Scientific Research Publishing

Journal of Signal and Information Processing, 2018, 9, 92-110
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

DM-L Based Feature Extraction and Classifier Ensemble for Object Recognition

Hamayun A. Khan

Faculty of Computer Studies, Arab Open University, Industrial Ardiya, Kuwait
Email: h.khan@arabou.edu.kw

How to cite this paper: Khan, H.A. (2018) DM-L Based Feature Extraction and Classifier Ensemble for Object Recognition. *Journal of Signal and Information Processing*, **9**, 92-110.
https://doi.org/10.4236/jsip.2018.92006

Received: July 7, 2017
Accepted: May 28, 2018
Published: May 31, 2018

Abstract

Deep Learning is a powerful technique that is widely applied to Image Recognition and Natural Language Processing tasks amongst many other tasks. In this work, we propose an efficient technique to utilize pre-trained Convolutional Neural Network (CNN) architectures to extract powerful features from images for object recognition purposes. We have built on the existing concept of extending the learning from pre-trained CNNs to new databases through activations by proposing to consider multiple deep layers. We have exploited the progressive learning that happens at the various intermediate layers of the CNNs to construct Deep Multi-Layer (DM-L) based Feature Extraction vectors to achieve excellent object recognition performance. Two popular pre-trained CNN architecture models *i.e.* the VGG_16 and VGG_19 have been used in this work to extract the feature sets from 3 deep fully connected multiple layers namely "fc6", "fc7" and "fc8" from inside the models for object recognition purposes. Using the Principal Component Analysis (PCA) technique, the Dimensionality of the DM-L feature vectors has been reduced to form powerful feature vectors that have been fed to an external Classifier Ensemble for classification instead of the Softmax based classification layers of the two original pre-trained CNN models. The proposed DM-L technique has been applied to the Benchmark Caltech-101 object recognition database. Conventional wisdom may suggest that feature extractions based on the deepest layer *i.e.* "fc8" compared to "fc6" will result in the best recognition performance but our results have proved it otherwise for the two considered models. Our experiments have revealed that for the two models under consideration, the "fc6" based feature vectors have achieved the best recognition performance. State-of-the-Art recognition performances of 91.17% and 91.35% have been achieved by utilizing the "fc6" based feature vectors for the VGG_16 and VGG_19 models respectively. The recognition performance has been achieved by considering 30 sample images per class whereas the proposed system is capable of achieving improved performance by considering all

sample images per class. Our research shows that for feature extraction based on CNNs, multiple layers should be considered and then the best layer can be selected that maximizes the recognition performance.

Keywords

Deep Learning, Object Recognition, CNN, Deep Multi-Layer Feature Extraction, Principal Component Analysis, Classifier Ensemble, Caltech-101 Benchmark Database

1. Introduction

Deep Learning is one of the most important areas of research that is currently finding adoption in wide and diverse fields [1] [2] [3]. Deep Learning techniques and models exist for Image Recognition tasks, Natural Language Processing (NLP) tasks, Control Systems Applications, Medical Applications and many other diverse tasks. Traditionally, Neural Networks have been the mainstay of Machine Learning algorithms and applications to solve complex classification and recognition problems. Due to the recent advancement in the field of Neural Networks, Deep Learning has become one of the most rapidly developing area in the general domain of Artificial Intelligence (AI) and Machine Learning (ML). A large number of frameworks and libraries exist for deep learning research and development such as Tensorflow, Theano, Torch, Caffe and Keras etc. [1] [2] [3]. Some of the popular Deep Learning systems and architectures include CNNs, the Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Deep Belief Networks (DBNs), Deep Stacking Networks (DSNs) and Generative Adversarial Networks (GANs). Many large international corporations are involved in the research and development of deep learning systems including Google, IBM, Microsoft, Amazon, Apple, Facebook, Baidu and others. Along with the enthusiasm for advancement in AI and ML research, there are concerns about the social, ethical and possible job loss issues due to expected rapid automation and deployment of intelligent machines and robots in the work place. Concerns about possible safety issues related to AI have also been raised by professionals and researchers of this field stressing for the advancement of safe AI research and development.

Object recognition is an important area of research in the field of ML since it forms an essential part of intelligent systems related to Computer Vision, Robotics and Autonomous Vehicles. Conventional Object Recognition systems generally employ Neural Networks to make decision about object recognition based on feature sets that have been extracted from the object images. Deep Learning systems are also being used for Object Recognition tasks. For Image and Object Recognition purposes, the most popular Deep Learning techniques are based on the Convolution Neural Networks (CNNs) [4]-[13]. The CNN utilizes a series of layers including the Convolutional layers, Pooling layers, the

Rectified Linear Unit (ReLU) layers, Fully Connected layers and Softmax layers to achieve recognition.

In this work we have built on the concept of extending the learning capabilities of pre-trained deep networks to other datasets, as proposed in [14] [15] by considering multiple deep layers for this purpose. Specifically, we have proposed to use Deep Learning systems *i.e.* pre-trained CNN models, to extract feature sets from object databases along with employing an external ensemble of classifiers to make decisions about object recognition. By utilizing pre-trained CNN models for feature extraction, we exploit the learning capabilities that pre-trained models possess and which they have acquired through the long training times on very large databases. Our proposed system uses multiple layers of CNN models for feature extraction and then it selects the best layer based on the classification performance. By employing the Classifier Ensemble technique for classification purposes, our proposed system exploits the power of group of classifier for decision making instead of a single classifier. By exploiting the aforementioned capabilities, our proposed object recognition system has achieved State-of-the-art recognition performance. Hence this work is an important contribution towards highlighting the significance of considering multiple layers of deep CNNs for feature extraction purposes and then selecting the best layer based feature vector that maximizes the classification performance.

The breakdown of this paper is as follows. Review of related work about Image and Object recognition has been provided in Section 2. Our proposed DM-L based Feature Extraction and Classifier Ensemble system has been introduced and discussed in Section 3 of this paper. Section 4 presents the experimental setup of the proposed system for simulations and the achieved results are also presented in this section. The summary of the results is discussed in Section 5 and Comparison of the results with the State-of-the-Art is presented in Section 6. Finally the Conclusions and Future Research Directions have been discussed in Section 7.

2. Related Work

Object Recognition is an essential task associated with Computer Vision, Robotics, Autonomous Vehicle Control or Driverless Cars and many other important tasks. Traditional Object Recognition techniques are based on the conventional ML strategy employing feature extraction from object images and performing recognition by using a standard classifier on the extracted features. The recognition performance of traditional approaches depends on the quality of extracted features as well as on the quality of the classifier system. Apart from traditional approaches, Deep Learning has also been extensively employed for Object Recognition using CNNs. CNNs are usually employed in two roles for object recognition purposes *i.e.* either as complete object recognition system or as feature extractors. In this work we have employed pre-trained CNN models for feature extraction purposes along with an external classifier ensemble.

LeCun *et al.* have applied CNNs to the task of object recognition in [4] and they have shown that CNNs performed the best compared to other object recognition techniques. Hinton *et al.* [5] have applied deep CNNs to the task of image classification on the Imagenet database and they have achieved the best performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The work of Hinton *et al.* thus ushered a flurry of research activities in the field of deep learning using deep CNNs for image recognition and classification tasks [6]-[12]. Gu *et al.* have presented a good survey on recent advances in CNNs in [13]. Since we have considered the benchmark Caltech101 database in this work, we discuss here a review of the major research efforts conducted by researchers using this database. Simonyan *et al.* have applied very deep convolutional networks for large scale image recognition purposes and they have made their best performing deep learning networks *i.e.* the VGG-16 and VGG-19 public for other researchers [14]. They have also shown in their work that their pre-trained models generalize well to other datasets such as Caltech-101 and Caltech-256. For generalization purposes, they have considered the penultimate layer for feature extraction and have used an SVM based external classifier to achieve a mean class recall of 92.7 ± 0.5 on the Caltech-101 database [14]. Zeiler *et al.* [15] have reported a classification accuracy of 86.5% on the Caltech-101 database in their work on visualization and understanding convolutional networks. They have presented strong useful feature visualizations and activations in their work. They have also discussed the strong generalization of features from one system trained on the ImageNet database to other databases such Caltech-101. Our current work builds on these concepts by considering multiple deeper layers of pre-trained CNN models for feature extraction, their subsequent dimensionality reduction and by selecting the best layer features based on performance of our Classifier Ensemble. He *et al.* [16] have considered the problem of visual recognition by applying Spatial Pyramid Pooling to Deep Convolutional Networks. They have applied their technique to the Caltech-101 database and have achieved a classification accuracy of 91.44%. Convolutional nets have been applied by Chatfield *et al.* in [17] to Caltech-101 database and a recognition accuracy of 88.4% has been achieved by their system. Object categorization through Group-Sensitive Multiple Kernel Learning has been considered by Yang *et al.* in [18]. They have reported a recognition performance of 84.3% on the Caltech-101 database.

Two of the major challenges associated with Deep Learning Systems involve long training times and requirement of large databases for effective learning. In this work, the above mentioned problems have been alleviated by using the pre-trained CNN models to extract feature vectors from these pre-trained models to save on the training time. Also, instead of using the Softmax layer as basis for classification, we feed the extracted feature sets to an external Classifier Ensemble. Hence, we will show in this work that by using our proposed new DM-L approach, a higher classification accuracy is achievable along with saving on the

training time of a Deep Learning system. It will be shown in this work that a State-of-the-Art classification accuracy is achievable on the Benchmark Caltech-101 dataset using the Feature Set extracted from pre-trained CNN architectures and the Classifier Ensemble technique.

3. The Proposed System

We propose a new Deep Multi-Layer (DM-L) Based Feature Extraction and Classifier Ensemble system for object recognition in this work. Our proposed system builds on the concepts in [14] [15] and it is based on utilizing the learning capabilities of advanced pre-trained CNN architectures by using deep multiple layers to extract features from image databases for Object Recognition purposes. Hence our system exploits the learning achieved by the various deep layers contained in a Convolutional Neural Network (CNN) for feature extraction. Since our proposed system makes use of the available pre-trained CNN models therefore it achieves substantial savings on the long training times associated with Deep Learning systems. It also uses a separate Ensemble of Classifiers approach to the classification task using the features extracted from the multiple layers namely "fc6", "fc7" and "fc8" layers of the CNN models instead of a single deepest fully connected layer. The block diagram of the proposed new DM-L based object recognition system is shown in **Figure 1** below with feature extraction from multiple layers. The main characteristics and major sub-blocks of the proposed system are discussed below.

3.1. Image Database

The Benchmark Object Recognition database that we have considered in this work is the Caltech-101 database from the Caltech object category datasets [19] [20]. This database consists of various objects of 102 classes, including the Google background object, and majority of the images in this database are of size 300 × 200 pixels. Most of the classes have about 50 images in the database while some classes have more number of images. As suggested by Fie Fei Li in [19], we have considered 30 images per class in all of our simulations and experiments. This restriction on the unified number of images per class ensures unified treatment of each class without preferential treatment of class with large number of images such as the "airplanes" category. A montage of various classes of objects from this benchmark database is shown in **Figure 2**.

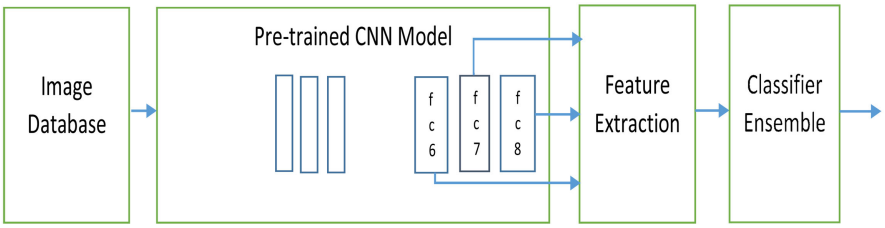

Figure 1. The proposed DM-L based feature extraction and classifier ensemble system.

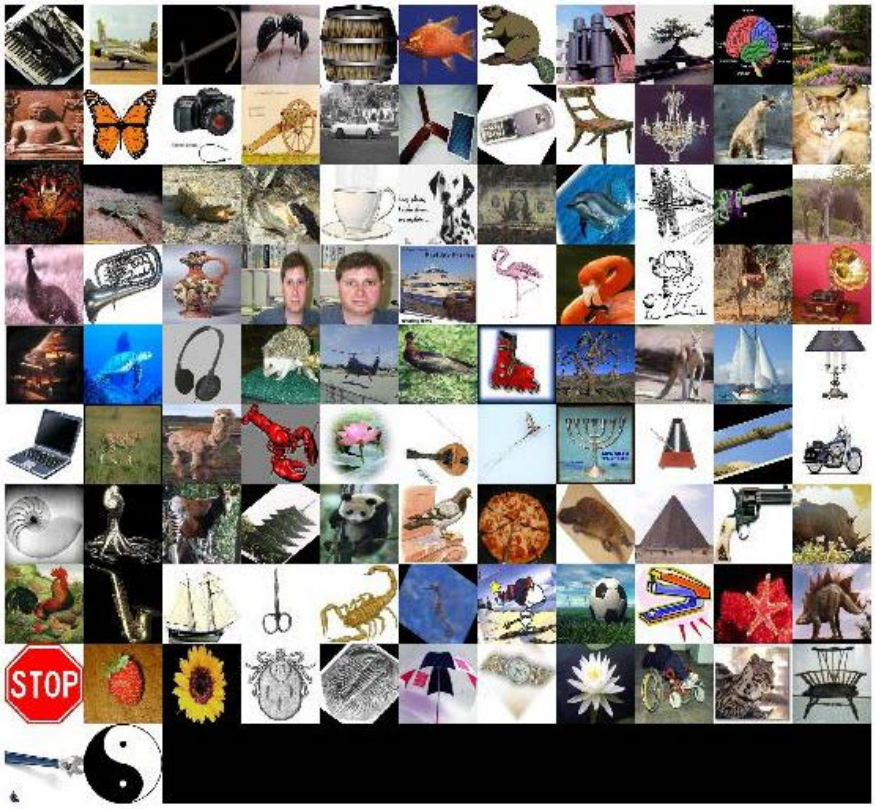

Figure 2. Montage of selected object images from Benchmark Caltech-101 database.

3.2. The Pre-Trained CNN Model

The CNNs are powerful Deep Learning systems that achieve learning by utilizing their deep architecture and extensive training on huge data sets. Our proposed system employs pre-trained CNN models to exploit their learning capabilities by using their deep layers for feature extraction from new databases through the process of activations. Our approach thus builds on the work of Simonyan *et al.* [14] and Zieler *et al.* [15] who studied the generalization of learning from one trained system to new databases through activations. We have extended this concept to extracting features from multiple layers, their Dimensionality Reduction and the subsequent use of a Classifier Ensemble instead of the Softmax layer for classification purposes. In order to overcome the shortcoming of very long training time required to train Deep Learning systems, the proposed DM-L based object recognition system makes use of popular pre-trained CNN models for Feature Extraction purposes. The pre-trained CNN models used in this work include the VGG_16 and VGG_19 models [14] and these are available as Matlab support packages [21] [22]. These models have been trained extensively on the ImageNet Benchmark object recognition database. The ImageNet database consists of about one million images of 1000 object classes and it is considered as a premier database in object recognition research. Since these models have already been trained on the ImageNet database, these models will be used in this work to extract features from our Object Recognition Benchmark Database *i.e.* the Cal-

tech-101 database. This methodology enables us to use the learning of the pre-trained models to extract features from the Caltech-101 database.

3.3. Feature Extraction Stage

The Feature Extraction stage consists of two major processes *i.e.* the DM-L based Feature Extraction and the subsequent Dimensionality Reduction of the extracted feature vectors.

3.3.1. DM-L Based Feature Extraction

A conventional Deep Learning system used for image object recognition purposes, such as a CNN, consists of a large number of deep layers through which learning is achieved. The learning achieved by the various layers is progressive in nature with the earlier layers learning certain basic characteristics of the images while the deeper layers achieve learning about more detailed characteristics of the images. The proposed system extracts features from the deepest 3 fully connected layers of a pre-trained CNN model. These multiple layers are named as "fc6", "fc7" and "fc8" with "fc8" being the deepest fully connected layer before the classification stage of the CNN. These DM-L based features are extracted from the pre-trained models using the "activations" function available in Matlab [21] [22]. It is emphasized here that conventional wisdom about CNNs suggests using the deepest fully connected layer for feature extraction but we are proposing to use multiple deep layers for feature extraction and then selecting the one that maximizes recognition performance. We use the Caltech-101 Benchmark Database to extract the feature vectors from the 3 mentioned deep multiple layers of the pre-trained CNN models. These features capture the learning achieved by the pre-trained CNN models on the new Caltech-101 database. As mentioned earlier, the fully connected layers of CNNs have been used before for recognition purposes [14] [15] but in this work we are proposing to use multiple layers of CNNs for feature extraction along with an external Classifier Ensemble for recognition purposes and this is a new novel approach that we have proposed in this work.

3.3.2. Dimensionality Reduction

The features extracted from the multiple deeper layers of the CNN models exhibit high dimensionality and their subsequent use will over burden the classification stage by confusing the classifiers. Hence the proposed DM-L based classifier includes a Dimensionality Reduction step in which the PCA technique is used to achieve Dimensionality Reduction. Only a small percentage of the computed features are retained without affecting the performance of the subsequent classification stage. Actually, the reduction of the dimensionality of the feature space helps both in speed of computations as well as it results in improved classification accuracy as will be demonstrated in the experimental section of this article. This stage of the proposed DM-L based Feature Extraction for object recognition system presents a significant improvement in the performance of the proposed

system since the extracted feature vectors are of high dimension and their use without dimensionality reduction will overburden and confuse the system.

3.4. The Classifier Ensemble Stage

The use of the external classifier ensemble in this work is a novel approach to CNN based feature extraction and object recognition system's implementation. The classification stage of a conventional CNN is based on the Softmax layer but our proposed system implements this stage through the use of the powerful ensemble of classifier technique available for classification purposes. The features extracted from the 3 deep multiple layers of the CNN Model are fed to the external classifier ensemble. Our proposed system uses an ensemble of 15 base classifiers based on Linear Discriminant Analysis (LDA) technique to achieve excellent classification performance as will be demonstrated in the experimental section. For training the Classifier Ensemble [23] we have used the popular Bagging technique for ensemble training. For validation purposes, we have used the 10-fold cross validation strategy and have taken the average of 25 simulation runs as the recognition performance of the proposed system. Based on the recognition performance, the system identifies the best feature vector from amongst the three multiple layer vectors for recognition purposes.

4. Experimental Setup and Results

The experimental setup of the proposed DM-L based object recognition system consists of the selection of benchmark object recognition database, the pre-trained CNN models, the selection of deep multiple layers for feature extraction, the dimensionality reduction size selection and the selection of the various parameters of the classifier ensemble that is used at the last stage of the proposed recognition system. The experimental setup of our proposed system is given in **Table 1**.

We have performed a number of simulations to assess the recognition performance of the proposed system and the results of the experiments have been presented in the sub-sections given below.

4.1. Results of the Proposed System Using the Pre-Trained VGG_16 Model

The VGG_16 Model [14] is a popular Deep Learning system that was originally trained on some one million object images consisting of 1000 classes of the ImageNet benchmark database. This pre-trained model is available as Matlab support package [21]. The various layers of the VGG_16 Model are shown in **Table 2**. This model consists of a number of Convolution layers, Pooling layers, ReLU layers and 3 Fully Connected layers. The ReLU layers have been customarily omitted from the table in order to present a concise representation of the model. The 3 Fully Connected layers *i.e.* "fc6", "fc7" and "fc8" are present in the last or deeper part of the model.

Table 1. System setup for the proposed DM-L based feature extraction and classifier ensemble.

Proposed System's Sub-Part/Entity	Parameter Value Selection
Benchmark Object Database	Caltech-101
Sample Images/Class used in Simulations	30
Pre-Trained CNN Models	• VGG_16 • VGG_19
CNN Deep Multiple Layers for Feature Extraction	• "fc6" • "fc7" • "fc8"
Dimensionality Reduction/Size	300
Dimensionality Reduction Strategy	PCA
Classifier Ensemble	• Base Classifiers: Linear Discriminant Analysis (LDA) • Number of Base Classifiers: 15 • Ensemble Training Technique: Bagging • Validation Strategy: 10-Fold Cross Validation
Number of Simulation Runs	25

4.1.1. Visualization of the Achieved Learning by Layers of VGG_16 on Caltech-101

We highlight the learning achieved by various layers of the pre-trained VGG_16 model on the Caltech-101 by computing the relevant activations of these layers. The activations computed from convolutional layers named "conv1_2" and "conv2_2" have been plotted as montage in **Figure 3** and **Figure 4** respectively.

It is clear from figures that the VGG_16 model's initial layers have achieved learning about the shapes of various classes of the Caltech-101 database. The subsequent layers learn more detailed features from images as they travel through the deeper layers of the system.

4.1.2. DM-L Based Feature Vector Construction from VGG_16

The features extracted from the three fully connected multiple layers *i.e.* "fc6", "fc7" and "fc8" of VGG_16 pre-trained model are denoted as "Features_VGG_16_FC6", "Features_VGG_16_FC7" and "Features_VGG_16_FC8" respectively. In order to reduce the dimensionality of the feature vectors we use the PCA technique which results in the following reduced dimensionality feature vectors:

$$\text{Features\_VGG\_16\_FC6\_Reduced} = \text{PCA}(\text{Features\_ VGG\_16\_FC6}) \quad (1)$$

$$\text{Features\_VGG\_16\_FC7\_Reduced} = \text{PCA}(\text{Features\_ VGG\_16\_FC7}) \quad (2)$$

$$\text{Features\_VGG\_16\_FC8\_Reduced} = \text{PCA}(\text{Features\_ VGG\_16\_FC8}) \quad (3)$$

The dimensions of the feature vectors extracted from the "fc6" and "fc7" layers are 4096x1 per example and those of the "fc8" are of dimension 1000x1. In our simulations we have also computed the recognition performance of the

Table 2. Pre-trained VGG_16 model for deep learning with "fc6", "fc7" and "fc8".

Layer Name	Type	Description
"input"	mage Input	224 × 224 × 3 images with "zerocenter" normalization
"conv1_1"	Convolution	64 3 × 3 × 3 convolutions with stride [1 1] and padding [1 1]
"conv1_2"	Convolution	64 3 × 3 × 64 convolutions with stride [1 1] and padding [1 1]
"pool1"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv2_1"	Convolution	128 3 × 3 × 64 convolutions with stride [1 1] and padding [1 1]
"conv2_2"	Convolution	128 3 × 3 × 128 convolutions with stride [1 1] and padding [1 1]
"pool2"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv3_1"	Convolution	256 3 × 3 × 128 convolutions with stride [1 1] and padding [1 1]
"conv3_2"	Convolution	256 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"conv3_3"	Convolution	256 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"pool3"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv4_1"	Convolution	512 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"conv4_2"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv4_3"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"pool4"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv5_1"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv5_2"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv5_3"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"pool5"	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0]
"fc6"	Fully Connected	4096 fully connected layer
"drop6"	Dropout	50% dropout
"fc7"	Fully Connected	4096 fully connected layer
"drop7"	Dropout	50% dropout
"fc8"	Fully Connected	1000 fully connected layer
"prob"	Softmax	softmax
"output"	Classification Output	crossentropyex with "tench", "goldfish", and 998 other classes

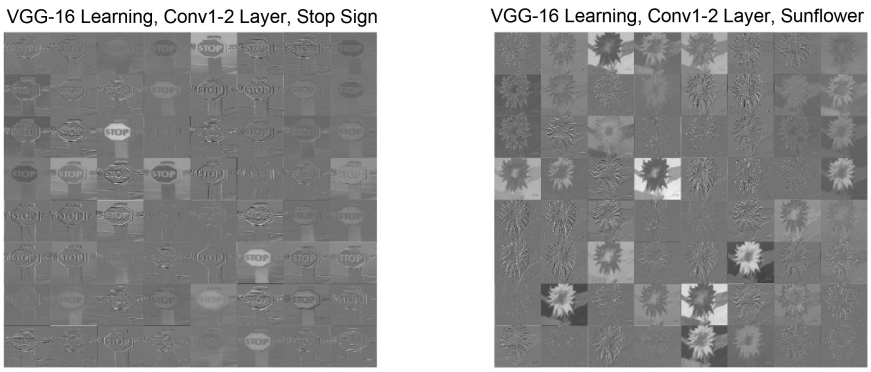

Figure 3. Learning exhibited by VGG_16's "conv1_2" layer on Caltech-101.

system for these feature vectors by dropping every tenth value for the "fc6" and "fc7" layers based feature vectors and every third value for the "fc8" layer based

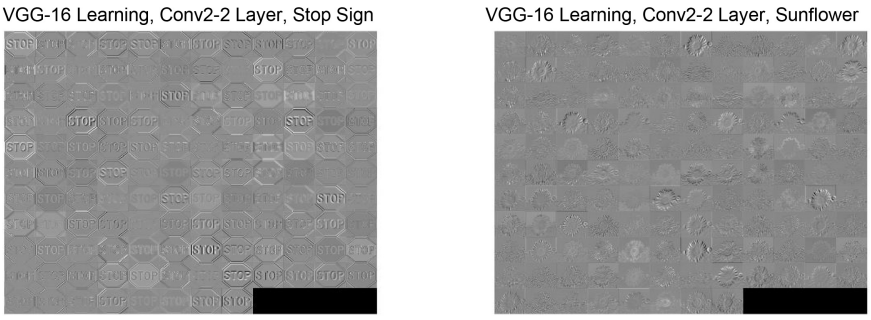

Figure 4. Learning Exhibited by VGG_16's "conv2_2" Layer on Caltech-101.

feature vectors. These feature vectors are expressed as:

$$\text{Features\_VGG\_16\_FC6\_Dropped} = \text{Features\_ VGG\_16\_FC6}[1{:}\,10{:}\,\text{end}] \quad (4)$$

$$\text{Features\_VGG\_16\_FC7\_ Dropped} = \text{Features\_ VGG\_16\_FC7}[1{:}\,10{:}\,\text{end}] \quad (5)$$

$$\text{Features\_VGG\_16\_FC8\_ Dropped} = \text{Features\_ VGG\_16\_FC8}[1{:}\,3{:}\,\text{end}] \quad (6)$$

The simulations for the PCA based dimensionality reduced feature vectors use a size of 300 for all 3 feature vectors.

4.1.3. Proposed System's Performance Using DM-L Based Features Extracted from VGG_16 Model

Instead of using the original Softmax Layer based classification scheme of the VGG-16 model, the proposed system uses an external Ensemble of Classifiers for final recognition decision. The multiple feature vectors are fed to the Classifier Ensemble system consisting of 15 base classifiers and the best layer feature vector is identified based on the recognition performance. The Ensemble uses Linear Discriminant Analysis (LDA) base classifiers and it uses the Bagging technique for ensemble training. We have performed 10-fold cross validation and have taken average of 25 runs as the system's recognition accuracy. Excellent recognition accuracy of 91.17% has been achieved for feature vectors extracted from the "fc6" layer. This performance has been achieved by using the PCA technique for dimensionality reduction thus enabling the classifier ensemble to achieve excellent results. The recognition performance of the new proposed DM-L based feature extraction and classifier ensemble technique using the pre-trained VGG-16 model is listed in **Table 3**.

The plots of the proposed system's recognition performance on the Caltech-101 database using the pre-trained VGG-16 Model are shown in **Figure 5** and **Figure 6**.

It is clear from the plots that "fc6" layer based feature vector has outperformed other layers based feature vectors in the object recognition task. Conventional wisdom about Deep Learning networks suggest that deeper layers would achieve more detailed learning about the recognition task at hand but the above results clearly highlight that consideration of multiple layers is advisable for feature extraction purposes. The above results highlight that the best layer to be considered for feature extraction for the VGG-16 based model is "fc6" layer and

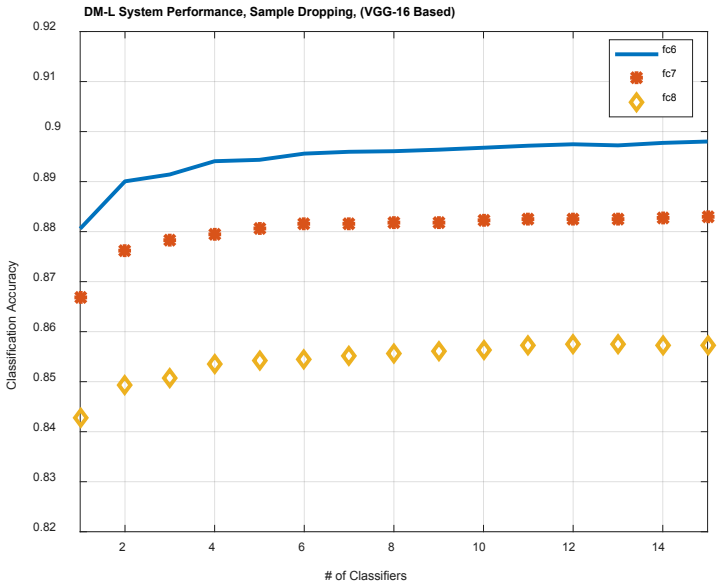

Figure 5. Performance using DM-L based features, sample dropping (VGG_16).

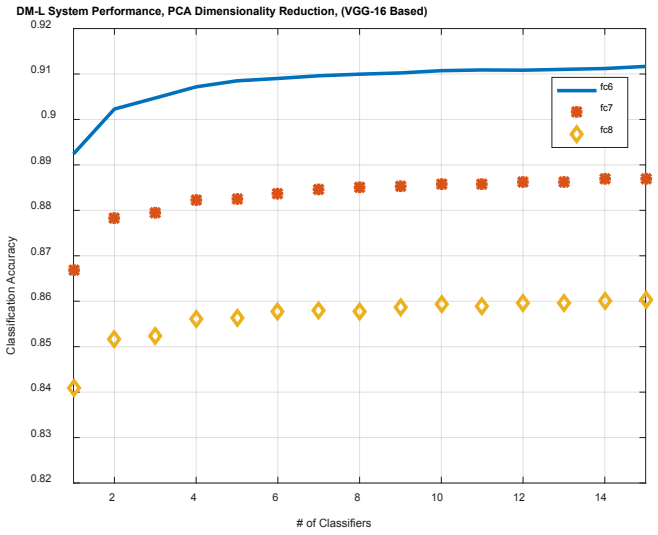

Figure 6. Performance using DM-L based features, dimensionality reduction (VGG_16).

Table 3. Proposed system's performance using DM-L based features extracted from VGG_16.

Feature Type	Recognition Accuracy (%)
Features_VGG_16_FC6_Dropped	89.80
Features_VGG_16_FC7_Dropped	88.29
Features_VGG_16_FC8_Dropped	85.72
Features_VGG_16_FC6_Reduced	**91.17**
Features_VGG_16_FC7_Reduced	88.69
Features_VGG_16_FC8_Reduced	86.04

not the "fc8" layer.

4.2. Results of the Proposed System Using the Pre-Trained VGG_19 Model

The second pre-trained CNN Model that we have considered in this work is the VGG-19 Model [14]. The VGG_19 Model is also another popular Deep Learning system that was originally trained on some one million object images consisting of 1000 classes of the ImageNet benchmark database. This pre-trained model is also available as Matlab support package [21]. The various layers of the pre-trained VGG_19 Model, with the ReLU layers omitted for concise representation, are shown in Table 4 below. The 3 Fully Connected layers *i.e.* "fc6", "fc7"

Table 4. The pre-trained VGG_19 model for deep learning with "fc6", "fc7" and "fc8".

Layer Name	Type	Description
"input"	Image Input	224 × 224 × 3 images with "zerocenter" normalization
"conv1_1"	Convolution	64 3 × 3 × 3 convolutions with stride [1 1] and padding [1 1]
"conv1_2"	Convolution	64 3 × 3 × 64 convolutions with stride [1 1] and padding [1 1]
"pool1"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv2_1"	Convolution	128 3 × 3 × 64 convolutions with stride [1 1] and padding [1 1]
"conv2_2"	Convolution	128 3 × 3 × 128 convolutions with stride [1 1] and padding [1 1]
"pool2"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv3_1"	Convolution	256 3 × 3 × 128 convolutions with stride [1 1] and padding [1 1]
"conv3_2"	Convolution	256 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"conv3_3"	Convolution	256 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"conv3_4"	Convolution	256 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"pool3"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv4_1"	Convolution	512 3 × 3 × 256 convolutions with stride [1 1] and padding [1 1]
"conv4_2"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv4_3"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv4_4"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"pool4"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"conv5_1"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv5_2"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv5_3"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"conv5_4"	Convolution	512 3 × 3 × 512 convolutions with stride [1 1] and padding [1 1]
"pool5"	Max Pooling	2 × 2 max pooling with stride [2 2] and padding [0 0]
"fc6"	Fully Connected	4096 fully connected layer
"drop6"	Dropout	50% dropout
"fc7"	Fully Connected	4096 fully connected layer
"drop7"	Dropout	50% dropout
"fc8"	Fully Connected	1000 fully connected layer
"prob"	softmax	softmax
"output"	Classification Output	crossentropyex with "tench", "goldfish", and 998 other classes

and "fc8" are present in the last part of the model. Some of the architectural differences between the pre-trained VGG_16 and VGG_19 Models are evident in the deeper convolutional layers as shown in Table 3 and Table 4.

4.2.1. Visualization of the Achieved Learning by Layers of VGG_19 for Caltech-101

Similar to the previous pre-trained CNN model, the activations computed from convolutional layers named "conv1_2" and "conv2_2" for the Caltech-101 database using the VGG-19 Model have been plotted as montage in Figure 7 and Figure 8 respectively.

The figures show the learning that has been achieved by the pre-trained VGG-19 based object recognition system's initial layers. More detailed learning is achieved as images travel through the deeper layers of the system.

4.2.2. DM-L Based Feature Vector Construction from VGG_19

Similar to the previous model, the features extracted from the three fully connected multiple layers *i.e.* "fc6", "fc7" and "fc8" of VGG_19 pre-trained model are denoted as "Features_VGG_19_FC6", "Features_VGG_19_FC7" and "Features_VGG_19_FC8" respectively. Similarly, in order to reduce the dimensionality of the feature vectors we use the PCA technique which results in the following reduced dimensionality feature vectors:

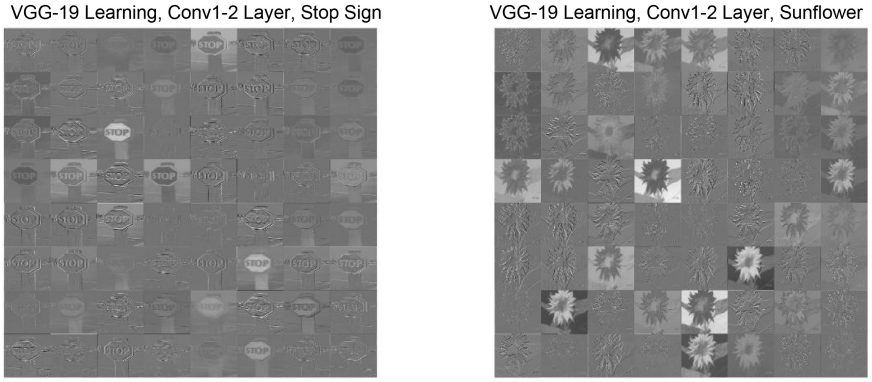

Figure 7. Learning exhibited by VGG_19's "conv1_2" layer on Caltech-101.

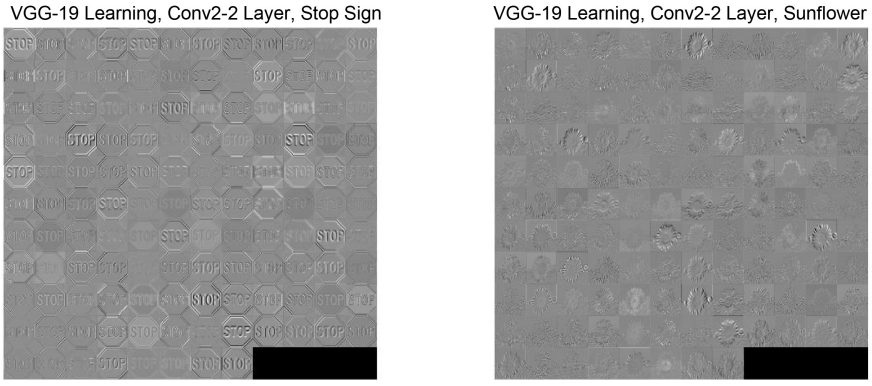

Figure 8. Learning exhibited by VGG_19's "conv2_2" layer on Caltech-101.

$$\text{Features\_VGG\_19\_FC6\_Reduced} = \text{PCA}(\text{Features\_ VGG\_19\_FC6}) \quad (7)$$

$$\text{Features\_VGG\_19\_FC7\_Reduced} = \text{PCA}(\text{Features\_ VGG\_19\_FC7}) \quad (8)$$

$$\text{Features\_VGG\_19\_FC8\_Reduced} = \text{PCA}(\text{Features\_ VGG\_19\_FC8}) \quad (9)$$

In this case too, the dimensions of the feature vectors extracted from the "fc6" and "fc7" layers are 4096x1 per example and those of the "fc8" are of dimension 1000x1. Here too, we have also computed the recognition performance of the system for these feature vectors by dropping every tenth value for the "fc6" and "fc7" layers based feature vectors and every third value for the "fc8" layer based feature vectors. These feature vectors are expressed as:

$$\text{Features\_VGG\_19\_FC6\_Dropped} = \text{Features\_ VGG\_19\_FC6}[1{:}\,10{:}\,\text{end}] \quad (10)$$

$$\text{Features\_VGG\_19\_FC7\_Dropped} = \text{Features\_ VGG\_19\_FC7}[1{:}\,10{:}\,\text{end}] \quad (11)$$

$$\text{Features\_VGG\_19\_FC8\_Dropped} = \text{Features\_ VGG\_19\_FC8}[1{:}\,3{:}\,\text{end}] \quad (12)$$

Like the previous model, the simulations for the PCA based dimensionality reduced feature vectors for the VGG-19 model also use a size of 300 for all 3 feature vectors.

4.2.3. Classifier Ensemble Performance Using DM-L Based Features Extracted from VGG_19 Model

Similar to the previous model, the feature vector extracted from the three multiple layers *i.e.* "fc6", "fc7" and "fc8" of the VGG_19 pre-trained model are fed to the Classifier Ensemble consisting of 15 base classifiers. Similarly, the best layer feature vector is identified based on the recognition performance. Once again, the feature vectors extracted from the "fc6" layer have achieved excellent recognition accuracy of 91.35%. The recognition performance of the new proposed system using the pre-trained VGG-19 model is listed in Table 5 below.

Figure 9 and Figure 10 show the plots of the proposed system's recognition performance using the pre-trained VGG-19 Model on the Caltech-101 database.

Once again, it is clear from the plots that "fc6" layer based feature vector has outperformed other layers based feature vectors in the object recognition task. For the VGG-19 Model also, the above results highlight that the best layer to be considered for feature extraction is "fc6" layer and not the last fully connected layer. It is mentioned here that VGG-19 is a deeper network compared to VGG-16 and hence it has achieved better performance compared to VGG-16

Table 5. Proposed system's performance using DM-L based features extracted from VGG_19.

Feature Type	Recognition Accuracy (%)
Features_VGG_19_FC6_Dropped	90.06
Features_VGG_19_FC7_Dropped	88.57
Features_VGG_19_FC8_Dropped	85.91
Features_VGG_19_FC6_Reduced	**91.35**
Features_VGG_19_FC7_Reduced	89.11
Features_VGG_19_FC8_Reduced	86.75

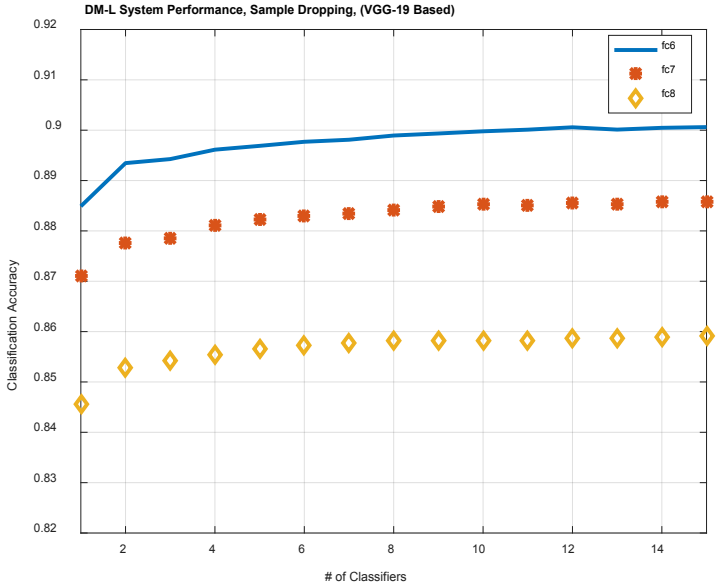

Figure 9. Performance using DM-L based features, sample dropping (VGG_19).

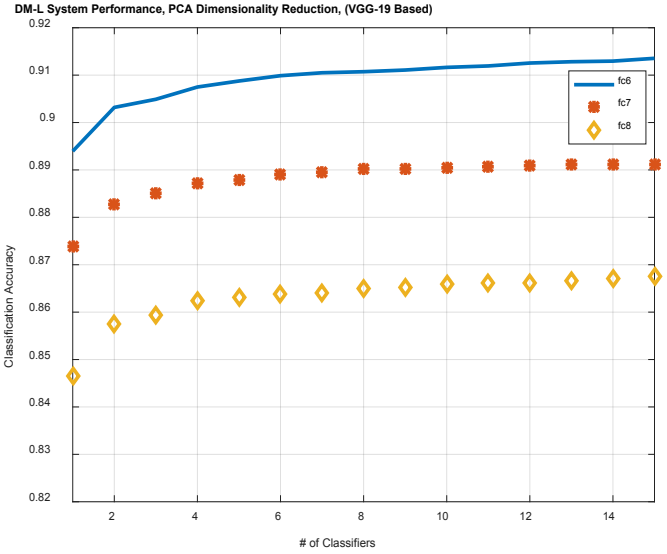

Figure 10. Performance using DM-L based features, dimensionality reduction (VGG_19).

based feature extraction.

5. Summary of Results

The results obtained from our simulations have been summarized in **Table 6**. It is clear from the results that the fully connected layer "fc6" has outperformed the other two deeper layers of the 2 pre-trained CNN models that we have considered in this work. The achieved results highlight the importance of considering multiple deeper layers for feature extraction purposes and then selecting the

Table 6. Summary of results of the proposed system's performance.

Feature Type	Recognition Accuracy (%)
Features_VGG_16_FC6_Dropped	89.80
Features_VGG_16_FC7_Dropped	88.29
Features_VGG_16_FC8_Dropped	85.72
Features_VGG_16_FC6_Reduced	**91.17**
Features_VGG_16_FC7_Reduced	88.69
Features_VGG_16_FC8_Reduced	86.04
Features_VGG_19_FC6_Dropped	90.06
Features_VGG_19_FC7_Dropped	88.57
Features_VGG_19_FC8_Dropped	85.91
Features_VGG_19_FC6_Reduced	**91.35**
Features_VGG_19_FC7_Reduced	89.11
Features_VGG_19_FC8_Reduced	86.75

layer that gives the best recognition performance.

6. Comparison of Results with State-of-the-Art

We have obtained results of 91.17% and 91.35% for the VGG-16 and VGG-19 pre-trained models respectively and our results are comparable with the state-of-the-art in [14] [15]. Our research has shown that it is prudent to use the multiple layers for feature extraction since the earlier fully connected layers may contain valuable information about the features of the images. The state-of-the-art in [14] has also removed the last fully connected layer *i.e.* the "fc8" in their work and our results also highlights this phenomena since the "fc6" based feature extraction has outperformed the last layer based feature vectors for object recognition purposes. Extracting feature vectors based on multiple layers of CNNs may be computationally expensive but the higher accuracies achievable by selecting the best layer for recognition may be worth the computational burden. The same is true for the classifier ensemble usage but the accuracy of the achieved results validates the use of the classifier ensemble for classification and recognition tasks.

7. Conclusions and Future Research Directions

In conclusion, it is mentioned that we have obtained comparable results with the state-of-the-art in using the deep layers of CNNs as feature extractors for object recognition purposes. The results have shown that using multiple layers for feature extraction and then selecting the best layer features is a wise approach to using deep layers of CNNs for feature extraction. In future, we plan to consider using the feature vectors extracted from different layers of CNNs in a combined or concatenated manner to evaluate the effect on recognition performance. Also, we plan to consider feature fusion approach to the extracted feature vectors since

the fusion of feature vectors from various layers may enhance the recognition performance of the system. We also plan to consider other pre-trained CNN models for feature extraction and compare the performance of subject feature vectors on other Benchmark Databases.

Acknowledgements

The research support and research opportunities provided by the Arab Open University (AOU) are highly appreciated and acknowledged by the author. This work has been made possible due to the positive research environment at AOU and a great desire on part of the management to develop a thriving research culture at AOU.

References

[1] Buduma, N. (2017) Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. O'Reilly Media, Sebastopol, CA.

[2] Bengio, Y.I., Goodfellow, J. and Courville, A. (2016) Deep Learning. MIT Press, Cambridge, MA. http://www.deeplearningbook.org

[3] Nielsen, M.A. (2015) Neural Networks and Deep Learning. http://neuralnetworksanddeeplearning.com/

[4] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**, 2278-2324. https://doi.org/10.1109/5.726791

[5] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) Imagenet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*, 1097-1105.

[6] Simonyan, K., Vedaldi, A. and Zisserman, A. (2013) Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv preprint arXiv:1312.6034.

[7] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014) Large-Scale Video Classification with Convolutional Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1725-1732. https://doi.org/10.1109/CVPR.2014.223

[8] Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y. and Yan, S. (2014) CNN: Single-Label to Multi-Label. arXiv preprint arXiv:1406.5726.

[9] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y. (2013) Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. arXiv preprint arXiv:1312.6229.

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., *et al.* (2015) Going Deeper with Convolutions. CVPR.

[11] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[12] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580-587. https://doi.org/10.1109/CVPR.2014.81

[13] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B. and Wang, G. (2015) Recent Advances in Convolutional Neural Networks. arXiv preprint arXiv:1512.07108

[14] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[15] Zeiler, M.D. and Fergus, R. (2013) Visualizing and Understanding Convolutional Networks. arXiv preprint arXiv:1311.2901

[16] He, K., Zhang, X., Ren, S. and Sun, J. (2014) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. arXiv preprint arXiv:1406.4729

[17] Chatfield, K., Simonyan, K., Vedaldi, A. and Zisserman, A. (2014) Return of the Devil in the Details: Delving Deep into Convolutional Nets. arXiv preprint arXiv:1405.3531

[18] Yang, J., Li, Y., Tian, Y., Duan, L. and Gao, W. (2009) Group-Sensitive Multiple Kernel Learning for Object Categorization. 12*th International Conference on Computer Vision*, Kyoto, 29 September-2 October 2009, 436-443.

[19] Fei-Fei, L., Fergus, R. and Perona, P. (2007) Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *Computer Vision and Image Understanding*, **106**, 59-70. https://doi.org/10.1016/j.cviu.2005.09.012

[20] Griffin, G., Holub, A. and Perona, P. (2007) Caltech-256 Object Category Dataset. http://authors.library.caltech.edu

[21] MATLAB Neural Network Toolbox. https://www.mathworks.com/help/nnet/ref/

[22] Machine Learning with MATLAB, MathsWorks. http://www.mathworks.com/solutions/machine-learning.html

[23] Zhou, Z., Wu, J. and Tang, W. (2002) Ensembling Neural Networks: Many Could Be Better than All. *Artificial Intelligence*, **137**, 239-263. https://doi.org/10.1016/S0004-3702(02)00190-X