

## Research Article

# Label Rectification Learning through Kernel Extreme Learning Machine

Qiang Cai,<sup>1,2,3</sup> Fenghai Li<sup>1,2,3</sup>,<sup>ORCID</sup> Yifan Chen,<sup>4</sup> Haisheng Li,<sup>1,2,3</sup> Jian Cao,<sup>1,2,3</sup>  
and Shanshan Li<sup>1,2,3</sup>

<sup>1</sup>School of Computer, Beijing Technology and Business University, 100048, China

<sup>2</sup>National Engineering Laboratory for Agri-product Quality Traceability, 100048 Beijing, China

<sup>3</sup>Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

<sup>4</sup>Canvard, Beijing Technology and Business University, Beijing 100048, China

Correspondence should be addressed to Fenghai Li; [lifenghai@st.btbu.edu.cn](mailto:lifenghai@st.btbu.edu.cn)

Received 2 December 2020; Revised 30 January 2021; Accepted 20 February 2021; Published 13 March 2021

Academic Editor: Zhili Zhou

Copyright © 2021 Qiang Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Along with the strong representation of the convolutional neural network (CNN), image classification tasks have achieved considerable progress. However, majority of works focus on designing complicated and redundant architectures for extracting informative features to improve classification performance. In this study, we concentrate on rectifying the incomplete outputs of CNN. To be concrete, we propose an innovative image classification method based on Label Rectification Learning (LRL) through kernel extreme learning machine (KELM). It mainly consists of two steps: (1) preclassification, extracting incomplete labels through a pretrained CNN, and (2) label rectification, rectifying the generated incomplete labels by the KELM to obtain the rectified labels. Experiments conducted on publicly available datasets demonstrate the effectiveness of our method. Notably, our method is extensible which can be easily integrated with off-the-shelf networks for improving performance.

## 1. Introduction

Image classification is a fundamental and challenging task in computer vision, which has received ever-increasing interests. In general, image classification is aimed at distinguishing the image categories according to their semantic information. It is widely applied in real-world applications, including face recognition [1], traffic sign detection [2], and other applications [3–7]. Traditional image classification methods often adopt hand-crafted features (e.g., SIFT [8] and HOG [9]) combined with classical classifiers (e.g., SVM [10]). However, they often have poor performance due to the limited representation ability of features.

Convolutional neural networks (CNNs) have witnessed great development on image classification tasks. In [11–13], they design small convolutional kernels to increase the layers of CNNs. The authors in [14, 15] build relations of different convolutional layers and alleviate vanishing gradient problem in deep networks by shortcut connections. SKNet [8]

allows each neuron to adaptively adjust its receptive field size based on multiple scales of input information by a dynamic selection mechanism. Res2Net [16] represents multiscale features at a granular level and increases the range of receptive fields for each network layer. All of the above-mentioned methods try to improve CNN's design to boost classification performance, but complex architectures and huge parameters often lead to redundant complicated loads and poorly trained models. These frameworks consist of two major components, i.e., a feature extractor and a softmax classifier. Feature extractor can get powerful feature representation, and these features can be normalized by softmax function to obtain predicted probabilities. However, the softmax function encourages the output scores dramatically distinctive which potentially leads to overfitting. Some studies consider improving the discriminative ability of features by replacing softmax with other machine learning algorithms [17–20], e.g., Support Vector Machine (SVM) or Extreme Learning Machine (ELM). Although considerable progress has been

achieved in image classification, existing traditional methods are still faced with some limitations: (1) extracting high-dimensional features in matrix manipulation is time-consuming and (2) most works rarely exploit the label-wise relation for image classification.

To solve these problems, we propose a novel method to learn to rectify label (LRL) by kernel extreme learning machine (KELM), which attempts to calibrate label output of CNN and obtain more accurate distribution. The KELM is an improvement of ELMs by adopting a kernel function for feature mapping. Because the kernel function can map labels to the infinite dimensional space, the separability of the label distribution can be further enhanced. Figure 1 schematically illustrates our framework. It consists of two steps, i.e., preclassification and label rectification. In preclassification, we put images into a trained CNN and generate corresponding classification results as incomplete labels. These labels may greatly deviate from their ground truth. Hence, we try to exploit label-wise relation by label rectification. The final classification result can be obtained through random kernel mapping and linear combination. It is worth noting that KELM is more suitable for multiclass classification against KSVM [21].

In summary, our main contributions are threefold:

- (i) To the best of our knowledge, it is the first time that a label rectification method is proposed for image classification, which can exploit label-wise relations for more correct distribution
- (ii) We present a novel image classification framework (LRL) that combines CNN with KELM, and it has well generalizability for different CNN architecture
- (iii) Our experiments on publicly available datasets also demonstrate the efficiency and effectiveness of our method

The rest of our paper is organized as follows. In Section 2, we introduce the preclassification. In Section 3, we introduce the way label rectification briefly. In Section 4, we show the experimental results. In Section 5, we draw a conclusion of this paper.

## 2. Preclassification

Deep convolutional neural networks (CNNs) have recently achieved unprecedented success in image classification. And it is generally believed that a CNN consists of two components: a feature extractor and a softmax classifier. The feature extractor can carry out depth construction to obtain strong representative capability for input images. Most existing efforts try to improve the architecture of feature extractor (increasing depth [22], multiscale kernel size [23], attention mechanism [24], etc.) to learn more complicated mappings. Some methods notice the limitation of the softmax classifier in nonlinear conditions. The softmax function encourages the output scores dramatically distinctive. And it makes model to assign full probability to the ground truth label for each training sample, which potentially leads to overfit-

ting. Therefore, they use other machine learning models (SVM [17], ELM [20], etc.) to replace it. However, all these methods ignore exploiting predicted label information which is generally regarded as final classification results of CNNs. In addition, according to existing observations, CNNs are sensitive to hyperparameters, so it is difficult to train the network well and it lacks adaptability in real-world applications.

The illustrations mentioned above motivate us to capture label information output by CNN to boost classification performance. This paper denotes the labels as incomplete labels because we conjecture these labels still have potential for improvement. In our method, we therefore extract labels instead of features from CNNs. It can be formulated as

$$L = F_{\theta}(I), \quad (1)$$

where  $F$  is a well-trained CNN,  $I$  is the input image,  $L$  is extracted labels, and  $\theta$  is the parameters of CNN. Compared with features in fully connected layers [25], extracted labels contain predicted scores of each image category, and they are relatively low dimensions (it depends on the number of all categories), so it is more efficient in subsequent operations. In addition, for a pretrained CNN, the proposed method can drastically improve their performance according to our experiments. It is worth noting that we do not use the softmax function to normalize the extracted labels in our method.

## 3. Label Rectification

Given incomplete label  $L$ , we aim to revise them by a model  $f$ . For efficiently training  $f$ , we utilize the ELM to rectify incomplete labels. The ELM was proposed by Huang et al. [21], it is a single-hidden layer feedforward neural network (SLFN) which randomly chooses hidden nodes and analytically determines the output weight of SLFNs. ELM tends to provide good generalization performance at extremely fast learning speed and the hidden layer need not be tuned. ELM consists of three layers: input layer, hidden layer, and output layer. The structure of it is shown in Figure 2.

Consider the incomplete label  $L$  is passed through an ELM network, the hidden layer can map it to large dimensionality that increases the universal infinite approximation ability of the ELM. The output function of ELM for generalized SLFNs can be described as

$$f_l(L) = \sum_{i=1}^l \beta_i h_i(L) = h(L)\beta, \quad (2)$$

where  $\beta = [\beta_1, \dots, \beta_l]^T$  is the output weight vector of hidden layer  $h(L) = [h_1(L), \dots, h_l(L)]$  and  $h_i(L)$  is the output of the  $i$ th hidden node output;  $h(L)$  maps the  $d$ -dimensional label  $L$  to the  $l$ -dimensional hidden layer feature. And the output functions of hidden nodes may not be unique. Different output functions may be used in different hidden neurons. The vector  $h_i(L)$  can be defined as

$$h_i(L) = G(a_i, b_i, L), \quad a_i \in R^d, b_i \in R, \quad (3)$$

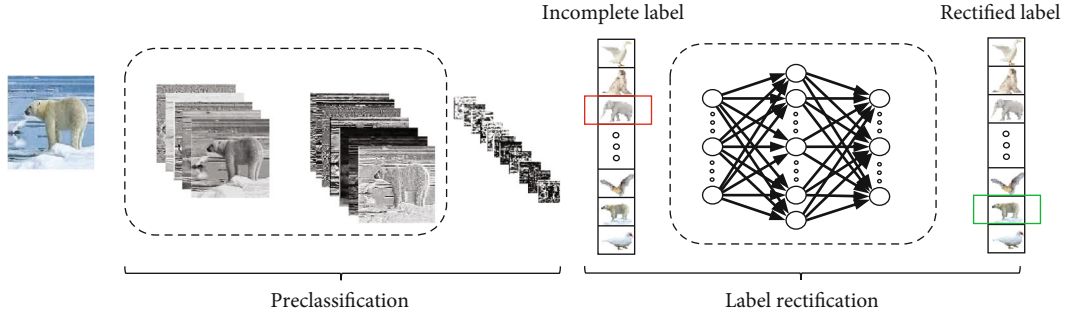


FIGURE 1: Framework of the proposed approach.

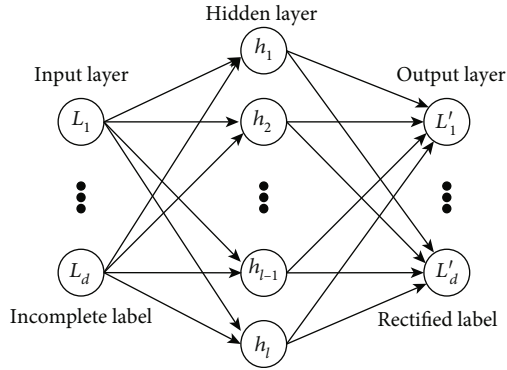


FIGURE 2: Architecture of the ELM.

where  $G(a, b, L)$  is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems and  $(a_i, b_i)_{i=1}^L$  are randomly generated according to any continuous probability distribution. The sigmoid function (4) is conventional kernel function for ELM.

$$G(a, b, L) = \frac{1}{1 + \exp(-(\alpha L + b))}. \quad (4)$$

The KELM is a species of ELMs that adopts the Gaussian kernel function (5) as ELM feature mapping. Because the kernel function can map labels to the infinite dimensional space, the separability of the label distribution can be further enhanced. This encourages a finite output that can generalize better.

$$G(a, b, L) = \exp(-b\|L - a\|^2). \quad (5)$$

KELM is to minimize the training error as well as the norm of the output weights. The KELM learning function uses the minimal norm least square method; it can be formulated as

$$\arg \min_{\beta} \|H\beta - Y\|, \quad (6)$$

where  $H$  is the hidden layer output matrix and  $Y$  is the target matrix of the training data. The minimal norm least square method was used in ELM, and it can provide the solution

of  $\beta$ :

$$\hat{\beta} = H^\dagger Y, \quad (7)$$

where  $H^\dagger$  is the Moore-Penrose generalized inverse of matrix  $H$ . The orthogonal projection method can be used to calculate  $H^\dagger$ : when  $HH^T$  is nonsingular,  $H^\dagger$  can be defined as

$$H^\dagger = (H^T H)^{-1} H^T, \quad (8)$$

or when  $HH^T$  is nonsingular,  $H^\dagger$  can be defined as

$$H^\dagger = H^T (HH^T)^{-1}. \quad (9)$$

As far as we get the value of  $\beta$ , we can pass through the incomplete label  $L$  to well-trained ELM or KELM, and the rectified label feature matrix  $L'$  can be defined as

$$L' = h(L_{\text{test}})\beta. \quad (10)$$

## 4. Experiments

**4.1. Label Rectified Evaluation.** To evaluate the effectiveness of the proposed method, we conduct experiments on CIFAR-10 and CIFAR-100 [26]. Those two datasets are widely used in image classification benchmark and consist of natural scene images. The size of all images is  $32 \times 32$  pixels. The training and test sets contain 50k and 10k images, respectively. For adequately evaluating our method, we utilize two classical CNNs, i.e., VGG19 [22] and ResNet50 [27]. We set two baselines [26, 28] for comparisons. Table 1 shows our experiment results.

LRL can promote CNN classification performance. On CIFAR-10, ResNet50 achieves 90.79% accuracy, while ResNet50-LRL (KELM) achieves 92.28% accuracy. On CIFAR-100, VGG19-LRL (KELM) outperforms VGG19 by 1.39% accuracy. From these comparisons, we can conclude the proposed method can effectively rectify incomplete label output by CNNs. Compared with LRL (ELM), LRL (KELM) has more potential for image classification tasks, and it demonstrates the RBF kernel is more appropriate to exploit label relations.

TABLE 1: Accuracy (%) on CIFAR-10 and CIFAR-100 datasets.

Model	CIFAR-10	CIFAR-100
Gao et al. [16]	88.34	62.20
ResNet [22]	89.44	—
FractalNet [20, 29]	89.92	64.66
Network in network [24]	89.59	64.32
VGG19	90.49	63.83
VGG19-LRL (ELM)	90.64	65.01
VGG19-LRL (KELM)	90.86	65.22
ResNet50	90.79	68.52
ResNet50-LRL (ELM)	92.11	71.44
ResNet50-LRL (KELM)	<b>92.28</b>	<b>72.27</b>

TABLE 2: Accuracy (%) on Caltech-256 dataset compared with other state-of-the-art methods.

Model	Top-1	Top-5
VGG19	70.54	87.81
ResNet18	73.22	86.96
ResNet18 with stochastic depth [32]	70.00	89.00
ResNet50	80.40	92.95
+Li et al.	<b>82.01</b>	93.07
+ELM	78.83	91.40
+KELM	81.96	<b>93.27</b>

**4.2. Comparison with State-of-the-Art Methods.** In this section, we compare our method with state-of-the-art methods. We run experiments on the Caltech-256 dataset [14]. It contains more classes and fewer samples than CIFAR, with 256 classes and a total of 30607 images. Each category has a minimum of 80 images. Following [20], we utilize 60 randomly selected images per class as the training dataset, the rest as the testing dataset. We resize all images to  $256 \times 256$  pixels. VGG19 [22], ResNet18 [27], Zhu et al. [20], Squeeze-Net [11], VGG19-BN [30], and Inception-V3 [31] are introduced for comparison, and we finally adopt ResNet50 for preclassification. Table 2 shows our experimental results.

Notably, ResNet50-LRL (KELM) also works well on Caltech-256 and outperforms all other state-of-the-art approaches. ResNet50 achieves up to 80.40% Top-1 and 92.95% Top-5 accuracy. Compared with Zhu et al. [20], our method outperforms it by 11.96% Top-1 and 2.4% Top-5 accuracy, and ResNet50-LRL (KELM) outperforms base model by 1.46% Top-1 and 0.32% Top-5 accuracy. Those experimental results demonstrate the superiority of our approach on image classification.

**4.3. Classifier Comparisons.** In our method, we adopt the KELM to rectify incomplete labels. In order to evaluate it, we compare it with other different classifiers, linear SVM, kernel SVM (KSVM), neural network (NN), and sigmoid function ELM (ELM). Besides, we introduce a method of Li et al. [20], which utilizes the KELM to replace the softmax

TABLE 3: Accuracy (%) and training times on the Caltech-256 with different methods.

Classifier	Promotion (%)	Time (s)
Li et al. [18]	+1.12	18.02
Linear SVM	-5.06	16.26
KSVM	-3.03	47.07
NN	-5.70	120.31
ELM	<b>-2.06</b>	<b>2.81</b>
KELM	<b>+1.07</b>	<b>4.58</b>

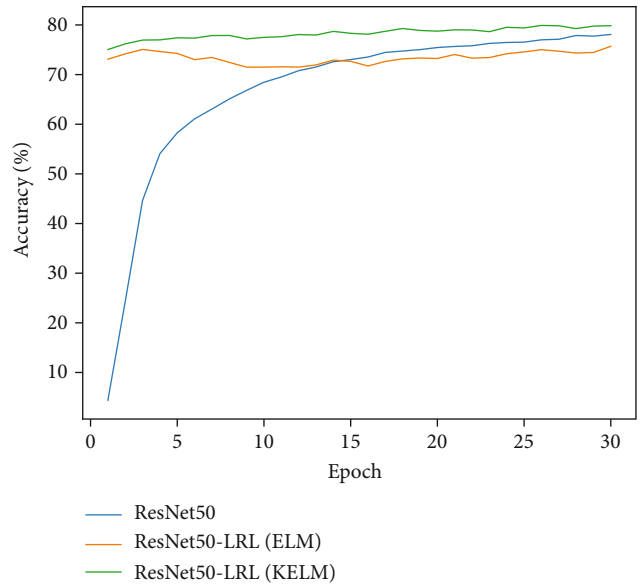


FIGURE 3: Training observation of ResNet50 on Caltech-256 dataset.

classifier that may cause two problems. First, features may contain more redundant information than labels. Second, the dimension of features is larger than labels, which consumes more time for training. In our method, we extract the labels whose dimension is equal to the image classes. We train the ResNet50 as preclassification. Table 3 exhibits our experimental results.

Apparently, Li et al. and LRL (KELM) are efficient in improving the performance of image classification. But other classifiers have a negative impact on improving accuracy. Li et al. can improve the accuracy of the base model by about 1.12%, and LRL (KELM) can improve about 1.07%. But, in contrast, experiments show that the LRL (KELM) has the advantage of high training speed. In conclusion, the LRL (KELM) achieves a better balance between accuracy and training time.

**4.4. Training Observation.** In our training phase, we apply a pretrained network (on ImageNet) into specific datasets (e.g., Caltech-256), so we can obtain a well-trained CNN for preclassification. But badly trained CNNs should be considered. Therefore, we apply the LRL method into each ResNet50's training epoch. We plot the Top-1 accuracy of



ResNet50, ResNet50-LRL (ELM), and ResNet50-LRL (KELM). The visual comparisons are provided in Figure 3.

Obviously, KELM-based LRL (green line) can substantially promote ResNet50 (blue line) performance in very early epochs. It manifests our method is effective for unwell-trained CNNs. Although the disparity of them becomes small, KELM-based LRL still outperforms ResNet50. Besides, we notice that ELM-based LRL (orange line) cannot improve ResNet50 performance after the 13th epoch.

**4.5. Implementation Details.** For the Caltech-256 dataset, we modify the final output of VGG19 and ResNet50 to 257. These CNNs are trained using a batch size of 32 for 50 epochs. For the CIFAR-10 and CIFAR-100 datasets, we replaced the fully connected layer of ResNet50 to 11 and 101 outputs. ResNet50 are trained 300 epochs using a batch size of 128, and the initial learning rate is set to 0.0001 and is divided by 10 at 30% and 75% of training epochs. All CNNs are trained by stochastic gradient descent. We use a weight decay of 0.01 and Nesterov momentum of 0.9. In all experiments, the images are randomly flipped and cropped before passing into the networks. In all our simulations on ELM with sigmoid additive hidden node and RBF hidden node,  $l = 1000$ . All the hidden node parameters are randomly generated based on uniform distribution. Experiments are conducted on an NVIDIA Titan Xp GPU.

## 5. Conclusion

In this paper, instead of designing complicated and redundant CNNs, we explore the label-wise relation for label rectification and then propose a method (LRL) to learn label rectification through kernel extreme learning machine to improve the accuracy of image classification. Compared with intermediate features, label-wise features contain less redundant information and have a lower dimension. LRL can achieve similar accuracy while saving more time during training and inference. In addition, by training observation, we find that LRL has a strong advantage in rectifying the labels of pretrained models. Extensive experiments conducted on public datasets demonstrate the effectiveness of LRL (KELM). As far as we know, this is the first label rectification approach for image classification. In the future, we will develop a method to learn label rectification from source dataset to target dataset directly.

## Data Availability

CIFAR10 and CIFAR100: <http://www.cs.toronto.edu/~kriz/cifar.html>; Caltech256: [http://vision.caltech.edu/Image\\_Datasets/Caltech256/](http://vision.caltech.edu/Image_Datasets/Caltech256/).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Qiang Cai and Fenghai Li contributed equally to this work.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFB0803700), National Natural Science Foundation of China (No. 61877002), Beijing Municipal Commission of Education PXM2019 014213 000007, Beijing Natural Science Foundation, and Fengtai Rail Transit Frontier Research Joint Fund 19L00005.

## References

- [1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, California, June 2019.
- [2] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, 2016.
- [3] J. Bernal, K. Kushibar, D. S. Asfaw et al., "Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review," *Artificial Intelligence in Medicine*, vol. 95, pp. 64–81, 2019.
- [4] X. Zhu, Z. Li, X. Li, S. Li, and F. Dai, "Attention-aware perceptual enhancement nets for low-resolution image classification," *Information Sciences*, vol. 515, pp. 233–247, 2021.
- [5] X. Zhu, Z. Li, X. Zhang, H. Li, Z. Xue, and L. Wang, "Generative adversarial image super-resolution through deep dense skip connections," *Computer Graphics Forum*, vol. 37, no. 7, pp. 289–300, 2018.
- [6] X. Zhu, Z. Li, X. Zhang, C. Li, Y. Liu, and Z. Xue, "Residual invertible spatio-temporal network for video super-resolution," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5981–5988, 2019.
- [7] X. Zhu, Z. Li, J. Lou, and Q. Shen, "Video super-resolution based on a spatio-temporal matching network," *Pattern Recognition*, vol. 110, p. 107619, 2021.
- [8] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 510–519, California, June 2019.
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, June 2005.
- [10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size," 2016, <http://arxiv.org/abs/1602.07360>.
- [12] D. G. Lowe, "Object recognition from local scale-invariant features," *ICCV*, vol. 99, pp. 1150–1157, 1999.
- [13] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," 2014, <http://arxiv.org/abs/1409.3215>.
- [14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Technical Report 7694, California Institute of Technology, Pasadena, 2007.
- [15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings*

- of the *IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, Hawaii, America, 2017.
- [16] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang, and P. Torr, “Res2net: a new multi-scale backbone architecture,” 2019, <http://arxiv.org/abs/1904.01169>.
  - [17] A. F. Agarap, “An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification,” 2017, <http://arxiv.org/abs/1712.03541>.
  - [18] X. Zhu, X. Zhang, X. Y. Zhang, Z. Xue, and L. Wang, “A novel framework for semantic segmentation with generative adversarial network,” *Journal of Visual Communication and Image Representation*, vol. 58, pp. 532–543, 2018.
  - [19] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
  - [20] Z. Li, X. Zhu, L. Wang, and P. Guo, “Image classification using convolutional neural networks and kernel extreme learning machines,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3009–3013, Athens, Greece, October 2018.
  - [21] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.
  - [22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, <http://arxiv.org/abs/1409.1556>.
  - [23] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 1–9, Boston, America, 2015.
  - [24] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2013, <http://arxiv.org/abs/1312.4400>.
  - [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 27, pp. 1097–1105, 2012.
  - [26] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Technical Report TR-2009, University of Toronto, Toronto, 2009.
  - [27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, America, 2016.
  - [28] S. Hou and Z. Wang, “Weighted channel dropout for regularization of deep convolutional neural network,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8425–8432, 2019.
  - [29] G. Larsson, M. Maire, and G. Shakhnarovich, “Fractalnet: ultra-deep neural networks without residuals,” 2016, <http://arxiv.org/abs/1605.07648>.
  - [30] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” 2015, <http://arxiv.org/abs/1502.03167>.
  - [31] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European conference on computer vision*, Lecture Notes in Computer Science, pp. 630–645, Springer, Cham, 2016.
  - [32] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in *European conference on computer vision*, pp. 646–661, Springer, 2016.