Learning Object Classifiers with Limited Human Supervision on a Physical Robot

Christopher Eriksen, Austin Nicolai, and William D. Smart
Robotics Program
Oregon State University
Corvallis, Oregon 97331
Email: {eriksenc, nicolaia, smartw}@oregonstate.edu

***Abstract*—In recent years, deep learning approaches have been leveraged to achieve impressive results in object recognition. However, such techniques are problematic in real world robotics applications because of the burden of collecting and labeling training images. We present a framework by which we can direct a robot to acquire domain-relevant data with little human effort. This framework is situated in a lifelong learning paradigm by which the robot can be more intelligent about how it collects and stores data over time. By iteratively training only on image views that increase classifier performance, our approach is able to collect representative views of objects with fewer data requirements for longterm storage of datasets. We show that our approach for acquiring domain-relevant data leads to a significant improvement in classification performance on in-domain objects compared to using available pre-constructed datasets. Additionally, our iterative view sampling method is able to find a good balance between classifier performance and data storage constraints.**

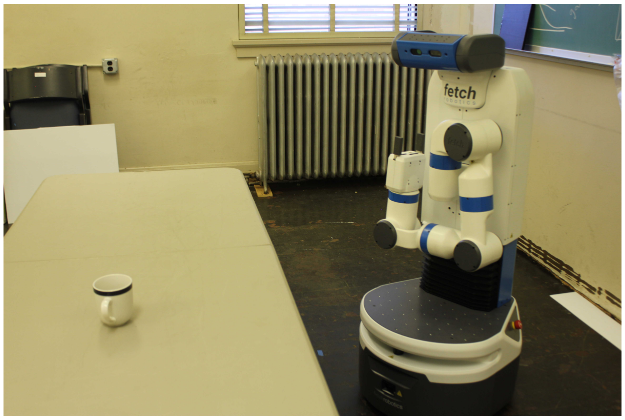


Fig. 1. Our autonomous data gathering framework capturing training data of a coffee mug using a Fetch robot.

I. Introduction

Recent years have seen a significant rise in performance of computer vision techniques as applied to a wide variety of robotics tasks. Deep learning approaches in particular have allowed for large performance gains in critical robotics applications such as object recognition and classification [1]–[5]. Despite the success of these models, they are limited in their applicability to robots operating in real world domains. Deep models must be supplied with large amounts of training data, which typically require a great deal of human effort to create. While large image corpora of common objects exist for training good general models [6]–[8], such datasets are unsuitable for representing the extent of variation found in robotics environments. Additionally, images contained online are often distinguishably different from image views a robot will be presented with during operation. For example, online images are typically taken in good lighting conditions, from human heights, and with the subject in center frame. Robots, however, might be operating at different heights, in a variety of lighting conditions, and with no predictable structure to the layout in their current view.

To address these challenges, we provide a framework by which a human user can direct a robot to autonomously capture domain-relevant data for use in training classifiers of household objects in the user's personal environment. By using a simple click interface, the user is able to intuitively specify which objects he or she would like the robot to learn. The robot then captures a series of image views for use in training without placing additional demands on the human's time. In this way, the robot is able to effectively learn the objects in one's personal environment without the intensive human effort of manually constructing a dataset.

Additionally, we present steps toward leveraging our data capture framework in a lifelong learning domain. While most previous work in image classification assumes a predetermined, static dataset for use in training in batch fashion, such a paradigm is of limited use in real world robotics domains. Given the extensive variation of possible configurations and viewpoints in personal environments, it is difficult to create a singular, environment-independent dataset that adequately represents the learning space. Recently, more work has started to approach the learning problem from a lifelong or never-ending learning perspective. By continuously incorporating accumulated knowledge, such approaches are able to increasingly learn the complexities of an environment over time. In this work, we present initial steps toward bringing our data capture method into a lifelong learning setting. By iteratively training only on images that improve learning performance, our method retains a limited training set of representative object views to lessen the storage and computational demands on repeated model training. Given the continuous acquisition of data in the lifelong learning paradigm, such data management is important for allowing model retraining to happen within a reasonable timeframe and not overwhelm storage resources. By comparing against capturing a pre-specified number of image views of each object instance, we show that our approach for intelligent data capture approach is able to find a good trade-off between classifier performance and storage constraints.

The rest of the paper is organized as follows. First, we provide relevant background on object recognition, lifelong object learning, and previous interfaces for teaching objects to robots (Section II). Next, we specify the details of our data collection and training framework (Section III). We then describe an experiment by which we compare our data collection strategy against networks trained on pre-constructed datasets as well as investigate the performance benefits of our iterative view sampling approach (Section IV). Finally, we discuss the result of our experiments (Section V) and suggest areas for future work in applying our data capture framework toward the lifelong learning of objects (Section VI).

II. Related Work

A. Object Recognition

Approaches to object recognition typically involve the supervised training of classification models on labeled training data. While object recognition is commonly applied to images, approaches handling other forms of data such as point clouds [9], [10] and RGB-D images [11], [12] exist as well. In recent years, deep learning approaches such as convolutional neural networks (CNNs) have been able to achieve impressive results in object recognition and related computer vision problems [1]–[5]. One benefit of such techniques is that they can effectively learn features that are important for the classification process, without requiring those features to be explicitly defined a priori.

Data driven methods depend on being supplied with an intensive amount of labeled training data, which typically requires a great deal of effort by humans to annotate. In the computer vision domain, a number of sizable image datasets have been produced over the years, which have served as standard testbeds for computer vision problems [6]–[8]. Despite their popularity, these datasets represent a relatively few number of classes and are limited in their applicability to real-world robotics applications. As an illustrative example, Sun and Fox [13] apply an object recognition classifier trained on the large-scale ImageNet dataset [6] to a much smaller RGB-D database specifically designed for robotics applications [14] and found that precision across the RGB-D dataset was 33% lower than across the same objects types than across the ImageNet test set. The authors note the significant difference in appearance between images in the more general ImageNet dataset and the robotics specific set, despite containing the same object classes. As supported in our experiments, we maintain that the large variability in real-world scenarios limits the applicability of even the relatively few robotics-specific datasets to unseen environments. To address this, we provide an interface for acquiring data relevant to a user's specific scenario that requires only a limited amount of human effort.

Domain adaptation and transfer learning methods provide complementary approaches for adapting data from a different, but related source domain to a target domain. However, these approaches still require matching target domain data to apply the transfer. In our work, we apply a related technique specific to deep learning known as finetuning, which allows feature representations learned for one task to be applied to new tasks [15]–[17]. As is common amongst a number of other computer vision applications, we leverage the large variability in the ImageNet dataset for training good generic image features that we repurpose for use in our learning problem. This has the added benefit of allowing our classifier to learn a good model using relatively few captured images compared to larger, pre-constructed datasets since the generic image features have already been learned and only need to be correctly associated for our target problem.

B. Lifelong Object Learning

While many object recognition systems in particular are trained once and then tested on a pre-defined dataset, personal robots that are to be deployed in longterm scenarios must be able to continuously expand and refine their knowledge throughout their deployment in what is often called *lifelong learning*. Recently, a number of approaches have investigated the online training of deep networks, by which a model can be trained iteratively on individual data instances as they become available instead of having to collect all training into a single set for batch training [18]–[21]. Still, many of these approaches require or could at least greatly benefit from maintaining a validation set of previously seen data for retraining so as not to forget previously learned knowledge. However, if continually expanding training sets are not well-managed, they can easily surpass available storage capacity and require increasingly impractical amounts of time to train over, but this has not been addressed in current work. Approaches specific to the lifelong learning of objects includes the Never-Ending Object learner of Sun and Fox [13], in which a semantic hierarchy of object names is constructed by using crowdsourcing to learn relations between objects. In this approach, exemplar images for labels in the hierarchy are continuously stored over time. While this supports the continual learning of their models, it similarly does not consider limitations regarding the continual acquisition of training data.

In addition to providing a framework for the collection of domain-relevant data, we take initial steps toward applying our work to the lifelong learning paradigm. In order to lessen demands on storage and training resources, our method limits data capture so as to store only a limited set of training images as are useful to the learning problem rather than capture an arbitrary and possibly superfluous number of images. While this approach does not increase the sustainability of continual data capture in the limit, these are our first steps toward constructing a representative set of training data that accounts for limitations in storage and training resources.

C. Object Instruction Interfaces

In addition to techniques for supporting the continual learning of a model over time, a number of interfaces have also been developed by which human partners can teach objects to robots in a progressive manner. In [22], Lopes and Chauhan present an interface by which a user collects training examples for a robot by using a mouse to point to an object in a robot's visual feed and providing the object's name. Using edge-filtering, the boundary of the object is extracted, and the example is fed into an incrementally trained classifier. While this approach is useful for achieving high quality annotations, a user must annotate every image used for learning. In our work, we lessen demands on a human user by requiring only a single interaction for multiple data capture instances of a given object. In work by Villamizar, et al. [23], [24], uncertain frames from a video stream pre-recorded on a robot are presented to a human user for annotation, with annotation consisting of drawing a bounding box around objects of interest and labeling them. Human interaction is minimized by presenting fewer queries as classifier performance improves. In our work, we aim to similarly minimize human interaction time but do not depend on data that is available from pre-recorded streams. Rather, our framework allows a robot to capture relevant data for the learning task at hand.

In [25], Azagra, et al. present a framework by which a human user teaches objects to a robot by either pointing at a given object and speaking its name, grabbing the object and moving it directly in front of the robots camera while also announcing its name, or by describing the object's placement within the robots current field of view. Similar work by Lim, et al. [26] describes an interactive object teaching interface that tracks objects on a table, displays them using the Robot Operating System (ROS) [27] visualization tool, RViz [28], and depends on human users to provide object labels. Users can identify an object using a corresponding tracking number or by physically pointing, which is detected by the robot using skeleton tracking. Since objects are being tracked, the user can provide additional views for training by moving the objects around in the robot's view. In our approach, we similarly use RViz as an easy-to-use interaction interface, but put less requirements on the user by having the robot sample views instead of depend on human demonstrations. Additionally, our approach only captures as much data as is needed to learn a given object.

While previous works provide intuitive interfaces allowing a human to teach object examples to robots in a progressive manner, they are dependent on the user for demonstrating different object views or are restricted by only considering pre-collected data. Our work builds on these approaches by using a similarly intuitive interface, yet autonomously captures new data instances for learning. Additionally, our approach only captures as much data as is useful for training the classifier.

III. Problem Formulation

In this section, we describe the steps of our framework for constructing an object classifier for use in a user's personal environment. First, the user must orient the robot toward the object he or she would like it to learn. Using the RViz graphical user interface [28], the user then selects the object in the robot's current view frame. The robot will then autonomously gather data of the object by randomly sampling a set of image views in the 3D space around the object. As part of our iterative learning approach, the robot will continuously evaluate its learning performance over captured data and halt training when performance stops improving. Captured images that are not trained on are then discarded such that a minimal set of images is maintained for future retraining. This process can then be performed again on new objects as the user points them out.

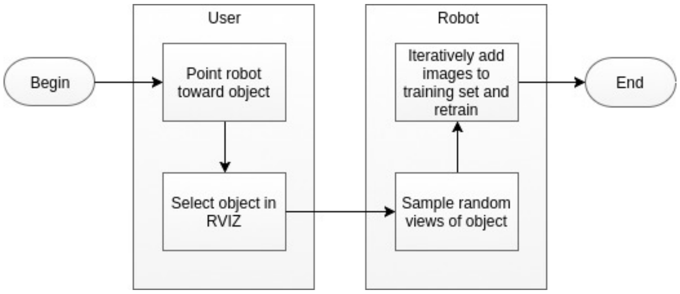
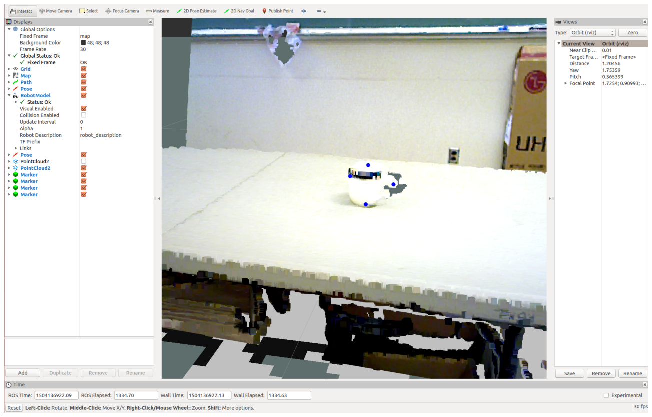

User
Robot
Begin
Point robot toward object
Select object in RVIZ
Sample random views of object
Iteratively add images to training set and retrain
End

Fig. 2. Flowchart of the proposed method. First, the user directs the robot toward an object to learn. The robot then captures a set of images by randomly sampling object views. Captured images are iteratively added to a growing set for classifier training until prediction performance across the set of captured images stops improving. The robot is then free to repeat the process for new objects.

A. User Interaction and Data Capture

In order to specify the object of interest, a user is presented with a point cloud of what the robot can see using RViz. The user then selects a set of points bounding the object inside the point cloud using the point publishing functionality (see Figure 3). For each specified point, a coordinate transform is calculated between it and the robot's current world map. These transforms are later used for finding the object of interest during image capture.

During data capture, the robot randomly samples 3D locations from which to view the selected object within an experimentally-tuned area around the object. For each random sample, the robot orients itself such that it is facing the specified object. We assume that the object stays relatively static during data capture. To localize objects during data capture, we additionally tried having the user place a fiducial marker (ARTag) next to the object of interest such that the robot could find the specified object using the relative location of the marker. However, we found that the robot could not robustly detect the markers (distinct as they were) at longer distances or extreme angles. Since the use of markers additionally placed another constraint on the human user, we forewent their use and depended on the robot's localization ability using the ROS Navigation package [29] to direct data capture.



Global Options	
Fixed Frame	map
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
Map	☑
Path	☑
Pose	☑
RobotModel	☑
✓ Status: Ok	
Visual Enabled	☑
Collision Enabled	□
Update Interval	0
Alpha	1
Robot Description	robot_description
TF Prefix	
Links	
Pose	☑
PointCloud2	□
PointCloud2	☑
Marker	☑
Marker	☑
Marker	☑
Marker	☑

Type: Orbit (rviz) Zero

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	1.20456
Yaw	1.75359
Pitch	0.365399
Focal Point	1.7254; 0.90993; ...

ROS Time: 1504136922.09 ROS Elapsed: 1334.70 Wall Time: 1504136922.13 Wall Elapsed: 1334.63 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options. 30 fps

Fig. 3. The user interface in RViz by which an end user picks points surrounding the object to be learned. A static transform between the selected points and the world map is consulted to direct data capture around the object.

B. Network Training

Using the data collection framework, the robot will capture a set number of images of the specified object. Our learning approach then interleaves iteratively adding a captured image to a growing training set and retraining the classifier on the training set. Classifier performance on all captured images of the object of interest is tracked, and training is halted once classification performance stops improving. In this way, a representative subset of captured images is stored to lessen the storage requirements and computational burden of future retraining. For our network architecture, we use Google's Inception network [30] as implemented in the Keras framework [31]. Since we perform object learning with relatively few images, we finetune the network after pre-training on the large-scale ImageNet dataset to learn generic image features. As an initialization step and point of comparison, we also pre-train class weights on scraped Flickr [32] images of our specified object classes.

IV. Experiments

We applied our approach to an experiment in which a human user directed a Fetch robot equipped with and RGB-D sensor to learn instances of six common household object categories. After placing an object on a table, the user oriented the robot toward the placed object and selected the object in the RViz GUI. The robot then randomly sampled the relevant space around the specified object to produce view locations (x, y, spine height) and planned a path between points using the ROS Navigation package [29]. Following this process, a random set of 100 image views were captured for each object. Using our learning approach, the robot would iteratively add a captured image to the growing training set and retrain the classifier until performance across the entire set of 100 images stopped increasing. For our experiment, data capture and training were performed disjointly so that captured images could be used for testing a variety of network parameters without having to repeat data capture. Network training was performed on four object instances of each of the six object classes. Data of a fifth object instance per class was additionally captured to test the generalization performance of learning. The deep learning training was performed using an NVIDIA GTX 1080 GPU.

To test the benefit of capturing in-domain data over utilizing pre-constructed datasets, we compared our approach against training on 1000 random images of the six object classes scraped from Flickr using the scraping implementation of Hays and Berg [33]. Images scraped from the web form a good baseline, as many household objects do not appear in specially constructed datasets. Baseline training was additionally used for initializing further training. We also tested against training on a subset of images corresponding to the selected object classes from a pre-constructed object dataset (the RGB-D dataset of Lai, et al. [14]) to represent scenarios in which such data is available. For this dataset, we similarly trained a network on four instances of each of the six object classes, leaving a fifth instance of each for testing generalization. Furthermore, while the RGB-D dataset contains both depth and RGB image data, we constrain our experiments to only using images for simplicity. Additionally, to test the benefits of our iterative learning approach, we compared resultant classifier performance from our method against batch training on a pre-specified number of images for each object instance.

V. Results and Discussion

To test our trained networks, we used all images for all five instances of each object class for both our captured data and the pre-constructed RGB-D dataset. This case tests a network's general performance across images in a target domain. While images from the corresponding sets appear in both training and testing, we note that we are interested in overfitting to a user's personalized environment. Additionally, we test our networks on the only held out instance of each object class to test generalization performance regarding related images in the user's general domain.

In Figure 4 and Table I, we compare the effect of training set on test set performance. We see that our results validate the proposition that training on domain-relevant data matters. We see that training on our captured data (using either batch training on all 100 images of each instance or our iterative approach) leads to significantly higher test performance on captured data when compared to training on either the Flickr or RGB-D baselines (roughly 70-80% test accuracy as opposed to around 40%). This is true when tested on all captured data as well as the unseen fifth instance of each object class. Similarly, we see that general test performance on the pre-constructed RGB-D dataset is highest when trained on related data. Strangely, the learning model seemed to highly overfit when trained on the RGB-D dataset, with high test performance not translating as well to the held out instances. In Table I, we additionally compare the number of images in each training set and the time it took to train our classifier on the associated dataset. Training time scaled linearly with number of images, with both baselines and the batch training over captured images taking orders of magnitude more time

TABLE I

TEST PERFORMANCE ON SPECIFIED TEST SETS AS WELL AS NUMBER OF IMAGES AND TRAINING TIMES FOR THE VARIOUS TRAINING SETS TESTED.

	Captured: All	Captured: Held Out	RGB-D [14]: All	RGB-D [14]: Held Out	Number of Images	Training Time (min)
Flickr baseline	38.0 ± 2.0%	37.2 ± 3.1%	69.3 ± 4.3%	67.7 ± 4.0%	6000	108.0 ± 0.0
RGB-D [14]	41.3 ± 1.2%	40.1 ± 0.8%	91.6 ± 0.5%	60.5 ± 2.4%	14782	261.0 ± 0.4
Batch captured	80.9 ± 1.0%	80.3 ± 1.7%	60.2 ± 2.1%	66.1 ± 3.1%	2400	42.5 ± 0.1
Iterative selection	70.71 ± 3.5%	71.5 ± 4.6%	60.6 ± 6.0%	63.8 ± 8.4%	142.7 ± 5.9	2.1 ± 0.1

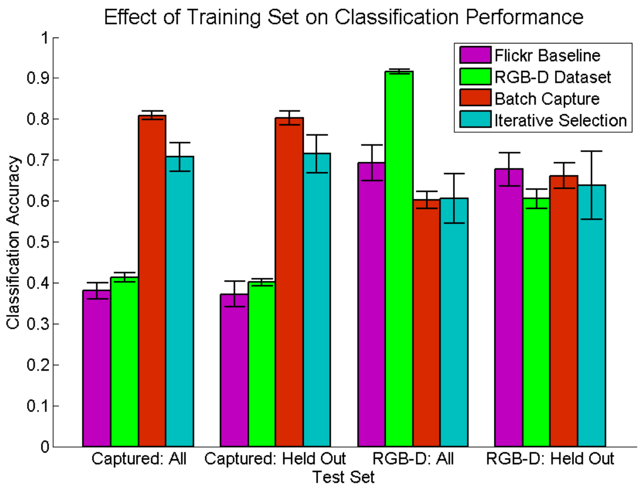


Fig. 4. Test accuracy on captured and RGB-D dataset data when learning models are trained on the Flickr baseline, RGB-D dataset images, and captured data. Training on in-domain data leads to a marked improvement over training on other datasets. Interestingly, training on the RGB-D dataset seemed to greatly overfit such that the higher test performance did not translate to the unseen examples within the dataset.

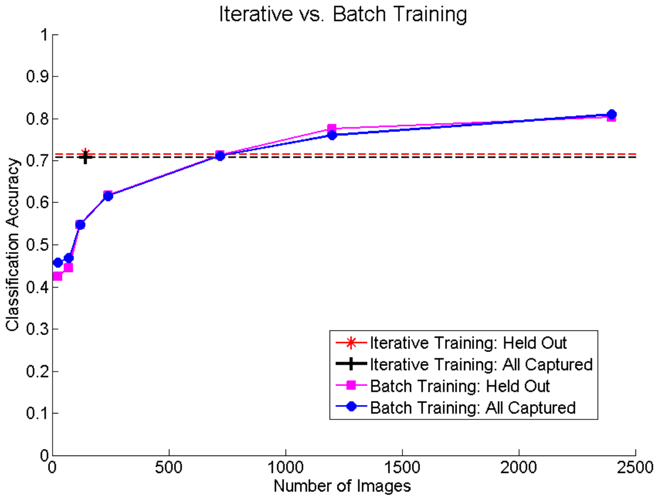


Fig. 5. Captured data test performance when the classifier is trained either iteratively or in batch on captured data. Batch performance is reported for a number of pre-specified number of images to train on. The iterative method reaches a nice trade-off between classifier performance and number of stored images, even outperforming the batch method when trained on a comparable number of images.

to train compared to our iterative approach due to their much larger size.

Figure 5 further illustrates the computational savings of our iterative training method compared to the batch approach. Classifier accuracy is reported for batch training over a specified number of images. We see that training on a larger number of images has diminishing returns, with an elbow in the curve forming in the range of a few hundred images. In comparison, we see that our iterative approach selects a number of images that roughly corresponds to the elbow in the curve. Additionally, this selection method outperforms the batch method when trained on a similar number of images. Since the iterative method is not constrained to treat each object instance equally, it is able to direct more training resources to objects it has a harder time learning, halting the learning only once a given object instance has been learned adequately. In this way, the iterative approach is able to achieve competitive performance with applying the batch approach over all captured images while using an order of magnitude fewer images (bringing network training time down from almost an hour to a couple of minutes in this case). The storage and computational savings associated with the iterative approach will allow it to scale much better to a lifelong learning scenario, allowing the framework to learn a much greater number of object classes given similar space and computation constraints.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we have presented a framework by which a human user can direct a robot to learn objects in his or her personal environment with relatively little effort. As shown by our results, such a method for capturing domain-relevant data leads to significantly higher classification performance compared to using available web-based datasets. By leveraging the ROS framework, our data capture method can be easily integrated with a large number of supported robots. We plan on releasing our framework implementation as a ROS package to further support end users.

Furthermore, this data capture framework allows us to begin investigation into the continual learning of object classes. In this paper, we have presented our first steps in this regard by lessening dataset burdens on storage and computational resources. By keeping track of learning performance over a larger collection of sampled images and only iteratively growing the training set while classification performance is increasing, our method is able to achieve good learning performance using relatively few images. While our method requires the robot to capture a number of images that are not actually added to the training set, the compounded time and space savings in retraining over a robot's lifecycle are more important. However, while this approach scales better over time than naively capturing a pre-specified number of images per instance, it still involves an unbounded collection of images. In future work, we are interested in investigating methods that are more explicit in maintaining representative subsets of each object class. By continually filtering such subsets to better represent captured data, training sets can be adjusted to meet storage or computational constraints. We are also interested in integrating this work with recent model architectures for lifelong deep learning in a longterm object learning experiment.

In future work, we would also like to make a number of improvements to the data capture implementation itself. Since the current approach relies completely on the robot's ability to localize to find specified objects, objects can appear out of frame when localization is erroneous. We would like to integrate this approach with image-based object trackers to make sure an object stays in view. Furthermore, while random uniform sampling of object views can be expected to represent the learning space fairly well, we are interested in testing whether more explicit view planning methods (e.g., space filling approaches, geometric planning around observable object volume) result in more representative datasets.

Acknowledgment

This work was supported by a NASA Space Technology Research Fellowship.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS 2012)*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016, pp. 770–778.

[3] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, 2015, pp. 3156–3164.

[5] S. Guadarrama, E. Rodner, K. Saenko, N. Zhang, R. Farrell, J. Donahue, and T. Darrell, "Open-vocabulary object retrieval." in *Proc. Robotics: Science and Systems (RSS 2014)*, vol. 2, no. 5, 2014, p. 6.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. IEEE, 2009, pp. 248–255.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. European Conference on Computer Vision (ECCV 2014)*, 2014, pp. 740–755.

[9] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010, pp. 998–1005.

[10] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.

[11] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*. Springer, 2013, pp. 387–402.

[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Proc. European Conference on Computer Vision (ECCV 2014)*, 2014, pp. 345–360.

[13] Y. Sun and D. Fox, "Neol: toward never-ending object learning for robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA 2016)*, 2016, pp. 1621–1627.

[14] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Proc. IEEE International Conference on Robotics and Automation (ICRA 2011)*, 2011, pp. 1817–1824.

[15] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition." in *Proc. International Conference on Machine Learning (ICML 2014)*, vol. 32, 2014, pp. 647–655.

[16] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1329–1335.

[17] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale, "Object identification from few examples by improving the invariance of a deep convolutional neural network," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4904–4911.

[18] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, "Compete to compute," in *Proc. Advances in Neural Information Processing Systems (NIPS 2013)*, 2013, pp. 2310–2318.

[19] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, p. 201611835, 2017.

[20] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," *arXiv preprint arXiv:1607.00122*, 2016.

[21] B. Goodrich and I. Arel, "Mitigating catastrophic forgetting in temporal difference learning with function approximation," in *Proc. Multidisciplinary Conf. on Reinforcement Learning and Decision Making*, 2015.

[22] L. S. Lopes and A. Chauhan, "How many words can my robot learn?: An approach and experiments with one-class learning," *Interaction Studies*, vol. 8, no. 1, pp. 53–81, 2007.

[23] M. Villamizar, A. Garrell, A. Sanfeliu, and F. Moreno-Noguer, "Modeling robot's world with minimal effort," in *Proc. IEEE International Conference on Robotics and Automation (ICRA 2015)*, 2015, pp. 4890–4896.

[24] ——, "Interactive multiple object learning with scanty human supervision," *Computer Vision and Image Understanding*, vol. 149, pp. 51–64, 2016.

[25] P. Azagra, Y. Mollard, F. Golemo, A. Murillo, M. Lopes, and J. Civera, "A multimodal dataset for interactive and incremental learning of object models," in *Proc. IEEE International Conference on Robotics and Automation (ICRA 2017)*, 2017.

[26] G. H. Lim, M. Oliveira, V. Mokhtari, S. H. Kasaei, A. Chauhan, L. S. Lopes, and A. M. Tomé, "Interactive teaching and experience extraction for learning about objects and robot activities," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2014)*, 2014, pp. 153–160.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[28] D. Hershberger, D. Gossow, and J. Faust, "Rviz." [Online]. Available: http://wiki.ros.org/rviz

[29] E. Marder-Eppstein, "Ros navigation." [Online]. Available: http://wiki.ros.org/navigation

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016, pp. 2818–2826.

[31] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io/

[32] Flickr. [Online]. Available: http://www.flickr.com

[33] J. Hays and T. Berg. Code for finding and downloading images on flickr. [Online]. Available: http://graphics.cs.cmu.edu/projects/im2gps/flickr_code.html