ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse128862019.pdf
https://doi.org/10.30534/ijatcse/2019/128862019

HOSVD based Hybrid CNN-ELM model for Scene Categorization

Ramesh Ragala[1], G. Bharadwaja Kumar[1]
[1]School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

ABSTRACT

Scene classification involves grouping the images without semantic overlap which is an arduous task. There has been large amount of work carried out using CNNs in the literature. But, in the recent years Convolutional Neural Networks (CNNs) have been used in combination with other classifiers such as SVM to achieve better accuracies. In this juncture, we have proposed a novel architecture that combines CNN and Extreme Learning Machines for the classification of scene images. Also, we have proposed efficient feature extraction technique since ELM is sensitive to the number of features. Our experimental results show that the proposed method is able to produce state-of-art results on SUN (Scene UNderstanding) - dataset for scene classification.

Key words: Scene classification, Convolutional Neural Networks (CNN), Extreme Learning Machines (ELM), Higher Order Singular Value Decomposition (HOSVD)

1. INTRODUCTION

One of the most highly skilled tasks of the human brain is to recognize the category of the object and understands the environment within few milliseconds. The human visual recognition mechanism can learn and memorize a wide variety of palaces and their patterns [1]. This has several applications in the real world, for example, remote sensing. Scene classification involves grouping the images without semantic overlap which is an arduous task. For such whole-image categorization tasks, bag-of-feature methods, which represent an image as an order-less collection of local features have been used in the earlier works. Later, feature descriptors such as SIFT, GIST, HOG, LBP and SSIM feature representations have been proposed for the scene classification tasks [2]. These traditional feature representation methods are not effective and hinder the performance of the algorithms by compelling more training time and complex models. In addition to that, there may be a huge semantic gap between low-level features and high-level semantic concepts due to the diversity in object category and the distribution complexity in the scene. Henceforth, Bag-of-Visual-Words (BoVW) model and Spatial Pyramid co-occurrence Model have become most popular methods in scene classification in the literature [3]. In the recent years, Convolutional Neural Networks based approaches have become more predominant for the scene classification. Bolei Zhou[4] et al. used convolutional neural network-based architectures for extracting both low-level and high-level features of images and use these feature set to classify the scene images having only places [4].

Convolutional Neural Network (CNN) [5] is a one of variation of feedforward neural network with deep architecture analogous to that of the connectivity pattern of neurons in the human brain which was motivated by the organization of the Visual Cortex. CNN is extensively used in many domains including speech recognition [6], face recognition [7], facial expression recognition [8], traffic signal recognition [9], etc. Even though, CNN has become popular, the fully connected layers of CNN have been replaced by SVM and other classification techniques in the literature to get better performance [10][11].

In this juncture, even though, Extreme Learning Machine (ELM) based classifiers have been used for classification tasks extensively in the literature, they have not been explored in combination with CNN models adequately. In this work, we would like to propose a novel architecture CNN-ELM and an efficient methodology that can be used for effective scene classification tasks.

2. PREVIEW OF CNN, ELM AND TENSORS

2.1 Convolutional Neural Networks

Many of the artificial intelligence domain applications such as visual recognition, speech recognition, etc. are using deep learning technologies to produce efficient and good performance [12]. Convolutional Neural Network is one of the most popular, representative and widely used deep learning architecture [13]. Initially, they have been proposed for large scale multi-classification tasks. LeNet-5 is first and full CNN developed by LeCun et al. [14-15] and used to classify handwritten digits. This architecture accomplished an effective representation of image, which makes it possible to recognize visual patterns directly from raw pixels without pre-processing [14]. In extension to this work, Krizhevsky et al [15], introduced a typical CNN architecture called AlexNet and produced substantial improvements over the past

methods in image classification tasks. Later, many extensions to the basic CNN architecture have been proposed such as ZFNet [16], VGGNet [17], ResNet [18], etc. to improve the performance of AlexNet on ImageNet dataset. The general architecture of CNN architecture is shown in Fig-1.

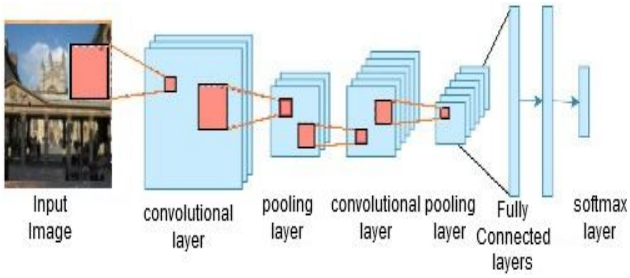

Figure 1: Basic Architecture of CNN

Even though ImageNet, ResNet architectures are able to produce state-of-art- results, the number of parameters that have to be estimated are very large. However, if one wants to run the model on a laptop, maybe without GPU, or even on mobile phone, it is herculean task. In this regard, a few CNN architectures having lesser number of parameters while sacrificing performance a little such as MobileNet, ShuffleNet have been proposed recently.

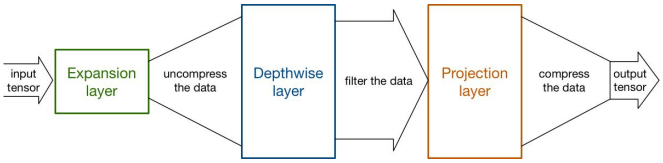

Figure 2: Basic Architecture of MobileNetV2[19]

The MobileNetV2 [19] architecture has been shown in Figure 2. In our proposed approach, MobileNetV2 is used to extract the features from the images. MobileNetV2 is a benchmarked and robust model for feature extraction in object detection and segmentation.

Peng Wang et al [20], proposed an improved CNN, which is used to extract features of Synthetic Aperture Radar (SAR) images and ELM algorithm is used to recognize and classify images. Recently, Lihua Ye et al [21], proposed a hybrid Deep Conventional Neural Network (DCNN) with ELM. This approach is used for Aerial scene classification problem. But simple ELM is very inefficient when the classification involves large number of features.

2.2 Basics of Extreme Learning Machine[25]

A training data set $X = \{a_i, r_i\}_{i=1}^{N}$, where $a_i = [a_{i1}, a2, \cdots a_{im}]^T \in R^m$ is a $m \times 1$ input vector and $r_i = [r_{i1}, r_{i2}, \cdots r_{ip}]^T \in R^p$ is a $p \times 1$ output vector. Then the output of the SLFN with K-hidden layer nodes can be expressed as

$$r_i = \sum_{l=1}^{K} \beta_l . \hbar(w_l . a_i + b_l), i = 1, 2, \cdots N \quad (1)$$

Where $\hbar(.)$ is an activation function, $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots \beta_{ip}]^T$ is the weight vector connecting the i[th] hidden layer and the output layer, $w_i = [w_{i1}, w_{i2}, \cdots w_{im}]^T$ is the weight vector connecting the i[th] hidden layer and the input layer and bi is the bias of the i[th] hidden layer. Then the equation (1) can be rewritten as

$$P\beta = R \quad (2)$$

Where $\beta = [\beta_1, \beta_2, \cdots \beta_L]^T \in R^{L \times p}$, $R = [r_1, r_2, \cdots r_N]^T \in R^{N \times p}$ and P is the output matrix of hidden layer.

Thus, P can be expressed as

$$P(w_1, \cdots w_L, b_1, \cdots; b_L, a_1, \cdots a_L) = \begin{bmatrix} \hbar(w_1 a_1) + b_1 & \cdots & \hbar(w_L a_1) + b_L \\ \vdots & \ddots & \vdots \\ \hbar(w_1 a_N) + b_1 & \cdots & \hbar(w_L a_N) + b_L \end{bmatrix} \quad (3)$$

Here we can compute the output weights using following expression

$$\beta = P^{\dagger} R \quad (4)$$

Here $P^{\dagger}$ (dimension as K × N) is the Moore-Penrose inverse of P (dimension as N × K).

The procedure of ELM can be specified in two stages. In the first stage the algorithm takes the random input weights and bias followed by hidden layer output matrix using Moore-Penrose pseudo inverse method. In the second stage, the algorithm calculates the most suitable output weights between hidden layer and output layer with using the results which are got from first stage. The complexity of ELM approach is directly depending on Moore-Penrose Pseudo inverse computation.

The Moore-Penrose Inverse [26] of P is a unique matrix satisfying the following equations for the real valued numbers:

$$PP^{\dagger}P = P \quad (5)$$

$$P^{\dagger}PP^{\dagger} = P^{\dagger} \quad (6)$$

$$(PP^{\dagger})^T = PP^{\dagger} \quad (7)$$

$$(P^{\dagger}P)^T = P^{\dagger}P \quad (8)$$

Matrix Decomposition, Orthogonal projection approaches, etc are used to compute the Pseudo inverse. Newton's method is the most popularly used in iterative method for approximating pseudo inverse [26].

Singular Value Decomposition (SVD) is the most commonly used method to compute Moore-Penrose inverse in ELM. The

Pseudo inverse computation using SVD consumes more resources such as computational time and memory [28]. Direct Methods and Iterative methods have been proposed in the literature to compute Moore-Penrose inverse in applications. Mostly, the calculation of pseudo inverse matrix using classical direct methods is not effective for large datasets [29]. So, in this proposed methodology, we are using Iterative approach i.e. Second-Order Iterative method [27] to compute the Moore-Penrose pseudo inverse. Iterative approach to compute pseudo inverse uses computational resources effectively [29].

2.3 Terminology of Tensors

As we are working on set of images for scene classification problem, the data which is generated from the MobileNetV2 Model have a greater number of features. So, this problem leads to curse of dimensionality which is a non-trivial task to train extreme learning machines. It also poses limitations like computational time and the memory. To avoid this problem, we propose decomposition of higher-order tensors to alleviate the curse of dimensionality. Hence in this proposed approach, Higher Order Single Value Decomposition (HOSVD) is used to reduce the features set dimension.

Tensors provide a natural and concise mathematical framework for formulating and solving problems if one has multi-dimensional matrices [30]. Thus, it is a generalized representation of multi-dimensional matrices. Simply, a data cube or three-dimensional data can be thought simplest representation of tensor. Even though, the tensor is originated in the psychometrics community in the 20th century, its utilization is wider in many applications including machine learning, signal processing, etc. due to increased computation capacity and a better understanding of multilinear algebra. Tensors and their decompositions are most useful in unsupervised learning [30].

The basic terminologies used in tensor are:

A. Order of Tensors: The number of dimensions of the corresponding tensor can be treated as the order of the tensor. Scalar can be a zeroth-order tensor, vector can be a first-order tensor and matrix can be the second-order tensor and the dimension number greater than three can be treated as an actual tensor. Usually tensors can be represented with a notation of bold Euler script letters in upper case like A $\in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, where *I*s is the number of elements in the corresponding dimension.

B. Tensors Indexing: Tensor indexing can be used to get the variety of sub-tensors like fiber, slices, etc.

C. Product operation in Tensors: The tensor product of two vectors represents a dyad (it is a special tensor), which is a linear vector transformation.

The Tucker decomposition (HOSVD) simply represents the data from tensor to higher-order PCA form [31]. It decomposes a tensor into a core tensor and a set of matrices [31]. The fig-3 shows the three-way array tucker decomposition. It is used in data compression and is very useful for the purposes of data distribution and data storage, when the data size far exceeds the terabyte range. The same concept has been used to reduce the number of features to a greater extent. Generally, this algorithm is implemented using alternating least squares. Another alternate approach could be a block coordinate descent type search method known as Maximum Block Improvement (MBI).

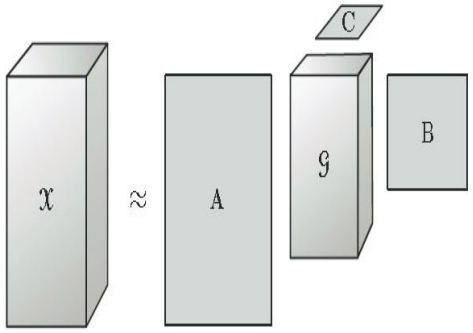

Figure 3: HOSVD Decomposition

The tensor $\Xi \in \mathbb{R}^{I \times J \times K}$, then

$$X \approx \varsigma \times_1 A \times_2 B \times_3 C$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_p \circ b_q \circ c_r$$

$$= [\varsigma; A, B, C]$$

Where $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$ and $C \in \mathbb{R}^{K \times R}$ are orthogonal matrices and $\varsigma \in \mathbb{R}^{P \times Q \times R}$ is a core tensor. Here P, Q, R are number of columns in factor matrices A, B and C respectively. The HOSVD algorithm specified in the fig-4

procedure HOOI($\mathcal{X}$, $R_1, \ldots, R_N$)
  initialize $A^{(n)} \in R^{I_n \times R}$ for $n = 1, \ldots, N$ using HOSVD
  repeat
    for n = 1,…,N **do**
      $\mathcal{Y} \leftarrow \mathcal{X} \times_1 A^{(1)T} \times_2 \cdots \times_{n-1} A^{(n-1)T} \times_{n+1} A^{(n+1)T} \times_{n+2}$
      $\hookrightarrow \cdots \times_N A^{(N)T}$
      $A^{(n)} \leftarrow R_n$ leading left singular vectors of $Y_{(n)}$
    end for
  until stopping criterion satisfied
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 A^{(1)T} \times_2 \cdots \times_N A^{(N)T}$
  return $\mathcal{G}, A^{(1)}, \ldots, A^{(N)}$
end procedure

Figure 4: HOSVD algorithm

3. PROPOSED APPROACH

The overall architecture of the proposed approach is specified in Figure-4.

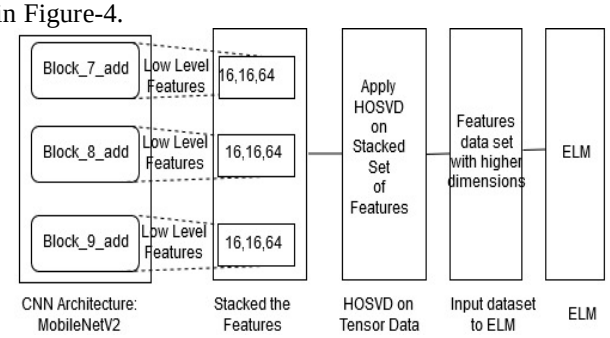

Figure 4: Proposed Methodology Architecture

MobileNetV2 model is used to extract the features from the images. In the present work, three layers are used to extract the features of the images from MobileNetV2. The features are extracted from block7, block8 and block9. The model summary of the MobileNetV2 with few activation functions is shown in Table-1. The full Model summary can be found in [19] .

Table 1: MobileNetV2 Model Summary

Layer Type	Output Shape	No. Parameters	Connected to
Input_1(Input layer)	256,256, 3	0	
Conv1_pad (ZeroPadding2D)	257,257, 3	0	input_1[0][0]
Conv1 (Conv2D)	128,128, 32	864	Conv1_pad[0][0]
Conv1_relu (ReLU)	128,128, 32	0	bn_Conv1[0][0]
block_2_add (Add)	64,6424	0	block_1_project_BN[0][0] block_2_project_BN[0][0]
block_4_add (Add)	32,32,32	0	block_3_project_BN[0][0] block_4_project_BN[0][0]
block_5_add (Add)	32,32,32		block_4_add[0][0] block_5_project_BN[0][0]
block_7_add (Add)	16,16,64	0	block_6_project_BN[0][0] block_7_project_BN[0][0]
block_8_add (Add)	16,16,64	0	block_7_add[0][0] block_8_project_BN[0][0]
block_9_add (Add)	16,16,64	0	block_8_add[0][0] block_9_project_BN[0][0]

All these features are stacked and then applied HOSVD decomposition to get the features set. Now this two-dimensional dataset is used as an input to the ELM for classification task. In the present work, we have not considered the conventional ELM which is not efficient for higher dimensional datasets. Generic ELM uses batch processing approach to process the data. The large amount of training data may not fit into main-memory before processing. Also, the computation of Moore-Penrose generalized inverse using conventional methods like SVD, matrix decomposition etc. for large matrices is difficult and returns out of memory error. There is also memory limitation when dealing with online and sequential incremental algorithms. Hence, we proposed Second Order Iterative method to compute the Moore-Penrose generalized inverse of hidden layer output matrix.

Let us assume the pseudo inverse of the P matrix is D. The procedural steps of Second Order Iterative method is as follows:

Step – 1: Initial matrix, $X_0 = \alpha P^T$ where $\alpha = 0<\alpha<2/\lambda_1(PP^T)$ here λ_1 is the highest Eigen value in range of A^T .

Step – 2: $D_{K+1} = MD_K+Q$ where $D_1 = X_0$ and $M = I - P^TP$ and $Q=P^T$.

Step – 3: The step-2 repeats till the P converges with residual error with 10^{-7}

This approach can be efficiently implemented in the conventional computers without needing GPU based architectures. The present work has been carried out on Lenovo ThinkPad with I5 processor and 8GB RAM.

4. RESULTS

In this research work, the subset of SUN [2] database is used for the evaluation purpose. The SUN database has a vast collection of images which covers many of the environmental scenes, places, etc. This dataset consists of 899 categories and 130,519 images. For our experimentation, we had taken 16 categories and 18293 images for scene classification task.

The following table-2 shows the description of the dataset

Table 2: Description of dataset used for Evaluation

S.No	Category	No. of Images
1	Airport Terminal	1091
2	Beach	1193
3	Bedroom	2073
4	Castle	1100
5	Church Outdoor	905
6	Conference Room	832
7	Dining Room	1181
8	Golf Course	813
9	House	954
10	Kitchen	1746
11	Living Room	2356
12	Market Outdoor	838
13	Playground	896
14	Restaurant	793
15	Shop Front	722
16	Skyscraper	800

The dimensions of the dataset, which is produced by HOSVD is 18293 ×12289. This feature descriptor dataset is given to ELM. The hidden layer output matrix dimension is 18293×K, where K is the number of hidden nodes in ELM. K is a hyper parameter in this case. Here, we have used Second Order Iterative method to solve the Moore-Penrose inverse for this high dimensional matrix, which consumes less memory and produce the results in faster than exiting conventional approaches. The following table shows the average accuracy of 10-fold cross-validation results.

Table 3: Summary of Results

	Training Time (Sec)	Testing Time (Sec)	Training Accuracy (%)	Testing Accuracy (%)
Conventional ELM	1238.67	6.32	91.12	89.23
Proposed ELM	901.27	5.93	91.12	89.23

5. RESULT ANALYSIS AND CONCLUSION

In the present work, we have proposed a novel architecture to efficiently combine CNN and ELM for scene classification task. We have used a few layers of MobileNetV2 for feature extraction and then applied tensor decomposition (HOSVD) to reduce the number of features and also effective feature representation. Then, we used a second-order iterative approach-based ELM method for the classification task which takes lesser computation time and resources without compromising on accuracy. From the result analysis, it can be observed that the proposed method is able to produce the same classification accuracy in less amount of time.

REFERENCES

1. T. Konkle, T. F. Brady, G. A. Alvarez, and A. Oliva. **Scene memory is more detailed than you think: the role of categories in visual long-term memory**. Psych Science, 2010. https://doi.org/10.1177/0956797610385359
2. J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. **Sun database: Large-scale scene recognition from abbey to zoo**. InProc. CVPR, 2010.
3. Liu, Y., Zhong, Y., Fei, F., Zhu, Q. and Qin, Q. **Scene classification based on a deep random-scale stretched convolutional neural network.** Remote Sensing, 10(3), p.444, 2018. https://doi.org/10.3390/rs10030444
4. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). **Learning deep features for scene recognition using places database.** In Advances in neural information processing systems (pp. 487-495).
5. Agarap AF, **An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification**, arXiv:1712.03541, Dec 10 2017.
6. Niu, X.X. and Suen, C.Y., **A novel hybrid CNN–SVM classifier for recognizing handwritten digits**. Pattern Recognition, 45(4), pp.1318-1325, 2012.
7. Y. Bengio**, Learning deep architectures for AI**, Found. Trends Mach. Learn., 1 (2009), 1–71. https://doi.org/10.1561/2200000006
8. S. Singhal, V. Passricha, P. Sharma, and R. K. Aggarwal, **Multi-level region-of-interest CNNs for end to end speech recognition**, Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 11, pp. 4615–4624, 2019
9. T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, **Adaptive very deep convolutional residual network for noise robust speech recognition,** IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 8, pp. 1393– 1405, 2018.
10. S. Kang, J. Lee, K. Bong, C. Kim, Y. Kim, and H.-J. Yoo, **Lowpower scalable 3-D face frontalization processor for CNNbased face recognition in mobile devices**, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 4, pp. 873–883, 2018.
11. C. Ding and D. Tao, **Trunk-branch ensemble convolutional neural networks for video-based face recognition**, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 1002–1014, 2018 https://doi.org/10.1109/TPAMI.2017.2700390
12. S. Xie and H. Hu, **Facial expression recognition with FRRCNN,** Electronics Letters, vol. 53, no. 4, pp. 235–237, 2017
13. J. Li, D. Zhang, J. Zhang et al., **Facial expression recognition with faster R-CNN**, Procedia Computer Science, vol. 107, no. C, pp. 135–140, 2017.
14. X. Mao, S. Hijazi, R. Casas, P. Kaul, and C. Rowen, **Hierarchical CNN for traffic sign recognition**, in 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, June 2016
15. F. Schroff, D. Kalenichenko and J. Philbin, **FaceNet: A Unified Embedding for Face Recognition and Clustering**, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), United States of America, 2015, 815–823. https://doi.org/10.1109/CVPR.2015.7298682
16. G. Huang, Z. Liu, L. van der Maaten, et al., **Densely Connected Convolutional Networks**, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), United States of America, 2017, 4700–4708.
17. Z. Q. Zhao, P. Zheng, S. T. Xu, et al**., Object Detection with Deep Learning: A Review**, IEEE Trans. Neural Networks Learn. Syst., (2019), 1–21.
18. Gu, Jiuxiang, et al. **Recent advances in convolutional neural networks**. Pattern Recognition 77 (2018): 354-377.

19. LeCun, Yann, et al. **Gradient-based learning applied to document recognition.** Proceedings of the IEEE 86.11 (1998): 2278-2324.
20. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov**, Improving neural networks by preventing co-adaptation of feature detectors**, CoRR abs/1207.0580.
21. M. D. Zeiler, R. Fergus, **Visualizing and understanding convolutional networks**, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 818–833 https://doi.org/10.1007/978-3-319-10590-1_53
22. K. Simonyan, A. Zisserman**, Very deep convolutional networks for large-scale image recognition**, Proceedings of the International Conference on Learning Representations (ICLR), 2015.
23. K. He, X. Zhang, S. Ren, J. Sun, **Deep residual learning for image recognition**, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
24. Rabanser, Stephan, Oleksandr Shchur, and Stephan Günnemann. **Introduction to tensor decompositions and their applications in machine learning** arXiv preprint arXiv:1711.10781 (2017).
25. Nair, Nikhitha K., and S. Asharaf. **Tensor decomposition based approach for training extreme learning machines**m, Big Data Research 10 (2017): 8-20.
26. Gower, Robert M., and Peter Richtárik. **Linearly convergent randomized iterative methods for computing the pseudoinverse**, arXiv preprint arXiv:1612.06255 (2016).
27. Sandler, Mark, et al. **Mobilenetv2: Inverted residuals and linear bottlenecks,** Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. https://doi.org/10.1109/CVPR.2018.00474
28. Wang, Peng, Xiaomin Zhang, and Yan Hao. **A Method Combining CNN and ELM for Feature Extraction and Classification of SAR Image,** Journal of Sensors 2019.
29. Ye, L., Wang, L., Sun, Y., Zhu, R., & Wei, Y. (2019). **Aerial scene classification via an ensemble extreme learning machine classifier based on discriminative hybrid convolutional neural networks features**. International Journal of Remote Sensing, 40(7), 2759-2783.
30. McDonnell, M. D., Tissera, M. D., van Schaik, A., & Tapson, J. (2014). **Fast, simple and accurate handwritten digit classification using extreme learning machines with shaped input-weights**. arXiv preprint arXiv:1412.8307.
31. Kolda, T. G., & Bader, B. W. (2009). **Tensor decompositions and applications**. SIAM review, 51(3), 455-500. https://doi.org/10.1137/07070111X