Journal on Artificial Intelligence
DOI:10.32604/jai.2021.027154
Article

Tech Science Press

Solving the Feature Diversity Problem Based on Multi-Model Scheme

Guanghao Jin[1], Na Zhao[1], Chunmei Pei[1], Hengguang Li[2], Qingzeng Song[3] and Jing Yu[1,*]

[1]School of Telecommunication Engineering, Beijing Polytechnic, Beijing, 100176, China
[2]Department of Mathematics, Wayne State University, Detroit, MI, 48202, USA
[3]School of Computer Science and Technology, Tiangong University, Tianjin, 300387, China
[*]Corresponding Author: Jing Yu. Email: yujing@edu.bpi.cn
Received: 11 January 2022; Accepted: 21 January 2022

Abstract: Generally, the performance of deep learning models is related to the captured features of training samples. When the training samples belong to different domains, the diverse features may increase the difficulty of training high performance models. In this paper, we built a new framework that generates multiple models on the organized samples to increase the accuracy of classification. Firstly, our framework selects some existing models and trains each of them on organized training sets to get multiple trained models. Secondly, we select some of them based on a validation set. Finally, we use some fusion method on the outputs of the selected models to get more accurate results. The experimental results show that our framework achieved higher accuracy than the existing methods. Our framework can be an option for the deep learning system to increase the classification accuracy.

Keywords: Deep learning; classification; distribution of labels; probability of labels

1 Introduction

Currently, deep learning models is utilized in many applications [1–5]. When utilizing the deep learning models for the classification, the performance of models is related to the captured feature. To get higher performance for the classification tasks, the structure should be well designed and the hyper-parameters should be well tuned [6–8]. When the training set contains the samples that belong to different domains, the diverse features may lower the performance of trained models [9,10]. Fig. 1 illustrates some samples that belong to different domains. As this figure introduced, the samples may be different in resolution or size. Furthermore, each label may have the different number of samples (For example, label dog has 1000 training samples while label deer has 10000 training samples), which is a challenge to the training of deep learning models. A general solution uses more layers to contain more features, which makes the structure bigger and more complicated. On the other side, that solution does not always work well as the computational resource is limited.

When using multiple models, we can make each of them well capture the features of corresponding domain to lower the demand of big structure. In this paper, we built a novel framework MMS (Multi-model scheme) that uses multiple models to improve the performance of classification. Our contribution can be summarized as the following. 1) Our framework can lower the requirement of computational resource. In our framework, the accuracy is increased by more models instead of increasing the structure of single big model. 2) We increase the error-tolerance of the classification. When using single model, the accuracy only depends on the output of this model. When using multiple models, there may be wrong results but also correct ones. By using fusion method, the classification accuracy may be increased. We evaluated the methods on some real datasets that include CIFAR-10 [11–13] and CIFAR-100 [14–16]. Furthermore, we also collected a real dataset where the samples have different resolution and each label has different number of samples. All of these experimental results proved the effectiveness of our framework.

We organize this paper by the following sections. Section 1 is the background and explains our contribution. Section 2 introduces the existing methods. In Section 3, based on some definitions, we explained our framework with some analyses. The experiment is organized in each part of Section 4. Section 5 shows the conclusion of this paper and follows the introduction of the future work.

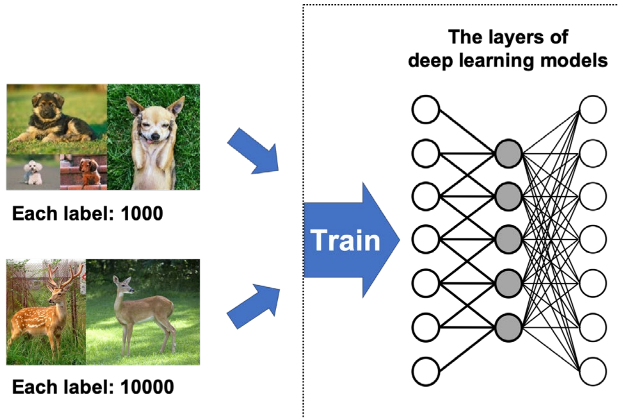

Figure 1: The samples have different resolutions and size; each label has different number of samples

2 Related Works

In this subsection, we firstly introduce some existing deep learning models. In these days, there are many deep learning models that are used in the image classification. Image classification is an important research field of computer vision, which extracts the features from the image to classify this image. The improvement of the accuracy is based on the huge amount of training samples. With the increase of the number or level of layers, the amount of calculation required by the training process also increases sharply. We select some of these models based on our computational resource as the baseline, which are VoVNet-39 [17], VGG16 [18] and ResNeSt50 [19].

• VoVNet-39: VoVNet is designed for the object classification, which consists of blocks, convolution layers and OSA modules [17]. An OSA module is to minimize MAC. VoVNet-39 has more OSA modules than the other types of VoVNet models where down-sampling is in the last.

•VGG16: VGG models is used for image recognition and classification [18]. The image is firstly passed through convolutional layers where the filters are used with a small receptive field. Then, spatial pooling is applied to the max-pooling layers, which is performed by a pixel sliding window. The final step is a soft-max layer and all of the hidden layers apply ReLU.

• ResNeSt50: A deep learning model for the classification, which applies split-attention block [19]. By the variety of these blocks, there are four versions of models. From ResNeSt50 to ResNeSt269, where the structure becomes bigger and more complicated for higher accuracy. Based on the size of testing samples and computational resource, we select ResNeSt50 from these ResNeSt models in this paper.

These models have been proved efficient in many utilizations. In some real applications, there are the cases that some samples belong to same label but there are big differences in the appearance. These differences include the size or resolution. A deep learning system should work well in these cases, which is important to the robustness and the level of intelligence. Deeper structure can be a solution while it is not always work because of the vanishing gradient problem [20–22]. Furthermore, big models need more computational resource, which limits the utilization of deep learning methods on some devices. Thus, applying multiple models in these cases can be a good choice.

The fusion method on multiple models can improve the performance of classification [23]. According to that paper, we found that the weighted voting method achieved the highest accuracy among all of the others. The weighted voting method is also utilized to produce the more reliable results [24]. A sliding window is utilized to a convolutional neural model and a long short-term memory model [25]. In this paper, we also apply a weighted voting method for higher accuracy.

3 Our Framework

Fig. 2 introduces our framework. Firstly, we try to increase the number of training sets by organizing the samples. Secondly, we select some existing deep learning models and train each model on each training set. Thirdly, we apply the fusion method to these trained models on the validation set. Then, we only select some of these models that can make the fusion method achieve the highest accuracy. Some hyper-parameters also computed based on this validation set. Finally, on a testing sample, the fusion method applies to the outputs of the selected models.

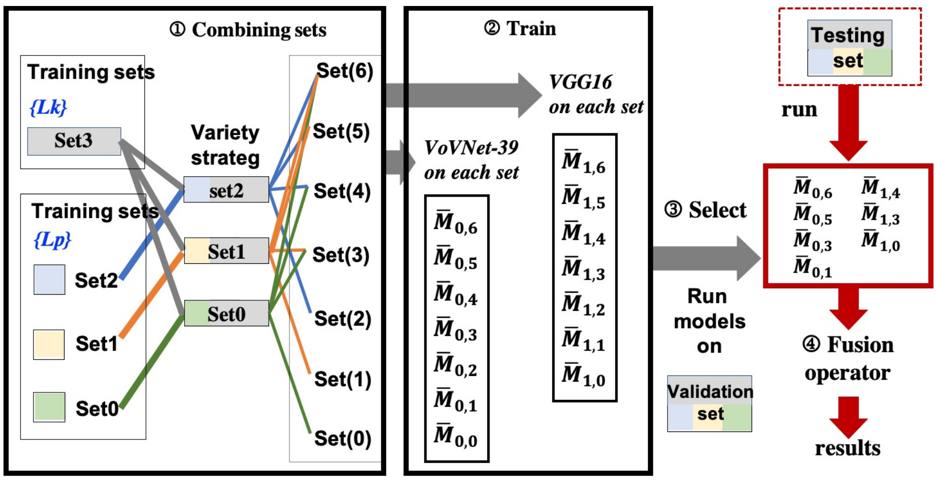

Figure 2: Our framework

We firstly give some definitions before the explanation. We define S_n as a sample and G_n as the label of S_n, which means the ground truth [26,27]. Generally, we can have $\forall\ G_n \in \{L_k\}$ where $k \in \{1,2,\ldots,N_{label}\}$ and L_k represents all of the labels. The classification task is to summarize S_n as L_k, which try to be the same as the ground truth G_n.

We define D_i as a domain where $i = \{1,2,\ldots,N_{domain}\}$ and N_{domain} as the number of these domains. Then we can train a deep learning model M_u on domain D_i to get a trained model $\bar{M}_{u,i}$. We define $R_{u,i,n} = \bar{M}_{u,i}(S_n)$ as the predicted label by the model $\bar{M}_{u,i}$ on a sample S_n. Then we can give the definition of deep learning task when using multiple models as the following: for a sample S_n, the task is to find out the highest accurate result $R_{u,i,n}$. Based on this definition, we can perform our method in the next subsection.

3.1 Combining the Training Sets

In this subsection, we introduce how to combine the training sets as the following steps. We set $\bar{D}_i$ as a training set that is derived from the domain D_i. There can be two kinds combinations. The first one is to make each combined training set contain the samples of all labels. Then, based on these combined training sets, we can arbitrary combine some of them to generate more training sets as the ① of Fig. 2 shows.

3.2 Train and Select Models

We can use the generated sets to train a deep learning model M_u. Then, on a combined training set $\{\bar{D}_i\}$, we can get a trained model $\bar{M}_{u,\{\bar{D}_i\}}$. The trained models may have different performance on the testing set. Some high accurate models may increase the accuracy while the others may reduce the accuracy. Thus, we should select the trained models that can make the fusion method achieve high accuracy. We can use a validation set to select the models and compute some hyper-parameters, which will be introduced in Subsection 3.3.

3.3 The Fusion Method

To use the fusion method, we should build a validation set that is labelled by the ground truths. Then we can use the fusion method that combine the outputs of models as the following:

$$W_{u,\{i\}} = 1/\left(1-\tilde{P}\left(\bar{M}_{u,\{i\}}(S_n)=G_n\right)\right), W_{u,\{i\}} = W_{u,\{i\}} / \sum W_{u,\{i\}}$$
$$L_x = \arg\max_{L_k} \sum_{\bar{M}_{u,\{i\}}(S_n)=L_k} W_{u,\{i\}} \tag{1}$$

where $\tilde{P}\left(\bar{M}_{u,\{i\}}(S_n)=G_n\right)$ presents the posterior accuracy of a trained model $\bar{M}_{u,\{i\}}$ on the validation set. Furthermore, we can also compute $W_{u,\{i\}}$ by this validation set.

Each trained model can well capture the features on the corresponding training set. On the other side, the trained model may achieve low accuracy on the other training sets. Thus, the selection of models plays important role to achieve high accuracy.

4 Experiments

We evaluated the methods on some real datasets. We trained the deep learning models by their default settings (without changing the structure). We set the number of epochs [28,29] as 10 when training these models. When we evaluate a random parameter, we evaluate 1000 times and compute the average as the final result.

4.1 The Evaluation on CIFAR-10

CIFAR-10 contains 50000 training samples and 10000 testing ones, which can be summarized as 10 labels and all of the samples belong RGB image [11–13]. 50000 training samples are to train the deep learning models. Then, we randomly select 1000 validation samples from 10000 testing samples. To obviously show the effect of domains to the models, we construct four domains by shifting the points of color channels. The samples of 4 domains are shown in Fig. 3.

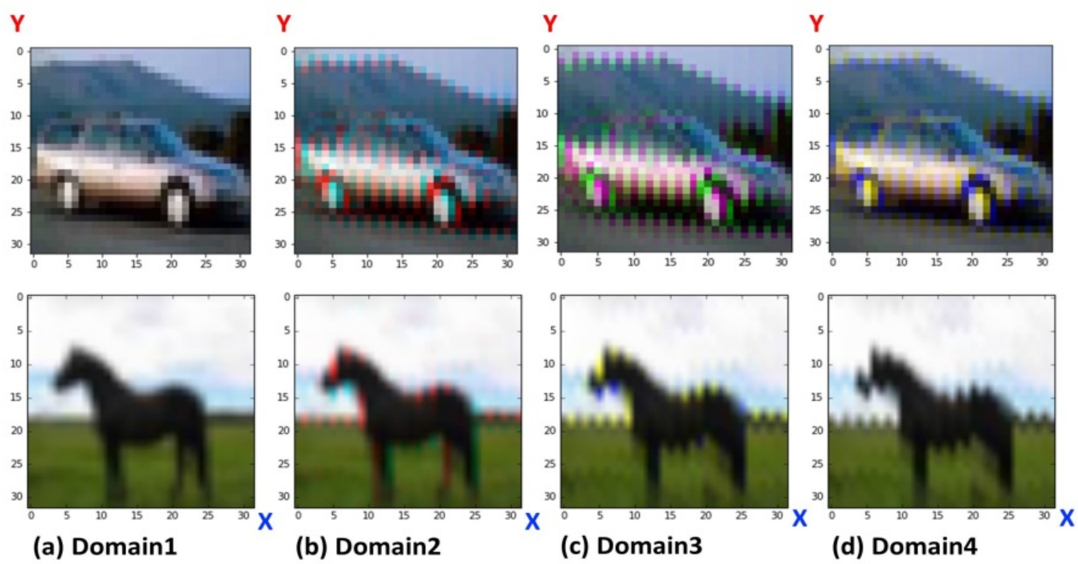

Figure 3: The shifted samples of the same label

As we can see in Table 1, our best method achieved higher accuracy: 2.1% than the best one of the existing methods. *Appeared Max* selects the final result that appeared max times among the outputs of models.

Table 1: Classification results on CIFAR-10

Models / Methods	VoVNet-39 [17]	VGG16 [18]	ResNeSt50 [19]
1-Model	82.72%	86.09%	79.39%
Appeared Max	68.89%	85.64%	70.79%
Fusion Method [25]		87.65%	
Our framework (MMS)		89.75%	

4.2 The Evaluation on CIFAR-10 and CIFAR-100

CIFAR-100 is similar to the CIFAR-10. This set has 100 classes and each class has 600 images [14–16]. 50000 training samples are to train the models. We combined the generated 4 domains of CIFAR-10 (introduced in the Subsection 4.1) and the samples of CIFAR-100 to evaluate the methods. In more details, we use the training samples of CIFAR-10 and CIFAR-100 to train the models. Then we randomly select 1000 of CIFAR-10 and CIFAR-100 for the validation set. The remained 19000 samples of these two datasets are for the testing.

Table 2: Classification results on CIFAR-10+CIFAR-100

Models / Methods	VoVNet-39 [17]	VGG16 [18]	ResNeSt50 [19]	VoVNet-39 [17]	VGG16 [18]	ResNeSt50 [19]
1-Model	83.36%	79.79%	79.36%	62.43%	43.87%	57.48%
Appeared Max	62.94%	62.06%	68.14%	58.23%	42.90%	48.10%
Fusion Method [25]		86.65%			65.05%	
Our framework (MMS)		89.64%			67.85%	

As we can see in Table 2, our best method achieved higher accuracy: On the samples of CIFAR-10, 2.99% than the best one of the existing methods; on the samples of CIFAR-100, 2.80% than the best one of the existing methods

4.3 The Evaluation on the Collected Dataset

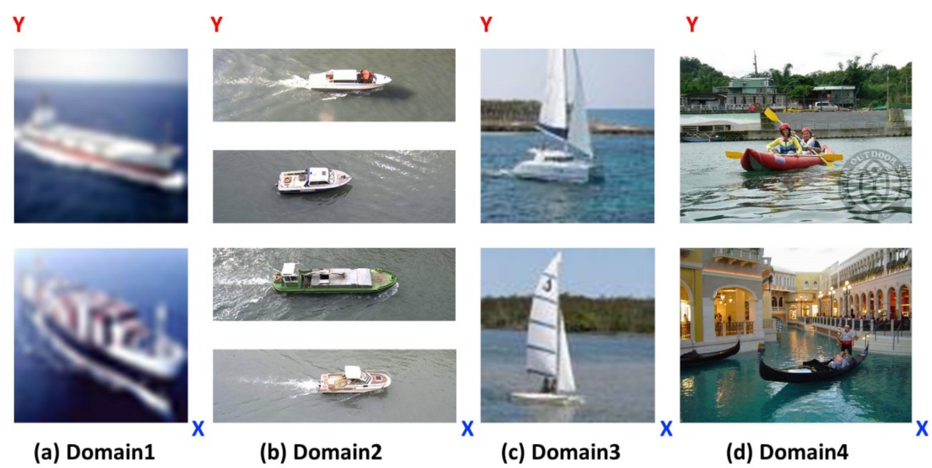

Figure 4: The examples of collected dataset

In this subsection, we construct 4 domains by collecting the samples from some real datasets. These samples have different resolutions and sizes. The labels are based on those of CIFAR-10 [11–13]. When there is only one dataset for the labels like *frog* or *deer,* the samples are separated into different domains by their appearances like the color.

Domain1 is CIFAR-10 dataset as we introduced in the Subsection 4.1. Domain2 has 76605 samples where we use 80% for training and 20% for testing. We collect Domain2 from the datasets [30-36] with the *frog* samples from https://github.com/jonshamir/frog-dataset.

Domain3 has 32089 samples where we use 80% for training and 20% for testing. The samples are from the datasets [32,36–39] with frog dataset from https://github.com/jonshamir/frog-dataset and horse dataset from http://www.laurencemoroney.com/horses-or-humans-dataset

Domain4 has 67980 samples where we use 80% for training and 20% for testing. The samples are from Imagenet dataset [40,41].

Table 3: Classification results on the collected dataset

Methods \ Models	VoVNet-39 [17]	VGG16 [18]	ResNeSt50 [19]
1-Model	82.68%	81.49%	80.39%
Appeared Max	60.22%	53.79%	57.2%
Fusion Method		84.62%	
Our framework (MMS)		86.75%	

Fig. 4 shows some samples of 4 domains, which belong to the same label but have high or low resolutions (also have different figure sizes, each label has different number of samples). We set 10000 samples as the validation set and 32121 samples as testing set. As we can see in Table 3, our best method achieved higher accuracy: 2.13% than the best one of the existing methods.

4.4 Analysis

We have evaluated our framework and the existing methods on real datasets. The selected deep learning models achieved different accuracies from low (which is about 44%) to high (which is about 86%). The experimental results show the effectiveness of our framework. We also evaluated our framework on some real samples of collected datasets, which also achieved higher accuracy than the existing ones.

When increasing the structure of a model is impossible for some reasons, the utilization of multiple models can be a choice as our framework proved. As we applied multiple models, the execution time may be increased, which is related to the number of models. This problem can be solved by executing these models at the same time by using parallel technology [42].

5 Conclusions

In this paper, we have introduced a novel framework that uses multiple models to increase the accuracy of classification. Our framework can be a choice to increase the classification accuracy especially when the training set contains the samples of multiple domains.

As the wide utilization of deep learning models, there will be cases that should combine the outputs of multiple models. These models may be installed in different devices or places. In the future work, we will do research about the parallel execution of multiple models, which is to solve the communication time between models.

Funding Statement: National Natural Science Foundation of China (Grant Nos. 61702281, 61802279, 6180021345 and 61702366), Natural Science Foundation of Tianjin (Grant Nos. 18JCQNJC70300, 19PTZWHZ00020, 19JCTPJC49200 and 19JCYBJC15800), Fundamental Research Funds for the Tianjin

Universities (Grant No. 2019KJ019), and the Tianjin Science and Technology Program (Grant No. 19PTZWHZ00020) and in part by the State Key Laboratory of ASIC and System (Grant No. 2021KF014) and Tianjin Educational Commission Scientific Research Program Project (Grant Nos. 2018KJ215 and 2020KJ112).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Fang and J. Q. Ouyang, "Classification algorithm optimization based on Triple-GAN," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 1–15, 2020.

[2] C. Song, X. Cheng, Y. X. Gu, B. J. Chen and Z. J. Fu, "A review of object detectors in deep learning," *Journal on Artificial Intelligence*, vol. 2, no. 2, pp. 59–77, 2020.

[3] Z. Chen, L. Yan, S. Yin and Y. Shi, "Vehicle license plate recognition system based on deep learning in natural scene," *Journal on Artificial Intelligence*, vol. 2, no. 4, pp. 167–175, 2020.

[4] B. Almaslukh, "Deep learning and entity embedding-based intrusion detection model for wireless sensor networks," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 1343–1360, 2021.

[5] A. K. Mishra and V. Ponnusamy, "Deep learning-based decoding and ap selection for radio stripe network," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 131–148, 2022.

[6] A. Schumann, S. Gong and T. Schuchert, "Deep learning prototype domains for person re-identification," in *2017 IEEE Int. Conf. on Image Processing (ICIP)*, vol. 2017, no. 1, pp. 1767–1771, 2017.

[7] S. W. Park, J. Park, K. Bong, D. Shin, J. Lee *et al.,* "An energy-efficient and scalable deep learning/inference processor with tetra-parallel MIMD architecture for big data applications," *IEEE Transactions on Biomedical Circuits & Systems*, vol. 9, no. 1, pp. 838–848, 2016.

[8] S. Huang, J. Wang, D. Wang, M. Zhang and H. You, "A thread-saving schedule with graph analysis for parallel deep learning applications on embedded systems," *IEEE International Conference on Smart Cloud*, vol. 1, no. 1, pp. 111–115, 2018.

[9] H. Venkateswara, J. Eusebio, S. Chakraborty and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," *CVPR*, vol. 2017, no. 1, pp. 5385–5394, 2017.

[10] O. Abdelwahab and A. Elmaghraby, "Deep learning based vs. markov chain based text generation for cross domain adaptation for sentiment classification," *IRI*, vol. 1, no. 1, pp. 252– 255, 2018.

[11] R. D. Gottapu and C. H. Dagli, "Efficient architecture search for deep neural networks," *Procedia Computer Science*, vol. 168, no. 1, pp. 19–25, 2020.

[12] H. Zhang and M. Xu, "Improving the generalization performance of deep networks by dual pattern learning with adversarial adaptation," *Knowledge-Based Systems*, vol. 200, no. 1, pp. 1–12, 2020.

[13] X. Li, M. Xu, J. Xu, T. Weise and Z. Wu, "Image retrieval using a deep attention-based hash," *IEEE Access*, vol. 8, no. 1, pp. 142229–142242, 2020.

[14] M. Y. Huang, C. H. Lai and S. H. Chen, "Fast and accurate image recognition using deeply-fused branchy networks," in *2017 IEEE Int. Conf. on Image Processing (ICIP)*, vol. 2017, no. 1, pp. 2876–2880, 2017.

[15] B. Stuhr and J. Brauer, "CSNNs: Unsupervised, backpropagation-free convolutional neural networks for representation learning," in *2019 18th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, vol. 1, no. 1, pp. 1613–1620, 2020.

[16] C. Termritthikun, Y. Jamtsho and P. Muneesawang, "An improved residual network model for image recognition using a combination of snapshot ensembles and the cutout technique," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 1475–1495, 2020.

[17] Y. Lee, J. Hwang, S. Lee, Y. Bae and J. Park, "An energy and GPU-computation efficient backbone network for real-time object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, vol. 2019, no. 1, pp. 752–760, 2019.

[18] Z. Liu, J. Wu, L. Fu, Y. Majeed and Y. Cui, "Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion," *IEEE Access*, vol. 8, no. 1, pp. 2327–2336, 2020.

[19] Q. Y. Xu, G. H. Qin, M. H. Sun and C. X. Meng, "Classification of drivers' head status based on improved ResNeSt," *Computer Vision and Pattern Recognition*, vol. 51, no. 1, pp. 704–711, 2020.

[20] W. Min, M. H. Frankosky, B. W. Mott, J. P. Rowe and J. Lester, "DeepStealth: Leveraging deep learning models for stealth assessment in game-based learning environments," in *Int. Conf. on Artificial Intelligence in Education Springer*, vol. 9112, no. 1, pp. 277–286, 2015.

[21] C. Ding, Y. Li, L. Zhang, J. Zhang, L. Yang *et al.,* "Fast-convergent fully connected deep learning model using constrained nodes input," *Neural Processing Letters*, vol. 49, no. 3, pp. 995–1005, 2019.

[22] K. Sharma, A. Alsadoon, P. Prasad, T. Al-Dala'in and D. Pham, "A novel solution of using deep learning for left ventricle detection: Enhanced feature extraction," *Computer Methods and Programs in Biomedicine*, vol. 197, no. 1, pp. 1–12, 2020.

[23] G. Aceto, D. Ciuonzo, A. Montieri and A. Pescapé, "Multi-classification approaches for classifying mobile app traffic," *Journal of Network and Computer Applications*, vol. 103, vol. 1, pp. 131–145, 2018.

[24] Y. You, F. Li and X. Li, "On redundant weighted voting systems with components having stochastic arrangement increasing lifetimes," *Operations Research Letters*, vol. 49, no. 5, pp. 777–784, 2021.

[25] X. Liu and A. G. Richardson, "Edge deep learning for neural implants: a case study of seizure detection and prediction," *Journal of Neural Engineering*, vol. 18, no. 4, pp. 1–16, 2021.

[26] A. Khodja, Z. Zheng, J. Mo, D. Zhang and L. Chen, "Rain to rain: Learning real rain removal without ground truth," *IEEE Access*, vol. 9, no. 1, pp. 57325–57337, 2021.

[27] Z. Rania, B. Sonia and Z. Ezzeddine, "Unsupervised image-adapted local fisher discriminant analysis to reduce hyperspectral images without ground truth," *IEEE Transactions on Geoscience and Remote Sensing*, vo. 58, no. 1, pp. 7931–7941, 2020.

[28] G. Lim, Y. Cheng, W. Hsu and M. L. Lee, "Integrated optic disc and cup segmentation with deep learning," in *IEEE Int. Conf. on Tools with Artificial Intelligence*, vol. 2016, no. 1, pp. 162–169, 2016.

[29] W. Song, W. Li, Z. Hua and F. Zhu, "A new deep auto-encoder using multiscale reconstruction errors and weight update correlation," *Information Sciences*, vol. 559, nol. 1, pp. 130–152, 2021.

[30] H. Zheng, J. Fu, M. Tao and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *2017 IEEE Int. Conf. on Computer Vision (ICCV)*, vol. 2017, no. 1, pp. 5219–5227, 2017.

[31] J. Krause, M, Stark, J. Deng and F. F. Li, "3D object representations for fine-grained categorization," in *IEEE Int. Conf. on Computer Vision Workshops IEEE*, vol. 1, no. 1, pp. 554–561, 2014.

[32] N. Zhang, J. Donahue, R. Girshick and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *European Conf. on Computer Vision*, vol. 8689, no. 1, pp. 834–849, 2014.

[33] C. Mash, P. C. Quinn, V. Dobson and D. B. Narter, "Global influences on the development of spatial and object perceptual categorization abilities: Evidence from preterm infants," *Infant Behavior & Development*, vol. 1, no. 1, pp. 84–102, 2002.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* "ImageNet large scale visual recognition challenge," *International Journal of Computer*, vol. 115, no. 1, pp. 211–252, 2014.

[35] D. D. Bloisi, L. Iocchi, A. Pennisi and L. Tombolini, "ARGOS-Venice boat classification," in *IEEE Int. Conf. on Advanced Video & Signal Based Surveillance*, vol. 1, no. 1, pp.1–6, 2015.

[36] F. C. Soon, Y. K. Hui, J. H. Chuah and J. Kanesan, "Semi-supervised PCA convolutional network for vehicle type classification," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 8267–8277, 2020.

[37] S. Zhu, X. Y. Wei and C. W. Ngo, "Collaborative error reduction for hierarchical classification," *Computer Vision & Image Understanding*, vol. 124, no. 1, pp. 79–90, 2014.

[38] O. M. Parkhi, A. Vedaldi, A. Zisserman and C. V. Jawahar, "Cats and dogs," in *IEEE Conf. on Computer Vision & Pattern Recognition IEEE Computer Society*, vol. 1, no. 1, pp. 3498–3505, 2012.

[39] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 1, pp. 4367–4375, 2018.

[40] D. Hou, J. Ma, S. Huang, J. Zhang and X. Zhu, "Classification of defective photovoltaic modules in ImageNet-trained networks using transfer learning," in *2021 IEEE 12th Energy Conversion Cong. and Exposition–Asia*, vol. 1, no. 1, pp. 2127–2132, 2021.

[41] L. Blaivas and M. Blaivas, “Are convolutional neural networks trained on ImageNet images wearing rose-colored glasses? A quantitative comparison of ImageNet, computed tomographic, magnetic resonance, chest X-ray, and point-of-care ultrasound images for quality,” *Journal of Ultrasound in Medicine*, vol. 40, no. 2, pp. 377–383, 2021.

[42] T. T. Leonid and R. Jayaparvathy, “Classification of elephant sounds using parallel convolutional neural network,” *Intelligent Automation & Soft Computing*, vol. 32, no. 3, pp. 1415–1426, 2022.