Incremental Online Learning of Objects for Robots Operating in Real Environments

Jose L. Part
HRI Lab, Heriot-Watt University
Edinburgh Centre for Robotics
Edinburgh, Scotland, UK
Email: jose.part@ed.ac.uk

Oliver Lemon
HRI Lab, Heriot-Watt University
Edinburgh Centre for Robotics
Edinburgh, Scotland, UK
Email: o.lemon@hw.ac.uk

***Abstract*—The ability of an object classifier to adapt to new data and incorporate new classes on the fly is of paramount importance for robots operating in the real world. This paper presents an approach for incremental online learning of real-world objects to be used by robots operating in real environments. We combined the representational power of Convolutional Neural Networks with the adaptability features of Self-Organizing Incremental Neural Networks. We evaluated our approach on the RGB-D Object Dataset in terms of classification accuracy and incremental learning of new classes. Our results show that whereas our method does not yet compete with the performance of state-of-the-art batch learning algorithms, it offers the important advantage of being able to adapt to new data and incorporate new classes on the fly. Finally, we aim at establishing a baseline on a publicly available dataset for comparing different approaches to realize online incremental learning in the context of robotics.**

I. INTRODUCTION

Whereas the object recognition problem has been extensively studied, the related problem of incremental online object learning has not received similar attention. This problem is particularly relevant in robotics since robots are expected to operate within a wide variety of unstructured environments and interact with virtually an infinite number of objects. Thus, assuming full prior knowledge of the environments or the objects involved is not sensible. A robot operating in a hospital is likely to require a different knowledge base than a robot keeping the inventory or restocking the shelves in a supermarket. Moreover, the latter is more likely to require to update its knowledge base more often as new products become available.

Deep learning architectures have shown impressive performance on the object recognition task, e.g., [1], [2], [3], [4], [5], but their application in robotics is still limited due to the requirement of large amounts of data, lots of computational power and the fact that they are trained with batch learning methods. Thus, if new data becomes available or we need to add a new class, the model has to be retrained.

In this work we propose an approach that combines the great representational power provided by deep Convolutional Neural Networks (CNNs) with the Self-Organizing Incremental Neural Network (SOINN) [6], an algorithm for incremental online learning that is able to incorporate new classes and adapt to new information dynamically.

The remainder of this paper is structured as follows: section II provides a summary of the related work, section III introduces the proposed approach, section IV presents the evaluation protocols and the results obtained, section V proposes directions for future work and finally, we conclude our work in section VI.

II. RELATED WORK

Currently, there is a high volume of work addressing object recognition for robotic applications, e.g., [7], [8], [9], [10], but they do not deal with non-stationary data and unknown categories, inherently present in the real world. These works mainly focus on achieving a good performance on a set of predefined objects but do not address the problem of learning new objects once the system is in operation. Most of the work in object recognition is based on off-line batch learning methods and assumes prior knowledge of the number of classes. This offers little flexibility, i.e., if something changes, the models need to be retrained.

Another body of work focuses on transfer learning techniques, e.g., zero-shot learning [11], [12] or one-shot learning [13]. These methods normally perform attribute-based classification and hence, rely on the availability of pre-trained attribute classifiers. In our work, we make the assumption that this "prior knowledge" is not available.

Skočaj et al. [14] proposed a system capable of incrementally learning object attributes through interactive dialogue with a tutor and showed high accuracy in the testing phase. However, the number of attributes was limited to 8 colours and 2 shapes and hence, it is difficult to assess how well the approach would scale to a bigger domain.

Pasquale et al. [15] addressed the problem of incremental learning but from a different perspective. They refer to incremental learning as the ability to update the model over time but only for those classes that were learned in the beginning.

Most of the literature treats the term incremental to refer to models that are capable to adapt to sequential (non-stationary) data, i.e., given a trained model, tune its parameters in order to represent the current distribution of the data. Our interpretation of incremental online learning on the other hand is closely related to the problem of life-long learning [16], i.e., being able

to incorporate and adapt to new information without destroying (forgetting) previously acquired knowledge.

Hence, in this work, we are not only interested in being able to adapt the model for known objects but also in being able to incorporate new categories to our model.

III. PROPOSED APPROACH

Our proposed approach consists of two main components, a preprocessing pipeline and a recognition pipeline. In the preprocessing step, the input data is prepared to be fed into the object recognition subsystem.

We make the following assumptions:

- The objects have been segmented out using depth information by a previous module.
- The images have been cropped around the object using the segmented region of interest.

In the following sections, we describe each of these components in more detail.

A. Preprocessing Pipeline

Before making the input data available to the recognition pipeline, we process it to match its format to the one expected by the CNN and to reduce the influence of the image background. The former is particularly important for the depth channel since the CNN expects a colour image as input.

To process the depth channel, we take the depth map, which encodes the distance to the camera in each pixel, and pass it through a recursive median filter, as proposed by Lai et al. [7], to eliminate missing values (stored as 0). The filter is applied only on the missing values by extracting the median of the non-missing values in a 5×5 kernel. Since most missing values are found in groups and near the border of objects, by iterating over them in order, we would be filling the filtered depth map always in the same direction, which would result in expanded or shrunk object edges. Hence, we iterate over the missing values randomly to compensate for this effect until there are none left.

Then, we compute and colourize the surface normals as suggested by Maday-Tahy et al. [17]. Surface normals provide more information about the structure and shape of objects than the mere depth map, and thus, are expected to lead to better classification results, particularly for category recognition. First, we compute the gradients of each pixel in the depth map using a Sobel filter with a 3×3 kernel in both directions (x, y). Then, we build the tangent vectors to the surface in each direction using the gradients and compute the normal as their cross product. Each component of the surface normals is then scaled to the range $[0, 255]$ and mapped to an RGB channel: $n_x \rightarrow R$, $n_y \rightarrow G$, $n_z \rightarrow B$.

Once the surface normals have been computed, we square and resize both the RGB and the surface normals images to the size expected by the CNN. We square the images using the approach proposed by Eitel et al. [10], which consists of replicating the image borders on both sides along the shorter dimension such that the object remains in the centre of the image.

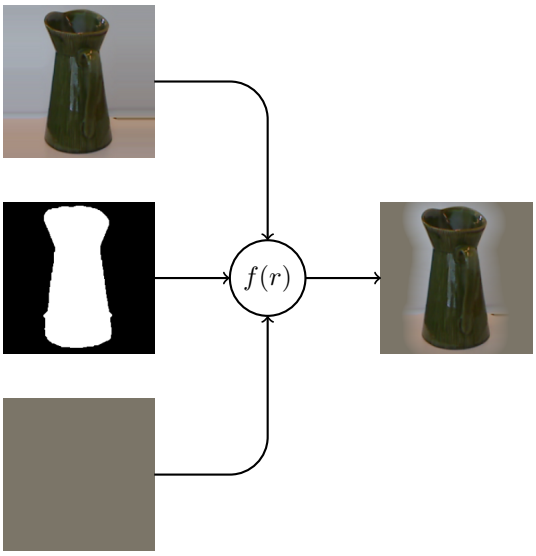

Fig. 1. Background fading operation. The RGB image background is faded into the *ImageNet ILSVRC12* mean image based on the binary mask and the function $f(r)$ in (2).

Finally, we apply a fading operation on the images to reduce the CNN response to the image background, similar to what Schwarz et al. [9] suggested. Essentially, we combined the resized image with the mean image computed over the training set used for the *ImageNet ILSVRC12* challenge using the interpolation scheme in (1) and the binary mask of the depicted object.

$$\mathbf{p} = f(r) \cdot \mathbf{p_0} + (1 - f(r)) \cdot \mathbf{p_m} \tag{1}$$

where $\mathbf{p}$ represents a pixel of the resulting image, $\mathbf{p_0}$ represents a pixel from the RGB image, $\mathbf{p_m}$ represents a pixel from the mean image, and $f(r)$ is defined as:

$$f(r) = \begin{cases} 1 & \text{if } r = 0 \\ 0 & \text{if } r > R \\ \frac{(R-r)^\beta}{R^\beta} & \text{otherwise} \end{cases} \tag{2}$$

where r is the distance from the current background pixel to the closest foreground pixel, β is set to 0.75 and R is set to 30 for the RGB image and 20 for the surface normals image.

The fading operation is illustrated for the RGB image in Fig. 1 and the full preprocessing pipeline is summarized in Fig. 2.

B. Recognition Pipeline

Our recognition pipeline is composed of a feature extractor and an incremental online classifier. In recent years, deep network architectures have become ubiquitous in computer vision tasks, particularly for object recognition/classification. They offer great representational power but they also come with inherent limitations. In general, these type of architectures require enormous amounts of data and take a considerable amount of time to be trained. They fall under the category

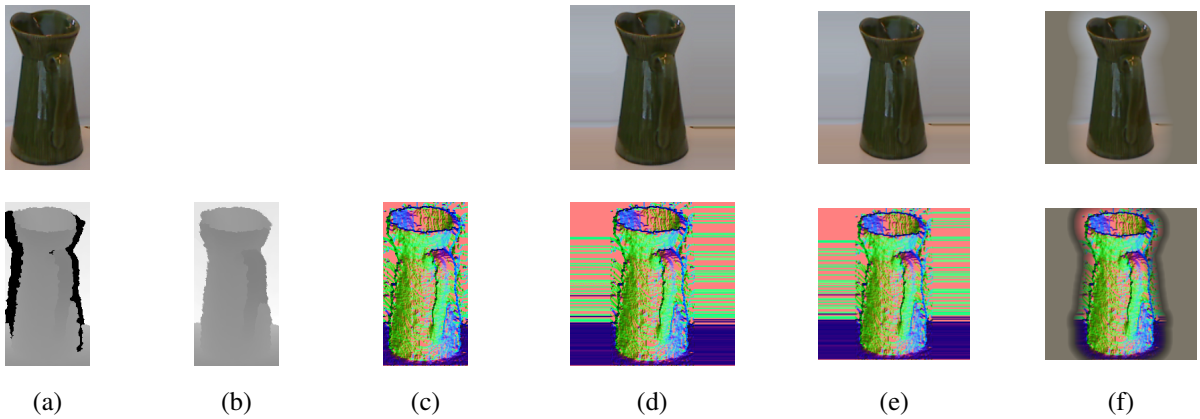

Fig. 2. Preprocessing pipeline. The upper row shows the preprocessing steps performed on the RGB image aligned with the preprocessing steps performed on the depth map shown in the lower row. The steps are (a) input image, (b) depth map filtering, (c) computation of surface normals, (d) squaring, (e) resizing and (f) background fading.

of batch learning methods, which means that they have to be trained iteratively on the full dataset, and the number of classes need to be defined in advance. All these factors render their use in robotic applications quite limited.

Fairly recently however, many research groups, e.g., [18], [19], [20], showed that it is possible to use CNNs that have been trained on massive datasets as off-the-shelf feature extractors for a variety of different tasks obtaining state-of-the-art results. This has the advantages that we no longer need to have access to huge amounts of data or the time and resources to train these models, but more importantly, that CNNs are capable of learning high-level semantic features that are more representative than hand-engineered features and do not require the same level of expertise to be developed.

For our experiments we selected the popular AlexNet pre-trained model based on the network proposed by Krizhevsky et al. [1] and publicly available in the *Caffe Model Zoo* [21]. This network was trained on a subset of the *ImageNet* dataset and is composed of five convolutional layers and three fully-connected layers. The last fully-connected layer is a softmax layer that outputs a probability distribution over all the classes. Since we are not interested in the classes for which the network was originally trained, we discarded the last layer. After running tests using the feature vectors obtained after the last convolutional layer and the two remaining fully-connected layers, we decided to use the features obtained after the last fully connected layer because these led to better classification results.

In order to deal with the particular nature of our problem, we based our approach on the Load-Balancing version of the Self-Organizing Incremental Neural Network (LB-SOINN) proposed by Zhang et al. [6]. LB-SOINN is an unsupervised learning method inspired by the Self-Organizing Map [22]. Each node in the network has an associated weight, which lives in the feature space of the data. Every time a new signal (feature vector) becomes available, the algorithm assesses whether a new node should be added to the network based on a similarity metric between the input signal and the weights of the two nearest nodes. If no new node is added, the weights and connections of the existing network are updated. In this manner, the topology of the network is continuously evolving to reflect the distribution of the input data.

Despite the fact that SOINN is an unsupervised learning method, we found that for our data, the clusters reported by the algorithm were not reliable. Hence, we decided to adapt it such that we could use it as a supervised method under the assumption that in a real robot learning scenario, it is likely that a "tutor", who can provide labels, is available.

Our recognition pipeline has two channels, one for the RGB image and one for the surface normals image. Once the feature vectors have been computed by each CNN, these are combined into one single vector that is used for training the incremental classifier. Fig 3 shows the full recognition pipeline.

IV. EVALUATION

We evaluated our approach in terms of classification accuracy and incremental learning of new objects. The classification accuracy is reported for both categories and instances whereas the incremental learning of new classes is reported only for categories.

In the following sections we describe the dataset used and the evaluation protocols along with the obtained results.

A. RGB-D Object Dataset

For the evaluation of our approach, we used the RGB-D Object Dataset [7]. This dataset contains 300 household objects organized into 51 categories. It was acquired with a Kinect-like camera by placing each object on a turntable and recording video sequences of a full rotation from three different heights of the camera at 30 Hz. As a result, there is a rich availability of data for each object. In addition to the RGB and depth images, the dataset also contains cropped versions around the object and binary masks obtained through automatic segmentation.

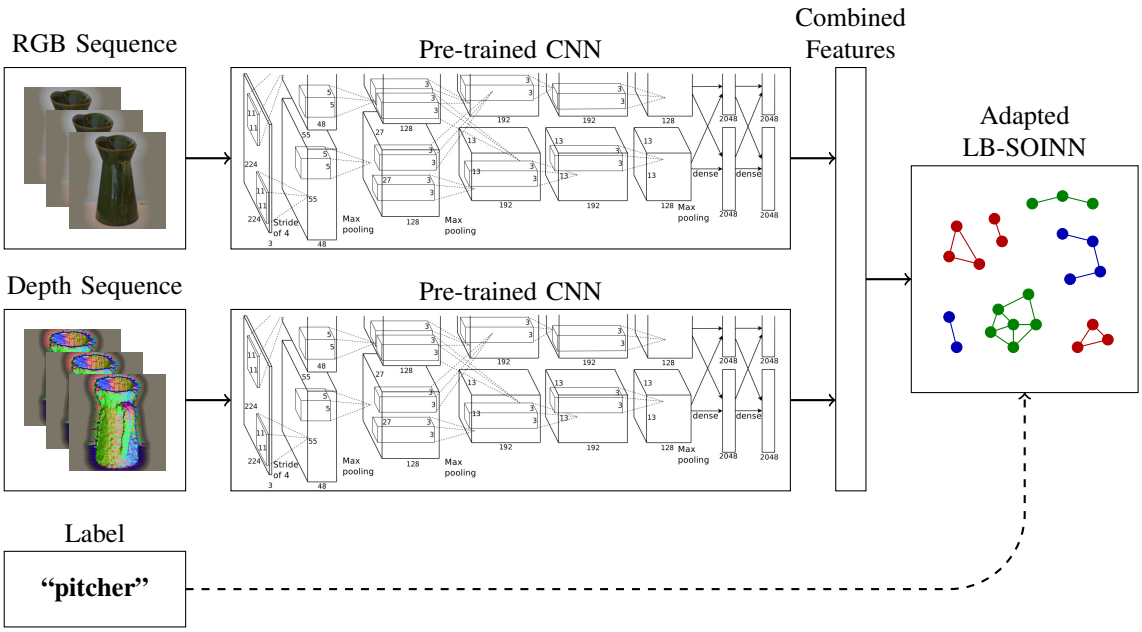

Fig. 3. Recognition pipeline. Preprocessed RGB images and colourized surface normals of the target objects are input sequentially to the corresponding CNN (the same CNN is used for both modalities), which outputs the computed feature vector. These feature vectors are then combined and the end result is used to train an adapted version of the Load-Balancing SOINN classifier [6] which learns the underlying topology of the data. During training, each example is accompanied by its label which can be obtained from interacting with a human tutor.

The intrinsic nature of this dataset makes it a very useful benchmark for robotic applications since for every object, there are multiple views and modalities available. Moreover, it is arranged hierarchically, i.e., every instance is associated to a category. This allows to study not only the problem of category recognition, which involves classifying new objects into the corresponding category, but also the problem of instance recognition, which involves classifying a previously seen object into the corresponding instance.

B. General Considerations

In order to evaluate our system, we considered two cases: using the full dataset and a subsampled version where we took every fifth frame of the video sequences, as proposed by Lai et al. [7]. The first case assumes that the processing of the data can cope with the high rate in which the data becomes available and gives a measure of the upper bound in accuracy that the method can reach. The second case is more realistic in terms of real-time processing capabilities since it effectively reduces the frame rate to 6 Hz ($\approx$ 167 ms per frame).

Since we are interested in the use of this system for robotic applications, the training is performed per instance, which means that consecutive frames correspond to the same object. This in fact simulates a situated interaction where the robot has access to one object at a time and can explore it from different viewpoints. This is fundamentally different from other approaches where batch methods are used and the full data is shuffled to avoid biases in the learning process.

We consider the RGB features, the depth features and two different combinations, the average and the concatenation of the features. The advantage of the former combination is that it reduces the dimensionality of the resulting feature vector, thus reducing the computational time required during training and evaluation. On the other hand, concatenating the features allows for a better representation of the objects since visual features and depth features are kept separated from each other.

C. Classification Accuracy

1) Category Recognition: For evaluating the category recognition task, we used two different strategies to split the dataset. The first one is known as Leave-One-Out (LOO) and it is commonly used in the literature [7], [9], [10]. For every category, one instance is left out for evaluation while the system is trained on the remaining instances. With this strategy, we run 10 experiments selecting one instance at random for every category in each run.

The second strategy consists of splitting the dataset at the image level in 60% for training and 40% for evaluation. This means that the system sees all the instances at training time but it is evaluated on different views (frames). The motivation for using this type of split is to establish an upper bound for the accuracy of our method.

The results are reported in Table I for different combinations of the previous situations. The first three rows correspond to the Leave-One-Out strategy. The first row corresponds to the use of the full dataset whereas the remainder two correspond to using the subsampled version of the dataset. The difference between these two is that for the second experiment we iterate 4 times over each instance, which is equivalent to a longer exposure to the object. The latter is to compensate for the reduction in data points. We show that a longer exposure improves the classification results since the model can form

TABLE I
CATEGORY CLASSIFICATION ACCURACY. "SS" MEANS THAT THE SUBSAMPLED VERSION OF THE DATASET WAS USED AND "4IT" MEANS THAT EACH OBJECT WAS ITERATED OVER 4 TIMES.

	RGB	Depth	RGB-D Average	RGB-D Concat.
LOO	73.7 ± 3.4	70.5 ± 2.3	84.5 ± 2.0	84.0 ± 2.3
LOO (SS)	70.3 ± 5.1	62.3 ± 3.8	78.8 ± 4.1	78.6 ± 4.2
LOO (SS, 4it)	71.9 ± 2.3	62.6 ± 3.0	81.0 ± 1.9	81.3 ± 1.8
Percent.	98.0 ± 1.6	84.0 ± 3.5	98.5 ± 0.6	98.6 ± 0.5
Percent. (SS)	87.5 ± 7.8	66.8 ± 2.7	89.8 ± 7.1	87.8 ± 4.4

more robust structures. The last two rows in the table show the results obtained when using the second splitting strategy (Percentages), also with the full dataset and the subsampled version respectively. As expected, the accuracy is much higher when all the instances are known even if we are evaluating on unseen images (views).

One of the interesting results that we can extract from Table I is that the accuracy obtained when using the average of the features is very similar to the one obtained when concatenating the features. However, the latter results in higher computational time, which is not desirable for real-time operation. Another interesting observation is that when the system is exposed for a longer period to the objects, the final accuracy is higher and the standard deviation lower.

In Table II we show a comparison with batch learning approaches for the task of category recognition on the RGB-D Object Dataset. As it can be seen, our approach still lags behind all the other methods but, as opposed to all of them, it offers the possibility to be trained online and incrementally. This feature is of utmost importance for robots operating in real environments.

2) Instance Recognition: For the evaluation of the instance recognition task we used the Leave-One-Sequence-Out (LSO) data split suggested by Lai et al. [7]. LSO consists of using, for every instance, the two sequences corresponding to the lower and higher heights of the camera for training, and the sequence corresponding to the middle height for validation. Hence, the system sees all the instances during training but it is tested on a sequence of views that it has not seen before. Table III summarizes the results. One particularly interesting result is that the averaged features get a slightly lower accuracy than the concatenated features. Moreover, the RGB features alone get the highest overall accuracy. This is due to the fact that instances of some categories present a high degree of similarity and can be difficult to distinguish from each other. In fact, in most cases, the distinguishing factor is the colour or texture, which are elements not captured by the depth features.

TABLE II
CATEGORY CLASSIFICATION ACCURACY. COMPARISON AGAINST BATCH LEARNING APPROACHES. THE CLASSIFICATION ACCURACY IS REPORTED FOR THE SUBSAMPLED VERSION OF THE DATASET AND THE LEAVE-ONE-OUT PARTITION STRATEGY. IN OUR METHOD, THE RGB-D RESULTS CORRESPOND TO USING THE AVERAGE OF THE FEATURE VECTORS.

	RGB	Depth	RGB-D	Incr.	Online
kSVM [7]	74.5 ± 3.1	64.7 ± 2.2	83.8 ± 3.5	x	x
KD [23]	77.7 ± 1.9	78.8 ± 2.7	86.2 ± 2.1	x	x
HKD [24]	76.1 ± 2.2	75.7 ± 2.6	84.1 ± 2.2	x	x
HMP [25]	82.4 ± 3.1	81.2 ± 2.3	87.5 ± 2.9	x	x
CNN-RNN [8]	80.8 ± 4.2	78.9 ± 3.8	86.8 ± 3.3	x	x
Schwarz et al. [9]	83.1 ± 2.0	-	89.4 ± 1.3	x	x
FusionNet (HHA) [10]	84.1 ± 2.7	83.0 ± 2.7	91.0 ± 1.9	x	x
FusionNet (jet) [10]	84.1 ± 2.7	83.8 ± 2.7	91.3 ± 1.4	x	x
FusionNet (SN) [17]	84.7 ± 3.7	88.0 ± 2.5	94.0 ± 2.4	x	x
Ours	70.3 ± 5.1	62.3 ± 3.8	78.8 ± 4.1	✓	✓
Ours (4it)	71.9 ± 2.3	62.6 ± 3.0	81.0 ± 1.9	✓	✓

TABLE III
INSTANCE CLASSIFICATION ACCURACY. "SS" MEANS THAT THE SUBSAMPLED VERSION OF THE DATASET WAS USED AND "4IT" MEANS THAT EACH OBJECT WAS ITERATED OVER 4 TIMES.

	RGB	Depth	RGB-D Avg.	RGB-D Concat.
LSO	83.1 ± 4.6	29.1 ± 2.0	79.1 ± 2.3	81.7 ± 2.6
LSO (SS)	67.2 ± 13.8	21.1 ± 1.2	59.6 ± 8.9	58.8 ± 7.6
LSO (SS, 4it)	77.7 ± 5.4	22.3 ± 0.8	70.8 ± 6.0	74.0 ± 7.6

D. Incremental Learning

An important contribution of our approach is the ability to train our model incrementally, i.e., to add new classes on the fly as the system interacts with the world. For evaluating our system in terms of incremental learning we used the Leave-One-Out splitting strategy. For every class, we chose one instance at random for evaluation and trained on the remaining instances. As opposed to the previous experiments, the evaluation was performed on all the learned classes immediately after each training on a new class. Hence, we expect a high accuracy at the beginning when only a few classes are known, and a decrease in the accuracy as the number of classes increases. Fig. 4 and Fig. 5 correspond to the incremental

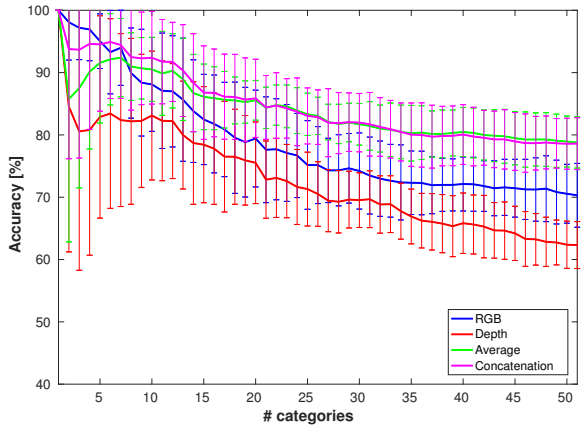

Fig. 4. Incremental classification accuracy. The evaluation was performed on the subsampled version of the dataset over 10 runs, where in each run one instance was selected at random for validation and the remainder were used for training.

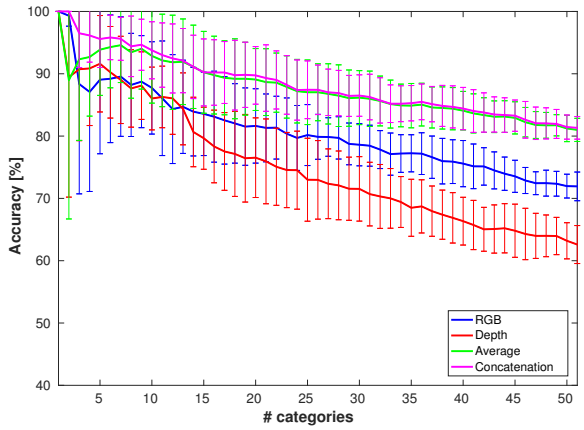

Fig. 5. Incremental classification accuracy. The evaluation was performed on the subsampled version of the dataset over 10 runs, where in each run one instance was selected at random for validation and the remainder were used for training. Every object was iterated over 4 times.

learning evaluation associated with the second and third rows in Table I respectively, and show the evolution of the learning process. It is interesting to notice that despite some minor turbulences in the beginning, the learning becomes quite stable and the standard deviation between different runs decreases as the number of objects increases.

V. DISCUSSION AND FUTURE WORK

In the previous section, we saw that our method has an acceptable performance in category recognition but fails substantially when it comes to instance recognition. This is actually quite understandable and in fact, the results are somehow complementary. On the one hand, a failure in category recognition generally means that there is a high intra-class variation between the instances used for training and the instance used for evaluation. On the other hand, for instance recognition, exactly the opposite holds, i.e., a bad classification result is generally associated with a low intra-class variation. For example, if an unseen instance of the category lemon is very similar to all the other instances in the category, this will result in very good accuracy when classifying it into the respective category. On the contrary, if we instead aim to distinguish it from the other lemon instances, the approach will fail considerably. In many cases, distinguishing between different instances is completely unnecessary but there are some cases in which it can be very important, for example, differentiating between your laptop and someone else's. Ideally, a hierarchical approach like the one proposed by Schwarz et al. [9] would be desirable, since it allows the system to first discriminate categories and, in a second step, discriminate instances. How to accomplish this behaviour in an online manner however is part of our ongoing research.

The main goal of this work was to propose a method for incremental online learning. Hence, it is worth noting that achieving state-of-the-art performance was not the main focus. This is in part reflected by our decision to adopt the CNN as an off-the-shelf feature extractor, i.e., we did not fine-tune any of its parameters. However, doing so, especially for the depth channel, could help considerably to improve the classification accuracy of our approach. Moreover, learning how to combine the features from the different modalities, as proposed by Eitel et al. [10], could be very beneficial. In their work, each channel is fine-tuned independently and, in a second stage, the parameters of a fusion layer are learnt through back-propagation. However, we argue that the training and evaluation of the incremental learning approach should then be done on a different dataset to test how well the features can generalise to different data.

Another potential improvement to the proposed method would be to use a different network for each category as proposed by Kawewong et al. [26]. Despite the fact that preliminary results have shown no significant improvement in the recognition accuracy with respect to the current configuration, we believe that adopting this scheme may aid to avoid destructive interference between classes [16] and facilitate the implementation of other desirable behaviours such as detecting when an object is from an unknown category.

Currently, our method is not able to identify when a new class is presented and relies on the user to provide this information. Future work is concerned with the proposal of approaches to assess when an object of a new class has been encountered and trigger a self-driven mechanism for requesting feedback from the user, thus reducing the overall load to the user (tutor cost).

In our evaluation for the category recognition task, we have made the assumption that all objects from the same category are learnt consecutively. This is definitely not true in a real situation and it would be interesting to evaluate how learning these instances in a random mixed order may affect the overall classification accuracy. Furthermore, it would be interesting to increase the number of instances and categories and observe how the overall accuracy evolves, i.e. whether it converges or keeps dropping below the current value. This is a very

important question in the context of life-long learning.

VI. CONCLUSIONS

In this paper we presented an approach for incremental online learning of objects that harnesses the representational power of deep convolutional neural networks. We showed that our method is capable of incrementally adding new classes and updating its model despite the fact of not being able to compete against the state-of-the-art in object recognition yet. However, we suggested a series of potential improvements that may help in achieving a better performance.

We also identified a few problems with the proposed method. In our work, the clusters reported by SOINN are not reliable. The edges that connect the nodes are nevertheless necessary to provide stability to the network and to be able to keep the number of nodes at a reasonable amount. A solution to this problem was to adapt the algorithm to be used as a supervised method. In a real interaction scenario, this is not a problem because the label is only needed in the beginning since the subsequent images correspond to the same object. Using an object tracking algorithm, the robot could assess whether the object is still the same or whether it has changed.

Due to the way the data is distributed in the feature space, we found that it was necessary to shuffle the images corresponding to the same instance in order for our algorithm to learn robust structures. As exposed by Schwarz et al. [9], the features obtained by the CNN are located in lower-dimensional pose manifolds where similar poses of an object remain close in the feature space as well. This makes it difficult for SOINN to learn robust structures if the data is provided in a purely sequential manner.

Finally, we would like to mention that ultimately, the goal of incremental online learning approaches should not be to replace batch learning approaches but to complement them, i.e., once a robot has been deployed, it should be able to evolve and incorporate new knowledge regardless of the level of coverage and granularity of the in-built recognition system.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their useful comments and remarks.

J. L. Part is supported by a James-Watt scholarship from the School of Mathematical and Computer Sciences at Heriot-Watt University. This work is partially supported by the ROBOTARIUM Grant (EPSRC Grant No. EP/J015040/1).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[2] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," 2014.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] H. Zhang, X. Xiao, and O. Hasegawa, "A load-balancing self-organizing incremental neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1096–1105, 2014.

[7] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset." in *ICRA*. IEEE, 2011, pp. 1817–1824.

[8] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification," in *Advances in Neural Information Processing Systems 25*, 2012.

[9] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D Object Recognition and Pose Estimation based on Pre-Trained Convolutional Neural Network Features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1329–1335.

[10] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015.

[11] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa, "Online incremental attribute-based zero-shot learning." in *CVPR*. IEEE Computer Society, 2012, pp. 3657–3664.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-Based Classification for Zero-Shot Visual Object Categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[13] E. Krause, M. Zillich, T. Williams, and M. Scheutz, "Learning to recognize novel objects in one shot through human-robot interactions in natural language dialogues," in *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[14] D. Skocaj, M. Kristan, A. Vrecko, M. Mahnic, M. Jancek, G.-J. M. Kruijff, M. Hanheide, N. Hawes, T. Keller, M. Zillich, and K. Zhou, "A system for interactive learning in dialogue with a tutor." in *IROS*. IEEE, 2011, pp. 3387–3394.

[15] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, "Teaching iCub to recognize objects using deep Convolutional Neural Networks." in *JMLR Workshop and Conference Proceedings*, vol. 43, 2015, pp. 21–25.

[16] F. H. Hamker, "Life-long learning cell structures–continuously learning without catastrophic interference," *Neural Networks*, vol. 14, no. 4-5, pp. 551–573, 2001.

[17] L. Madai-Tahy, S. Otte, R. Hanten, and A. Zell, *Revisiting Deep Convolutional Neural Networks for RGB-D Based Object Recognition*. Springer International Publishing, 2016, pp. 29–37.

[18] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[19] 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 *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 647–655.

[20] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3320–3328.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 675–678.

[22] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464–1480, 1990.

[23] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 821–826.

[24] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *CVPR 2011*, June 2011, pp. 1729–1736.

[25] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for RGB-D based object recognition," in *International Symposium on Experimental Robotics (ISER)*, 2012.

[26] A. Kawewong, R. Pimpup, and O. Hasegawa, "Incremental Learning Framework for Indoor Scene Recognition," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013, pp. 496–502.