Pose Invariant Object Recognition Using a Bag of Words Approach

Carlos M. Costa(✉), Armando Sousa, and Germano Veiga

INESC TEC and Faculty of Engineering, University of Porto, Porto, Portugal
{carlos.m.costa,germano.veiga}@inesctec.pt, asousa@fe.up.pt

Abstract. Pose invariant object detection and classification plays a critical role in robust image recognition systems and can be applied in a multitude of applications, ranging from simple monitoring to advanced tracking. This paper analyzes the usage of the Bag of Words model for recognizing objects in different scales, orientations and perspective views within cluttered environments. The recognition system relies on image analysis techniques, such as feature detection, description and clustering along with machine learning classifiers. For pinpointing the location of the target object, it is proposed a multiscale sliding window approach followed by a dynamic thresholding segmentation. The recognition system was tested with several configurations of feature detectors, descriptors and classifiers and achieved an accuracy of 87% when recognizing cars from an annotated dataset with 177 training images and 177 testing images.

Keywords: Object recognition · Image feature analysis · Clustering · Machine learning

1 Introduction

Pose invariant object detection is a critical component in automated systems that require robust detection and classification of classes of objects within cluttered environments. It also plays a pivotal role on extracting information from images by providing the classification of the objects along with their position. Given its generalization properties, this kind of systems can be adapted to a multitude of tasks, and an efficient implementation could be used in real-time applications.

Several approaches were suggested over the years, ranging from the more computationally intensive solutions that compare patches of the image to a database of objects in several poses, to the more efficient techniques that uses classifiers to try to detect several variations of the target object [1–3]. This paper focuses on the later and provides an analysis of the application of the Bag of Words model to object detection and classification.

The system relies on an initial setup phase for training a classifier that later on can be used for recognizing the target objects. It starts by building

A. Ollero et al. (eds.), *ROBOT 2017: Third Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 694, https://doi.org/10.1007/978-3-319-70836-2_13

a visual vocabulary using the feature descriptor clusters of the training images. This vocabulary represents characteristic structures of the target object and will be used as the n-dimensional descriptor space to describe an image. Using this vocabulary, a database of samples is built for training a machine learning classifier. This classifier creates a descriptor model that later on can be employed to detect the target object in test images. After the setup of the recognition system, the detection of the target objects along with their location in the image relies on a sliding window technique [4]. This approach uses the trained classifier to scan the image with windows of different size. In the end, a voting mask with the probable locations of the targets is retrieved and in conjunction with a dynamic thresholding method the locations are extracted. This approach achieved promising results and can be used to recognize objects in different perspective views even if they are within cluttered environments. To allow the fine tuning of the system configuration, several feature detector and descriptors can be selected in conjunction with a range of machine learning classifiers.

In the following section it will be presented a brief overview of other approaches that can be used to perform pose invariant object recognition. Later on, Sect. 3 will provide a detailed description of the implemented system. Then, the results along with the respective analysis will be discussed in Sects. 4, 5 and 6. Finally, Sect. 7 will give a brief set of conclusions and possible future work.

2 Related Work

There are numerous approaches for detecting objects that can appear in several perspective views. Ranging from the very simple template matching to highly advanced systems relying on point cloud perception. The most basic method to perform pose invariant object detection is template matching. In this method, a database of images taken from several points of view is used to scan the test image and try to detect the target object. The problem of this approach is that it requires the image to be scanned with this database in several scales and orientations, which causes it to be very inefficient.

To solve the scale and orientation problem [5], feature detection and description algorithms can be used [6]. In this approach the database is only scanned once. Moreover, since the feature detection describes the image as feature points, the size of data to be compared is reduced drastically, and as a result, it is orders of magnitude more efficient than template matching. In this method, it is critical that the matching of descriptors is filtered in order to remove outliers, using for example a ratio test [5] or a homography computed using Random Sample Consensus. Other approaches suggest the construction of an Implicit Shape Model [3], that takes into consideration the relative position of interesting structures in the target object, in order to build a 3D representation that can then be used to recognize the intended objects. Other methods use image strip features [7] to speed up recognition by focusing in structural parts of the target object or even Haar wavelets and edge orientation histograms [8].

For recognition of specific 3D objects, a more advanced approach using point clouds matching can be used. In this technique, 3D point clouds are matched

using algorithms such as the Iterative Closest Point [9]. Besides recognizing the object, this method also allows the identification of the position of the camera in relation to the target object. However, this approach may not be suitable for general classification of objects, because it was designed to search for a particular 3D geometry. Moreover, it takes considerable computation time to extract 3D point clouds from images, unless the point clouds are retrieved directly from the environment using 3D sensors (such as LIDARs).

After reviewing the existing approaches, it was clear that an efficient and general recognition system should rely on machine learning algorithms in order to be able to successfully recognize the intended category of objects within cluttered environments. One way to implement such system is by employing the bag of words model in conjunction with classifiers. As shown in [10], this approach has promising results and good efficiency.

3 Recognition System

The recognition system is comprised with a setup phase, in which a classifier is trained with samples built with a visual vocabulary, and a recognition phase, in which the classifier is used to identify new instances of the target objects. In the next sections it will be provided a detailed description of the main steps required to successful recognize categories of objects in cluttered environments.

3.1 Preprocessing

To improve the detection of good features and ensure that the system has robust recognition even when the images have considerable noise, a preprocessing step was applied. In a first phase, most of the noise was removed using a bilateral filter. This filter was chosen because it preserves the edges of the image blobs, which are very valuable structures in the detection of feature points. After the noise was reduced, a Contrast Limited Adaptive Histogram Equalization (CLAHE) filter was applied to increase the contrast. This can improve the recognition of the system when the images are taken in low light environments. This technique has better results over the simple histogram equalization because it can be applied to images that have areas with high and low contrast and also limits the spread of the noise. Finally, the brightness was adjusted to correct images that were too dark or too bright.

3.2 Visual Vocabulary

The Bag of Words model [1] had its inception in the document classification realm, but its concepts can be extended to image recognition by treating image features as words. As such, a visual vocabulary must be built from the target objects feature descriptors. In this stage, each image in the vocabulary image list set is preprocessed, and for each ground truth mask of the target objects, it is computed the feature points and their associated descriptors. These extracted

descriptors are then grouped using the k-means clustering algorithm in order to obtain the visual words of the vocabulary. There are several algorithms to select features from images. The supported feature detectors are SIFT, SURF, GFTT, FAST, ORB, BRISK, STAR and MSER. For describing these features there is also several algorithms that aim to be scale and rotation invariant. The supported feature descriptors are SIFT, SURF, FREAK, BRIEF, ORB and BRISK. The matching of these descriptors can be performed using either a brute force or a heuristic approach. In the brute force approach, each descriptor in the image is compared with all descriptors in the reference image to find the best correspondence. In the heuristic approach (that relies on the FLANN library), several optimizations are employed to speed up the computations. These optimizations can be related to the appropriate selection of which descriptors to match, and to the use of efficient data structures to speed up the search (such as k-d trees).

3.3 Training Samples

Before a classifier can be used, it must be trained with several annotated samples of the target objects. As such, a training database was built using the vocabulary of the visual words computed earlier. In this stage, each manually annotated image of the training set list is preprocessed, its feature points are computed and separated into the corresponding classes (target or background) according to the ground truth masks. These manually annotated masks specify if a region belongs to the target objects or the background (shown in Fig. 1 right image as red and black respectively). After having the segmented keypoints, it is computed the associated descriptors and the visual vocabulary is built. The results are a set of normalized histograms of the visual words present in each training image, associated with the corresponding labels, that will inform the classifier which class the training samples belong to (target or background).

3.4 Classifier Training

After having the training samples, a classifier can be trained in order to build a model of the distribution of the target object visual words descriptors. This model can then be used to predict with acceptable accuracy if the target objects are in an image or not. There are several machine learning classifiers to perform object recognition. The included classifiers are Support Vector Machines, Artificial Neural Networks, Normal Bayes Classifiers, Decision Trees, Boosting, Gradient Boosting Trees, Random Trees and Extremely Randomized Trees.

3.5 Object Recognition

For achieving scale invariant object recognition it was implemented a sliding window technique with Regions of Interest (ROI) with several sizes. In this method the image is scanned with ROIs from left to right and from top to

bottom in a column by column and row by row approach. In order to be scale invariant, the image is scanned several times with a increasing ROI size and the ROI movement increment was carefully chosen for ensuring overlapping of successive ROIs. During the image scanning, the trained classifier is used to evaluate if the target object is present or not within each image ROI. If the classifier predicts that the object is within the ROI with high confidence, then the voting mask cells within the ROI are incremented. In the end of the image scanning, the voting mask are used in conjunction with a dynamic thresholding method to pinpoint where are the target objects in the image. After having the image binarized into target and background classes, a blob detection algorithm is used to retrieve the bounding boxes of the target regions.

3.6 Evaluation of Results

To evaluate the results of the object recognition system, an image test set was used, in which the computed voting masks were compared with the target objects ground truth masks (in the right side of Fig. 1 is an example of a ground truth mask for the left side image).

In this stage, each pixel in the computed voting masks was compared to the ground truth masks, in order to see if the result was a true positive (correctly detected that there was a target object), true negative (correctly labeled background), false positive (incorrectly labeled background as belonging to a target object) or false negative (missed regions that belonged to the target objects and were labeled as background). With each of these measures acquired for each test image, the accuracy, precision and recall was computed.

To allow fast testing of the system, it was implemented an automatic evaluation module that analyzes all the test images and collects both intermediate and final results.

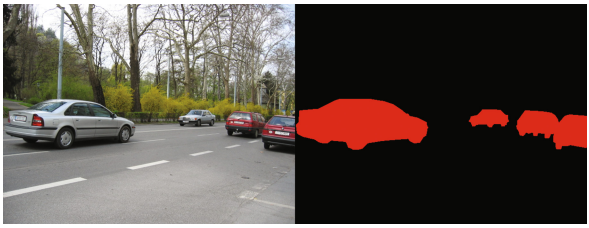

Fig. 1. Example of dataset image (left) and associated ground truth masks (right). The manually annotated red regions represent target objects while the black regions represent background.

4 Methodology

The results were collected using a Clevo P370EM, with an i7-720QM CPU, NVIDIA GeForce GTX 680M GPU and 16 GB of RAM DDR3 (1600 MHz).

It was used the Graz-02 dataset of car images, from which it was retrieved 177 images to build the vocabulary and the training samples, and another 177 images for testing the recognition system.

The visual vocabularies were built with a 1000 word size, and all the intermediate results (vocabulary, training samples, and classifiers) were saved to xml files to speedup future uses of the system.

The OpenCV algorithms were used with the default parameters with the exception of the SVM classifier, in which the maximum number of iterations was set to 100000. Also, the Artificial Neural Networks were configured to have 20 neurons in the intermediate layer. Moreover, for binary descriptors, the FLANN matcher was modified to use the multi probe LSH index search, and the BFMatcher to use Hamming distances.

The sliding window technique used 482 regions of interest per image. These patches start at 20% of the image size, and after each scan of the image, (in which the patch moves at 25% increments of its own size), the patch grows 10% (in relation to the image size).

5 Results

In Figs. 2, 3, 4, 5, 6, 7 and 8 and Tables 1 and 2 are presented some representative results of the recognition of the target objects in several perspective views and in different types of environments. On the right side of the images it is presented the voting masks. These masks start with count 0 (black) and every time a classifier predicts that the target object is present in that ROI, the pixels in the masks are incremented (becoming increasingly brighter in the images). As such, brighter regions indicate that a lot of ROIs were marked as containing the target object,

Fig. 2. Results obtained with STAR detector, SIFT extractor, FLANN matcher and ANN classifier (right image with the voting masks and left image with the overlaid results)

Fig. 3. Results obtained with STAR detector, SURF extractor, FLANN matcher and SVM classifier (right image with the voting masks and left image with the overlaid results)

Fig. 4. Results obtained with STAR detector, FREAK extractor, FLANN matcher and SVM classifier (right image with the voting masks and left image with the overlaid results)

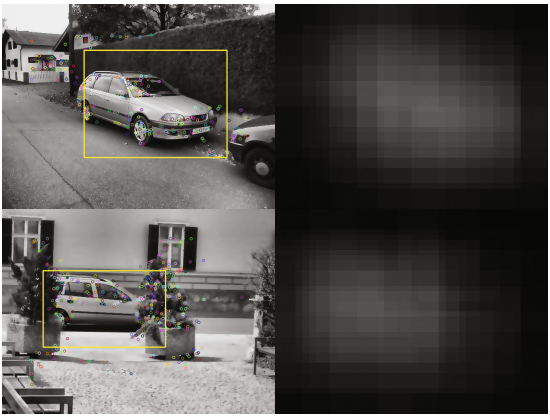

Fig. 5. Results obtained with STAR detector, SIFT extractor, FLANN matcher and SVM classifier (right image with the voting masks and left image with the overlaid results)

Fig. 6. Results obtained with SURF detector, SURF extractor, FLANN matcher and ANN classifier (right image with the voting masks and left image with the overlaid results)

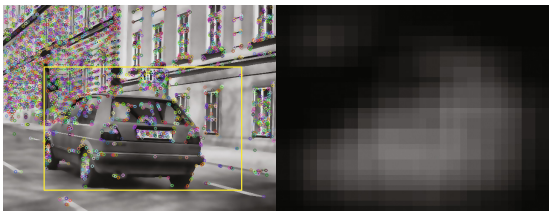

Fig. 7. Results obtained with FAST detector, SURF extractor, FLANN matcher and ANN classifier (right image with the voting masks and left image with the overlaid results)

Fig. 8. Results obtained with ORB detector, ORB extractor, FLANN matcher and ANN classifier (right image with the voting masks and left image with the overlaid results)

and with a dynamic threshold, it can be extracted the regions in the image in which the target objects reside (presented as an overlaid yellow rectangle on the left images, along with the detected keypoints as colored circles).

6 Analysis of Results

In Tables 1 and 2 it is shown the detailed results that were obtained in the testing of the recognition system. It is presented both the recognition performance and

Table 1. Object recognition configurations and performance results

Test ID	Feature detector	Feature descriptor	Feature matcher	Classifier	Accuracy	Precision	Recall
1	STAR	SIFT	FLANN	Neural Network	0.874	0.234	0.162
2	STAR	SURF	FLANN	SVM	0.855	0.271	0.214
3	STAR	SURF	BFMatcher	SVM	0.854	0.299	0.234
4	STAR	SIFT	FLANN	SVM	0.847	0.306	0.362
5	STAR	BRIEF	FLANN	SVM	0.841	0.276	0.277
6	ORB	ORB	FLANN	Neural Network	0.839	0.206	0.195
7	STAR	FREAK	FLANN	SVM	0.815	0.274	0.279
8	SURF	SURF	FLANN	Neural Network	0.815	0.168	0.202
9	SIFT	SIFT	BFMatcher	Neural Network	0.794	0.217	0.296
10	SIFT	SIFT	BFMatcher	SVM	0.784	0.242	0.385
11	SIFT	SIFT	FLANN	SVM	0.776	0.251	0.411
12	ORB	ORB	FLANN	SVM	0.739	0.239	0.549
13	SIFT	SURF	FLANN	SVM	0.714	0.219	0.543
14	SIFT	SURF	BFMatcher	SVM	0.705	0.213	0.528
15	GFTT	FREAK	FLANN	SVM	0.699	0.201	0.478
16	MSER	SURF	FLANN	SVM	0.672	0.241	0.735
17	FAST	FREAK	FLANN	SVM	0.666	0.204	0.596
18	BRISK	BRISK	FLANN	SVM	0.661	0.213	0.682
19	SIFT	BRIEF	FLANN	SVM	0.616	0.187	0.661
20	SIFT	FREAK	BFMatcher	SVM	0.606	0.188	0.696
21	SIFT	FREAK	FLANN	SVM	0.605	0.191	0.717
22	SIFT	BRIEF	BFMatcher	SVM	0.601	0.191	0.732
23	BRISK	FREAK	FLANN	SVM	0.579	0.191	0.801
24	SURF	SURF	FLANN	Decision Tree	0.578	0.175	0.648
25	SURF	SURF	FLANN	Random Tree	0.503	0.172	0.847
26	SURF	SURF	FLANN	Boosting Tree	0.499	0.171	0.845
27	SURF	SURF	FLANN	Extremely Random Tree	0.469	0.167	0.864
28	ORB	ORB	FLANN	Normal Bayes Classifier	0.446	0.165	0.886
29	SURF	SURF	FLANN	Gradient Boosting Tree	0.423	0.161	0.897
30	SIFT	BRISK	FLANN	SVM	0.421	0.159	0.889

Table 2. Object recognition temporal performance results (dataset with a group of 177 images for training and another one of 177 images for testing)

Test ID	Vocabulary build time	Training samples build time	Classifier training time	Classifier test time
1	00 min 31.204 s	00 min 44.265 s	00 min 00.028 s	15 min 14.323 s
2	00 min 21.251 s	00 min 17.901 s	00 min 38.217 s	03 min 02.452 s
3	00 min 20.932 s	00 min 17.985 s	00 min 37.934 s	03 min 33.083 s
4	00 min 31.204 s	00 min 44.265 s	00 min 36.318 s	09 min 43.652 s
5	00 min 20.131 s	00 min 20.105 s	00 min 35.184 s	03 min 46.283 s
6	01 min 25.694 s	00 min 43.962 s	00 min 00.188 s	17 min 04.451 s
7	00 min 20.824 s	00 min 24.739 s	00 min 36.273 s	05 min 22.562 s
8	00 min 37.574 s	00 min 35.434 s	00 min 00.201 s	13 min 03.423 s
9	01 min 46.338 s	01 min 32.902 s	00 min 00.234 s	43 min 00.362 s
10	01 min 40.631 s	01 min 30.025 s	00 min 49.265 s	41 min 43.748 s
11	01 min 46.338 s	01 min 32.902 s	00 min 50.727 s	42 min 32.801 s
12	01 min 25.695 s	00 min 43.966 s	00 min 44.078 s	16 min 56.037 s
13	01 min 17.674 s	00 min 43.966 s	00 min 51.802 s	27 min 05.743 s
14	01 min 11.727 s	00 min 39.477 s	00 min 50.481 s	26 min 21.736 s
15	01 min 01.011 s	00 min 40.011 s	00 min 50.479 s	40 min 07.149 s
16	00 min 22.772 s	00 min 20.369 s	00 min 47.321 s	07 min 41.181 s
17	00 min 56.567 s	01 min 49.256 s	00 min 54.863 s	51 min 38.865 s
18	00 min 21.704 s	00 min 30.616 s	00 min 47.038 s	13 min 12.818 s
19	01 min 03.294 s	00 min 47.819 s	00 min 48.438 s	29 min 37.773 s
20	01 min 08.355 s	00 min 38.618 s	00 min 49.269 s	25 min 22.225 s
21	01 min 06.325 s	01 min 00.102 s	00 min 53.349 s	35 min 35.147 s
22	01 min 05.877 s	00 min 38.599 s	00 min 50.382 s	25 min 04.586 s
23	00 min 30.058 s	00 min 29.093 s	00 min 45.131 s	11 min 03.882 s
24	00 min 37.188 s	00 min 34.271 s	00 min 00.064 s	18 min 05.666 s
25	00 min 37.073 s	00 min 43.967 s	00 min 00.199 s	16 min 17.609 s
26	00 min 37.495 s	00 min 43.962 s	00 min 00.956 s	15 min 41.621 s
27	00 min 35.759 s	00 min 43.969 s	00 min 00.491 s	18 min 33.911 s
28	01 min 24.585 s	00 min 26.650 s	00 min 05.779 s	27 min 22.274 s
29	00 min 37.207 s	00 min 43.964 s	00 min 04.295 s	17 min 23.841 s
30	01 min 08.126 s	01 min 00.105 s	00 min 49.559 s	45 min 40.242 s

also the temporal performance in order to evaluate if the recognition was good enough and also if it can be used for real time applications.

From the analysis of the results, it can be seen that the best accuracy (87.4%) was achieved by combining the STAR feature detector, the SIFT feature extractor, the FLANN matcher and the Artificial Neural Network classifier. This can be attributed to the superior feature description of the SIFT algorithm due to its scale and orientation invariance, and to the fact that the

Neural Network classifier can achieve better generalization of models. Nevertheless, the second best accuracy result (85.5%), which was achieved with the STAR feature detector, the SURF feature extractor, the FLANN matcher, and the Support Vector Machine classifier, was 5 times faster to analyze all the test images. This is greatly due to the application of a faster feature extractor (SURF), and the usage of the more efficient classifier (the SVM shifted the computation time to the training stage, in which it was more than 1300 times slower than the best result, but since this is computed only once, it is an acceptable cost for the overall usage of the system).

From the output of the system it can also be seen that the preprocessing stage helped in the selection of better feature points by reducing the noise and correcting the contrast and brightness. This can be seen in the Fig. 9, in which the mud in the car was reduced and the pavement was smoothed.

Fig. 9. Effect of removing noise and improving contrast and brightness (original image on the left, preprocessed on the right)

7 Conclusions

The presented Bag of Words approach to pose invariant object recognition has shown promising results and good versatility to handle different shapes of cars in different views. Its efficiency, accuracy and flexibility make it a viable solution for recognition of classes of objects with variable geometry.

The clustering of descriptors obtained with scale and rotation invariance significantly contributed to the accuracy and robustness of the recognition and in conjunction with the versatility of the bag of words model, allowed the system to recognize the target objects within cluttered environments.

These results can be further improved if a more advanced and precise location detection algorithm is used (instead of the sliding window approach). This can be achieved by either improving this method, or by considering its result as an initial step in identifying the target objects. For example, the peak in the voting mask could be used has the centroid of a more advanced segmentation technique, in order to retrieved the real location and contour of the target objects.

Acknowledgments. This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the Portuguese funding agency, FCT - Fundao para a Ciência e a Tecnologia as part of project UID/EEA/50014/2013.

References

1. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision (2004)
2. Jang, D.M., Turk, M.: Car-Rec: a real time car recognition system. In: IEEE Workshop on Applications of Computer Vision (2011)
3. Thomas, A., Ferrar, V., Leibe, B., Tuytelaars, T., Schiel, B., Van Gool, L.: Towards multi-view object class detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2006)
4. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: object localization by efficient subwindow search. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (2004)
6. Ponce, J., Lazebnik, S., Rothganger, F., Schmid, C.: Toward true 3D object recognition. In: Congres de Reconnaissance des Formes et Intelligence Artificielle (2004)
7. Zheng, W., Liang, L.: Fast car detection using image strip features. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
8. Gerónimo, D., Sappa, A.D., López, A., Ponsa, D.: Adaptive image sampling and windows classification for on-board pedestrian detection. In: International Conference on Computer Vision Systems (2007)
9. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
10. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision (2003)