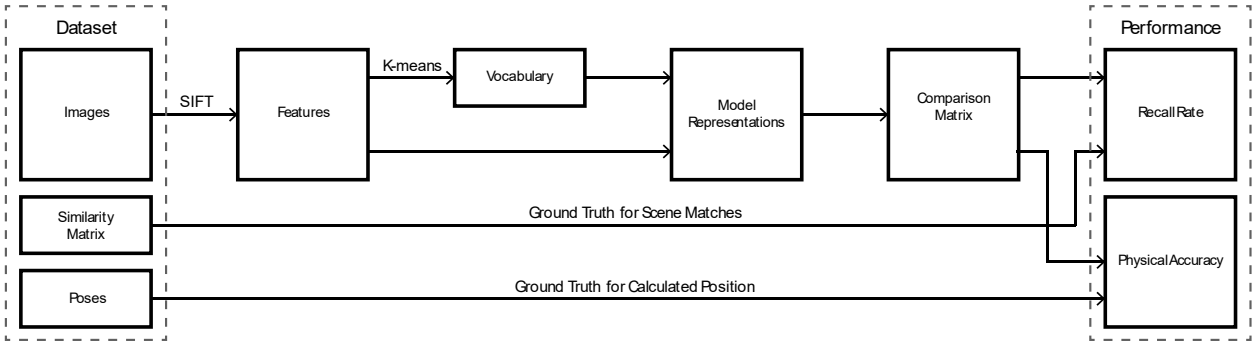
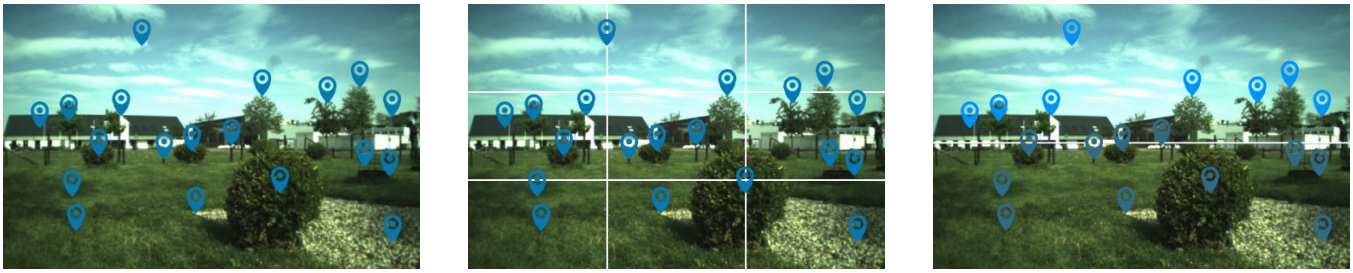


Figure 1. Schematic of the location retrieval process



a) b) c)

Figure 2. Illustration of a) the baseline model, b) the grid model, and c) the skyline model on an image from W17

2.5.2 Grid

One method to add positional information of an image's features is to group them by the part of the image they have been detected in. When the same segmentation is applied to two images, a feature in one section of the query image must then also occur in that same section of an image depicting the same scene. A straightforward segmentation is a rectangular grid, evenly dividing an image into cells.

The grid model is implemented by splitting the image into N×N cells and computing the bag-of-words vector individually for each cell. The total representation of the image is then a concatenation of the representations of all cells. These vectors are compared just like in the baseline model to find the difference between two images.

Since a grid of 1×1 cell is no different from the baseline, the minimum value of N is thus 2. The maximum value of N is dependent on the size of the source images, and the chosen algorithm. The SIFT algorithm describes a feature with an area of 16×16 pixels. Ideally, features on the border between two cells should not be considered as in either cell, but this is beyond the scope of this research. A minimum cell size of 32×32 pixels is chosen to somewhat negate this. In the TB-places data set, each image is 752×480 pixels. Dividing the image height by the minimum cell size gives a maximum value for N of 480 / 32 = 15. Due to memory limitations, however, a maximum value of N = 8 is used.

2.5.3 Skyline

Outdoor scenes such as these found in the TB-Places data set often include the sky. Features disturbing the skyline are highly contrasting, making them easily identifiable for extraction. Key points derived from these landmarks in the sky could prove more reliable than key points closer to the ground.

Similar to the grid model, the image is split into sections, one section above the skyline, and another below. These sections of a query image are again compared to their respective counterparts for an image in the data set. To represent a difference in importance the two resulting differences of these two sections are averaged by weight, in which the skyline can be more or less represented. This combined value is computed by equation 2, where a represents the weight of features in the sky. The value of a is bounded by $0 < a < 1$. For this research, a step size of 0,1 for a is used, and value 0,5 skipped as it is identical to the baseline model.

$$\Delta_{total} = a \times \Delta_{sky} + (1 - a) \times \Delta_{ground}$$

Equation 2. Combining sky and ground difference

A second parameter required in the skyline model is the height of the skyline in the data set. For the TB-Places data sets, the skyline will be defined as the horizon. While the horizon is not clearly visible in each image, measuring its height in various images from the data set resulted in a height of 50% from the top of the image. The actual orientation of the camera at the point of taking the picture was not taken into account and assumed to be facing straight forward. Landing at a skyline height of 50% then matches the intuition; a camera facing directly forward close to the ground will capture about 50% below and 50% above the horizon.

3. EVALUATION

3.1 Data set

The TB-Places data set is used for this research, as it depicts the scenery and environmental conditions TrimBot would be operating in. The data set consists of the public subsets W16, W17, and W18 recorded in Wageningen, and a private set R17 recorded in Renningen. This research will train on the W17 set, and test on both the W17 and W18 sets. Figure 2 visualizes how the different models would apply to an example image from the W17 set.

3.2 Vocabulary Size

Before staring the experiments, the vocabulary size needs to be determined. For this, the total squared error of the trained k-means classifier is measured at various sizes around the value estimated through equation 1. The W17 set contains a total of 19.873.064 features, leading to an estimated vocabulary size of 3.152.

3.3 Testing

The performance of a model on a data set is tested by querying each image in the set against the trained model. The representation of the query image for the chosen model is compared to the representations of all other images in the testing set. The resulting differences are then sorted from most to least similar to find which images in the data set the model determines to be most similar. The data set then provides a ground truth matrix of which images indeed depict the same location as the query image. This matrix is used to evaluate the correctness of the result from a query.

First, the W17 data set is tested against itself to analyse normal performance. Next, images from the W18 data set are queried against the W17 data set to determine how each model performs under changing conditions in the same environment.

3.4 Metrics

Querying each image in a set against the model yields an ordered set of matches. The top K of these matches are then used for evaluation. The performance of a model is defined by the recall rate at K, where a query is considered positive if there is at least one result depicting the same location in the top K matches. In operation, TrimBot would only consider the most similar result, however, to better evaluate the performance, a larger range of values for K have been selected. Additionally, the physical location of the best match can be compared to that of the query image's ground truth to compute the physical accuracy and precision. The physical distance is defined by the Euclidean distance between the computed and true locations.

4. RESULTS

4.1 Vocabulary Size

The measurements of the first experiment are shown in figure 3 below. This shows the total squared error after training a k-means classifier on the W17 data set. From the results of figure 3 below there is no indication of a clear structure in the error of the classifier. Application of the elbow method is not possible, therefore a cluster count of 3.000 will be used for further experiments as a balance between distortion and computation cost.

Figure 3. Error measurements for training k-means classifier on W17 at various cluster counts

4.2 Vector Comparison

To determine the influence of different vector comparison functions two full tests have been done on the W17 set. Table 1 highlights the differences between the Euclidean distance and cosine similarity functions for each of the models. The full measurements can be found in Appendix A. Figures 4 & 5 provide a visual comparison for the grid and skyline models in comparison to the baseline. From these measurements, it is clear that comparing two vectors using cosine similarity yields the best result, and this function will be used for all further experiments.

Table 1. Recall rate at K on W17

Model	Euclidean distance		Cosine similarity	
	K=1	K=5	K=1	K=5
base	40,501%	50,411%	88,445%	96,018%
grid-2	**31,832%**	**38,582%**	**93,844%**	**98,767%**
grid-3	23,886%	29,622%	93,250%	98,666%
skyline-0.2	41,615%	53,215%	79,412%	91,916%
skyline-0.3	44,611%	55,407%	83,522%	94,319%
skyline-0.4	**46,063%**	**56,248%**	86,929%	96,036%
skyline-0.6	43,862%	54,202%	90,309%	97,616%
skyline-0.7	40,829%	51,169%	**91,286%**	98,027%
skyline-0.8	36,125%	47,744%	91,140%	**98,118%**

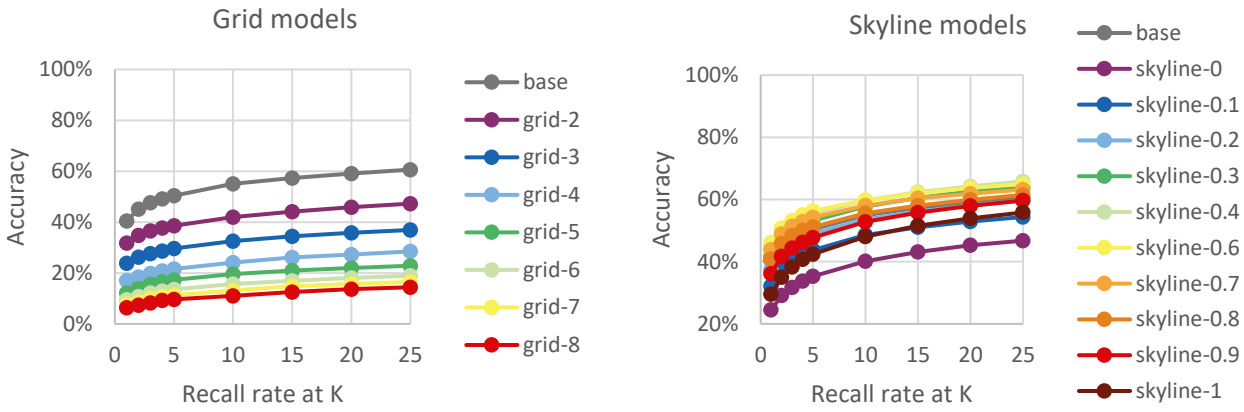


Figure 4. Recall rates at K on W17 using Euclidean distance

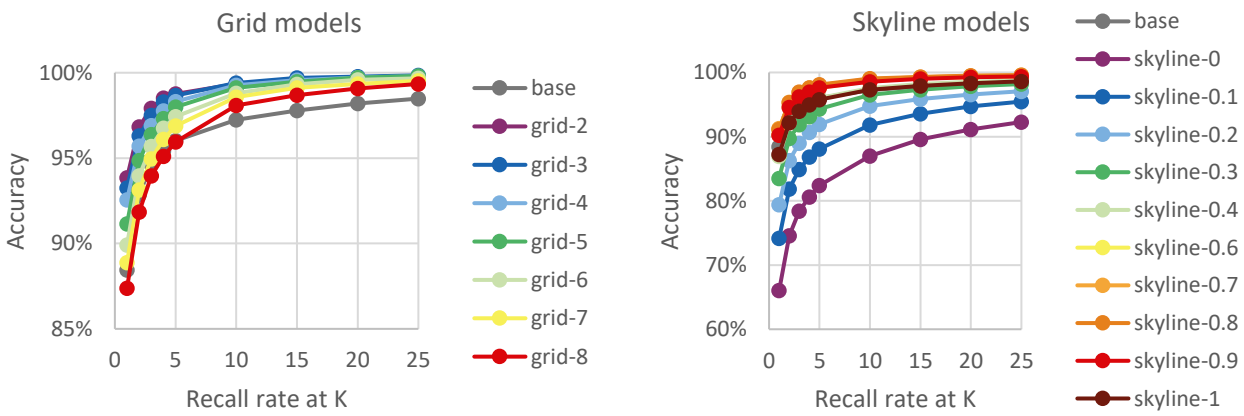


Figure 5. Recall rates at K on W17 using cosine similarity

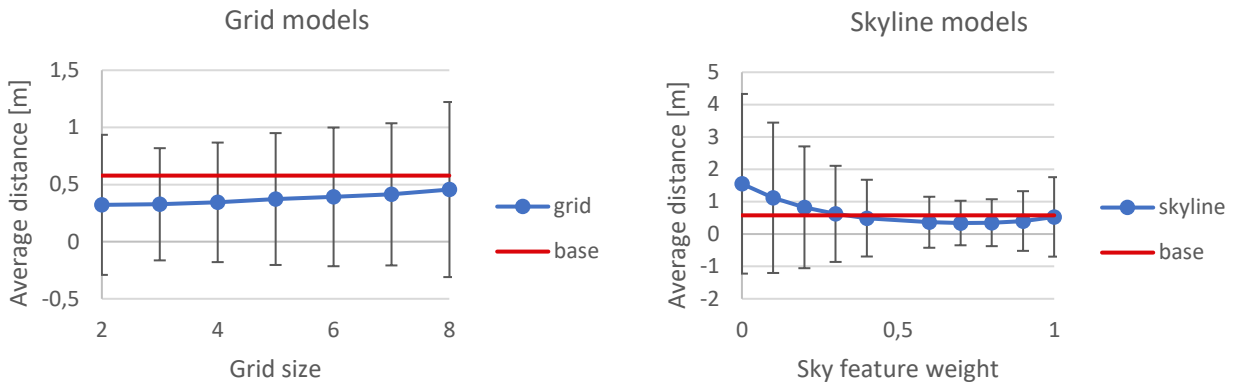


Figure 6. Physical distances of all models on W17

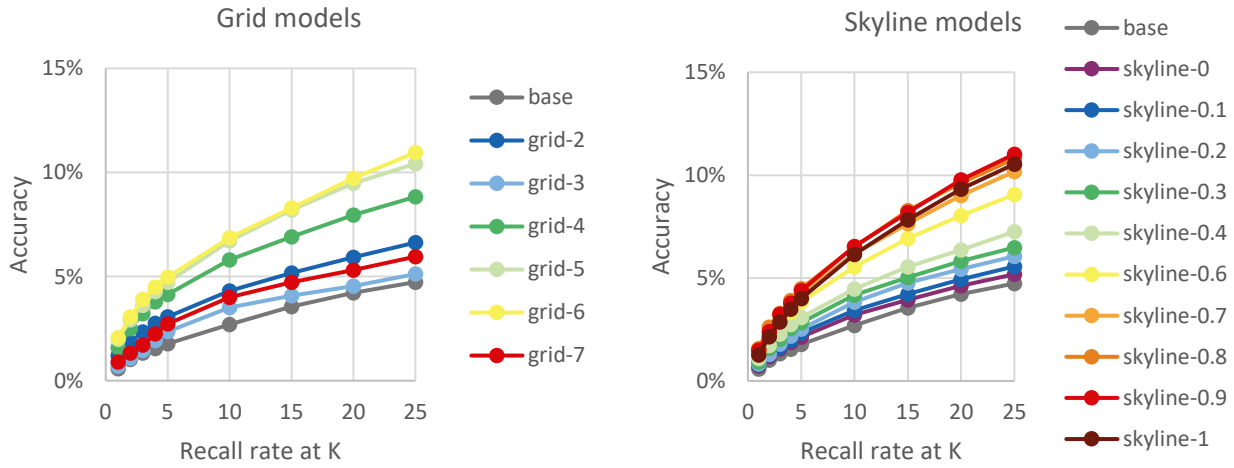


Figure 7. Recall rates at K testing W18 against W17

4.3 Physical distance

Additionally, the Euclidean distance between the location of the computed best match compared to the true location of the query image has been recorded. Figure 6 compares the various parameters of the grid and skyline model against the baseline on the average distance to the true location.

Table 2. Recall rate at K testing W18 on W17

Model	K=1	K=5
base	0,577%	1,784%
grid-2	1,206%	3,073%
grid-3	0,707%	2,365%
grid-4	1,593%	4,144%
grid-5	1,988%	4,279%
grid-6	**2,066%**	**4,496%**
skyline-0.2	0,825%	2,530%
skyline-0.3	0,920%	2,834%
skyline-0.4	1,089%	3,086%
skyline-0.6	1,341%	3,841%
skyline-0.7	1,532%	4,214%
skyline-0.8	**1,580%**	**4,492%**

4.4 Generalization

Finally, the models trained on W17 have been tested with W18 to validate whether they are robust against changes in environmental conditions in the same scenes. Figure 7 shows the performance of each of the models when testing W18. Table 2 shows several of the best performing models in detail. The performance of all models, however, is far below that of the normal W17 tests. The complete measurements for this experiment can be found in Appendix B.

5. DISCUSSION

In comparison to similar research, the performance of the implemented models holds up well. One research applying and testing vocabulary trees on a custom urban environment data set achieves a recall rate just under 70% for the top match and around 78% for the top five matches [21]. Research that also aims for location retrieval introduces a new indoor parking garage data set covering 1200 square meters. When training and testing on this data set, positional accuracy of 0.749 meters is achieved [11]. A most fitting comparison is, however, the work of María Leyva-Vallina et al, applying several convolutional neural networks on the same TB-Places data set [16]. This research presents an average precision of 0.7055 on the W17 set, and 0.2339 when testing the W18 set with networks trained on W17. While not directly comparable to the results of this research's experiments, it shows a better generalization in performance when testing W18 against W17

than this research, which demonstrated high performance on W17, but an abysmal one on W18.

One factor that can play a role in the difference in performance between W17 and W18 is that of image preprocessing. In the experiments, the images have not been preprocessed, and are taken straight from the cameras as TrimBot would have captured them. However, this means that any changes in illumination are not compensated, and have a clear negative effect on the performance. Introducing a preprocessing step to somewhat equalize the lighting could improve tests using W18.

A few more remarkable observations can be made in the measurements of W18. Increasing the grid size no longer decreases the performance, but generally increases it instead. This is with exception of sizes 3 and 7, where the performance dips. However, if it were caused by important features on the borders between cells, a comparable dip could be expected in multiples of these numbers, which is not the case. For the skyline model, all values of *a* now outperform the baseline, and peaks at an even higher value of *a*. This further highlights the importance of large features in the sky, which in the W18 data set are less affected by the different lighting conditions.

6. CONCLUSION

As shown in table 1, the baseline model achieves an 88,445% recall rate within the top match using cosine similarity. Since TrimBot has multiple cameras onboard, simultaneously photographing its environment, a baseline bag-of-words approach is a viable option for scene recognition.

Table 1 also shows that grouping features into evenly distributed cells can improve the performance to reach a 93,844% recall rate for the top match for a 2x2 grid, with the performance slightly decreasing as the number of cells increases. In addition to this, the skyline model has also shown to be a reasonable improvement to the baseline model, peaking at a recall rate of 91,286% for the best match with a sky features weight of 0,7.

The average physical distance, as shown in figure 6, indicates that the best performing parameters are close to determining the location of the robot, but not always close enough to distinguish between standing next to one plant and another. The high standard deviation also indicates that if the best match does not match the true scene, the calculated location is far from the truth.

When testing the generalization of the models, is has been shown that even though the second data set has been recorded in the same garden, the different environmental conditions have shown that the vocabulary extracted from W17 is not applicable for generalization.

Lastly, both models have shown to be capable of outperforming a baseline bag-of-words model. The position of where a feature was found is indeed valuable information when analyzing whether two images depict the same scene. This information is however not enough to make a such a bag-of-words model robust enough against changes in illumination as seen between W17 and W18.

7. REFERENCES

[1] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, “Discovering objects and their location in images,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, vol. I, pp. 370–377, doi: 10.1109/ICCV.2005.77.

[2] C. Olah *et al.*, “The Building Blocks of Interpretability,” *Distill*, vol. 3, no. 3, p. e10, Mar. 2018, doi: 10.23915/distill.00010.

[3] U. S. a Department Of Defense, “Global Positioning System Standard Positioning Service,” *Www.Gps.Gov*, no. September, pp. 1–160, 2008, [Online]. Available: http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[4] “TrimBot2020 Project – Cutting Hedge Research.” http://trimbot2020.webhosting.rug.nl/ (accessed Jun. 15, 2020).

[5] M. Leyva-Vallina, N. Strisciuglio, M. Lopez Antequera, R. Tylecek, M. Blaich, and N. Petkov, “TB-places: A data set for visual place recognition in garden environments,” *IEEE Access*, vol. 7, pp. 52277–52287, 2019, doi: 10.1109/ACCESS.2019.2910150.

[6] D. G. Lowe, “Object recognition from local scale-invariant features,” 1999. doi: 10.1109/iccv.1999.790410.

[7] S. Krig, *Computer vision metrics: Survey, taxonomy, and analysis*, vol. 9781430259. Apress Media LLC, 2014.

[8] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, “Discovering objects and their location in images,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, vol. I, pp. 370–377, doi: 10.1109/ICCV.2005.77.

[9] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000, doi: 10.1109/34.888718.

[10] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, May 2001, doi: 10.1023/A:1011139631724.

[11] J. M. Ciou and E. H. C. Lu, “Indoor positioning using convolution neural network to regress camera pose,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, Jun. 2019, vol. 42, no. 2/W13, pp. 1289–1294, doi: 10.5194/isprs-archives-XLII-2-W13-1289-2019.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018, doi: 10.1109/TPAMI.2017.2711011.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.

[14] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, and Á. García-Martín, “Semantic-aware scene recognition,” *Pattern Recognition*, vol. 102, Jun. 2020, doi: 10.1016/j.patcog.2020.107256.

[15] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez, “Appearance-invariant place recognition by discriminatively training a

convolutional neural network," *Pattern Recognition Letters*, vol. 92, pp. 89–95, Jun. 2017, doi: 10.1016/j.patrec.2017.04.017.

[16] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Place Recognition in Gardens by Learning Visual Representations: Data Set and Benchmark Analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11678 LNCS, pp. 324–335, 2019, doi: 10.1007/978-3-030-29888-3_26.

[17] J. Farooq, "Object detection and identification using SURF and BoW model," *2016 International Conference on Computing, Electronic and Electrical Engineering, ICE Cube 2016 - Proceedings*, pp. 318–323, 2016, doi: 10.1109/ICECUBE.2016.7495245.

[18] J. Liu, "Image Retrieval based on Bag-of-Words model," Apr. 2013, Accessed: May 01, 2020. [Online]. Available: http://arxiv.org/abs/1304.5168.

[19] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: Speeded up robust features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 3951 LNCS, pp. 404–417, doi: 10.1007/11744023_32.

[20] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008*, pp. 125–134, 2008, doi: 10.1109/ISMAR.2008.4637338.

[21] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, doi: 10.1109/CVPR.2007.383150.

APPENDIX

A. COMPLETE VECTOR COMPARISON MEASUREMENTS

Table A1. Recall rate at K on W17 using Euclidean distance

Model	K=1	K=2	K=3	K=4	K=5	K=10	K=15	K=20	K=25
base	40,501%	45,077%	47,598%	49,132%	50,411%	54,996%	57,335%	59,116%	60,632%
grid-2	**31,832%**	**34,792%**	**36,527%**	**37,742%**	**38,582%**	**41,962%**	**44,072%**	**45,862%**	**47,251%**
grid-3	23,886%	26,206%	27,612%	28,690%	29,622%	32,517%	34,481%	35,787%	36,911%
grid-4	17,053%	18,442%	19,812%	20,771%	21,547%	24,123%	26,151%	27,293%	28,480%
grid-5	12,185%	13,875%	15,564%	16,588%	17,254%	19,547%	20,981%	21,922%	22,853%
grid-6	9,454%	10,815%	12,121%	12,934%	13,491%	15,610%	16,935%	18,095%	18,917%
grid-7	7,810%	8,705%	10,047%	10,751%	11,281%	13,190%	14,733%	15,665%	16,524%
grid-8	6,394%	7,326%	8,285%	9,207%	9,609%	11,025%	12,468%	13,637%	14,395%
skyline-0	24,443%	29,074%	31,686%	33,76%	35,267%	40,153%	43,122%	45,278%	46,757%
skyline-0.1	32,125%	37,368%	40,281%	42,108%	43,707%	48,548%	51,023%	52,905%	54,375%
skyline-0.2	37,331%	42,620%	45,543%	47,579%	48,986%	53,590%	56,111%	58,120%	59,664%
skyline-0.3	41,615%	46,977%	49,735%	51,790%	53,215%	57,718%	60,477%	62,523%	63,939%
skyline-0.4	44,611%	49,808%	52,375%	54,083%	55,407%	59,691%	**62,349%**	**64,240%**	**65,802%**
skyline-0.6	**46,063%**	**50,767%**	**53,233%**	**55,051%**	**56,248%**	**59,837%**	62,020%	63,701%	64,971%
skyline-0.7	43,862%	48,895%	51,324%	52,914%	54,202%	58,075%	60,413%	61,874%	63,171%
skyline-0.8	40,829%	45,771%	48,374%	50,027%	51,169%	55,544%	58,075%	59,883%	61,472%
skyline-0.9	36,125%	41,624%	44,355%	46,218%	47,744%	52,740%	55,736%	57,892%	59,655%
skyline-1	29,558%	34,956%	38,281%	40,747%	42,391%	48,064%	51,553%	53,955%	55,828%

Table A2. Recall rate at K on W17 using cosine similarity

Model	K=1	K=2	K=3	K=4	K=5	K=10	K=15	K=20	K=25
base	88,445%	92,857%	94,629%	95,424%	96,018%	97,242%	97,790%	98,191%	98,484%
grid-2	**93,844%**	**96,830%**	**97,917%**	**98,520%**	**98,767%**	99,351%	99,626%	99,726%	99,799%
grid-3	93,250%	96,301%	97,497%	98,283%	98,666%	**99,397%**	**99,689%**	**99,772%**	**99,845%**
grid-4	92,547%	95,698%	96,904%	97,762%	98,310%	99,260%	99,562%	99,726%	99,799%
grid-5	91,158%	94,867%	96,383%	97,324%	97,991%	99,114%	99,470%	99,717%	99,808%
grid-6	89,916%	93,962%	95,689%	96,757%	97,424%	98,794%	99,324%	99,580%	99,671%
grid-7	88,884%	93,140%	94,958%	96,100%	96,885%	98,575%	99,114%	99,342%	99,525%
grid-8	87,386%	91,852%	93,962%	95,113%	95,944%	98,082%	98,685%	99,077%	99,342%
skyline-0	66,049%	74,571%	78,425%	80,599%	82,426%	86,975%	89,578%	91,149%	92,273%
skyline-0.1	74,160%	81,841%	84,874%	86,829%	88,071%	91,834%	93,579%	94,693%	95,460%
skyline-0.2	79,412%	86,271%	88,984%	90,692%	91,916%	94,775%	95,853%	96,575%	97,077%
skyline-0.3	83,522%	89,715%	91,880%	93,140%	94,319%	96,493%	97,333%	97,881%	98,191%
skyline-0.4	86,929%	92,245%	94,264%	95,332%	96,036%	97,579%	98,228%	98,548%	98,758%
skyline-0.6	90,309%	94,666%	96,328%	97,159%	97,616%	98,648%	99,087%	99,260%	99,397%
skyline-0.7	**91,286%**	**95,342%**	96,849%	97,589%	98,027%	98,977%	99,269%	99,470%	99,543%
skyline-0.8	91,140%	95,332%	**96,958%**	**97,616%**	**98,118%**	**99,050%**	**99,324%**	**99,516%**	**99,580%**
skyline-0.9	90,254%	94,593%	96,200%	96,986%	97,634%	98,557%	98,995%	99,260%	99,370%
skyline-1	87,258%	92,181%	93,962%	94,949%	95,753%	97,296%	97,890%	98,292%	98,621%

B. COMPLETE GENERALIZATION MEASUREMENTS

Table B1. Recall rate at K on testing W18 on W17

Model	K=1	K=2	K=3	K=4	K=5	K=10	K=15	K=20	K=25
base	0,577%	1,024%	1,328%	1,545%	1,784%	2,704%	3,554%	4,214%	4,735%
grid-2	1,206%	1,831%	2,348%	2,760%	3,073%	4,318%	5,169%	5,924%	6,631%
grid-3	0,707%	1,081%	1,454%	1,936%	2,365%	3,515%	4,084%	4,539%	5,121%
grid-4	1,593%	2,495%	3,224%	3,784%	4,144%	5,802%	6,909%	7,959%	8,831%
grid-5	1,988%	2,929%	3,619%	4,279%	4,726%	6,709%	8,211%	9,487%	10,420%
grid-6	**2,066%**	**3,073%**	**3,910%**	**4,496%**	**4,978%**	**6,870%**	**8,298%**	**9,721%**	**10,953%**
grid-7	0,903%	1,328%	1,714%	2,257%	2,734%	4,006%	4,726%	5,307%	5,958%
grid-8	Too computationally expensive								
skyline-0	0,733%	1,193%	1,562%	1,866%	2,144%	3,211%	3,936%	4,630%	5,182%
skyline-0.1	0,803%	1,285%	1,645%	1,983%	2,339%	3,424%	4,218%	4,943%	5,564%
skyline-0.2	0,825%	1,293%	1,801%	2,179%	2,530%	3,832%	4,765%	5,429%	6,058%
skyline-0.3	0,920%	1,593%	2,027%	2,552%	2,834%	4,179%	5,030%	5,824%	6,484%
skyline-0.4	1,089%	1,736%	2,278%	2,738%	3,086%	4,474%	5,546%	6,362%	7,256%
skyline-0.6	1,341%	2,200%	2,864%	3,355%	3,841%	5,550%	6,935%	8,033%	9,057%
skyline-0.7	1,532%	2,400%	3,133%	3,676%	4,214%	6,180%	7,651%	9,009%	10,172%
skyline-0.8	**1,580%**	**2,613%**	**3,285%**	**3,910%**	**4,492%**	6,501%	**8,285%**	9,612%	10,797%
skyline-0.9	1,497%	2,404%	3,242%	3,793%	4,392%	**6,540%**	8,193%	**9,777%**	**11,019%**
skyline-1	1,272%	2,152%	2,864%	3,485%	3,997%	6,145%	7,833%	9,330%	10,537%