IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 26, 2016, accepted November 18, 2016, date of current version January 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2639543

Object Recognition Using Deep Convolutional Features Transformed by a Recursive Network Structure

HIEU MINH BUI[1,2], MARGARET LECH[1], EVA CHENG[1], (Member, IEEE), KATRINA NEVILLE[1], (Member, IEEE), AND IAN S. BURNETT[3], (Senior Member, IEEE)

[1]School of Engineering, RMIT University, GPO Box 2476, Melbourne VIC 3001 Australia
[2]Center of Technology, RMIT University, 702 Nguyen Van Linh, District 7, Ho Chi Minh, Vietnam
[3]Faculty of Engineering and Information Technology, University of Technology Sydney, PO Box 123, Broadway NSW 2007, Australia

Corresponding author: H. M. Bui (s3372651@rmit.edu.vn)

This work was supported in part by the Center of Technology, RMIT University, Vietnam, and in part by RMIT University, Melbourne, Australia.

ABSTRACT Deep neural networks (DNNs) trained on large data sets have been shown to be able to capture high-quality features describing image data. Numerous studies have proposed various ways to transfer DNN structures trained on large data sets to perform classification tasks represented by relatively small data sets. Due to the limitations of these proposals, it is not well known how to effectively adapt the pre-trained model into the new task. Typically, the transfer process uses a combination of fine-tuning and training of adaptation layers; however, both tasks are susceptible to problems with data shortage and high computational complexity. This paper proposes an improvement to the well-known AlexNet feature extraction technique. The proposed approach applies a recursive neural network structure on features extracted by a deep convolutional neural network pre-trained on a large data set. Object recognition experiments conducted on the Washington RGBD image data set have shown that the proposed method has the advantages of structural simplicity combined with the ability to provide higher recognition accuracy at a low computational cost compared with other relevant methods. The new approach requires no training at the feature extraction phase, and can be performed very efficiently as the output features are compact and highly discriminative, and can be used with a simple classifier in object recognition settings.

INDEX TERMS Machine learning, pattern recognition, neural networks, knowledge transfer.

I. INTRODUCTION

Extraction of discriminative features from input images is one of the most challenging tasks in object recognition systems. Much effort has aimed at determining optimal feature sets for a specific task, based on the attributes of objects to be recognized and classifiers to be used. Many of these features produced very promising results [1], [2]. However, due to the ambiguity and lack of general task-independent rules for optimal feature selection, the process of data classification has been recently dominated by various approaches using neural networks. The important advantage of these neural network approaches is that during the training process the network self-determines the optimal set of features from the data. The disadvantage is that large training data sets may be required and thus the training process could be very lengthy.

Neural networks have been shown to provide excellent performance in multiple image classification benchmarks, ranging from simple feature datasets such as MNIST [3] to complicated challenges such as ImageNet [4]. However, more recent research in computer vision has demonstrated the dominating power of a neural network methodology known as deep learning [5]. Similar to the design of optimal image descriptors being cumbersome in the past, the design and training of deep learning structures today is also a big challenge. This is particularly a challenge when the performance of the design is very sensitive to the implementation details, which is often the case [6]. Fortunately, using an implementation of already trained structures is quite straightforward and if the power of the network depth can be applied to other classification scenarios, then this offers great advantages.

The progress of deep learning approaches in recent years has been spectacular. One of the biggest breakthroughs that catalyzed the recent wave of neural network deep learning research may be tracked to the work of Krizhevsky *et al.* [7], who proposed ImageNet Classification with Deep Convolutional Neural Networks (CNNs) as a combination of convolutional and fully connected neural networks applied with data augmentation and training techniques. The AlexNet was the winner of the ImageNet Large Scale Visual Recognition Challenge 2012, and continues to be the source of inspiration for winners in years thereafter. While AlexNet has been outperformed by later proposals [23], [27], [28], the approach is still valuable as a good compromise between simplicity and performance.

Much research has been conducted to apply deep neural networks to other computer vision tasks; however, most tasks require either a modification of network parameters, or extra adaptation layers in the networks tailor the application into the target task [8]–[10]. Further, it takes time and care to train new layers or fine-tune the pre-trained network model, especially when the structure is deep. With limited amounts of labeled data in small datasets, it is thus difficult to manage the potential issue of overfitting when tuning these deep structures. This intuition could be one of the factors explaining why the current classification performance of fine-tuned systems often show only moderate results [8]–[10]. The Recursive Neural Network (RNN) [11] offers one possible solution for this problem via a systematic transformation of data with added non-linearity and randomness, which eliminates the costs required for adaptive training and label knowledge.

The RNN method proposed by Socher *et al.* [11], was shown to be able to capture repetitive data such as the information contained in images and speech. Since features extracted from CNN layers of a deep network still encode spatial cues from the original images, the RNN provides a mechanism to explore these cues further without increasing the costs of training and refinement. The RNN structure is somewhat similar to the CNN structure, but instead of using learned weights, the RNN uses randomly initialized weights. In addition, the RNN has non-overlapping receptive fields, which is different to the CNN. Processing through the RNN is simple and extremely fast, and shows the ability to map the data into a more separable space [11]. This characteristic is particularly helpful in multi-class object classification tasks [12].

This work proposes a new efficient method for formulating an object recognition system by combining a deep trained network model with the RNN structure, which will be referred to as AlexNet-RNN throughout this paper. The proposed approach has the potential to significantly speed up the network adaptation process while maintaining a high level of performance.

The remainder of this paper is organized as follows. Section II briefly reviews previous studies that are most relevant to the current work, and outlines the contributions of this work. Section III describes in detail the structure of the proposed object recognition system, and Section IV describes experimental validation of this new system, presented alongside with the results and discussion. Finally, Section V concludes the paper.

II. RELATED WORK

As our work has been conducted on the Washington RGBD image dataset [13], we will review recent research that has also been tested on this dataset and then outline the contributions of the current study.

A. PREVIOUS WORK

Research conducted on the Washington RGBD (W-RGBD) image dataset investigated both hand designed features and features generated by neural networks. One of the first object recognition techniques evaluated on the Washington RGBD dataset utilized a wide range of features including SIFT, textons and color histograms. These features were classified using the random forest classifier and provided 74.7% accuracy for the RGB data [13]. Later studies based on the hand-designed features were not able to improve classification accuracy for the W-RGBD dataset much further, and since then machine learning approaches have emerged. Sparse coding [14] and clustering based convolutional extractors [12], [15] have increased the classification performance to 85.2%. The recently proposed Fisher Kernel approach [16] further increased to the accuracy to 86.8%, which can be considered as the current state of the art result.

Another significant stream of research includes transfer learning, which is extensively reviewed in [29]. Deep learning approaches have been shown to be able to capture high-level features providing both representative and discriminative information from images to facilitate different vision tasks. A possible explanation for this capability is presented in DiCarlo's hypothesis [30], where the image distributions are disentangled as they pass through layers of a deep neural network. Direct application of deep learning into many machine vision tasks are not possible due to the requirement of prohibitively large collections of labeled training data. However, the visualizations in [9] and [23] indicate that, as the distribution of objects is transformed from overlapped space to separable space in a deep network, intermediate representations can be used as generic features to semantically describe the object in the input image. There have been research efforts aimed at adapting a pre-trained deep network into specific object classification tasks, both with and without fine tuning, but the results were only moderate [17], [18]. Therefore, investigating the concept of adapting a deep network trained on a large labeled dataset to a new task represented by a smaller dataset is a current research challenge.

As already mentioned, the AlexNet approach to object recognition was one of the most influential breakthroughs that directed the research community back into deep learning. Intensive tests have been conducted to examine the activation weight characteristics of each layer of the AlexNet in

relation to visual recognition tasks, where it was observed that the activation weights taken from the fully connected layer right after the convolutional chain are the best features for object recognition purposes [9]. The OverFeat feature extractor proposed in [8] has further explored this issue by applying a trained convolutional extractor on different scales of input images during the inference step. Razavian *et al.* [10] have reported very promising results when using features extracted by OverFeat to recognize objects from multiple image datasets. Similarly, [19] and [20] have successfully applied features extracted by the AlexNet in object detection and localization tasks. Pre-trained deep structures can be used as feature extractors as mentioned above, or the structures can be fine-tuned to adapt the network from the source task to the target task (with the assumption of similarity in data distribution between two tasks). Examples of such methods applied to AlexNet are [21] and [22].

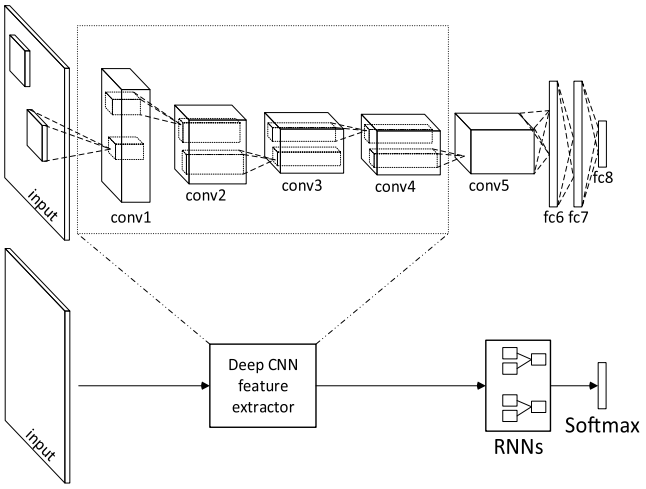

FIGURE 1. Structures of AlexNet (top) and structure of our system with part of AlexNet applied as the feature extractor (bottom).

B. CURRENT STUDY CONTRIBUTIONS

The contributions of the current study can be summarized as follows:

1. Proposing a new efficient feature extraction method using a deep convolutional network structure trained on a large dataset for object recognition tasks represented by a smaller dataset. The method combines the well-known AlexNet with a RNN structure.
2. Re-evaluating the use of AlexNet as a feature extractor to determine applicability of the 'best layer' rules proposed in [9].

This study can be fully reproduced using the code provided at: https://github.com/hieu-bm/deep_CRNN

III. METHOD

A. PROPOSED AlexNet-RNN APPROACH

A number of studies have confirmed that intermediate layers in a deep network can capture features that provide a good tradeoff between representation and object independence [9], [22], [23]. In this work, several low-level layers of the AlexNet trained on the ImageNet data set were selected, and each of these layers were examined as a black-box feature extractor. The full structure of the AlexNet is shown in the top half of Fig. 1. The network was trained using fixed size RGB images from ImageNet [4], as explained in [6] and [7]. The network structure consists of 8 layers, where the first 5 layers (conv1, conv2, conv3, conv4, conv5) are convolutional and the remaining 3 layers (fc6, fc7, fc8) are fully-connected. The last fully-connected layer (fc8) has the form of a Softmax classifier to categorize an input image into one of the classes used in training. The proposed new object recognition structure, with part of AlexNet embedded as the feature extractor, is illustrated in the bottom half of Fig. 1. This new structure uses the same input format as the AlexNet and consists of several low-level layers of trained AlexNet. These layers thus act collectively as the feature extractor. An RNN unit containing multiple RNN structures, is then added to further process the extracted features, before feeding them into the Softmax classifier that performs recognition on the target data set.

As a part of the new structure, two alternative versions of the AlexNet were used to extract the deep image features: the original version of the AlexNet-2012 [7], and the AlexNet-2014 [6]. The AlexNet-2014 [6] was more densely connected but had a smaller number of CNN weights in the intermediate layers compared to the AlexNet-2012 version described in [7]. In this paper, the AlexNet-2014 was applied in most experiments unless stated otherwise, and the original AlexNet-2012 was used as a reference point.

There were three key reasons behind the use of AlexNet-2014 over the original AlexNet-2012 in this paper:

1) It provides slightly higher performance than AlexNet-2012 on multiple datasets;
2) It is computationally cheaper; and,
3) Its last three convolutional layers have the same size, which allows for size-independent transferability and comparison of features between layers.

The pre-trained models of the AlexNet-2014 and AlexNet-2012 were adapted from the MatConvNet project [24]. These models were fully trained on the ImageNet 2012 [4] dataset to achieve performances consistent with the results reported in [6] and [7].

The main feature that differentiates our work from related studies is the incorporation of the RNN unit, which consists of an assembly of separate RNN structures processing features provided by the pre-trained deep convolutional network.

As illustrated in the bottom half of Fig. 2, the RNN structure is quite similar to the structure of the CNN (top half of Fig. 2). Both structures divide the input data into patches of equal size, compute element-wise products of each patch with a shared array of weights, add the outcomes together and then process the sum through a sigmoidal or other type of squashing function. The RNN is different to the CNN in

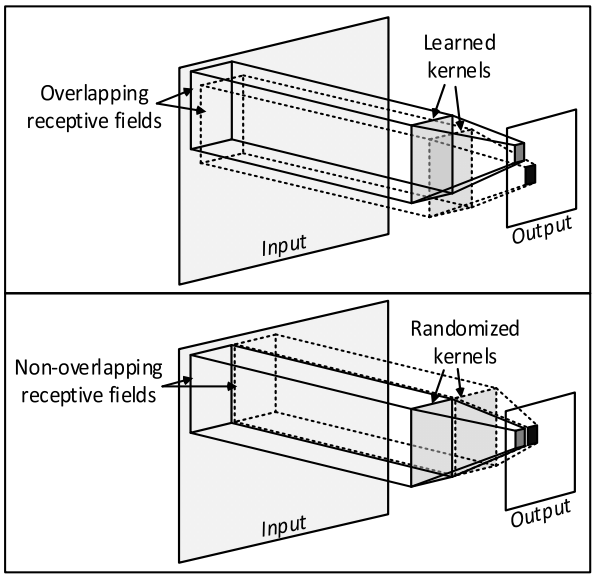

FIGURE 2. Structure of the CNN layer (top) and the RNN layer (bottom).

two aspects. Firstly, the RNN set of weights is randomly initialized based on the input data structure and kept permanently unchanged while the CNN set of weights is learned from the data. Secondly, the RNN uses non-overlapping input patches while the CNN typically uses densely overlapped patches. Due to the random attribution, the RNN does not require training, thus it is easy and quick to deploy. Due to the non-overlapping attributes of patches, the RNN is computationally less expensive than the CNN. The recursive structure allows RNNs to capture repetitive patterns in the input, while the random weights and nonlinear squash function help to differentiate between class clusters.

In this work, different squash functions were used for the RNN and CNN. The CNN layers of both AlexNet-2012 and AlexNet-2014 used the *ReLU* function given as,

$$y = \max(0, x) \tag{1}$$

while the RNN used the *tanh* squash function given as,

$$y = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2}$$

where x denotes the sum of products of input data with the weight set, and y is the squashed value of the sum, which is used as input to a later processing layer. ReLU was used in the AlexNet as it allows for faster training time for large structures [7]. The RNN does not require training, thus fast training is no longer an advantage. In addition, ReLU does not provide the necessary nonlinearity to transform the data, which makes the squash function unsuitable for RNN nodes.

In this work, the pre-trained AlexNet is used as a black-box feature extractor, therefore no fine tuning was involved. As indicated in [9], the fully connected layer number 6 of the AlexNet provided the highest quality features for the object classification task. In a fully-connected layer, each neuron connects in the same way to all neurons of the previous layer, and as a result the spatial information that describes the input data was largely discarded. The current study thus explores the possibility of utilizing the remaining spatial characteristics of data processed by the deep network through application of the RNN structure. Therefore, it is necessary to examine the features produced by previous CNN layers, where spatial information still remains. Consequently, this work separately analyzes and compares the efficiency of applying RNN processing to CNN features and to fully-connected features with regards to an object recognition task.

Features produced by a deep neural network layer, either CNN or fully-connected, typically have the form of a 3-dimensional matrix of size $w \times h \times d$. The 3rd, 4th, and 5th CNN layers of both AlexNet-2012 and Alex-Net-2014 set $w = h = 13$, whilst the value of d was set to either 384 for the 3rd and 4th layers of AlexNet-2012 and to 256 in other cases. The 6th fully-connected layer in both versions of the AlexNet set $h = d = 1$ and $w = 4096$. In cases where a fully-connected layer was used to extract features to be passed to the assembly of RNNs, the output matrix was set to have $w = h = 8$ to ensure compatibility with the assembly of RNNs. Each RNN within the assembly randomly mapped the input feature array into smaller feature sub-sets of size $1 \times 1 \times d$ each. The RNN outputs were then concatenated to form the final representation of the original input image.

The output features from the RNNs were used to train a classifier. The Softmax classifier powered by the Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm was applied to minimize the cross entropy error. The implementation of the Softmax classifier was adapted from the minFunc library described in [25].

B. EVALUATION OF THE PROPOSED APPROACH

The proposed object classification approach was validated and tested using the Washington RGBD image dataset (W-RGBD) [13], which contained images of 300 different objects placed on a turntable, and captured from around 200 different views. Each object view was described by a set of 3 images including RGB, depth, and mask. Only the RGB image was considered in this study as this mode is the most popular in practical applications. Each image primarily depicted the object, with occasional small elements of the background captured by the cropping box placed around the object. The 300 objects were grouped into 51 different categories or object classes. The object recognition task was thus to assign the correct class labels to a test set of unseen (not used in the training process) images depicting different object instances.

The complete W-RGBD dataset was split into mutually exclusive training and testing subsets. There was no validation involved, therefore training was stopped if any of the following conditions matched: 1) the maximum number of iterations was reached; 2) the gradient value had fallen below a threshold; 3) the change in cost function evaluation between consecutive iterations had fallen below a threshold.

The iteration limit and thresholds were kept as default from the minFunc implementation [25]. To ensure valid comparisons, the sizes of both subsets and the splitting rule were consistent with [12], [13], [16]. There are 10 preconfigured splitting profiles, with each profile completely removing one chosen instance (around 200 images) of each class for testing, and using the remaining instances in training. The classification results presented in this study were averaged over these 10 training and classification partitioning settings.

IV. EXPERIMENTS

A. EFFECT OF THE RNN UNIT SIZE ON THE OBJECT RECOGNITION ACCURACY

In this subsection, we evaluate the effect of using the RNN unit in combination with a pre-trained CNN (see Fig. 1) on the object recognition accuracy. Deep CNN features from images in the W-RGBD dataset were extracted at the 4th layer of the pre-trained AlexNet-2014. The receptive field size of RNN was 13×13.

TABLE 1. Effect of using RNN on recognition accuracy (applied on activations of layer 4 of AlexNet-2014).

Number of RNNs used	Accuracy ± Standard Deviation (%)	Feature size
128	89.34 ± 1.61	32768
64	89.22 ± 1.3	16384
8	86.72 ± 1.47	2048
No RNN	86.37 ± 1.45	43264

As shown in Table 1, using the RNN as in the proposed approach can effectively improve the recognition accuracy by approximately 3%, and also produces a more compact representation of object's image. By using 8 RNN structures alone, this system is already able to generate a feature set that is 20 times smaller in size but provides a competitive performance to the raw CNN feature set.

Figure 3 shows a plot of classification accuracy versus the number of RNNs used, ranging from 1 to 128. For the purposes of comparison, Fig. 3 also includes results for closely related approaches including grayscale clustering based on the CRNN proposed in [26], and the RGB clustering based on the CNN proposed in [12]. It can be observed that the AlexNet-RNN proposed in this paper provided a significant improvement in recognition accuracy compared to both the RGB clustering based on CRNN and the grayscale clustering based on CRNN.

It is important to note that clustering-based convolutional weights (both RGB and grayscale) were trained very specifically for the target dataset, while the AlexNet-based convolutional weights were trained on the ImageNet dataset, which has a different data distribution than the target dataset. This provides strong evidence of the power of deep convolutional features in terms of discrimination and generalization. Nevertheless, the performance of all three models presented in Fig. 3 becomes constant above around 64 RNNs, which may either imply a processing limit of the RNN processing, the Softmax classifier, or both.

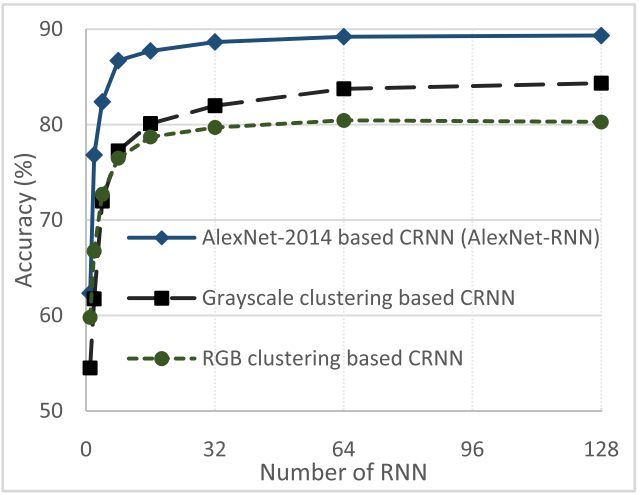

FIGURE 3. Recognition performance versus number of RNN.

TABLE 2. Performance comparison of proposed method with previous works.

Method	Accuracy (%)	Feature size
EMK feature & Histograms + SVM [13]	74.7 ± 3.6	>1500
RGB-CRNN + Softmax [12]	80.8 ± 4.2	16384
SP-HMP + SVM [14]	82.4± 3.1	590000
Deep-CNN + SVM [18]	83.1 ± 2.0	5096
Grayscale-CRNN + Softmax [26]	84 ± 2.9	16384
Fine-tuning Deep-CNN + Softmax [17]	84.1 ± 2.7	4096
CNN-SPM-RNN + SVM [15]	85.2 ± 1.2	4000
CNN-Fisher + SVM [16]	86.8 ± 2.2	1568000
AlexNet-RNN+ Softmax (current proposal)	**89.3 ± 1.6**	**32768**
AlexNet-RNN+ SVM (current proposal)	**89.7 ± 1.7**	**32768**

B. COMPARISON WITH RELATED STUDIES

This subsection compares the proposed AlexNet-RNN approach with previous approaches. The experiment settings are kept unchanged from subsection IV-A. As shown in Table 2, the AlexNet-RNN has significantly surpassed the current state-of-the-art approach CNN-Fisher [16] by nearly 3%. In addition, the features provided by the AlexNet-RNN are two orders of magnitude lower in size than the size of the features generated by the next best performer CNN-Fisher+SVM. This smaller feature data space allows for the use of a simpler classifier and thus increases classification speed. In this regard, many of experiments in our work use

the L-BFGS Softmax classifier, which is cheaper to train and faster to use compared to a Support Vector Machine (SVM). If the SVM is used as classifier, the accuracy can be slightly increased, however, with a significantly longer training time. Empirically, the SVM configured in a one-vs-one coding design takes about 10 hours to be trained, while the Softmax classifier needs only 1 hour to be trained for the same task.

Table 2 contains two other recently proposed methods (described in [17] and [18], respectively), which have also utilized activations from a trained deep network as object descriptors to classify image objects from the W-RGBD dataset. In [17], the entire AlexNet was trained on ImageNet and then fine-tuned on the W-RGBD dataset. In terms of computational cost, provided that the 4 first layers of the deep network are shared with the model proposed in this paper, this approach has to pay for one extra convolutional layer of size $256\times13\times13$ and two fully connected layers of size 4096. The convolutional layer itself is larger than the structure proposed in this paper with 128 RNNs, not to mention the additional effort required to fine-tune the network in the training phase.

In [18], the input images to the deep CNN were preprocessed to reduce the effect of the background obscuring the object in the image, and then activations from layers 6 and 7 were concatenated and passed to the SVM classifier. This method requires knowledge of the object mask for preprocessing, which is not always available in practice. In a similar way to [17], this method needs to accommodate the cost of one convolutional and two fully connected layers, in addition to a complicated multiclass structure of SVM classifiers. In contrast, the proposed AlexNet-RNN approach does not require any preprocessing other than scaling the images to the target network input size. The proposed approach also does not alter any of the pre-trained network parameters. As shown in Table 2, these attributes of the AlexNet-RNN provide high computational efficiency combined with high accuracy object classification.

C. DETERMINING THE MOST EFFICIENT LAYERS FOR FEATURE EXTRACTION

This subsection compares the performance of features extracted from different layers of AlexNet to determine which layer provides the most discriminative output vector. Features from four intermediate layers of an 8-layer network were examined individually in this experiment, with respect to the object classification accuracy. These layers included the last 3 CNN layers (layer 3, 4, 5) and the first fully connected layer (layer 6). The AlexNet-2014 was used instead of the AlexNet-2012, as the model provided convolutional features of the same size across three layers 3, 4 and 5, which eliminated the effect of size in comparing features across layers. In addition, these layers were previously reported to provide a good balance between generalization and discrimination for image representation [9]. The lowest-level layers cannot produce a good abstracted representation of the image, while the highest-level layers provide features that are too specific to the dataset on which the network was trained on.

In this subsection, the receptive field size of RNNs was kept fixed at 13×13 for processing features from the 3rd, 4th, and 5th layers, while the receptive field size applied on features from 6th layer was 8×8.

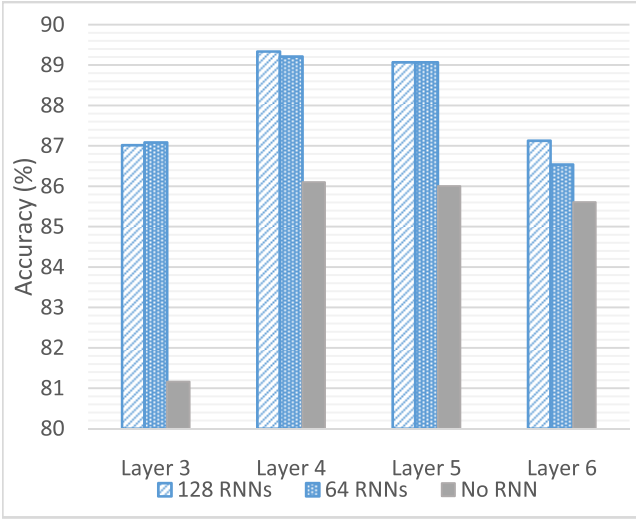

FIGURE 4. Recognition accuracy for features from several selected layers of AlexNet-2014.

Figure 4 shows that features from layer 4 of the AlexNet-2014 provided the best performance among all other selected layers, regardless of whether the RNN was used. It is important to restate at this point that since the AlexNet-2014 was used, the sizes of feature sets from the three selected CNN layers (layers 3, 4 and 5) were equal. In particular, there were 43264 features before the RNN and 32768 features after the RNN processing. The size of features generated by layer 6 was smaller (4096 weights before the RNN), but these features led to a significant drop in performance compared to layers 4 and 5.

TABLE 3. Performance difference of features from AlexNet-2014 and AlexNet-2012 across layers.

	No RNN		128 RNNs	
Layer	AlexNet-2012	AlexNet-2014	AlexNet-2012	AlexNet-2014
L4	84.93%	86.10%	88.30%	89.34%
L5	85.45%	86.00%	88.35%	89.07%
L6	83.61%	85.61%	86.22%	87.13%

Table 3 indicates that the AlexNet-2012 has shown very similar trends, however, the best performing layer was layer number 5. In general, provided that the AlexNet-2012 and the AlexNet-2014 have very similar configurations, this result appears to be different to [9], where features from layer 6 of the AlexNet-2012 were reported to provide the highest object recognition accuracy.

One of the potential disadvantages of using the fully connected layer 6 is that activations from this layer do not

contain the spatial information that is present in the lower layers. Meanwhile, the RNN component of the AlexNet-RNN structure was designed to capture repetitive patterns in the time domain, therefore the RNN processing was not necessarily expected to generate an advantage on features collected from this layer. However, as can be seen in Fig. 4, applying the RNN processing to features from layer 6 still improved classification performance by almost 2% compared to the AlexNet without the RNN processing. One of the possible factors contributing to this improvement could be the randomness of weights of the RNNs.

Further, in this experiment, we reshaped the output of the fully connected-layer, which was originally a vector of size 4096, into a 3-dimensional matrix of several different sizes. By varying the number of RNNs used we maintained the same feature size of 8192 weights for classification in all deformation settings. It was observed that reshaping the 6^{th} layer activation vector into an $8 \times 8 \times 16$ matrix and using the RNN receptive field size of 8×8resulted in the highest recognition performance. These results are presented in Table 3 and in Fig. 4, and further exploration of these findings will be conducted in our future work.

D. CLASSIFICATION ACCURACY: AlexNet-2012 vs AlexNet-2014

The experimental settings in this subsection are similar to settings in previous subsection IV-C. The average recognition rates obtained when using the AlexNet-2012 and AlexNet-2014 models are shown in Table 3. Features from AlexNet-2014 provide slightly higher recognition accuracy compared to the features from AlexNet-2012. More importantly the best AlexNet-2014 features are those extracted from layer 4, while the best AlexNet-2012 features are the features from layer 5, which involves the computation of one more convolutional stage of size $13 \times 13 \times 256$. Layers 3 and 4 of AlexNet-2012 are also 1.5 times larger compared to the corresponding layers of AlexNet-2014, thus demanding higher computational resources. At the same time, with RNN processing applied, both AlexNet-2012 and AlexNet-2014 network models outperform previous state of the art results achieved by the more complex CNN-Fisher (see Table 2).

E. EFFECT OF IMAGE NORMALIZATION

Since some but not all of the object recognition studies applied brightness and contrast normalization to the input data, the experiments described in this paper have tested both cases. The mean values were subtracted from the image intensities and the results were divided by the standard deviation of the data set. The results showed that the normalized images provided slightly lower classification results compared to the original unnormalized images.

To be specific, Table 4 shows the object classification accuracy using AlexNet-RNN with 128 RNNs and activation from layer 4. The results indicate that normalization based on the ImageNet, which was the data set used to train the AlexNet, led to better results than normalization based on the W-RBGD data; however, both cases were outperformed by classification performed without image normalization.

TABLE 4. Object classification accuracy using AlexNet-2014 with 128 RNNs and activation from layer 4.

No image normalization	Normalized against ImageNet data	Normalized against W-RBGD data
89.34%	88.63%	88.00%

V. CONCLUSION

This paper presented a new method of feature extraction from a deep Convolutional Neural Network (CNN) trained on a large dataset (AlexNet) and combined with the Recursive Neural Network structure (AlexNet-RNN).

Object recognition experiments conducted on the Washington RGBD image data set have shown that the proposed method has the advantages of structural simplicity combined with the ability to provide state of the art performance at a low computational cost compared to the AlexNet.

The proposed approach requires no training during the feature extraction stage, and can be performed very efficiently. The output features are compact and highly discriminative, and can be used with a simple multi-class classifier in object recognition settings.

Experimental results showed that the activation weights from the 4^{th} layer of the pre-trained AlexNet-2014, when combined with RNN processing, can achieve the highest recognition accuracy on the RGB images from the W-RGBD dataset.

ACKNOWLEDGMENT

The authors would like to thank Dr M. Schmidt for the useful advice on the use of his well-known minFunc library. They also want to thank and acknowledge Datalogic Vietnam for their support for our object recognition research.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[3] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," presented at the 22nd Int. Joint Conf. Artif. Intell., vol. 2. Barcelona, Spain, 2011.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. (2014). "Return of the devil in the details: Delving deep into convolutional nets." [Online]. Available: https://arxiv.org/abs/1405.3531

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. (2013). "OverFeat: Integrated recognition, localization and detection using convolutional networks." [Online]. Available: https://arxiv.org/abs/1312.6229
[9] J. Donahue *et al.* (2013). "DeCAF: A deep convolutional activation feature for generic visual recognition." [Online]. Available: https://arxiv.org/abs/1310.1531
[10] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
[11] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 129–136.
[12] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 665–673.
[13] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 1817–1824.
[14] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for RGB-D based object recognition," in *Experimental Robotics*. Springer, 2013, pp. 387–402.
[15] Y. Cheng, X. Zhao, K. Huang, and T. Tan, "Semi-supervised learning and feature evaluation for RGB-D object recognition," *Comput. Vis. Image Understand.*, vol. 139, pp. 149–160, Oct. 2015.
[16] Y. Cheng, R. Cai, X. Zhao, and K. Huang, "Convolutional Fisher kernels for RGB-D object recognition," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2015, pp. 135–143.
[17] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 681–687.
[18] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1329–1335.
[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.
[21] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Computer Vision—ECCV*. Springer, 2014, pp. 584–599.
[22] M. Long and J. Wang. (2015). "Learning transferable features with deep adaptation networks." [Online]. Available: https://arxiv.org/abs/1502.02791?mode=reply
[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV*. Springer, 2014, pp. 818–833.
[24] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," presented at the 23rd ACM Int. Conf. Multimedia, Brisbane, Australia, 2015.
[25] M. Schmidt. (2012). *minFunc: Unconstrained Differentiable Multivariate Optimization in MATLAB*. [Online]. Available: http://www.di.ens.fr/mschmidt/Software/minFunc.html
[26] H. M. Bui, M. Lech, E. Cheng, K. Neville, and I. S. Burnett, "Using grayscale images for object recognition with convolutional-recursive neural network," in *Proc. IEEE 6th Int. Conf. Commun. Electron. (ICCE)*, 2016, pp. 321–325.
[27] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[28] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385
[29] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2014.
[30] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

HIEU MINH BUI received the bachelor's degree in electronics and communication from the Da Nang University of Technology, Vietnam, in 2011, and the master's degree in electronic and computer engineering from RMIT University, Vietnam in 2014, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include computer vision and machine learning.

MARGARET LECH received the M.S. degree in physics from Maria Curie-Skłodowska University, Poland, and the Ph.D. degree in electrical engineering from the University of Melbourne, Australia. She is currently an Associated Professor with the School of Engineering, RMIT University, Australia. Her research interests include psychoacoustic, speech and image processing, system modeling, and optimization.

EVA CHENG (M'05) received the Ph.D. degree in telecommunications engineering from the University of Wollongong, Australia. She is currently a Lecturer with the School of Engineering, RMIT University, Australia. Her research interests include multimedia signal processing, 3D video/audio recording and reproduction, computer vision, and speech/audio processing.

KATRINA NEVILLE (M'10) received the Ph.D. degree from RMIT University, Australia. She is currently a Lecturer with the School of Engineering, RMIT University, Australia. Her research interests include speech and image processing and signal processing for communication applications.

IAN S. BURNETT (SM'02) received the Ph.D. degree from the University of Bath, Bath, U.K., in 1992. He is currently a Professor and the Dean of the Faculty of Engineering and Information Technology with the University of Technology Sydney, Australia. His current research interests include speech processing, 3D audio reproduction, recording and transmission, semantic media content description, and 3D video processing and quality of multimedia experience. He is a member of the Editorial Board of the IEEE Multimedia. He was a Chair of multimedia description schemes at Moving Picture Experts Group.