YoloSPoC: Recognition of multiple object instances by using yolo-based proposals and deep SPoC-based descriptors

Patricio Loncomilla and Javier Ruiz-del-Solar

Advanced Mining Technology Center, Universidad de Chile, Chile
{ploncomi,jruizd}@ing.uchile.cl

Abstract. The recognition of particular objects instances (e.g. my coffee cup or my wallet) is an important research topic in robotics, as it enables tasks like object manipulation in domestic environments in real-time. However, in recent years most efforts have been aimed to solve generic object detection and object class recognition problems. In this work, a method for performing recognition of particular objects instances, named YoloSPoC, is proposed. It is based on generation of high-quality object proposals by using YOLOv3, computing descriptors of these proposals using a MAC (Maximal Activation of Convolutions) based approach, recognizing the object instances using an open-set nearest neighbor classifier, and filtering of overlapping recognitions. The proposed method is compared to state-of the-art methods based on local features (SIFT and ORB based methods) using two datasets of home-like objects. The obtained results show that the proposed method outperforms existing methods in the reported experiments, being robust against conditions like (i) occlusions, (ii) illumination changes, (iii) cluttered backgrounds, (iv) presence of multiple objects in the scene, (v) presence of textured and non-textured objects, and (vi) object classes not available when training the proposal generator.

Keywords: Object recognition; object instance recognition; Robocup@Home.

1 Introduction

In recent years, advances in computer vision have been impressive, as convolutional neural networks (CNN), and in general deep learning, has enabled the use of large datasets for training models with millions of parameters. Then, hand-engineered systems have fallen in favor of flexible network architectures trained with large datasets, where each specific task to be solved is specified by means of an optimization loss.

One of the hottest research topics is generic object recognition, in which objects are classified into a set of possible categories, each having thousands of labeled examples. The accuracy of these systems is constantly rising, having outperformed humans in this task. The same applies to the task of object detection in images, where impressive results have been obtained using deep-based approaches such as YOLO [7].

However, in some applications like manipulation of objects in domestic environments (see Figure 1), the recognition of particular objects instances (e.g. my coffee cup, my wallet, my key chain), using just one or very few stored views (the number depends on the objects symmetry) of the object, is required. In this case it is difficult to apply the standard deep learning pipeline, because it is not always possible to obtain hundreds or thousands of images of the object instances to be recognized. Also, the number of particular objects to recognize can be much higher than those in standard datasets like COCO [14], which consider only 80 possible object categories. This topic has not been fully addressed by the computer vision or the deep learning communities.

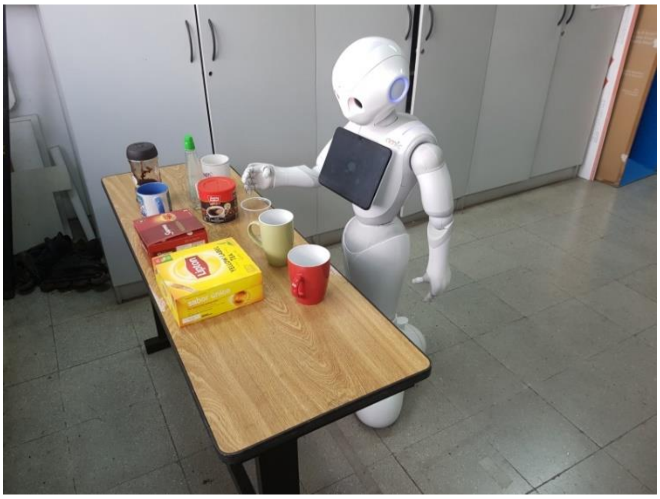

Fig. 1. Pepper robot detecting instances of domestic objects.

In this work, we analyze this problem and propose a method for solving the particular object instance recognition task. The method is based on the use of YOLOv3 [8] for generating high-quality proposals, and computing descriptors for each proposal using a MAC (Maximal Activation of Convolutions) based procedure [4][5]. Then, the descriptors in the current test image are compared to reference descriptors using an open-set nearest-neighbor classification procedure, followed by a post-processing step, which filters multiple, overlapping recognitions of the object instance.

This paper is organized as follows: Section 2 presents the related work, and in Section 3 the proposed method is described. Then, in Section 4 results are presented and analyzed, and in Section 5 conclusions are drawn.

2 Related Work

While general object recognition has been successfully addressed by the computer vision community by using CNNs, recognition of particular object instances has not been at the aim of current research. However, some works address this problem by following different approaches.

In [9][10], the SIFT L&R object recognizer, which uses matches between local descriptors followed by geometric verification, is used for comparing several local descriptor implementations. This object recognizer can detect successfully textured objects and can deal with cluttered backgrounds. However, it is not able to recognize objects which are untextured, or far from the camera.

In [12], a system which can recognize object instances from RGBD data is presented. For classifying images in which the object is cropped, local descriptors (spin images, SIFT) are extracted for representing shape, while SIFT is extracted from images. An EMK feature is generated for pooling information from the descriptors, and a classifier (SVM or random forest) is applied to predict the label from the object. For detecting objects, a sliding window approach using a HoG-based detector is tested. Also, in this work two benchmarks named *Washington RGB-D object dataset* and *Washington RGB-D scene dataset* are introduced.

It must be also noted that, in previous RoboCup conferences, there are not papers related to particular object instance recognition using deep learning techniques.

3 Proposed Method

In this work, we aim to solve a particular object instance recognition by using methods based on CNNs and nearest-neighbor classification. By changing the paradigm from local descriptors to global descriptors based on CNNs, we aim to recognize objects that are very hard to recognize by using just local descriptors.

Thus, the proposed method is based on four main blocks: (i) computation of region proposals, (ii) computation of global descriptors, and (iii) recognition of object instances using an open-set nearest-neighbor classification scheme, which can reject detections generated by objects outside the training dataset, and (iv) suppression of redundant detections. The blocks that define the method are shown in Figure 2.

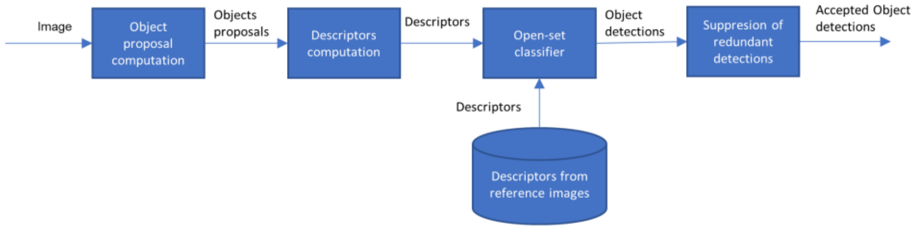
Image
Object proposal computation
Objects proposals
Descriptors computation
Descriptors
Open-set classifier
Object detections
Suppresion of redundant detections
Accepted Object detections
Descriptors
Descriptors from reference images

Fig. 2. Block diagram of the system

3.1 Object proposal detection

The generation of object proposals is based on the YOLOv3 object detector [8]. It is a CNN, which takes an image as input and generates a list of bounding boxes as output, each considering a confidence and probabilities of belonging to each class. The system works by considering an initial set of bounding boxes spread over the image, and then, by using regression, each bounding box is fitted to an object present in the image. Also, both probabilities of belonging to each object class, and an objectness score are computed for each proposal. Finally, non-maximal suppression is applied over the resulting bounding boxes. YOLOv3 is improved respect to previous YOLOs, by considering multiple scales and modifying the loss function for improving accuracy on small objects. The network is trained on large datasets like COCO [14], which includes about 200,000 labeled images, divided into 80 classes. As YOLOv3 is trained to detect a predefined set of classes, it is not feasible to use it as a general proposal generator, which must be class-agnostic. However, this problem can be handled by lowering the confidence threshold from 0.5 to 0.01, which enables detection of most of the objects to recognize, at the cost of generating misdetections that need to be handled by an open-set classifier. As YOLOv3 assigns labels to the detections, and this work is aimed at detecting objects that can be manipulated by a robot, detections with labels related to non-manipulable objects (like furniture) are discarded.

3.2 Descriptors computation

Global descriptors are computed for describing the proposals. The used methodology [4][5] is aimed to another application, image retrieval, in which it achieves state-of-the-art performance. However, it can be used successfully for generating global descriptors. The method consists on computing features using CNNs over the object proposals and then pooling them channel-wise into a descriptor [4][5]. The used architecture consists of the convolutional layers of a ResNet-101 network [16], which is a high performant CNN that includes residual blocks containing shortcut connections, as shown in Figure 3. The network is pretrained on ImageNet [15], but it can be fine-tuned by using a Siamese network approach, which consists on modeling the network as an embedding function, which maps inputs into a metric space. This procedure is shown in Figure 4.

The use of channel-wise pooling operations enables the system to be robust against small translations and small changes in the scale of the objects. Also, for improving robustness against scale change and translation, these descriptors can be computed for several sub regions of the image, and then summed up.

These descriptors are very robust, and able to outperform all previous methods on image retrieval. For computing the global descriptors, different variants of pooling described in [5] are used, including MAC (maximum activations of convolutions), SPoC (sum-pooled convolutional features) and gem (generalized mean). The CNN Image retrieval toolbox [11], which implements the computation of global descriptors, is used for computing a descriptor for each proposal. Also, this toolbox is able to train the network by using a Siamese network approach, by using Structure from Motion on outdoor scenes.

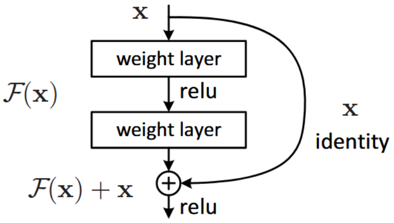
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 3. Residual block used in ResNet networks, which can include more than 100 layers. Figure taken from [16] with IEEE permission, license number 4606550286412.

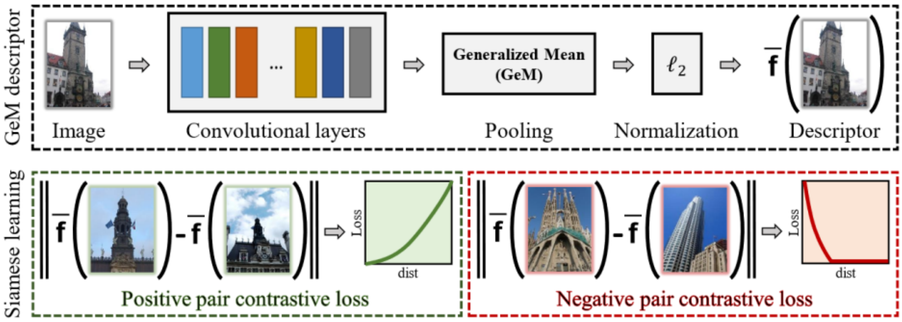

Fig. 4. Architecture of the system used for training global descriptors, by using a Siamese Network approach. The convolutional layers are based on a ResNet101 network. Figure taken from [5] with IEEE permission, license number 4606550559202.

3.3 Open-set nearest neighbor classifier

A nearest neighbor classifier is used for assigning labels to each proposal from the test image, by using its global descriptor. As the environment could contain objects different that those stored in the training database, the system must be able to reject unknown objects.

Open-set classifiers can deal with data different from that available during training, labeling them as unknown. An open-set classifier is used for classifying the proposals, and for rejecting those related to objects outside the training dataset. A nearest neighbor classifier considering a second-to-first nearest neighbor distance ratio test [3] is used for rejecting detections that does not correspond to any of the known objects.

$$\frac{d_2}{d_1} < threshold \rightarrow reject$$

where d_1 is the Euclidean distance between the test descriptor and the nearest descriptor in the database, and d_2 is the Euclidean distance between the test descriptor and the nearest descriptor from a different object in the database. Then, in case the two distances are similar, the classifier cannot be confident about the class of the object, and the detection is discarded.

3.4 Suppression of redundant detections

The invariance of the descriptor against different translations and scales is useful for managing occlusions and different object poses. However, it can generate multiple detections from a same object, because several small proposals intersecting the true object can be labeled as positives. Then, a post-processing step is required. Detections related to a same object are sorted by its size. Then, when the bounding box from a given detection is completely contained inside another bounding box related to the same object, it is eliminated. A tolerance threshold of 30 pixels is used for deciding when a given object detection is contained inside another object.

4 Experimental results

4.1 Datasets

There are several available datasets for particular object recognition, like Washington RGB-D Scenes Dataset [12], or the GMU Kitchens Dataset [13]. These datasets include a high amount of training instances, and provide depth information for detecting the objects. The Washington RGB-D Scenes Dataset [12] consists of 8 scenes, each of which consists on a sequence of images from a moving camera. Each sequence can contain up to 300 different possible domestic objects from the Washington RGB-D Object dataset. The Scenes dataset is not aimed at detecting object instances from few views, as around 830 training images per object are provided, for 300 different objects. However, in this work we compare on this dataset by using only 12 views for each object, without using parametric classifiers.

A dataset named DSLL, introduced in [10], is composed of two sets: a set of reference images (views of each of the reference objects) and a set of test images, which can contain some of the reference objects inside. The set of reference images contains 40 objects, each represented by 12 views. Only a 33% of the objects are contained in the 80 classes from COCO. As shown in Table 1, it consists of several test sets, each considering variations in the actual conditions of a domestic environment.

Previous results using this dataset [10] were based on comparing if the predicted affine geometric transformation associated to a detection is similar enough to the ground truth one [10]. However, object detectors based on CNNs provide a bounding box, but not an affine transformation. In consequence, results related to SIFT L&R and SURF L&R are recomputed, by using the usual 0.5 IoU threshold criterion.

Table 1. Test sets defined in DSLL Dataset

ID	**Description of test set conditions**
S1	One object, white background, normal illumination, no occlusion.

S2	One object, white background, low illumination, no occlusion.
S3	One object, colored background, normal illumination, no occlusion.
S4	One object, cluttered background, normal illumination, no occlusion.
S5	One object, white background, normal illumination, 50% occlusion.
M1	Six objects, colored background, normal illumination.
M2	Six objects, cluttered background, normal illumination.

4.2 Experiments on DSLL Dataset

The proposed method, YoloSPoC, is compared to state-of-the-art object recognition method based on global features. The system is composed by four components: (i) a proposal generator based on YOLOv3, (ii) global descriptors generator based on channel-wise CNN pooling, (iii) an open-set nearest neighbor classifier, and (iv) elimination of redundant detections. CNNs using three pooling flavors from [5] are tested: SPoC (based on average pooling), MAC (based on max pooling) and gem (based on generalized mean).

The specific methods under comparison are:

(i) SIFT L&R: Object detector based on SIFT local descriptors with geometric verification [10].

(ii) ORB L&R: Object detector based on ORB local descriptors with geometric verification [10].

(iii) YoloSPoC: The proposed method. It is based on YOLOv3 for generating proposals, CNN descriptors with SPoC pooling, an open-set nearest neighbor classifier, and suppression of redundant detections.

(iv) YoloSPoC-gem: Similar to YoloSPoC, but using gem pooling

(v) YoloSPoC-triplet: Similar to YoloSPoC-gem, but using a CNN learned by using triplet learning in a dataset of edifications [5].

Several experiments were performed for evaluating the performance of the proposed algorithms. In all of the methods, the following parameters are used: threshold for *YOLOv3* confidence 0.01, threshold for nearest neighbor ratio 0.95. Also, *YOLOv3* detections labeled as objects which cannot be manipulated by a robot are discarded: {diningtable, refrigerator, sink, chair, bench, toilet, bed}.

The first set of experiments consists of comparing all the methods on the DSLL dataset. Results are shown in Table 2. From this data, it can be concluded that the YoloSPoC-based variants largely outperforms methods based on local descriptors (YoloSPoC 0.843 v/s SIFT L&R 0.322). Also, YoloSPoC outperforms both YoloSPoC-gem and YoloSPoC-triplet. Finally, the use of triplet learning [1][2] is shown not to be useful when a large domain shift exists between the dataset used for training the network, and the dataset in which the network is used. In our case, the datasets being considered consists of domestic objects, and the network YoloSPoC-triplet was no trained on this kind of data, but on outdoor scenes. Also, the system is able to work even when only

33% of the objects belong to classes from the COCO dataset (used for training YOLOv3). Then the system is able to deal successfully with new object classes not available during training.

Table 2. F1-scores on DSLL, per subset

Subset	Method				
	YoloSPoC	***YoloSPoC-triplet***	***YoloSPoC-gem***	***SIFT L&R***	***ORB L&R***
S1	0.861	0.836	0.836	0.679	0.632
S2	0.840	0.841	0.864	0.329	0.244
S3	0.902	0.910	0.944	0.312	0.208
S4	0.923	0.877	0.857	0.247	0.169
S5	0.695	0.620	0.631	0.465	0.337
M1	0.863	0.841	0.861	0.143	0.090
M2	0.814	0.829	0.817	0.076	0.045
Mean	0.843	0.822	0.830	0.322	0.246

YOLOv3 is trained on the COCO dataset, which considers 80 object classes. Then, degradation of proposal's quality can be expected when detecting objects not contained in COCO. Then, the second set of experiments analyses which would be the performance of the proposed method in case of having ideal proposals. In order to test this, the real ground truth boxes are used instead of the YOLO-based proposals. Results are shown in Table 3. The proposed systems achieve impressive F1-scores over 0.92, except in the dataset S5 which includes occlusions. The best variant is again YoloSPoC, and the use of triplet learning is not useful again because of the domain shift.

Table 3. F1-Scores on DSLL, considering truth bounding boxes, per subset

Sub-set	Method				
	Yolo-SPoC	***Yolo-SPoC-triplet***	***Yolo-SPoC-gem***	***SIFT L&R***	***ORB L&R***
S1	1.0	1.0	1.0	0.679	0.632
S2	1.0	0.981	0.997	0.329	0.244
S3	0.987	0.978	1.0	0.312	0.208
S4	0.961	0.939	0.975	0.247	0.169
S5	0.865	0.836	0.823	0.465	0.337
M1	0.965	0.936	0.955	0.143	0.090
M2	0.928	0.916	0.919	0.076	0.045
Mean	0.941	0.931	0.934	0.322	0.246

The following conclusions can be drawn of the results presented in Tables 2 and 3:

(i) Multiple objects: The performance of YoloSPoC does not degrade when multiple objects are present in the images. In Table 2 a F1-score of 0.839 is obtained in multiple objects cases (M1, M2) v/s 0.847 in single object cases.

(ii) Occlusions: YoloSPoC is able to detect occluded objects, but its performance is slightly decreased with occlusions. In Table 2, a F1-score of 0.695 is obtained in occluded cases (S5) v/s 0.867 when no occlusions occur.

(iii) Illumination changes: The performance of YoloSPoC is almost not affected by changes in illumination. In Table 2, the F1-score is 0.840 in cases of variable illumination (S2), v/s 0.843 in cases of no illumination changes.

(iv) Cluttered backgrounds: The performance of YoloSPoC is almost not affected by cluttered backgrounds. In Table II, a F1-score of 0.869 is obtained when cluttered backgrounds are considered (S4, M2) v/s 0.832 when uniform backgrounds are used.

(v) Non-textured objects: YoloSPoC is able to detect non-textured objects. In Table III, YoloSPoC can get a perfect F1-score on cases S1 and S2, in which non-textured objects are present. Then, the SPoC descriptor is useful for detecting both textured and non-textured objects.

Examples of different conditions (i), (ii), (iii), (iv), (v) are shown in Figure 5.

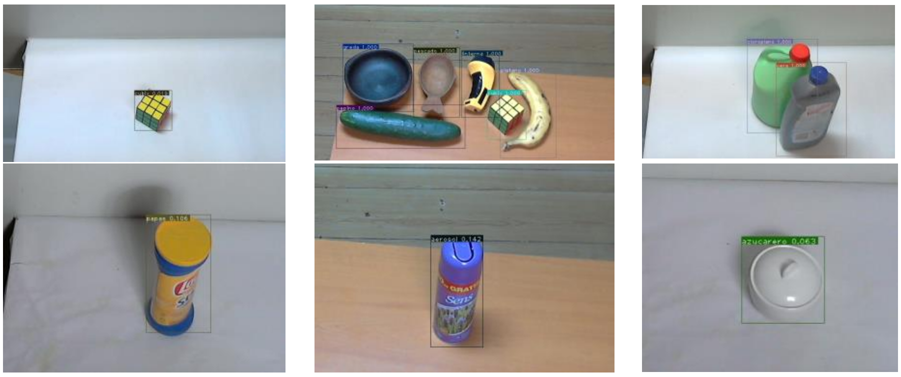

Fig. 5. Objects to recognize under different conditions: single object, multiple objects, occlusions, illumination changes, cluttered background and non-textured objects.

Finally, the third set of experiments consists on analyzing the dependence between the F1-score and the nearest neighbor ratio. Results are shown in Table 4. It can be noted that the best nearest neighbor ratio threshold for detections considering YOLO proposals is 0.95. However, when ground truth bounding boxes are used instead of YOLO, the best nearest ratio threshold is 1.0, which is equivalent to not using the nearest neighbor ratio at all. In consequence, the use of the nearest neighbor ratio is not useful in this case. Then, it can be concluded that the nearest neighbor ratio is useful for discarding proposals not containing objects from the dataset, because they are only present when YOLO-based proposals are used.

Table 4. F1-scores over all subsets using different methods, using different nearest neighbor ratios, both for YOLO proposals and for ground truth bounding boxes

Variant	**Method**				
	Yolo-SPoC	***Yologem triplet***	***Yologem***	***SIFT L&R***	***ORB L&R***
YOLO proposals @ 0.90	0.770	0.769	0.763	0.322	0.246
YOLO proposals @ 0.95	0.834	0.820	0.837	0.322	0.246
YOLO proposals @ 0.98	0.824	0.816	0.820	0.322	0.246
YOLO proposals @ 1.00	0.803	0.774	0.795	0.322	0.246
GT proposals @ 0.90	0.875	0.871	0.860	0.322	0.246
GT proposals @ 0.95	0.947	0.928	0.938	0.322	0.246
GT Proposals @ 0.98	0.966	0.940	0.952	0.322	0.246
GT proposals @ 1.00	0.967	0.940	0.952	0.322	0.246

4.3 Washington RGB-D Scenes Dataset experiments

The Washington RGB-D Scenes dataset [12] includes eight sequences, each generated by a moving camera. A total set of 300 object instances can be included in the sequences. Each object is represented by around 830 views.

In this work, we compare YoloSPoC against the instance object detector proposed in [12], which uses a variant of HoG descriptors applied over sliding windows and a linear SVM classifier for detecting and classifying the objects. Only 12 views per training object are used when testing our system, while for [12] around 830 views of each training object are used, as the last system is unable to work when few views are available. The precision-recall curves are shown in Figure 6. Results indicate that our system, using only 12 views per object, improves largely on the original work, which uses around 830 views per object. Also, the original system [12] requires the use of background images from the same scenes for training a classifier, while our system is able to work without any kind of extra training, and it is shown to be robust against different backgrounds.

Note that, in YoloSPoC, predictions do not have a meaningful score, because nearest neighbors are used for labeling the object proposals. Then, the precision-recall curve is generated by modifying the nearest neighbor ratio threshold.

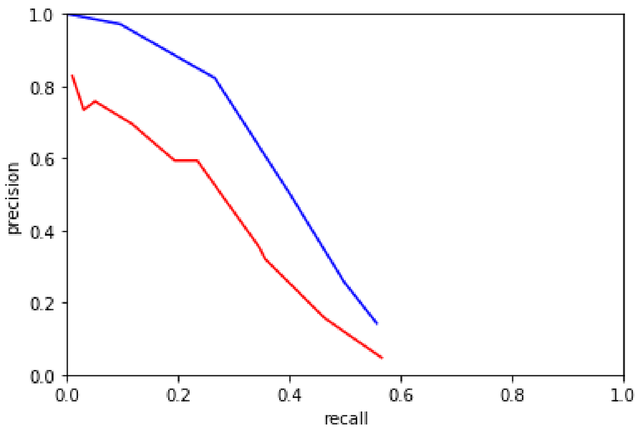

Fig. 6. Precision-recall curves over all WRGBD Scenes dataset. The original system [12] using RGB data is shown in red, and the proposed system YoloSPoC is shown in blue.

5 Conclusion

In this work, a novel method named YoloSPoC for solving the particular object recognition task is proposed. It is composed by an object proposal generator based on YOLOv3, a SPoC CNN used for generating global descriptors, a nearest neighbor classifier, a nearest neighbor ratio test for rejecting misdetections, and a final post-processing step for eliminating multiple detections from a same object. The method is compared against several state-of-the-art methods based on local descriptors and geometric verifications. Extensive experiments are performed, by using the DSLL dataset, which contains images of objects which can be manipulated by a robot, captured under different conditions, and also by using the Washington RGB-D Scenes dataset. In the reported experiments, the proposed system is shown to consistently outperforms previous methods on all of the tests performed by a large margin. YoloSPoC is robust against conditions like (i) multiple objects, (ii) occlusions, (iii) illumination changes, (iv) cluttered backgrounds, (v) non-textured objects, and (vi) object classes not available when training the proposal generator.

The use of triplet learning is not useful in our case because of domain shift, as the Siamese network used for computing global descriptors was trained on outdoor images, while our application is related to manipulable objects. Also, the nearest neighbor ratio test is shown to be useful for discarding proposals not associated to objects from the dataset. Future work includes evaluating the system in other datasets and fine-tuning the networks that compute global descriptors by using triplet learning, on a new dataset of domestic objects to be built.

Acknowledgement

This work is partially funded by Fondecyt 1161500 and CONICYT PIA grant

AFB18004.

References

1. Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML Deep Learning Workshop. Vol. 2. 2015.
2. Elad Hoffer and Nir Ailon. Deep Metric Learning Using Triplet Network. International Workshop on Similarity-Based Pattern Recognition, 84-92, 2015 .
3. David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60, 2 (November 2004), 91-110. DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94
4. Giorgos Tolias, Ronan Sicre, Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. ArXiv:1511.05879, 2016.
5. Filip Radenović, Giorgos Tolias, Ondřej Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1655-1668, 1 July 2019
6. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems (NIPS), 91-99, 2015
7. J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. Computer Vision and Pattern Recognition (CVPR), 2017
8. Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement. ArXiv:1804.02767, 2018
9. Luz Martínez, Patricio Loncomilla, Javier Ruiz-del-Solar. Object Recognition for Manipulation Tasks in Real Domestic Settings: A Comparative Study. In: Bianchi R., Akin H., Ramamoorthy S., Sugiura K. (eds) RoboCup 2014: Robot World Cup XVIII. RoboCup 2014. Lecture Notes in Computer Science, vol 8992. pp 207-219. Springer, Cham
10. Patricio Loncomilla, Javier Ruiz-del-Solar, Luz Martínez. Object recognition using local invariant features for robotic applications: a survey. Pattern Recogn. Vol 60, pp. 499-514, 2016
11. CNN Image Retrieval in PyTorch: Training and evaluating CNNs for Image Retrieval in PyTorch. https://github.com/filipradenovic/cnnimageretrieval-pytorch
12. Kevin Lai and Liefeng Bo and Xiaofeng Ren and Dieter Fox. A large-scale hierarchical multi-view RGB-D object dataset. 2011 IEEE International Conference on Robotics and Automation, 2011, 1817-1824
13. Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, Jana Kosecka. Multiview RGB-D Dataset for Object Instance Detection. International Conference on 3DVision (3DV) 2016
14. COCO Dataset. http://cocodataset.org
15. ImageNet. http://www.image-net.org
16. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)