Localized Deep Extreme Learning Machines for Efficient RGB-D Object Recognition

Hasan F. M. Zaki[1,3], Faisal Shafait[2], Ajmal Mian[1]
[1]Computer Science and Software Engineering
The University of Western Australia, Crawley, WA, Australia
[2]National University of Science and Technology, Islamabad, Pakistan
[3]Mechatronics Engineering, International Islamic University Malaysia, Malaysia
hasan.mohdzaki@research.uwa.edu.au
faisal.shafait@seecs.nust.edu.pk
ajmal.mian@uwa.edu.au

***Abstract*—Existing RGB-D object recognition methods either use channel specific handcrafted features, or learn features with deep networks. The former lack representation ability while the latter require large amounts of training data and learning time. In real-time robotics applications involving RGB-D sensors, we do not have the luxury of both. In this paper, we propose Localized Deep Extreme Learning Machines (LDELM) that efficiently learn features from RGB-D data. By using localized patches, not only is the problem of data sparsity solved, but the learned features are robust to occlusions and viewpoint variations. LDELM learns deep localized features in an unsupervised way from random patches of the training data. Each image is then feed-forwarded, patch-wise, through the LDELM to form a cuboid of features. The cuboid is divided into cells and pooled to get the final compact image representation which is then used to train an ELM classifier. Experiments on the benchmark Washington RGB-D and 2D3D datasets show that the proposed algorithm not only is significantly faster to train but also outperforms state-of-the-art methods in terms of accuracy and classification time.**

I. Introduction

Visual object recognition has many applications in robotics and surveillance. Conventional RGB and grayscale images have long been used for object recognition [1], [2], [3]. However, the appearance of objects change significantly with variations in viewpoint, illumination and occlusions. Some of these problems can be better addressed if the 3D shape of the object can be acquired. This is now possible with the availability of many real-time RGB-D sensors in the market, which sense the depth information in addition to the RGB appearance of the objects. Therefore, there has been a lot of research interest in RGB-D based object recognition in the past decade [4], [5], [6], [7], [8], [9], [10]. On one hand, with the availability of rich depth information such as geometrical shape, viewpoint changes and structural variation recognition capability can be significantly improved. On the other hand, low resolution images coupled with large amounts of noisy data make the recognition in this context far from a trivial task.

One of the key modules for visual recognition is the design of feature representation [11], [12]. In the last decade, visual representation methods have evolved from conventional hand-crafted techniques such as the design of feature detectors and descriptors (e.g. SIFT [13], HOG [14])), feature encoding schemes (e.g. Bag of Features (BoF) [2], [1]) to more sophisticated learning techniques such as those based on dictionary learning [15] and supervised deep networks (e.g. Convolutional Neural Networks (CNN) [3], [16], [17], [18]). These research directions have largely focused on representation learning from RGB images, while the RGB-D domain remains relatively unexplored. Therefore, designing expressive feature representations for RGB-D visual recognition is still an open problem.

For RGB-D image based visual recognition, earlier methods are based on shape descriptors which code salient information from the depth domain. For example, Bo et al. [5] proposed five distinctive kernel descriptors based on different cues of depth images. Browatzki et al. [19] jointly sampled 2D and 3D features into a compact representation through BoF and Multilayer Perceptron (MLP). Lai et al. [4] extracted SIFT and spin images and separately encoded them using the Efficient Match Kernel (EMK). More recently, Ali and Marton [20] evaluated different 3D object descriptors such as Point Feature Histograms (PFH), Viewpoint Feature Histogram (VFH) and Ensemble of Shape Functions (ESF) and their combination with feature selection methods. In the same fashion, Swadzba and Wachsmuth [21] combined 3D features extracted from planar surfaces with a holistic gist feature for indoor scene classification. However, these methods are manually engineered and heavily dependent on the quality of the chosen features. Moreover, they are domain and task specific thus cannot easily be generalized across different applications [20], [22].

Recently, researchers have proposed learning algorithms to automatically extract features from RGB-D images. Blum et al. [6] adapted k-means based convolutional descriptors from [23] to the RGB-D domain. Instead of this shallow network, Bo et al. [8] extracted rich representations from different channel maps using a hierarchical network which utilized sparse coding as layer-wise building blocks. Socher et al. [7] proposed a combined single-layer CNN and multiple fixed-tree

978-1-4673-6795-0/15/$31.00 ©2015 IEEE

Recursive Neural Networks (RNNs) to separately learn RGB and depth features. The classification is realized by connecting the concatenated learned features to a softmax classification layer. Building on this deep network, recent works have also proposed additional pre- and post-processing techniques [9], [10] to further enhance the performance. Despite the powerful representation of these learning methods, they share the same notable limitation. That is, the overall processing time for not only training, but also feature extraction and classification is expensive and thus prohibitive for real-time applications.

To fill this gap, we propose an efficient framework to learn deep generalized features from RGB-D images. In particular, we formulate our method based on a fast learning scheme namely Extreme Learning Machines (ELM) [24], [25]. ELM distinguishes itself from other neural networks by solving a simple cost function based on closed form least squares instead of optimizing a difficult non-convex problem. Besides, ELM avoids expensive iterative weights update that takes place in back-propagation-based networks by simply generating random hidden layer weights beforehand and keeping them constant throughout the training phase, resulting in a very efficient training and testing scheme. In our proposed method, ELM is leveraged for both learning deep representation as well as classification. By using this scheme, we achieve superior performance in terms of recognition accuracy and processing time.

The rest of this paper is organized as follows: Section II provides the fundamentals of ELM and the ELM-based unsupervised feature learning. This section also presents the framework of the proposed Localized Deep ELM (LDELM) with its corresponding model training, feature extraction, pooling and classification methods. We evaluate the proposed method in Section III on two challenging datasets of RGB-D object category and instance classification [4], [19] and compare our results with state-of-the-art methods. Section IV summarizes our work.

II. Proposed Methodology

Extreme Learning Machines (ELM) was originally proposed to solve supervised classification and regression tasks and proved to be an order of magnitude faster than its competitors like Support Vector Machines (SVM) [25], [24], [26]. Later, ELM was extended to address various unsupervised learning problems such as clustering [27], manifold and representation learning [28]. In this work, we employ ELM as an auto-encoder for unsupervised feature learning as well as a non-linear classifier. In the following sections, we first present the basics of supervised ELM and then explain how it is used as an auto-encoder for unsupervised feature learning. Finally, we present the proposed localized deep ELM feature learning technique in detail and demonstrate how we learn efficient high-level representations of the RGB-D images for classification.

A. Supervised Extreme Learning Machines

Consider a single-layer feed-forward neural network (SLFNN) with input, hidden, and output layers. Given a set of labelled examples $\mathbf{x} = \{x^{(i)}, l^{(i)}\} \in \mathbb{R}^D$, i $\in$ 1,…,N_x, the mapping of the input to the hidden layer activations $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^{N_h}$ is given by

$$\mathbf{h}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \sigma(\sum_{i=1}^{N_x} W^{(i)} x^{(i)} + b^{(i)}). \tag{1}$$

In 1, $\mathrm{W} \in \mathbb{R}^{N_h \times D}$ is the connecting weight between the input and the hidden neurons and $b^{(i)}$ defines the bias term. The parameter $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}$ is randomly initialized from a continuous probability distribution (e.g. a uniform distribution in [0,1]), hence the latent activation is called *randomized features*. The function $\sigma(.)$ is a non-linear activation function such as a Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ or Hyperbolic Tangent $\sigma(z) = \frac{sinh(z)}{cosh(z)} = \frac{e^{2z}-1}{e^{2z}+1}$. In our implementation, we used a Sigmoid function.

Note that the randomized weights for the input layer are generated only once and then replaced with the transpose of the matrix calculated through regularized regression (details to follow) which has a closed form solution. The weights are never updated throughout the training process making ELM training extremely efficient. Moreover, the random initialization of the hidden layer in ELM avoids the need for prior knowledge or assumptions on the data distribution. Thus, the parameters linking the input and the hidden layers can be initialized even without any knowledge of the input data [25]. This is in contrast with other conventional SLFNNs where the hidden layer mapping must be explicitly learned and updated in multiple epochs.

Define $\boldsymbol{\beta} \in \mathbb{R}^{N_h \times N_c}$ as the connection weights between the hidden layer and the output layer, the output response $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^{N_x \times N_c}$ for the network is given by

$$\mathbf{y}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}. \tag{2}$$

Since, the input weights are randomly initialized, this leaves the output weight vector $\boldsymbol{\beta}$ as the only free parameter that needs to be optimized. The objective function of ELM jointly minimizes the norm of the output weight as well as the error between the output response and the corresponding class labels. Minimizing the norm of the output weights, as in ridge regression, adds stability to the solution and improves the generalization capability of the network [25]. Thus, we define the cost function of a regularized ELM as

$$\min_{\boldsymbol{\beta}} \mathcal{J}_{ELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\|\mathbf{h}\boldsymbol{\beta} - \mathbf{L}\|^2, \tag{3}$$

where $\|.\|$ denotes the Euclidean norm and the first term regularizes against over-fitting. By adding the coefficient λ to the diagonal of $\mathbf{h}^T\mathbf{h}$ and $\mathbf{h}\mathbf{h}^T$, the output weight $\boldsymbol{\beta}$ can be estimated as

$$\boldsymbol{\beta} = \begin{cases} \left(\frac{1}{\lambda}\mathbf{I} + \mathbf{h}^T\mathbf{h}\right)^{-1}\mathbf{h}^T\mathbf{L}, & \text{if } N_x > N_h, \\ \mathbf{h}^T\left(\frac{1}{\lambda}\mathbf{I} + \mathbf{h}\mathbf{h}^T\right)^{-1}\mathbf{L}, & \text{else}, \end{cases} \tag{4}$$

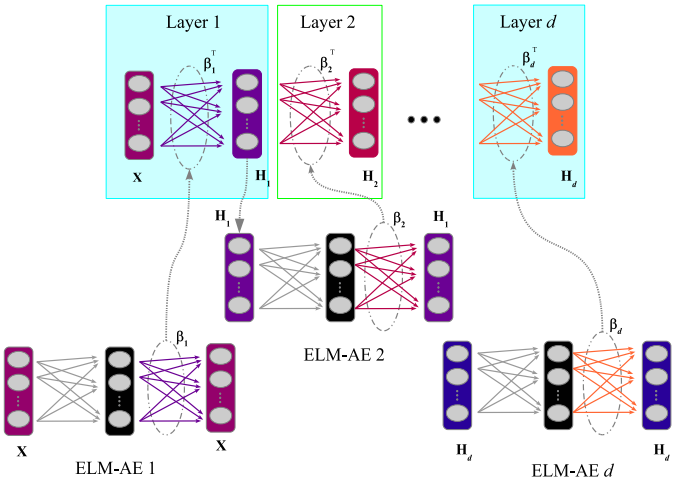

Fig. 1. Greedy layer-wise training of a d-layer DELM. The output weights learned during individual ELM-AEs are transposed and define the weights of corresponding layer of DELM.

where $\mathbf{I}$ is an identity matrix of dimension $\mathbb{R}^{N_h^2}$. When $\lambda = 0$, the cost function has a standard closed-form least square solution [24], [25], $\boldsymbol{\beta} = \mathbf{h}^{\dagger}\mathbf{L}$, where $\mathbf{h}^{\dagger}$ is the generalized inverse of matrix $\mathbf{h}$ or commonly known as Moore-Penrose pseudo-inverse. $\mathbf{h}^{\dagger}$ can be calculated using orthogonal projection method if $\mathbf{h}^T\mathbf{h}$ is nonsingular ($\mathbf{h}^{\dagger} = (\mathbf{h}^T\mathbf{h})^{-1}\mathbf{h}^T$) or if $\mathbf{h}\mathbf{h}^T$ is nonsingular ($\mathbf{h}^{\dagger} = \mathbf{h}^T(\mathbf{h}\mathbf{h}^T)^{-1}$) [25]. This formulation of ELM leads to a more *generalized universal approximation* of the data and provides robustness to the solution against different parameter settings [25].

B. Learning Deep Localized Features

1) Unsupervised Feature Learning with Extreme Learning Machines: Although ELM comes with the essence of neural network topology, little work has been done to explore its effectiveness in the task of feature learning, i.e. disentangling meaningful representation from the input data. On the contrary, feature learning has been extensively studied on other neural networks such as Deep Autoencoders (DAE) [29], [30], [12], [31], Deep Belief Net (DBN) [32] and Convolutional Neural Networks (CNN) [11], [3], [18], [16], [17]. These back-propagation based feature learners have shown superior performance across a variety of applications. However, they are generally computationally expensive in training, and still susceptible to resolve to a poor local minimum due to their complex non-convex objective functions especially when given insufficient training data [12], [33], [32]. On the other hand, the initial random projection in ELM may not be the most optimal one, the subsequent optimization for β is guaranteed to find the most optimal closed-form solution for that particular random projection [28].

Specifically, the supervised setting of ELM can be adapted to the unsupervised learning task. This can be done by considering a single layer ELM as a specialized auto-encoder (hereinafter referred to as ELM-AE), i.e. the network projects the latent activations back into its original input space and thus optimizes the corresponding mapping weights. Mathematically, this is given by

$$\boldsymbol{\beta}_{AE} = \begin{cases} \mathbf{h}^{\dagger}\mathbf{x}, & \text{for } \lambda = 0. \\ \left(\frac{1}{\lambda}\mathbf{I} + \mathbf{h}^T\mathbf{h}\right)^{-1}\mathbf{h}^T\mathbf{x}, & \text{if } N_x > N_h \text{ for } \lambda \neq 0. \\ \mathbf{h}^T\left(\frac{1}{\lambda}\mathbf{I} + \mathbf{h}\mathbf{h}^T\right)^{-1}\mathbf{x}, & \text{if } N_x < N_h \text{ for } \lambda \neq 0. \end{cases} \tag{5}$$

Note that instead of predicting the posterior of the class labels as in 4, the ELM is forced to minimize the *reconstruction error* of the input training data. In order to avoid learning random basis due to the randomized hidden projection or letting the network to just learn to reconstruct unimportant identity input, some constraint are imposed. More precisely, we restrict the randomness of the hidden layer activations by projecting the random parameters onto orthogonal hyperplanes. This improves the generalization of the ELM-AE as the orthogonal vectors are linearly independent and preserve the dot product of the variables. The hidden layer activation $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^{N_h}$ with orthogonal random parameters can be computed as follows:

$$\begin{aligned} \mathbf{h}(\mathbf{x}) &= \sigma(\boldsymbol{\theta}_{AE}^T\mathbf{x}) = \sigma(\mathbf{W}_{AE}\mathbf{x} + \mathbf{b}_{AE}) \\ s.t. \quad & \mathbf{W}_{AE}^T\mathbf{W}_{AE} = \mathrm{I}, \mathbf{b}_{AE}^T\mathbf{b}_{AE} = 1 \end{aligned} \tag{6}$$

In the case where equidimensional representation is desired (i.e. $D = N_h$), we impose another criterion to ensure the orthogonality of the output weight $\boldsymbol{\beta}$. To find the unique solution, we consider the task as equivalent to solving a standard Orthogonal Procrustes problem.

$$\begin{aligned} \boldsymbol{\beta}_{AE} &= \arg\min_{\boldsymbol{\beta}} \|\mathbf{x} - \mathbf{h}\boldsymbol{\beta}_{AE}\|_F \\ s.t. \quad & \boldsymbol{\beta}_{AE}^T\boldsymbol{\beta}_{AE} = \mathrm{I} \end{aligned} \tag{7}$$

To minimize the above cost function, we first compute $\mathbf{h}^T\mathbf{x} = U\Sigma V^*$ using Singular Value Decomposition (SVD). We hence obtain the orthogonal output weight using a closed-form solution $\boldsymbol{\beta}_{AE} = UV^*$. Similar to the traditional auto-encoders, the training of ELM-AE is completely unsupervised and it has been shown that ELM-AE can learn meaningful features comparable to other unsupervised methods such as Singular Value Decomposition (SVD) and dictionary learning but with much lower training time [28].

2) Learning Deep Extreme Learning Machines and Feature Encoding: We propose an efficient framework to extract rich representations from local regions based on ELM-AE. By learning localized features, we address three problems that normally occur in RGB-D object recognition. Firstly, we address the problem of sparsity of training data. Secondly, the learned localized features are more robust to small distortions [32] such as occlusions and viewpoint variations. Moreover, we effectively learn a set of features from small subpatches hence avoiding the need to learn a large number of training parameters. Figure 2 depicts an overview of the proposed method.

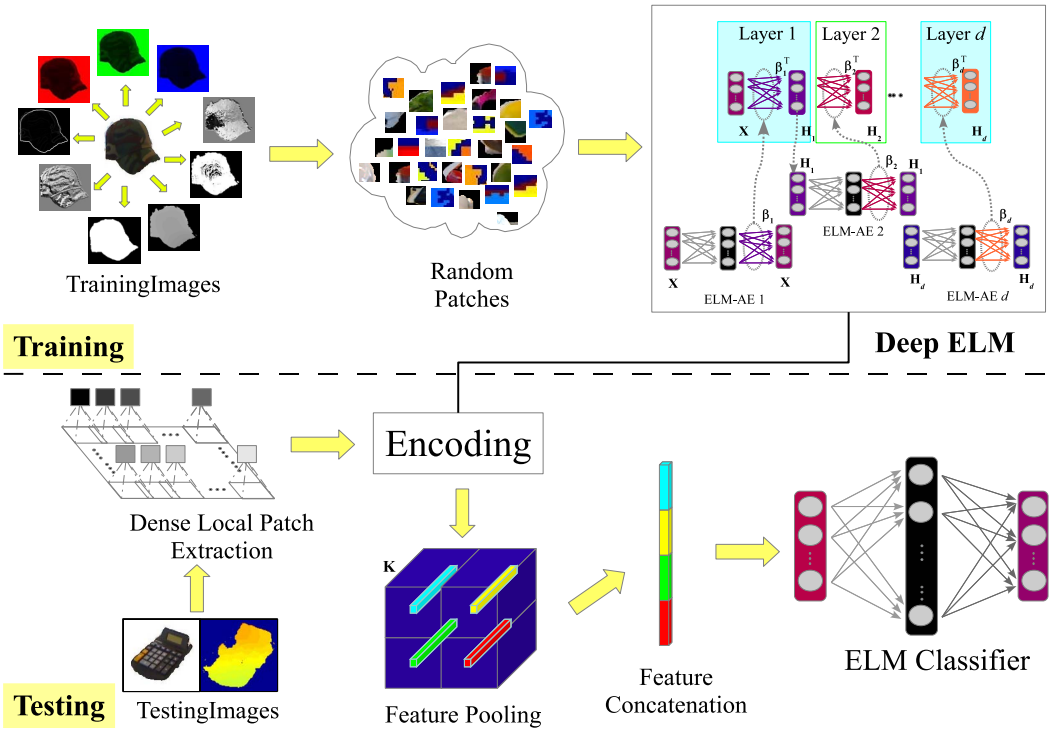

Fig. 2. Illustration of the proposed method for a single channel. Each RGB-D image is decomposed into its derivatives including the colour, depth and respective gradient magnitude and directions and also the binary channel (if available). Then, the proposed LDELM is learned for each channel (see text) where the resultant concatenated features from all channels are used as the final feature vector for classification.

We first define a multi-layer ELM in order to learn deep representation from the local regions. Considering single ELM-AE as the building blocks, we construct a deep ELM (DELM) by stacking multiple ELM-AEs hierarchically using *greedy layer-wise training* [12], [28]. Specifically, the output weight learned using the $n-$th ELM-AE is transposed and used to initialize the $n-$th layer of the DELM. As illustrated in Figure 1, we then feed-forward the $n-$th layer to produce the $n-$th feature which is then used as the input to the $(n+1)-$th ELM-AE. In contrast to traditional deep networks, DELM does not need fine-tuning through back-propagation as a result of its closed-form learning scheme, hence reducing the computational cost.

Starting from a training set of RGB-D images, we separate each channel into individual maps. In addition, we also take the binary maps and calculate the gradient maps from the grayscale and depth images. That is, given the horizontal gradient D_x and vertical gradient D_y, the gradient maps consist of both the magnitude $\sqrt{{D_x}^2 + {D_y}^2}$ and the direction of the gradient $\text{atan}(D_y/D_x)$ (see Figure 2 for visualization) . Then, we randomly sample p^2 patches from each individual map. Using these patches, we learn the hierarchical DELM models. Instead of capturing low-level features like other shallow networks, we hypothesize that learning deep models from local regions will extract higher level discriminative features similar to part-based models [34] and mid-level representations [35]. Since DELMs are computationally very efficient to train, we train a separate DELM for each channel.

Given a new probe image of size x-by-y pixels, we densely sample overlapping p^2 local patches with stride s and feed-forward the patches for each channel map through its corresponding DELM to extract K-dimensional vectors. These vectors are then organized into a cuboid such that the depth of the cuboid equals to K and the spatial location of the vectors corresponds to the center of the patch in the original image. Thus the XY-dimensionality of our cuboid is $(x-p+1)$-by-$(y-p+1)$. We then spatially divide the cuboid into four equal quadrants and sum-pool the vectors in each quadrant. Each map is represented by the concatenated $4K$-dimensional vector. Prior to the classification, the vectors from different channel maps are further concatenated to produce a discriminative and high dimensional vector. Finally, these vectors are used for classification.

For classification, we train an ELM classifier with a single hidden layer as explained in Section II-A. The ELM classifier is trained on the learned features extracted from the labelled training data using our Deep ELM autoencoders.

III. Experimental Results

We present experimental results of the proposed method on two benchmark RGB-D object recognition datasets: Washington RGB-D [4] and 2D3D [19]. We carefully followed the experimental protocols set forth by the authors of the corresponding datasets and benchmarked the presented algorithm against other related state-of-the-art methods [4], [5], [6],

TABLE I
COMPARISON WITH THE STATE-OF-THE-ART ON THE WASHINGTON RGB-D DATASET [4]. AVERAGE ACCURACIES (%) AND STANDARD DEVIATIONS OF 10 TRIALS ARE REPORTED.

Recognition Type	Category Classification			Instance Classification		
Method	Depth	RGB	RGB-D	Depth	RGB	RGB-D
Linear SVM [4]	53.1 ± 1.7	74.3 ± 3.3	81.9 ± 2.8	32.3	59.3	73.9
Kernel SVM [4]	64.7 ± 2.2	74.5 ± 3.1	83.8 ± 3.5	46.2	60.7	74.8
Random Forest [4]	66.8 ± 2.5	74.7 ± 3.6	79.6 ± 4.0	45.5	59.9	73.1
CNN-RNN [7]	78.9 ± 3.8	80.8 ± 4.2	86.8 ± 3.3	N/A	N/A	N/A
Depth Kernel [5]	78.8 ± 2.7	77.7 ± 1.9	86.2 ± 2.1	54.3	78.6	84.5
CKM [6]	N/A	N/A	86.4 ± 2.3	N/A	82.9	90.4
SP+HPM [8]	81.2 ± 2.3	**82.4 ± 2.1**	87.5 ± 2.9	51.7	92.1	92.8
SSL [9]	77.7 ± 1.4	81.8 ± 1.9	87.2 ± 1.1	N/A	N/A	N/A
Proposed LDELM	**81.6 ± 0.7**	78.6 ± 1.8	**88.3 ± 1.6**	**55.2**	**92.8**	**93.5**

[8], [7], [9], [19]. The reported results from other algorithms are taken from the original papers without re-implementation. However, we use exactly the same split used by Bo *et al*. [8] for Washington RGB-D dataset.

A. Experimental Settings:

We use raw images from the Washington RGB-D dataset and resize the images in the 2D3D dataset to 250×250. For LDELM model training, we optimize our parameters on a smaller subset of the Washington RGB-D dataset and then keep them constant across all experiments for both datasets. These include the stride $s = 1$, DELM depth $d = 3$, $\lambda = 0$, patch size $p = 9$ and number of hidden neurons as $\{100, 1000, 200\}$ to represent different hierarchical feature transformations. We also use the same number of random patches (500,000) for model training. For classification, the parameters were chosen empirically using a grid search $\lambda \in [1e8, 1e11]$, $N_h \in [5e3, 13e3]$ with grid step size $1e1$ and $1e3$ respectively.

B. Washington RGB-D Object Dataset

The first experiment was performed on the Washington RGB-D object dataset [4] which is the largest RGB-D object recognition dataset to date. The dataset consists of 41,877 images of 300 instances that are organized into 51 different categories. Each object is placed on a turntable platform while varying the camera position to output sequences from $30°, 45°$ and $60°$ angle above the horizon. We performed two sets of experiments including *category classification* (categorizing previously unseen test instances) and *instance classification* (categorizing known object instances).

1) Category Classification: We report the performance of the proposed method in terms of recognition accuracy over 10 different trials. For each trial, we randomly select one instance per class as the testing set while the rest were used for training. Table I shows the performance of the proposed LDELM compared to existing methods. Except for RGB-only inputs, our LDELM outperformed other methods. One interesting observation from the results is that only depth kernel based method [5] have similar results to ours where the depth-only recognition accuracy is higher than the RGB-only recognition accuracy. We hypothesize that there is little coherence within RGB space from which the ELM classifier tried to find the non-linear relationship. We also observe that our LDELM gave more stable performance across different splits as reflected by the low standard deviation of the recognition accuracy. Figure 3 shows the confusion matrix for the classification task on one of the trials of the Washington RGB-D object dataset and some examples of typical misclassified instances by the proposed algorithm are depicted in 4.

2) Instance Classification: Following the experimental protocol of Lai et al. [4], we evaluate the sequence captured from the $45°$ angle and train the algorithm on $30°$ and $60°$ sequences for each instance. As shown in Table I, our algorithm outperformed all competing methods in terms of accuracy. It is worth noting that, for instance classification, the contribution of RGB channels is higher than the depth channel since colour information stays more stable across different viewing angles compared to depth information. Since there are no random training/testing splits involved in this experiment, there is no performance bias in the results which may occur as a result of different choices of testing sets. Finally, the results of this experiment are important for various robotics applications where the robot must be able to recognize an object from different viewpoints.

3) Execution Time: We perform an additional experiment to compare the computational performance of the proposed method with the two methods whose code is available namely the Convolutional-Recursive NN (CNN-RNN) [1] and Hierarchical Matching Pursuit (SP+HMP) [2]. For the compared methods, we use their optimized parameters provided by the respective authors. All methods were implemented in MATLAB and tested on the same 64-bit, 2.5GHz machine. Table II compares the time required for training the models (networks), training the classifiers, feature extraction and classification. The results clearly show that the proposed LDELM is significantly faster than the existing methods for all individual modules. The online feature extraction + classification time is over four times faster than CNN-RNN and about 7 times faster than SP+HMP.

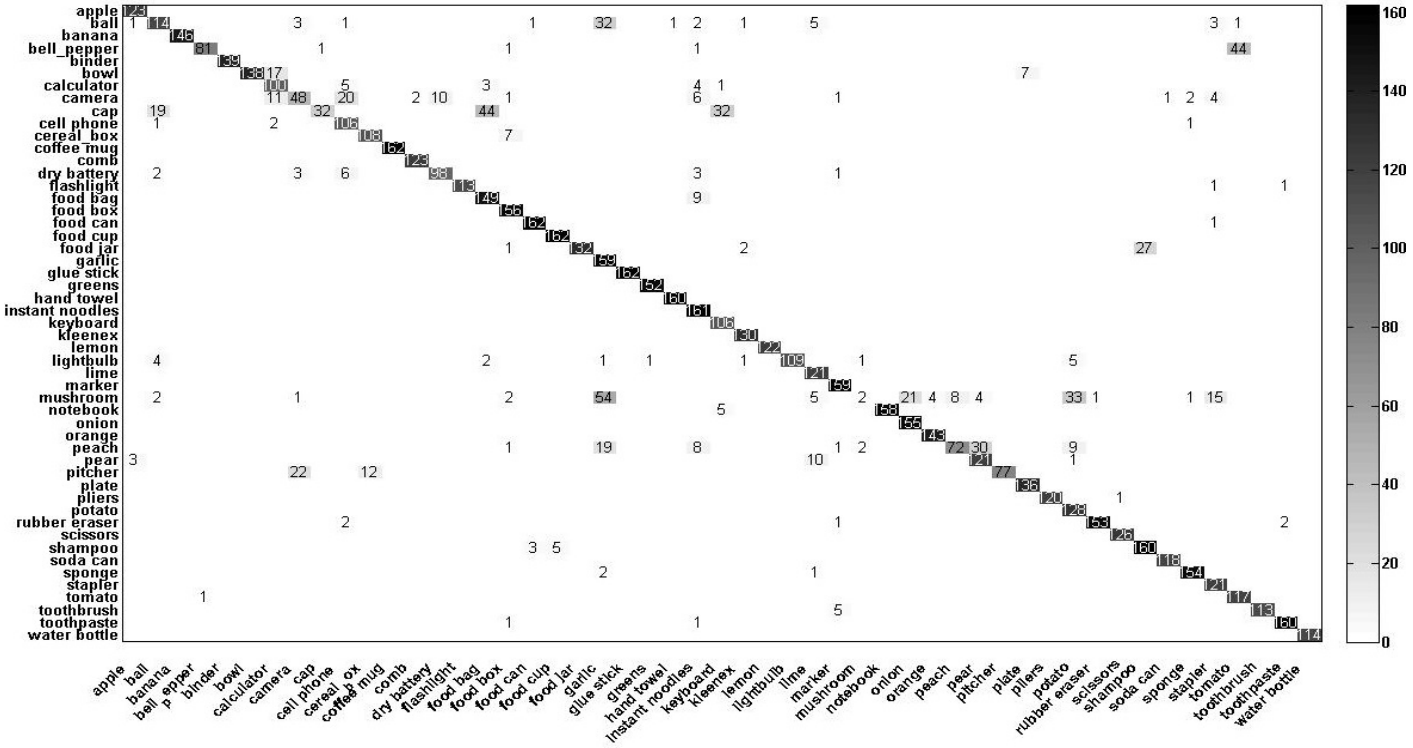

Fig. 3. Left: Confusion matrix for category classification task in Washington RGB-D object dataset. The vertical axis denotes predicted classes and the horizontal axis consists of ground truth labels. Right: Selected off-diagonal entries (from top): 1) *cap* misclassified as *food bag*, 2) *foodjar* misclassified as *shampoo*, 3) *mushroom* misclassified as *garlic*.

TABLE II
COMPARISON OF EXECUTION TIME (IN SECONDS). THE TRAINING TIME FOR DEEP LEARNING METHODS IS EXTREMELY HIGH AND THEY ARE UNABLE TO PERFORM REALTIME CLASSIFICATION. LDELM HAS THE FASTEST TRAINING TIME AND CAN PERFORM FEATURE EXTRACTION AND CLASSIFICATION IN REALTIME.

Module	CNN-RNN	SP+HMP	LDELM
Training models	$> 10^4$	$> 10^3$	**625.3**
Training classifier (training instances $\approx$ 34000)	$> 10^3$	111.9	**92.19**
Feature extraction (per image)	2.1809	3.4089	**0.4892**
Classification (per image)	5×10^{-4}	6×10^{-4}	$\mathbf{1 \times 10^{-4}}$

Fig. 4. Left: Selected off-diagonal entries (from left, top vs bottom): 1) *cap* misclassified as *food bag*, 2) *foodjar* misclassified as *shampoo*, 3) *mushroom* misclassified as *garlic*.

C. 2D3D Object Dataset

For the second experiment, we evaluated the proposed LDELM on the 2D3D object dataset [19]. This dataset has a relatively smaller number of instances compared to the Washington dataset. It has 18 object categories from 163 object instances and is a challenging dataset because it only consists of highly textured objects such as books, computer monitors and drink cartons. Following the procedure of Browatzki et al. [19], we exclude the classes phone and perforator due to their low number of samples. We also combine the classes fork, knife and spoon into a joint class of silverware to make a final dataset of 156 instances and 14 classes. For the evaluation, we randomly choose six instances per class for training and use the remaining for testing. However, if the number of instances is less than six in one class, we ensure that at least one instance is excluded from training and is available for testing. Moreover, only 18 RGB-D frames are randomly picked for each instance in both the training and testing sets.

[1]available: http://www.socher.org/

[2]available: http://research.cs.washington.edu/istc/lfb/software/hmp/index.htm

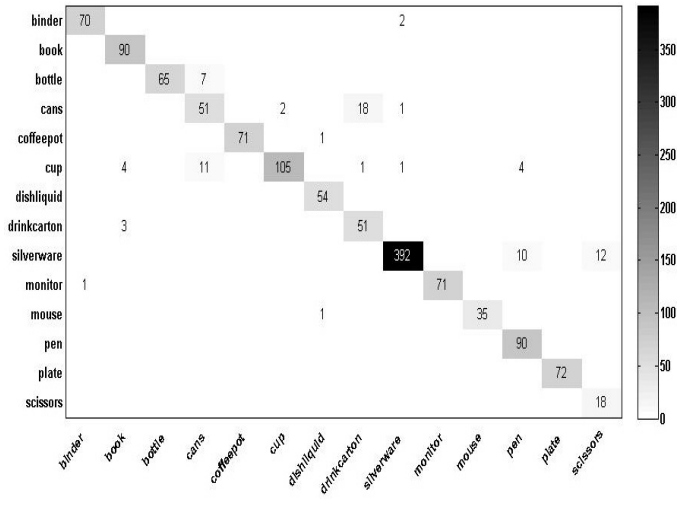

Fig. 5. Confusion matrix for category classification task in 2D3D dataset. The vertical and horizontal axes represent the predicted labels and ground truth, respectively. For instances with low classification accuracy such as *silverware*, some of the depth images are very noisy while small number of the images are almost completely flatten.

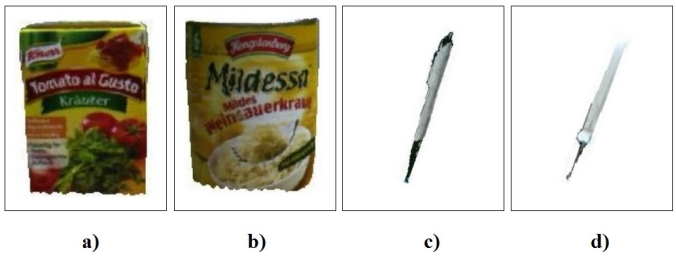

Fig. 6. Selected outliers for category classification task on 2D3D dataset [19]. The figures depict typical visual recognition challenges such as large inter-class similarity in terms of shape and appearance and low resolution input data. From left: a) *drink carton* misclassified as b) *cans* and c) *pen* misclassified as *silverware (fork)*.

Table III shows that our proposed LDELM outperforms the state-of-the-art algorithms including deep learning methods [8], [10] and the fusion of handcrafted features [19]. In this case, our algorithm achieved the highest performance on individual RGB and depth based classification as well as the combined RGB-D based classification. Note that this dataset comprises very low resolution images and fewer number of training samples. For some instances, the depth information is almost completely flat or missing. This is also the case for some RGB images where the colours of the objects are very dark which makes it extremely difficult to distinguish between the object and the background. However, our results show that the missing information from one domain is efficiently complemented by information from the other domain to achieve higher accuracy by our method. Additionally, the performance of our method does not degrade due to limited training samples. We present the confusion matrix on 2D3D dataset in Figure 5 and show selected outliers in Figure 6 for analysis.

TABLE III
OBJECT CATEGORIZATION ACCURACY (%) ON THE 2D3D DATASET [19]. THE PROPOSED LDELM OUTPERFORMS HAND CRAFTED FEATURES AND DEEP LEARNING METHODS.

Method	Depth	RGB	RGB-D
2D+3D [19]	74.6	66.6	82.8
SP+HPM [8]	87.6	86.3	91.0
Subset+RNN [10]	90.2	88.0	92.8
Proposed LDELM	**91.6**	**90.3**	**94.0**

IV. CONCLUSION

We presented an efficient algorithm for learning rich discriminative features from RGB-D images by combining the power of deep network representation with the efficiency of Extreme Learning Machines (ELM). Experimental results on two benchmark datasets, Washington RGB-D object dataset and 2D3D dataset show that the proposed method outperforms existing state-of-the-art in terms of accuracy and computation efficiency. With reduced training/test time and high recognition accuracy, our algorithm opens up the possibility of employing deep networks for real-time RGB-D object learning and recognition applications.

ACKNOWLEDGMENT

This research work is supported by the Ministry of Higher Education Malaysia and Australian Research Council (ARC) grant DP110102399.

REFERENCES

[1] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. BMVC*, 2011.
[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *proc. CVPR*, vol. 2, 2006, pp. 2169–2178.
[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014.
[4] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. ICRA*, 2011, pp. 1817–1824.
[5] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. IROS*, 2011, pp. 821–826.
[6] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Proc. ICRA*, 2012, pp. 1298–1303.
[7] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. NIPS*, 2012, pp. 665–673.
[8] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for RGB-D based object recognition," in *Proc. ISER*, 2012.
[9] Y. Cheng, X. Zhao, K. Huang, and T. Tan, "Semi-supervised learning for RGB-D object recognition," in *Proc. ICPR*, 2014, pp. 2377–2382.
[10] J. Bai, Y. Wu, J. Zhang, and F. Chen, "Subset based deep learning for RGB-D object recognition," *Neurocomputing*, 2015.
[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE PAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.
[12] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Proc. NIPS*, vol. 19, p. 153, 2007.
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[15] N. Akhtar, F. Shafait, and A. Mian, "Repeated constrained sparse coding with partial dictionaries for hyperspectral unmixing," in *Proc. WACV, 2014*. IEEE, 2014, pp. 953–960.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[17] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. NIPS*, 2014, pp. 487–495.

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. ICLR*, April 2014.

[19] B. Browatzki, J. Fischer, B. Graf, H. Bulthoff, and C. Wallraven, "Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011, pp. 1189–1195.

[20] H. Ali and Z.-C. Marton, "Evaluation of feature selection and model training strategies for object category recognition," in *Proc. IROS*, Sept 2014, pp. 5036–5042.

[21] A. Swadzba and S. Wachsmuth, "Indoor scene classification using combined 3D and gist features," in *Proc. ACCV*, 2011, pp. 201–215.

[22] H. Ali, F. Shafait, E. Giannakidou, A. Vakali, N. Figueroa, T. Varvadoukas, and N. Mavridis, "Contextual object category recognition for RGB-D scene labeling," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 241–256, 2014.

[23] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. AISTATS*, 2011, pp. 215–223.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[25] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[26] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[27] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541 – 2551, 2011.

[28] L. L. C. Kasun, H. Zhou, and G.-B. Huang, "Representation learning with extreme learning machine for big data," *IEEE Trans. on Intelligent Systems*, vol. 28, no. 5, pp. 31–34, 2013.

[29] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification." in *Proc. BMVC*, 2012, pp. 1–12.

[30] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[32] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[33] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[34] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *Proc. ICCV*, 2011, pp. 1307–1314.

[35] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. ECCV*, 2012, pp. 73–86.