RESFEATS: RESIDUAL NETWORK BASED FEATURES FOR IMAGE CLASSIFICATION

A. Mahmood[⋆], *M. Bennamoun*[⋆], *S. An*[⋆], *F. Sohel*[†]

[⋆] The University of Western Australia [†]Murdoch University

ABSTRACT

Deep residual networks have recently emerged as the state-of-the-art architecture in image classification and object detection. In this paper, we propose new image features (called ResFeats) extracted from the last convolutional layer of the deep residual networks pre-trained on ImageNet. We propose to use ResFeats for diverse image classification tasks namely, object classification, scene classification and coral classification and show that ResFeats consistently perform better than their CNN counterparts on these classification tasks. Since the ResFeats are large feature vectors, we explore dimensionality reduction methods. Experimental results are provided to show the effectiveness of ResFeats with state-of-the-art classification accuracies on Caltech-101, Caltech-256 and MLC datasets and a significant performance improvement on MIT-67 dataset compared to the widely used CNN features.

Index Terms— Deep learning, residual networks, object classification, scene classification, coral classification

1. INTRODUCTION

Deep convolutional neural networks (CNNs) have shown outstanding results on challenging image classification and detection datasets since the seminal work of [1]. Off-the-shelf image representations learned by these deep networks are powerful and generic. These generic features have been used to solve numerous visual recognition problems [2, 3]. Given the promising performance of these off-the-shelf CNN features, they have become the mainstream image features for solving most computer vision problems [4].

Recent evidence [5, 6, 3] suggests that off-the-shelf CNN features have outperformed previous handcrafted features for datasets with a limited amount of training data. These features are domain independent and can be transferred to any specific target task without compromising on performance [4]. Network width, depth and optimization parameters along with the network layer from which these features are extracted play a key role in the effectiveness of transfer learning. This paper attempts to provide an answer to the following question: *What are the criteria to select an initial deep network (pre-trained on ImageNet) to extract generic features in order to maximize performance and transferability across domains?* To answer this question, we hypothesise that a better optimized and a high performing deep network on ImageNet should result in more powerful and generic image representations. One such network is the deep residual network (ResNet) presented in [7].

ResNets are easier to train as opposed to other CNN architectures *e.g.* VGGnet [8]. For example, a 152-layer ResNet, which is 8 times deeper than VGGnet, is still less complex and can be learnt faster than VGGnet. Moreover, a 34-layer ResNet contains 3.6 billion multiply-add operations whereas a 19-layer VGGnet has 19.6 billion multiply-add operations (less than 20%) [7]. Very deep networks are known to cause overfitting and saturation in accuracy. However, residual learning and the identity mappings (shortcut connections) [9] in ResNets have been shown to overcome these problems. This enables ResNets to achieve outstanding results in image detection, localization and segmentation tasks [7]. In this paper, we explore the discrimination power of the image representations extracted from pre-trained ResNets. We name these off-the-shelf ResNet features as **ResFeats**.

The main contributions of this paper are: **(i)** We introduce ResFeats, which are image features extracted from pre-trained ResNets and test them on diverse image classification tasks including objects, scenes and corals. **(ii)** We analyse the performance of ResFeats extracted from ResNet-50 with those extracted from a deeper 152-layer ResNet. **(iii)** We propose a compact 2048-dimensional generic feature vector obtained after dimensionality reduction which is half of the size of the traditional CNN based feature vector (4096 dimensions). **(iv)** We show that ResFeats achieve a superior classification accuracy compared to off-the-shelf CNN features. We also provide experimental evidence that our proposed method achieves state-of-the-art performance on three out of the four popular and challenging image classification datasets.

2. RELATED WORK

Recent success stories [1, 8, 3, 10] have established deep CNNs as the mainstream method to solve challenging computer vision tasks. However, training a network from scratch requires a large amount of training data, time and GPUs. Donahue *et al.* [3] and Zeiler and Fergus [5] provided evidence that the generic image representations learned from pre-trained CNNs outperform previous state-of-the-art hand crafted features. However, they did not experiment on a large number of computer vision datasets. Razavian *et al.* [2] built on the concept of generic CNN features and proved that off-the-shelf CNN features outperform existing methods. They experimented with more than 10 datasets for tasks such as image classification, object detection, fine grained recognition, attribute detection and visual instance retrieval.

Chatfield *et al.* [11] evaluated the performance of CNN based methods for image classification and compared their methods with previous feature encoding methods. Their findings established that deeper CNN performed better than the shallower models of the same network trained on augmented data, where VGGnet [8] was used as the source CNN in their work. They improved the classification accuracies of popular datasets such as VOC, Caltech-101 and Caltech-256. He *et al.* [6] used spatial pyramid pooling of CNN features to further improve the classification accuracy on the Caltech datasets and reported state-of-the-art object classification results.

Scene classification is quite different from object classification due to the presence of multiple objects in a single scene. These object instances can be of varying size and pose, and can be located at different locations in a number of possible layouts in the test image. Consequently, the state-of-the-art performance on scene

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

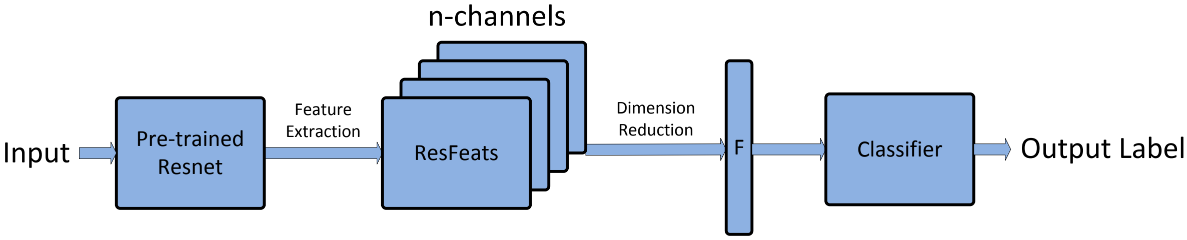

Fig. 1. Block diagram of the proposed method. F is the final feature vector obtained after dimension reduction.

datasets such as MIT-67 (81% in [12]) is comparatively lower than the performance on object classification datasets (93.4% for Caltech-101 in [6]). Cimpoi *et al.* [12] proposed Fisher Vector (FV) pooling of a deep CNN filter bank (FV-CNN) for texture and material classification. They achieved an accuracy of 81% on MIT-67 dataset (an improvement of 10% over previous state-of-the-art).

Coral classification is a task which is very different from the source dataset on which deep networks are pre-trained (ImageNet in this case). Despite this dissimilarity, off-the-shelf CNN features have improved the results of existing methods of coral classification [13, 14, 15], thereby demonstrating their strength for transfer learning. The baseline performance on MLC dataset was first reported in [16]. In [13], a hybrid (hand-crafted + CNN) feature vector was proposed to improve the classification accuracy on this dataset.

3. PROPOSED METHOD

In the following subsections, we describe various steps that are involved in our proposed method with a block diagram in Fig. 1.

3.1. Deep Residual Networks

Deep residual networks are made up of residual units. Each residual unit can be expressed as:

$$y_i = h(x_i) + F(x_i, w_i) \tag{1}$$

$$x_{i+1} = f(y_i) \tag{2}$$

where F is a residual function, f is a ReLU function, w_i is the weight matrix, and x_i and y_i are the inputs and outputs of the i-th layer. The function h is an identity mapping [7] given by:

$$h(x_i) = x_i \tag{3}$$

The essential idea behind residual learning is the branching of the paths for gradient propagation. For CNNs, this idea was first introduced in the form of parallel paths in the inception models of [17]. Residual networks share a few similarities with the highway networks [18] such as residual blocks and shortcut connections. However, the output of each path in the highway network is controlled by a gating function, which is learned during the training phase.

The residual units in ResNets are not stacked together as is the case with convolutional layers in a conventional CNN. Instead, shortcut connections are introduced from the input of each convolutional layer to its output. The use of identity mappings as shortcut connections decreases the complexity of the residual networks resulting in deep networks that are faster to train. ResNets can be seen as an ensemble of many paths, instead of viewing it as a very deep architecture. However, all of these network paths in the ResNets are not of the same length. Only one path goes through all of the residual units. Moreover, all of these signal paths do not propagate the gradient which accounts for the faster optimization and training of ResNets.

3.2. ResFeats

This section introduces ResFeats and elaborates on the process to extract those features from deep residual networks. Generally, the image representations extracted from the deeper layers of a CNN capture higher level features and increase the classification performance [5]. A typical residual unit in a ResNet consists of a block of three convolutional layers [7]. Unlike the conventional CNN features which usually are the activations of the fully connected layers [2], ResFeats are the outputs of residual units. This output is a vector of the form $w \times h \times d$ where w and h is the width and height of the resulting feature vector and d is the number of channels in the convolutional layer. Thus ResFeats can be considered as 2-D arrays of local features with d dimensions. The local spatial information of this feature vector will be lost when it is propagated to the fully connected layer. Therefore, we do not use the activations of the FC layer of ResNet as a feature vector.

Fig. 2 shows the architecture of the ResNet-50 deep network which we have used for feature extraction. The learned weights of the deeper layers are usually more class specific *e.g.* the fully connected layer of ResNet-50. We were interested in the classification performance of the output vector of the 5th convolutional layer and we call it Res5c. Res5c features extracted from the 152-layer ResNet tend to perform better than their ResNet-50 counterparts. The classification results of these features are reported in Sec. 4.

3.3. Dimensionality Reduction and Classification

The outputs of the convolutional layers are much larger in size than the traditional 4096-dimensional CNN based features, for example, the Res5c feature vector is $7 \times 7 \times 2048$ in dimension (more than 100k elements). In order to reduce the computational costs associated with the manipulation of large feature vectors, we propose two methods for dimension reduction. The **first** method involves implementing a shallow CNN network with one convolutional layer, one max-pooling layer and two fully-connected (FC) layers. We will refer to this network as sCNN in the rest of the paper. The first convolutional layer consists of small filters (*i.e.* 1×1) along 512 channels. This layer reduces the dimension of Res5c to $7 \times 7 \times 512$ which is of the same size as the output of the last convolutional layer of VGGnet [8]. The stride is set to 1 and the padding is set to zero for the convolutional layer. This layer is then followed by a max-pooling

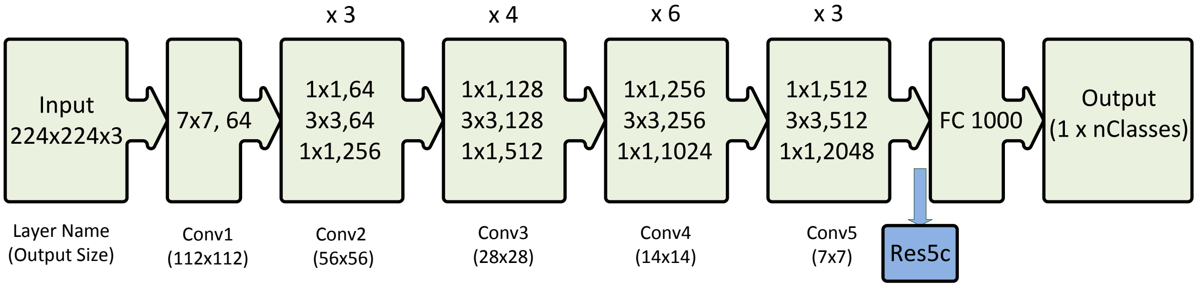

Fig. 2. ResNet-50 architecture [7] shown with the residual units, the size of the filters and the outputs of each convolutional layer.

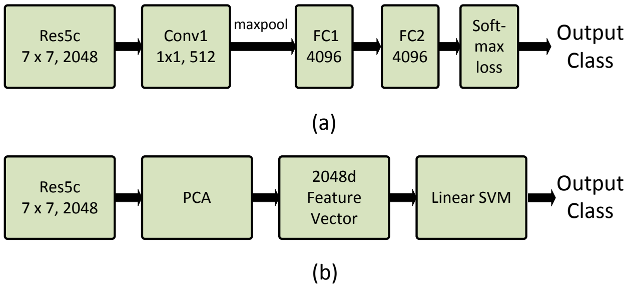

Fig. 3. Dimension reduction and classification pipelines: (a) sCNN with two convolutional layers and two fully connected layers. (b) PCA-SVM.

layer, two FC layers and a soft-max layer for classification. The resulting shallow CNN is very similar to the FC portion of the VGGnet (configuration D [8]). The resulting sCNN is initialized with random weights and is then trained for each dataset specifically. Fig. 3 (a) shows the architecture of sCNN along with the dimensions of the layers used for Res5c.

In the **second** proposed method for dimension reduction, we use the Principal Component Analysis (PCA) algorithm to reduce the Res5c feature vector to an n-dimensional vector. Here n is the number of channels in the convolutional layer from which ResFeats are extracted. For example, Res5c ($7 \times 7 \times 2048$) is reduced to a 2048-dimensional vector by PCA. The resulting feature vectors are then classified using a linear support vector machine (SVM) classifier. We were motivated to use PCA-SVM classification pipeline due to its popularity to classify off-the-shelf CNN features [2, 4, 12]. Fig. 3 (b) shows the pipeline for PCA-SVM module for Res5c.

4. EXPERIMENTS AND RESULTS

4.1. Datasets

Object Classification: Caltech-101 [19] contains 9,144 images, divided into 102 categories. The number of images for each category varies between 31 and 800 images. In our experiments, we used 30 images from each class for training and the remaining images were used for testing. Caltech-101 is a very popular dataset for object classification.

Object Classification: Caltech-256 [20] contains 30,607 images, divided into 257 classes (256 objects +1 background). Each category has at least 80 images. This dataset is less popular but more challenging compared to Caltech-101. In our experiments, following [5], we used 30 and 60 images from each class for training and the rest of the images were used for testing.

Scene Classification: MIT-67 [21] is a very challenging and popular dataset for indoor scene classification. It consists of 15,620 images belonging to 67 classes. The number of images varies between 101 and 738 per class. We followed the standard protocol [21] which uses a subset of 6700 images (100 per class) for training and testing. There are 80 images from each class in the training set. The remaining 20 images per class are set for testing.

Coral Classification: Moorea Labelled Corals (MLC) [16] contains 2055 images collected over three years: 2008, 2009 and 2010. It contains random point annotation *(x, y, label)* for the nine most abundant labels, four non coral and five coral classes. We have used 87,428 patches from the year 2008 for training and the remaining 43,832 patches from the same year for testing. This is a challenging dataset since each class exhibits a large variability in shape, color and scale.

4.2. Performance Analysis: CNN features vs ResFeats

Table 1 compares the performance of ResFeats with their CNN counterparts for a given dataset. The overall classification accuracy is used to evaluate the performance. To keep the comparison fair, standard train-test splits are used for all datasets. For a fair comparison of classification performance, we only consider the methods which have used CNN features without any post-processing. We compare the CNN features with ResFeats extracted from a 50-layer ResNet and a deeper 152-layer ResNet. ResFeats-50 consistently outperform the CNN features by a margin of at least 4%. Table 3 also shows that ResFeats-152 further improves the classification accuracy by 1-2%. We conclude that ResFeats perform significantly better than the corresponding CNN based features. Moreover, ResFeats extracted from a deeper ResNet perform better than the ones extracted from shallower ResNets.

4.3. Image Classification Results

The experiments above compare our ResNet based feature representation with off-the-shelf CNN features. In this section, we compare the performance of ResFeats with other state-of-the-art methods for each dataset.

Caltech-101: We randomly select 30 images per class for training and compare our results with the other existing methods in Table 2. ResFeats with a PCA-SVM classifier beats the current state-of-the-art (He *et al.* [6]) by 1.3%. It is worth mentioning here that the

Dataset	CNN Features	ResFeats-50	ResFeats-152
Caltech 101 (30)	86.5 [5]	91.8	**92.6**
Caltech 256 (30)	70.6 [5]	75.4	**78.0**
Caltech 256 (60)	74.2 [5]	79.3	**81.9**
MIT-67	58.4 [2]	71.1	**73.0**
MLC	72.9 [14]	78.8	**80.0**

Table 1. Performance comparison of the baseline CNN features with the baseline ResFeats. The number in the parenthesis denotes the number of samples per class that is used for training.

Method	Cal-101 (30)
Bo *et al.* [22]	81.4
Zeiler & Fergus [5]	86.5
Chatfield *et al.* [11]	88.4
He *et al.* [6]	93.4
ResFeats-50 + sCNN	91.8
ResFeats-152 + sCNN	92.6
ResFeats-152 + PCA-SVM	**94.7**

Table 2. Performance evaluation on Caltech-101 dataset. The number in the parenthesis denotes the number of samples per class that is used for training.

authors in [6] used the spatial pyramid pooling layer in their network to achieve a 93.4% accuracy. We, however, have achieved state-of-the-art accuracy without adding any post-processing modules to ResFeats. This demonstrates the superior classification power of ResFeats.

Caltech-256: We randomly select 30 and 60 images per class for training and report the classification accuracies in Table 3. Our method (both classification modules) outperforms the current state-of-the-art in both experiments. Table. 3 reports an absolute gain of 8.9% and 4.5% on previous state-of-the-art methods on Caltech-256 datasets with 30 and 60 training samples per class respectively.

MIT-67: We report our results on the standard split (80 train, 20 test) on MIT-67 in Table 4. Table 4 shows that ResFeats perform better than all the previous methods except [12]. The best performing method on MIT-67, Cimpoi *et al.* used deep filter banks that are extracted from VGGnet at multiple scales followed by a Fisher Vector (FV) encoding to achieve state-of-the-art performance on MIT-67. However, it is important to note that applying FV encoding to ResFeats is computationally expensive because of the large size of ResFeats (Res5c has more than 100k elements). Also, this method extracted features from the last convolution layer of VGGnet by using multiple sizes of each training image. In contrast, we only use a fixed size (224 × 224) to extract ResFeats. Using FV on the ResFeats has potential to further improve the performance.

Method	Cal-256 (30)	Cal-256 (60)
Bo *et al.* [22]	48.0	55.2
Zeiler & Fergus [5]	70.6	74.2
Chatfield *et al.* [11]	–	77.6
ResFeats-50 + sCNN	75.4	79.3
ResFeats-152 + sCNN	78.0	81.9
ResFeats-152 + PCA-SVM	**79.5**	**82.1**

Table 3. Performance evaluation on Caltech-256 dataset. The number in the parenthesis denotes the number of samples per class that is used for training.

Method	MIT-67
Razavian *et al.* [2]	58.4
Gong *et al.* [23]	68.9
Azizpour *et al.* [4]	71.3
Hayat *et al.* [24]	74.4
Cimpoi *et al.* [12]	**81.0**
ResFeats-50 + sCNN classifier	71.1
ResFeats-152 + sCNN classifier	73.7
ResFeats-152 + PCA-SVM	75.6

Table 4. Performance evaluation on MIT-67 dataset.

Method	MLC
Beijbom *et al.* [16]	74.0
Khan *et al.* [14]	75.2
Mahmood *et al.* [13]	77.9
ResFeats-50+ sCNN classifier	78.8
ResFeats-152 + sCNN classifier	80.0
ResFeats-152 + PCA-SVM	**80.8**

Table 5. Performance evaluation on MLC dataset.

MLC: We use the same experimental protocol for MLC dataset as given in [16]. Table 5 shows the classification accuracies for MLC dataset achieved by previous methods. Our proposed method achieves an accuracy gain of 6.8% over the baseline performance of [16]. Off-the-shelf ResFeats outperform the cost-sensitive CNN of [14] and multi-scale hybrid feature (CNN + hand-crafted feature) approach of [13].

5. CONCLUSION

In this paper, we used features extracted from deep ResNets off-the-shelf to address three image classification tasks: object, scene and coral classification. We investigated the effectiveness of transfer learning of the ResFeats. We showed that the ResFeats extracted from the deeper layers of a ResNet perform better than the shallower ResFeats. We experimentally confirm that our proposed features are powerful and consistently outperform the CNN off-the-shelf features. Finally, we improve the state-of-the-art accuracy on Caltech-101, Caltech-256 and MLC datasets. It is worth to further investigate the prospective applications of ResFeats for other computer vision tasks such as object localization, image segmentation, instance retrieval and attribute detection.

6. ACKNOWLEDGEMENTS

This research was partially supported by Australian Research Council Grants (DP150104251 and DE120102960). The authors also acknowledge NVIDIA for providing a Titan- X GPU for the experiments involved in this research.

7. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.

[3] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition.,” in *ICML*, 2014, pp. 647–655.

[4] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson, “From generic to specific deep representations for visual recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–45.

[5] Matthew D Zeiler and Rob Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *Computer Vision–ECCV 2014*, pp. 346–361. Springer, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[8] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Identity mappings in deep residual networks,” *arXiv preprint arXiv:1603.05027*, 2016.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 580–587.

[11] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[12] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi, “Deep filter banks for texture recognition and segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3828–3836.

[13] Ammar Mahmood, Mohammed Bennamoun, Senjia An, Ferdous Sohel, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert B. Fisher, “Coral classification with hybrid feature representations,” in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 519–523.

[14] Salman H Khan, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri, “Cost sensitive learning of deep feature representations from imbalanced data,” *arXiv preprint arXiv:1508.03422*, 2015.

[15] Ammar Mahmood, Mohammed Bennamoun, Senjia An, Ferdous Sohel, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert B. Fisher, “Coral classification with hybrid feature representations,” in *OCEANS*. IEEE, 2016.

[16] Oscar Beijbom, Peter J Edmunds, David Kline, B Greg Mitchell, David Kriegman, et al., “Automated annotation of coral reef survey images,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1170–1177.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[18] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber, “Highway networks,” *ICML Workshop*, 2015.

[19] Li Fei-Fei, Rob Fergus, and Pietro Perona, “One-shot learning of object categories,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[20] Gregory Griffin, Alex Holub, and Pietro Perona, “Caltech-256 object category dataset,” 2007.

[21] Ariadna Quattoni and Antonio Torralba, “Recognizing indoor scenes,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 413–420.

[22] Liefeng Bo, Xiaofeng Ren, and Dieter Fox, “Multipath sparse coding using hierarchical matching pursuit,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 660–667.

[23] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *Computer Vision–ECCV 2014*, pp. 392–407. Springer, 2014.

[24] Munawar Hayat, Salman H. Khan, Mohammed Bennamoun, and Senjian An, “A spatial layout and scale invariant feature representation for indoor scene classification,” *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4829–4841, Oct 2016.