2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Selecting CNN Features for Online Learning of 3D Objects

Monika Ullrich[1], Haider Ali[1,2], Maximilian Durner[1], Zoltán-Csaba Márton[1] and Rudolph Triebel[1,3]

Abstract— **We present a novel method for classifying 3D objects that is particularly tailored for the requirements in robotic applications. The major challenges here are the comparably small amount of available training data and the fact that often data is perceived in streams and not in fixed-size pools. Traditional state-of-the-art learning methods, however, require a large amount of training data, and their online learning capabilities are usually limited. Therefore, we propose a modality-specific selection of convolutional neural networks (CNN), pre-trained or fine-tuned, in combination with a classifier that is designed particularly for online learning from data streams, namely the Mondrian Forest (MF). We show that this combination of trained features obtained from a CNN can be improved further if a feature selection algorithm is applied. In our experiments, we use the resulting features both with a MF and a linear Support Vector Machine (SVM). With SVM we beat the state of the art on an RGB-D dataset, while with MF a strong result for active learning is achieved.**

I. Introduction

Service robots operating alongside people are permanently confronted with changing environments including unknown objects to interact with. Hence, a robust robotic perception system that is able to categorize new object instances into known categories is crucial. However, such dynamic environments lead to an infinite amount of possible future situations which makes an offline training phase infeasible. One solution to counteract this issue, is to update the perception model by including novel/uncertain samples in the training set. This places a specific challenge on the perception system which can be split into two parts. Firstly, the system has to select the few encountered samples that carry information beyond the already learned model, i.e., on which improvements can be made. To this end, the classifier's uncertainty estimate has to be a reliable indicator of model (in)sufficiency such that efficient selection of the relevant additional training samples is secured. Being restrictive with the added training data leads to short online training phases and requires fewer semantic annotations (here category labels) from a human supervisor, is hence generally desirable. Secondly, the system has to be capable of all-over improvement based on that small amount of additional training samples.

In this work we elaborate an *online* and *active* learning method for RGB-D category recognition, i.e. a method where the training process is performed persistently during the operation of the robot and not much data is needed for the training process. Therefore we combine feature sets known for their expressiveness, namely deep-learned features [1], [2], and an online learnable classifier. Most of the approaches dealing with CNNs for object recognition are only using RGB images, and the resulting deep-learned features tend to be highly robust under various circumstances, such as change of scale, viewpoint and illumination. However, as already shown in [1], depth data can also add valuable information.

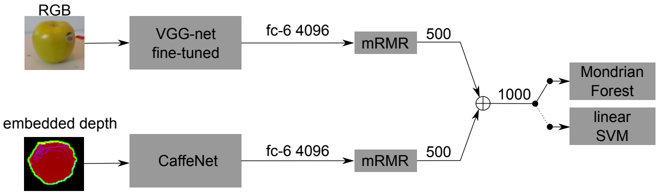

Fig. 1. Overview of the proposed object recognition pipeline. The 500 most discriminative dimensions of the features extracted from CNNs (for RGB and depth data) are selected and concatenated before classification.

Therefore, the approach presented in this work combines the RGB and depth features as outlined in Figure 1, which leads to a performance boost. These are independently extracted, as our experiments showed that different architectures work better for the different modalities. Then the resulting feature vectors are concatenated and used for classification. We chose two CNN architectures, namely the BVLC CaffeNet model [3] and the 19-layers VGG-Net model [4]. While VGG-Net outperforms CaffeNet on standard image recognition tasks, the design choice to reduce convolution sizes (in favor of more layers) is specifically optimized for RGB data. Depth images, and standard 3-channel embeddings of it, have considerably lower "frequencies", thus larger window sizes are an advantage. Additionally, while it makes sense to fine-tune a CNN pre-trained on ImageNet even with very different RGB information [5], depth data is just too different and training from scratch [6] or just using the pre-trained features is preferable (here we opted for the latter). Narr *et al.* [7] showed a promising *stream-based active learning approach* – a particular form of online learning – applying a so-called *Mondrian Forest* (MF) [8] as classifier. However, as we will show experimentally, the MF is not well suited for dealing with the high dimensionality of features extracted by the deep networks. To counteract this effect, several feature selection approaches are evaluated, which additionally leads to a much shorter learning time.

This work consists of several parts, such as feature selection, deep-learned features for object category recognition as well as the proposed Active Learning (AL) approach. We also conducted our experiments with a Support Vector Machine (SVM) as a baseline, as it is well-known to achieve

[1] Institute of Robotics and Mechatronics, Dep. of Perception and Cognition, German Aerospace Center (DLR), 82234 Weßling, Germany {<firstname.lastname>}@dlr.de

[2] Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA {hali@jhu.edu}

[3] Dep. of Computer Science, TU Munich, 85748 Garching, Germany {triebel@in.tum.de}

978-1-5386-2682-5/17/$31.00 ©2017 IEEE

high performances combined with CNN features [9], [5]. In contrast to [10], here the feature selection improved even the SVM results when concatenating the RGB and depth components. Applying the SVM classifier to our feature set, we were able to outperform state-of-the-art methods on the RGB-D benchmark provided by Lai *et al.* [11]. Although such performance was not reached with the MF classifier, our main focus is to use deep-learned features in combination with AL through MF for the application in robotics.

II. Related Work

Nowadays, applying CNNs on RGB sensor data for object recognition is a common practice. However, as inter alia Lai *et al.* [11] already showed for hand-crafted features, depth data could add valuable information to improve the recognition performance. This is why, recently several publications deal with the integration of depth information into a CNN pipeline. In most of the cases, a CNN trained on RGB data is adapted to be suitable for depth data. The strategy for this varies between fusing the information before [1], [12], [13], [14] and after the classification [15]. Combining before classifying means, the concatenation of features or some other kind of feature fusion. Otherwise the classifier predictions have to be ensembled either by naive methods or an additional classifier as shown in [16]. Since pre-trained CNNs normally expect 3-channel RGB input data, the depth image has to be preprocessed. Zaki *et al.* [1], [12] have added one channel describing the magnitude of the gradients, the other describing the direction of the gradients. Wang *et al.* [13] compute the surface normals from the depth images and combine them with the original depth image to a 3-channel input. Song *et al.* [6] also use the normals and their relation to a (for their application) known upright axis. In another work Wang and Siddiqi [9] use separate CNNs for pre-processed depth data. One of them uses surface normals as an input, the other curvatures which are the first and second order derivatives of the depth, respectively. They do not consider the raw depth values at all.

In robotic applications we are faced with limited amounts of training data, hence, a straightforward training of a CNN poses difficulties. This leads to the pragmatic approach of using one of the several publicly available pre-trained CNNs [3], [4] as feature extractors, which outperforms highly tuned state-of-the-art handcrafted features [17]. Another solution when confronted with limited data is an updateable classifier and/or an AL framework. A good overview of this topic is given by Settles [18]. In the work of Wang *et al.* [19] a Gaussian Process Classifier (GPC) performed image segmentation in an active manner. Kapoor *et al.* [20] also used a GPC to perform object categorization. Yang *et al.* [21] proposed a modified binary SVM which is also adaptable; in our earlier experiments [22] for the multi-class extension, however, the computational gain was largely lost.

III. Methodology

An adapting robotic system requires interaction with the human, but it should request input from the user as rarely as possible, which leads to the question of how to select the data samples to be annotated. In the following we will present two selection strategies, then the CNNs used for feature extraction, and their use for category recognition.

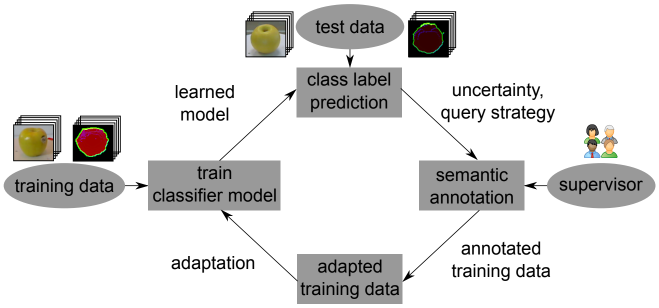

Fig. 2. General flowchart of an AL framework.

A. Active Learning with Mondrian Forest

The goal of AL is to use less data for training while achieving similar accuracies with the reduced data. The general flowchart is shown in Figure 2.

Initially the underlying classifier is trained with some initial training set before the learning model enters into the continuous update loop. Given the classifier model for each test sample, the probability distributions over the possible class labels are computed, which results in the corresponding class label prediction. In the next step a query strategy is applied to obtain new training data which is a subset of the former test data set. After the semantic annotation by the user, the annotated training data is used to adapt the classifier model. This can be either retraining the classifier with the adapted training data or updating the model online. While the retraining requires the storage of all training samples gathered so far, the online manner is only in need of the newly selected training data, requires reduced computational time. To sum up, the goal of AL is to use less time, less samples and less memory while the resulting classifier should be as good as the one trained with all the data. It is also possible that the resulting classifier is even better when training on the chosen subset of the training data instead of using all data. One of the reasons for this is that over-fitting can be reduced when less similar data is used. Additionally, the classifier is updated with exactly those samples that are uncertain, resulting in a better classifier in terms of confidence estimation. However, everything hinges on the data selected as adaptation data.

We refer to this as the *query strategy*. In his book, Settles [18] summarized various query strategies. Among these, the mostly applied one is the *uncertainty sampling*. As basis the uncertainty h has to be computed. Given the probability distribution $\mathbf{p}$ over class labels for test sample $\mathbf{x}$, we compute the so-called *best-vs-second-best* (BVSB) measure: $h(p) = p_{i_2}/p_{i_1}$ where p_{i_1} represents the highest, p_{i_2} the second highest probability. Based on the uncertainty, one can define the classification confidence as $1 - h$.

Given the confidences, in this work two different query strategies are applied, which will be introduced in the following. In the standard uncertainty sampling (denoted as AL_{thres}) a confidence threshold ϑ decides whether the sample is considered as additional training data $(1 - h < \vartheta)$.

The second criterion for AL (denoted as AL_{perc}) organizes the data into batches and sorts the data within each batch according to the confidence of the classifier. A certain percentage per batch is selected for the additional training data, meaning that the most uncertain samples in this batch will be used. Since both approaches show different pros and cons, both are considered in the experimental section.

B. *Modality-Specific Convolutional Neural Networks*

We use two ImageNet pre-trained CNNs namely VGG-19-layers-model [4] (in the following denoted as *VGG-net*) and the BVLC CaffeNet Model [23], [3] (in the following denoted as *CaffeNet*) for feature extraction. Pre-trained CNNs can easily be reused for other datasets, if the target dataset is somehow similar to the one the network was trained on [1]. As our target dataset contains RGB images of object categories, as in ImageNet, these CNNs are suitable for our purposes. The VGG-net [4] consists out of 19 layers including 16 convolutional (conv.) layers, three fully-connected (fc) layers followed by a softmax layer at the end. The CaffeNet [3], inspired by AlexNet [23], consists of five (larger) conv. layers as well as three fc layers.

Fine-tuning the network is a good possibility to get more discriminative features with respect to the desired object classes when the source and target data is similar enough. To this end, the fc layers are initialized randomly and their weights are adapted based on the performance of the softmax output on a given validation set. Several evaluation sets are obtained by applying the leave-one-out method, and in each split the full training data is divided into a training and validation set. In the following the ImageNet pre-trained networks are indicated as *pre-trn* and the adapted ones as *fine-tnd*. However, in the case of depth, the ImageNet data is too dissimilar, and fine-tuning results in so called *catastrophic forgetting*, i.e., the original generality of the feature maps is lost [6], hence no fine-tuning was used.

C. *Features for the RGB-D Category Recognition Pipeline*

Figure 1 shows the recognition pipeline of our approach. The RGB image is forwarded into the fine-tuned VGG-net and the features from a fc layer are extracted. In parallel, a pre-processed depth image, the so-called *embedded depth* image, is fed into a pre-trn CaffeNet. Specifically, the depth data is augmented from one to three channels as in [1], since this is the required input format for the network. One channel contains the magnitude of the gradients, the other describes the direction of the gradients. Also in this case the features are extracted from a fc layer. Since high dimensionality leads to a high computational load, and in the case of MF to a decrease of performance, the 500 best features are selected independently for both input modalities. The features are then concatenated, and a SVM as baseline or a MF are applied to predict the class label of the current test sample. During the experiments several modifications in terms of used fc layer, feature selection method and type of learning (active or not) are shown.

TABLE I

ACCURACY INFLUENCE OF CNN TYPE, INPUT DATA AND FC-LAYER (CORRESPONDING FEATURES OF BOLD CONSIDERED IN EXPERIMENTS)

Network Type	Input Data	fc-6	fc-7	fc-8
CaffeNet	Depth-embedded	**75.1**	74.7	72.1
	Mask	68.0	66.5	64.3
	RGB	81.5	79.7	76.6
VGG-net	Depth-embedded	73.4	71.7	68.9
	Mask	65.9	63.3	60.5
	RGB	**88.7**	86.5	84.5

IV. EXPERIMENTS AND EVALUATION

In this section we want to evaluate the performance of our online learning approach. We conducted two entry experiments for defining the optimal setup namely the best layer for feature extraction and the best performing feature-selection approach. Since in both experiments the expressiveness of the CNN features should be analyzed, a well-known linear SVM is used as classifier, and results should be transferable to other classifiers, e.g. the MF.

Throughout the experiments the publicly available RGB-D Object Dataset [11] is used, since it is a multi-view dataset that is widely used for benchmarking in 3D object recognition. The dataset can be used for object category and object instance recognition, but we only consider object category recognition in this work. It consists of 300 household objects in 51 categories that were captured using the Microsoft Kinect camera. For all conducted experiments we use the 10 fold data splits following the leave-one-out procedure as proposed by [11]. As also proposed by the authors, only every 5^{th} image is taken into account.

A. *Feature Selection*

The described dataset is used to extract features from all the fc layers of the network to find the most suitable layer representing our data. Features are extracted using both CNNs (CaffeNet and VGG-net) for RGB, the binary mask of the objects and for the introduced embedded depth image with the binary mask applied to it. Testing the features for one split with the linear SVM, the most suitable network as well as the best fc layer for RGB and depth data can be defined. The results in Table I show that (the most abstract/generic) fc-6 layer provides the best features, no matter what input image we use. For RGB the best results are obtained by the VGG-net, while for embedded depth by CaffeNet (thanks to the larger conv. window sizes). The following experiments will only consider this configuration. We do not consider the binary mask separately, as it is already included in the embedded depth.

As earlier experiments showed and also stated by [7], the MF is not suitable for high dimensional feature vectors.

TABLE II

ACCURACY [%] OF FEATURE SELECTION METHODS ON ONE SPLIT (BEST PERFORMANCES IN BOLD, SECOND BEST IN ITALIC)

selection method	fc-6	fc-7	fc-8
all features	**88.7**	86.5	**84.4**
PCA	82.9	82.1	82.0
mRMR	*88.6*	**87.5**	*84.0*
infFS	87.3	*87.4*	83.5

Hence the extracted feature set has to be reduced to make it suitable for the MF. Therefore we select the 500 most discriminative features using the *Maximal Relevance* (MaxRel), the *minimal Redundancy Maximal Relevance* (mRMR) [24], the *Infinite Feature Selection* (infFS) [25] as well as a *Principle Component Analysis* (PCA). Since the top 500 features selected by MaxRel and mRMR differ in the ordering alone, only mRMR is considered in the following. These experiments are conducted on one split only, for RGB alone using a SVM on the features by the pre-trn VGG-net.

As can be observed in Table II and as already seen in the previous experiment, features from the fc-6 layer perform best. While the infFS and PCA feature reduction negatively affect the performance, the accuracy value of the mRMR feature set reaches almost the same as using all 4096 features. In the fc-7 layer the reduced feature set even outperforms using all features. Reducing features using PCA leads to worse results than using one of the feature selection algorithms. Based on these results, only the mRMR feature selection is considered in the following experiments.

B. Mondrian Forest as Active Learning Classifier

Given the findings of the previous subsection, the performance of the MF can be analyzed and compared with the baseline SVM results. To this end, several feature sets and sensor inputs are applied. RGB and depth features are evaluated independently as well as their concatenation to yield RGB-D results. While the RGB features are obtained from both the ImageNet pre-trn as well as the fine-tnd VGG-net, depth features are only extracted from the pre-trn CaffeNet. As feature sets the best 500 dimensions obtained by mRMR and the whole feature vector are used. Concatenating both input data, results in an 8192-dimensional feature vector, with feature selection in an 1000-dimensional one.

The optimal parameters (here number of trees n) for the 1000-feature MF are optimized based on one split for each feature set individually and can be looked up in Table III. In both cases the fine-tnd features result in a lower optimal number of trees, the *Pearson Correlation Coefficient* between n and the accuracy (MF with mRMR in Table V) was $R^2 = -0.969$. This implies a higher quality of the fine-tnd features.

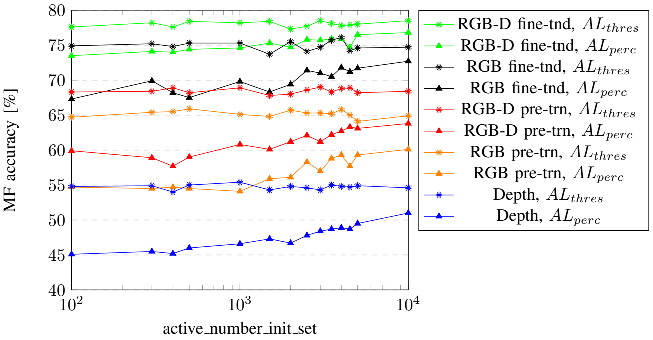

Fig. 3. Influence of the initial training set's size on accuracy (for one split). More deeply evaluated results on 10 splits are shown in Table V

TABLE III

MF PARAMETERS (NUMBER OF TREES AND INITIAL TRAINING DATA)

	RGB		Depth	RGB-D	
	pre-trn	fine-tnd	pre-trn	pre-trn	fine-tnd
number of trees	20	16	22	18	15
AL_{thres}	3000	500	300	300	3000
AL_{perc}	4500	5000	3000	5000	3500

When using MF for AL, it is initialized with a set of training data. In order to define an appropriate amount, various sizes (100 to 10000) of initial training samples are evaluated. Then, one round of adaptation based on the two query strategies form section III-A is applied on the remaining training data. As can be seen in Figure 3, AL_{thres} always outperforms AL_{perc} after one round of adaptational training. One reason for this is the higher amount of additional training samples AL_{thres} selects (all samples below a certain threshold) than what is selected by AL_{perc}, which is also shown in Table IV. As the amount of initial training data increases, the accuracy improves as well, which verifies the stated argument. For the following experiments, the initial training data size and the optimal number of trees obtained for the non-AL case are shown in Table III.

In subsection III-A we introduced two different query strategies. Both of them depend on a parameter (confidence threshold or a percentage of data) to be optimized. Hence, for both strategies several values for the corresponding parameter are evaluated using the previously estimated size of the initial training set (Table III). The results of the optimization are shown for RGB-D fine-tnd in Table IV where also the performance of the MF without AL (*no AL*) is depicted.

TABLE IV

PARAMETER INFLUENCE ON ACCURACY OF AL STRATEGIES (MEAN AND STANDARD DEVIATION OVER 10 SPLITS) FOR RGB-D FINE-TND

AL strategy		accuracy [%]	samples selected
no AL		79.5 ± 1.5	34416 ± 229
AL_{thres}	conf = 0.1	79 ± 1.7	22799 ± 627
	conf = 0.3	79.2 ± 1.8	25698 ± 633
	conf = 0.5	79.2 ± 1.3	28405 ± 877
	conf = 0.7	78.8 ± 2.1	31367 ± 312
	conf = 0.9	78.7 ± 2.2	34005 ± 171
AL_{perc}	perc = 5%	73.2 ± 2.6	5081
	perc = 10%	75.4 ± 1.4	6631
	perc = 20%	76.4 ± 1.5	9731
	perc = 50%	77.9 ± 3.7	19031
	perc = 70%	78.4 ± 2.2	25231

TABLE V

ACCURACIES [%] OF SVN AND MF VARIANTS OVER 10 SPLITS

	Classifier	RGB		Depth	RGB-D	
		pre-trn	fine-tnd	pre-trn	pre-trn	fine-tnd
SVM	all feat.	86.5 ± 3.2	86.3 ± 1.9	77.7 ± 2.3	90.8 ± 1.3	92.9 ± 1.1
	mRMR	84.7 ± 2.8	86.2 ± 1.9	76.6 ± 2.0	91.0 ± 1.1	93.1 ± 1.1
MF	all feat.	58.6 ± 4.5	72.4 ± 3.0	51.4 ± 1.8	61.3 ± 4.6	75.7 ± 4.0
	mRMR	65.5 ± 3.9	76.0 ± 1.5	55.6 ± 1.5	65.5 ± 1.4	79.5 ± 1.5
	mRMR + AL_{thres}	64.5 ± 3.4	76.6 ± 1.5	55.6 ± 1.3	65.8 ± 1.4	79.2 ± 1.3
	mRMR + AL_{perc}	65.2 ± 3.5	76.8 ± 1.0	54.6 ± 1.0	65.8 ± 1.7	78.4 ± 2.2

TABLE VI

AVG. COMPUTATION TIMES FOR EXPERIMENTS IN TABLE V.

	Selection method	RGB		Depth	RGB-D	
		pre-trn	fine-tnd	pre-trn	pre-trn	fine-tnd
SVM	all feat.	16 h	7 h	22 h	28 h	22 h
	mRMR	1.5 h	1.5 h	2.5 h	3.5 h	3 h
MF	all feat.	80 min	20 min	2 h	6 h	3 h
	mRMR	2 min	80 s	3.5 min	6 min	4 min
	mRMR + AL_{thres}	115 s	91 s	208 s	431 s	4 min
	mRMR + AL_{perc}	142 s	79 s	182 s	357 s	220 s

As already mentioned, it can be observed that AL_{perc} results in fewer additional training samples than AL_{thres}. AL_{perc} using the 50% or 70% most uncertain samples, reaches almost the result of non-AL MF. Furthermore it can be observed that adding more samples to the additional training set improves the accuracy. As already observed in the previous experiment, the AL_{thres} results in a higher accuracy, however accompanied with a higher amount of used data. As conclusion we can say that for both strategies an optimal parameter can be found which almost performs as good as the non-AL MF by using less samples. Hence, both the memory consumption and the computation time decrease, which we will show later.

C. Experimental Results

Table V shows the averaged accuracies on the 10-fold leave-one-out splitting by [11] using several pipeline modifications. The results show that using 500 (RGB or Depth) respectively 1000 (RGB-D) selected features instead of the full feature vector (4096 or 8192) does not degrade the results dramatically. This shows, inter alia, that dimensions of CNN features can be reduced without losing information but improving the computational time.

Looking on the SVM performances in Table V, the feature reduction does not lead to a huge performance difference. The performance is even increased for the RGB-D cases. Because of the fact that SVM can cope with irrelevant features, removing them with feature selection does not have a huge impact in the results. The experimental results on MF however show that using selected features for MF instead of the whole features, improves the accuracy which was already shown in [7]. The authors state that MFs need low-dimensional features representing only important information. The results also suggest that depth features for MF are not useful on their own but combining them with RGB features can improve the classification. However, the improvement is not as high as for the SVM classifier.

Since we want to focus on robotics applications, especially online learning, the computation times for the model training and testing are an interesting measure. For classification tasks in the field of robotics, a low computation time is important as robots need to interact with their environment in real-time. Table VI shows the computation times for the experiment shown Table V. Although the linear SVM results in the highest accuracy it is the one with the highest training time as well. Therefore, the usage of SVM in robotic online applications is not recommended. Conversely, the MF classifiers are much faster than SVM, but not as accurate. Hence one of the future objectives should be a deeper investigation into the behavior of MF or other online trainable classifiers to obtain the reduced computation time at a higher performance.

A comparison of our best accuracy using linear SVM to the state-of-the-art is shown in Table VII. The concatenation of reduced fine-tnd CNN features for RGB and depth improves the state-of-the-art to 93.1%, which is 2% more than the so far best method proposed by [1]. Many of the methods in Table VII are applied to pre-processed depth data. [1] improve their result for depth when they combine depth and point cloud data from 79.4% to 85%. They use the same pre-processing method for depth we are using, but not only extract the features from the fc-6 layer but also from the conv. layers for all data types. Instead of this, we fine-tune a CNN for RGB and achieve comparable results, and even better results for RGB-D. The inconsistent inter-modality behavior of several methods support our finding that specific solutions need to be developed for the two s.t. they provide complementary information for RGB-D classification.

V. CONCLUSIONS AND FUTURE WORK

In this work the combination of feature selection from layers of modality-specific CNNs, and the AL during classification was shown. In order to make the feature dimensionality suitable for the applied AL method, the MF, we reduced it with mRMR. As baseline the same feature extraction and selection pipeline was classified by a linear SVM which not only outperformed the (active) MF results but also is state-of-the-art for RGB-D object category recognition on the used dataset. However, AL with MF has specific advantages for mobile robotics: less training data, capability of online adaptation as well as reduced computational times. These should be investigated more deeply by comparing the MF to other online and/or active classifiers. In future work also the behavior of the MF should be improved in order to reach comparable results to the presented linear SVM. Furthermore the transferability of the results to testing on an independent dataset should be evaluated, where the adaptation of the

TABLE VII

AVERAGED ACCURACY WITH STANDARD DEVIATION (10 SPLITS) FOR CATEGORY RECOGNITION ON THE WASHINGTON RGB-D DATASET.

Method	Publication	RGB	Depth	RGB-D
EMK-SIFT [11]	ICRA '11	74.5 ± 3.1	64.7 ± 2.2	83.8 ± 3.5
Depth Kernel [26]	IROS '11	77.7 ± 1.9	78.8 ± 2.7	86.2 ± 2.1
CKM [14]	ICRA '12	-	-	86.4 ± 2.3
CNN-RNN [27]	NIPS '12	80.8 ± 4.2	78.9 ± 3.8	86.8 ± 3.3
HMP [28]	ISER '13	82.4 ± 2.1	81.2 ± 2.3	87.5 ± 2.9
SSL [29]	ICPR '14	81.8 ± 1.9	77.7 ± 1.4	87.2 ± 1.1
LDELM [12]	DICTA '15	78.6 ± 1.8	*81.6 ± 0.7*	88.3 ± 1.6
subset-RNN [2]	Neurocomp. '15	82.8 ± 3.4	**81.8 ± 2.6**	88.5 ± 3.1
CaRFs [15]	ICRA '15	-	-	88.1 ± 2.4
CNN-colourized [30]	ICRA '15	83.1 ± 2.0	-	89.4 ± 1.3
CIMDL [13]	arXiv '16	-	-	89.6 ± 2.1
Hypercube Pyramid [1]	ICRA '16	**87.6 ± 2.2**	79.4 ± 2.6	*91.1 ± 1.4*
fine-tnd CNN + SVM (mRMR)	this work	*84.7 ± 2.8*	76.6 ± 2.0	**93.1 ± 1.1**
fine-tnd CNN + MF (mRMR)	this work	76.0 ± 1.5	65.5 ± 1.4	79.5 ± 1.5

original classifier is more important [10], [16]. In [31] it was shown that adapting a classifier with newly gathered data during execution outperforms the initial classifier after a few correctly labeled samples. However, a linear SVM was applied there which had to be re-trained in every adaptation round, whereas the MF would be a more scalable solution.

ACKNOWLEDGMENT

This work has partly been funded by the European Union under the H2020 framework programme grant agreement no. 645403 (RobDREAM).

REFERENCES

[1] H. Zaki, F. Shafait, and A. Mian, "Convolutional hypercube pyramid for accurate RGB-D object category and instance recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[2] J. Bai, Y. Wu, J. Zhang, and F. Chen, "Subset based deep learning for RGB-D object recognition ," *Neurocomputing*, vol. 165, 2015.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[5] M. Durner, S. Kriegel, S. Riedel, M. Brucker, Z.-C. Marton, F. Balint-Benczedi, and R. Triebel, "Experience-based optimization of robotic perception," in *IEEE International Conference on Advanced Robotics (ICAR)*, Hong-Kong, China, July 2017, Best Paper Finalist.

[6] X. Song, L. Herranz, and S. Jiang, "Depth CNNs for RGB-D scene recognition: Learning from scratch better than transferring from RGB-CNNs," in *31st AAAI Conference on Artificial Intelligence*, 2017.

[7] A. Narr, R. Triebel, and D. Cremers, "Stream-based active learning for efficient and adaptive classification of 3d objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[8] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014.

[9] C. Wang and K. Siddiqi, "Differential geometry boosts convolutional neural networks for object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '16, 2016.

[10] H. Ali and Z. C. Márton, "Evaluation of feature selection and model training strategies for object category recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2014, pp. 5036–5042.

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

[12] H. Zaki, F. Shafait, and A. Mian, "Localized deep extreme learning machines for efficient RGB-D object recognition," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2015.

[13] Z. Wang, R. Lin, J. Lu, J. Feng, and J. Zhou, "Correlated and individual multi-modal deep learning for RGB-D object recognition," *CoRR*, vol. arXiv preprint arXiv:1604.01655, 2016.

[14] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[15] U. Asif, M. Bennamoun, and F. Sohel, "Efficient RGB-D object categorization using cascaded ensembles of randomized decision trees," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[16] M. Durner, Z. Márton, U. Hillenbrand, H. Ali, and M. Kleinsteuber, "Active classifier selection for RGB-D object categorization using a Markov Random Field ensemble method," in *Proc. of the 9th Int. Conf. on Machine Vision (ICMV 2016)*, ser. SPIE 10341, 2017.

[17] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '14, 2014.

[18] B. Settles, *Active Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.

[19] D. Wang, C. Yan, S. Shan, and X. Chen, "Active learning for interactive segmentation with expected confidence change," in *Proc. of the 11th Asian Conference on Computer Vision (ACCV '12).* Daejeon, Korea, November 5-9, 2012: Springer Berlin Heiderberg, 2013.

[20] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, Jun 2010.

[21] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. of the 15th ACM International Conference on Multimedia*. ACM, 2007, pp. 188–197.

[22] J. R. Nuricumbo, H. Ali, Z.-C. Márton, and M. Grzegorzek, "Improving object classification robustness in RGB-D using adaptive SVMs," *Multimedia Tools and Applications*, vol. 75, no. 12, 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012.

[24] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug 2005.

[25] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

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

[27] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, 2012.

[28] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for RGB-D based object recognition," in *Experimental Robotics: The 13th Int. Symp. on Experimental Robotics (ISER)*, P. J. Desai, G. Dudek, O. Khatib, and V. Kumar, Eds. Springer, 2013, pp. 387–402.

[29] Y. Cheng, X. Zhao, K. Huang, and T. Tan, "Semi-supervised learning for rgb-d object recognition," in *22nd International Conference on Pattern Recognition (ICPR)*, Aug 2014, pp. 2377–2382.

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

[31] F. Balint-Benczedi, Z.-C. Marton, M. Durner, and M. Beetz, "Storing and retrieving perceptual episodic memories for long-term manipulation tasks," in *IEEE International Conference on Advanced Robotics (ICAR)*, Hong-Kong, China, July 2017, Best Paper Finalist.