Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 3197 – 3211
© TÜBİTAK
doi:10.3906/elk-2102-101

A hybrid approach based on transfer and ensemble learning for improving performances of deep learning models on small datasets

Tunç Gültekin*, **Aybars Uğur**
Computer Engineering Department, Ege University, İzmir, Turkey

Received: 21.02.2021 • **Accepted/Published Online:** 19.07.2021 • **Final Version:** 30.11.2021

Abstract: The need for high-volume data is one of the challenging requirements of the deep learning methods, and it makes it harder to apply deep learning algorithms to domains in which the data sources are limited, in other words, small. These domains may vary from medical diagnosis to satellite imaging. The performances of the deep learning methods on small datasets can be improved by the approaches such as data augmentation, ensembling, and transfer learning. In this study, we propose a new approach that utilizes transfer learning and ensemble methods to increase the accuracy rates of convolutional neural networks for classification tasks on small data sets. To this end, we generate different-sized sub-networks by fragmenting an existing large pre-trained network then gather those networks to form an ensemble. For ensemble scoring, we also suggest two new methods. Conducted experiments with the proposed technique, on a randomly sampled Cifar10 small subset dataset, reveals promising results.

Key words: Ensemble learning, transfer learning, convolutional neural network, small dataset, deep learning, VGG16

1. Introduction

In recent years, deep learning methods perform state of art results in many different machine learning tasks. However, the requirements of these methods are much more than conventional machine learning techniques. In order to learn complex relationships and features, deep learning methods need plenty of training of samples (tens of thousands depending on the complexity of the problem). As a result of this, using special hardware such that GPU or TPU becomes a strict requirement for acceptable training times. Large dataset requirement makes it harder to apply deep learning techniques especially for the domains where it is difficult to collect many training samples such as medical image analysis. In literature, there exist several approaches applying deep learning methods successfully, especially convolutional neural networks, to small data sets. These approaches can be examined under three main categories: data augmentation, ensembling, and transfer learning. Data augmentation methods take the advantages of rotation, crop, zoom, shear transformation [1] PCA multipliers [2] techniques to produce new images and the data set size is increased [3] . For the ensemble methods, outputs of trained networks with different initial weights for the same data set are combined, and final results are produced. Whereas, in transfer learning approaches, the layers, in other words, weights of a deep learning model [2, 4], which were previously trained on a large data set, are transferred to a new network. By using this technique, high accuracy rates can be achieved with a short amount of training effort [5, 6]. This study has two folds. Firstly, we propose a new approach that uses transfer learning and ensemble methods to improve the performance of deep learning methods on small data sets. Secondly, we introduce two new methods for

*Correspondence: tuncgultekin@gmail.com

calculating the ensemble score in ensemble methods and compare them on a small data set, which is obtained by random sampling of a well-known Cifar10 dataset.

2. Related work

In deep learning literature, to improve model performances, transfer and ensemble learning approaches have been employed. In one of the studies that make use of both transfer and ensemble methods [7], classification is performed through an ensemble of deep neural networks, and those networks were constructed by transferring layers from pretrained networks.

The transfer learning approach is one of the most preferred methods for improving model performances on small data sets, and this technique is applied in various areas. For instance, in LIGO gravitational wave detectors, transferred weights from pre-trained VGG19 [25], ResNet[26], Inception V3 [27] networks were used for distinguishing actual signal from noisy data [8].

In another study [9] for medical image classification, which is another domain where the data is limited, subnetworks were created by using pretrained VGG16[25], and ResNet networks, and these networks were combined with a handcrafted neural network to constitute an ensemble. The final classification results were obtained by the majority voting method. In [5], the features which were transferred from the ImageNet dataset was employed for thoracoabdominal lymph node (LN) detection and interstitial lung disease (ILD) classification. In [10], they made use of CNN ensembles to identify vessels that appear in retinal images. To this end, random patches were cropped from retinal images, and 12 different CNN models were trained. Then, prediction results were calculated by taking the simple arithmetic mean of CNN models' posterior output probabilities. Similarly, in [33–35], to increase the performance of the classification system on limited number of medical images, the transfer learning and ensemble methods were used together. To create ensembles, they took the advantages of pretrained VGGNet, ResNet and DenseNet architectures.

Hierarchically, CNN ensembles were also shown to be effective. In [11], 2 different CNN ensembles were created and used for coarse and fine grade classification respectively. The outputs of those CNNs, which have common layers, were combined by averaging their last layers' outputs. In [12], deep learning algorithms were combined with support vector machine (SVM) ensembles for semantic event detection. Hence, support vector machine ensembles were fed with feature vectors, which were created by deep learning algorithms. The final results were obtained by taking the weighted average of SVM outputs, and aforementioned weights were determined with respect to the performances of each support vector machine on the validation set.

In the transfer learning approach, layers' weights are taken from a model that is trained on a different large data set. In the literature, there exist different techniques that employ transfer learning. For an emotion prediction task, transfer learning was applied by using two-staged fine-tuning [13]. Combinations of different data sets can also be used for fine-tuning operations; in [14], different deep convolutional neural network architectures such as VGG-19[25], AlexNet[28] and GoogleNet[29] were focused and fine-tuned with a common loss function. Transferred features from different pretrained networks can also be used as a single large feature vector. In [15], the feature vectors of the several pretrained models were concatenated to obtain better feature representations for medical image datasets. In [31], the transfer learning approach adapted to AdaBoost algorithm and used with CNN models to improve classification performance on imbalanced datasets. For each iteration of the AdaBoost, the CNN weights were transferred from the previous iteration's CNN classifier. As a different domain than medical imaging, the CNN ensembles, which were created via transfer learning, were used for fault diagnosis in [32].

Instead of ensemble weighting, ensemble performance can be improved by member selection. In [16], they proposed an ensemble selection framework that takes the advantages of meta-learning. In addition to utilizing transfer learning and ensemble learning methods to improve performance on small data sets, efficient neural network architectures can be obtained by applying iterative optimization methods such as evolutionary algorithms and reinforcement learning. For the automated deep neural network architecture generation, a special reinforcement learning method, namely Q-Learning, was used, and sequential basic layer types (convolutional, fully connected, pooling, softmax) were combined with Q-learning. To this end, various layer types were modeled as nodes on an acyclic graph, and the connections between the layers were shown as graph node connections.

In CoDeepNEAT [18], one of the studies that take the advantages of evolutionary algorithms for neural network generation, new network architectures were created and hyperparameter optimization was performed with genetic algorithm technique. The method starts with extremely simple network architecture and iteratively improves it by adding new layers to the network. In a similar study [19], the network performance was improved by changing the layer connections of LeNET reference CNN architecture by utilizing a genetic algorithm. This method employs fixed length chromosomes, and the reference model is evolved from a chain-network to a multipath-network.

3. Methodology

In this study, we take the advantages of both transfer and ensemble learning methods to improve the classification accuracy of deep convolutional neural networks (CNN) on small data sets. To this end, we firstly create a heterogeneous CNN ensemble by transferring weights of different convolutional layers from a pretrained VGG16 model, later on, we combine the outputs of each ensemble item by employing different scoring methods.

This study differs from its counterparts in terms of ensemble creation, ensemble scoring, and the goal. As a goal; rather than a specific use-case, we aim to propose an end-to-end framework to improve the image classification performance for small data sets. For ensemble creation, we generate different-sized sub-networks by fragmenting an existing large pretrained network then gather those networks to form an ensemble. This approach allows us to create a sufficiently diverse predictor set to reduce overfit effects. To calculate the ensemble's output, we propose different approaches and evaluate them. In the following sections, the details of the method are explained and experiment results are discussed.

3.1. Creating an ensemble by transferring different layers of a pretrained network

In the first stage, we transfer different layers from a VGG16 network that has been trained on the ImageNet dataset and create varying sized submodels. For each submodel m, we transfer the layers from pretrained source VGG16 network by starting from the first layer to layer t_m. Thanks to the varying $t$values, we introduce heterogeneity to our ensemble. Submodel creation and layer transfer processes are shown in Figure 1. The model, which is created by transferring the first t layers from VGG16 network is denoted as m_t (Eq. 1).

$$m_t = \{l_0, l_1, \ldots, l_t\} \qquad (Eq.1)$$

$$m_{t+1} = \{l_0, l_1, \ldots, l_{t+1}\}$$

$$m_{t+2} = \{l_0, l_1, \ldots, l_{t+2}\}$$

$$where\ l_t\ denotes\ t^{th}\ layer\ of\ VGG16$$

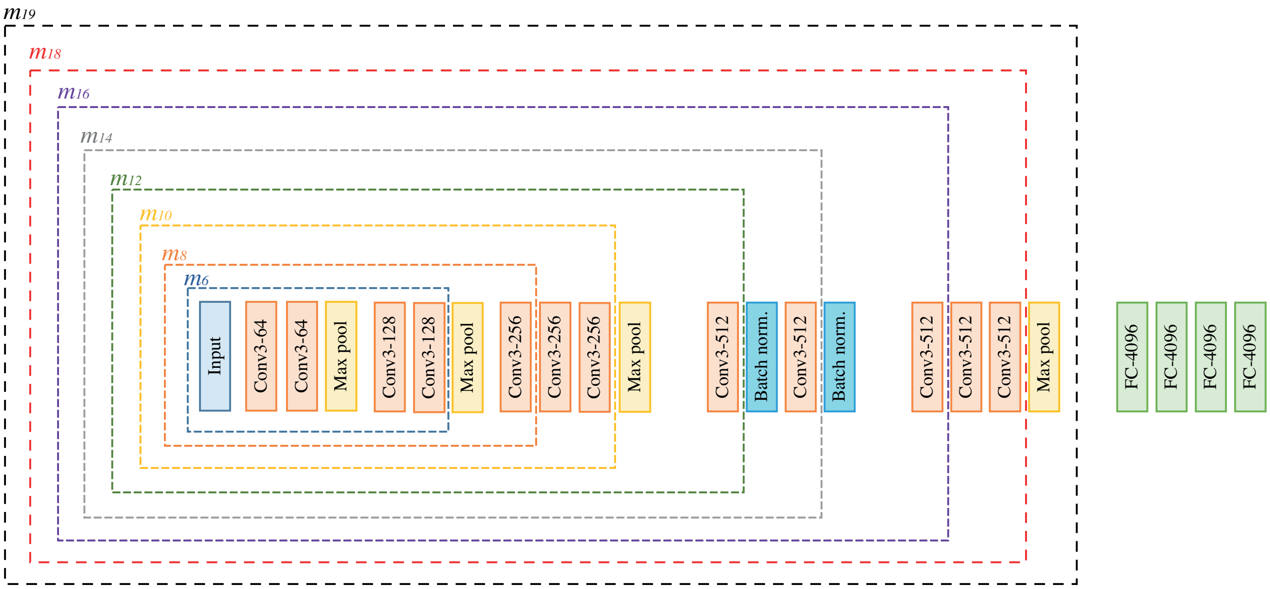


Figure 1. Transferred layers from pre-trained VGG16 network for sub-model creation.

In case of data input scale mismatch between source and target datasets, the performance of the transfer learning becomes limited. To address this issue, we add ‘BatchNormalization’ layers after each convolutional layer while transferring layer weights. The inserted ‘BatchNormalization’ layers to a submodel that includes the first 6 layers m_6 of VGG16 are shown in Figure 2. The number of transferred layers directly affects the classification accuracy and generalization ability of the model. When a few layers are transferred, the resulting network does not learn well enough (underfit). Similarly, depending on the target dataset characteristics, transferring all layers of the source model does not always guarantee the best transfer learning performance, since the resulting model might memorize the data (overfit). The risk of overfitting is dependent on the destination task’s domain specificity and the number of training samples. Hence, complex models which contain many layers are more prone to overfitting on a limited data domain. In [38], it was showed that altering the CNN architecture depth improves accuracy on limited datasets.

As shown in Figure 3, for a randomly sampled (with even class distribution) Cifar-10 data set, including 750 training and 150 test images, the highest classification accuracy is achieved when the first 16 layers of the VGG16 network are transferred. The last data point in Figure 3 corresponds to the results of a network trained by transfer learning without any layer removal. To obtain a CNN based robust classification ensemble, we create different-sized 8 new m_t models by transferring the first t layers from VGG16 network. These t values for m_t models defined as 6, 8, 10, 12, 14, 16, 18, and 19. The motivation behind this different-sized CNN ensemble approach is minimizing overfit effects which occurs due to the limited data and providing a general framework regardless of target domain for applying CNN models to small datasets. CNN ensembles can also be formed by gathering same-sized large models; however, depending on the characteristics of the target set, it reduces generalization capability and classification performance. The experimental results for homogeneous and heterogeneous CNN ensembles on small datasets are presented in the Experiments Section.

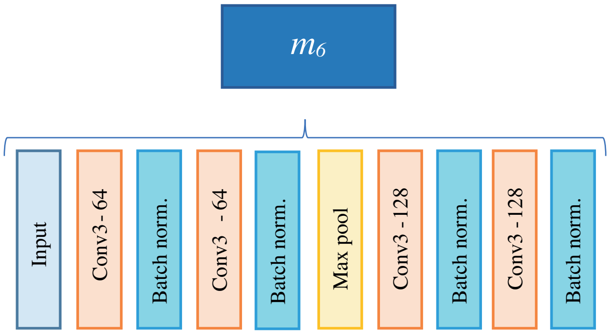


Figure 2. The ‘BatchNormalization’ layers which are added to m_6 submodel.

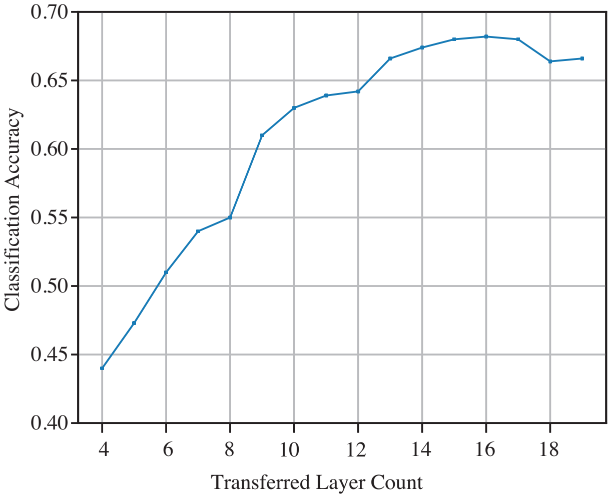


Figure 3. Single model accuracy with respect to transferred layer count from the reference model.

3.2. Ensemble scoring

Ensemble-based classification systems require a scoring mechanism to merge the ensemble's items' prediction results and to produce the final output. Thereby different approaches such as simple averaging, majority voting, etc. exist in the literature. Under the scope of this study, we propose a new approach, which takes the advantages of transfer and ensemble learning for improving the performances of deep learning models on small datasets. We aim to have an end-to-end solution and ensemble scoring is one of the important parts of it. Therefore we also suggest various ensemble scoring approaches and investigate their effects on our approach. Besides applying existing ensemble scoring techniques, we propose two new approaches, namely “majority voting with ensemble elimination” and “neural network-based ensemble weighting” to calculate the ensemble's prediction result. The overall architecture of the proposed method is shown in Figure 4.

3.2.1. Ensemble scoring via probability distribution based majority voting

As a first approach, we calculate the final prediction result of the ensemble by employing a majority voting schema. Unlike previous studies, we make use of output probabilities of each ensemble item, instead directly counting the frequencies of the top answers. The calculation of the ensemble's final prediction result $y_{ensemble}$ (2) is shown in Eq. 2 where the total model count is T, the total class count is S and the output classification

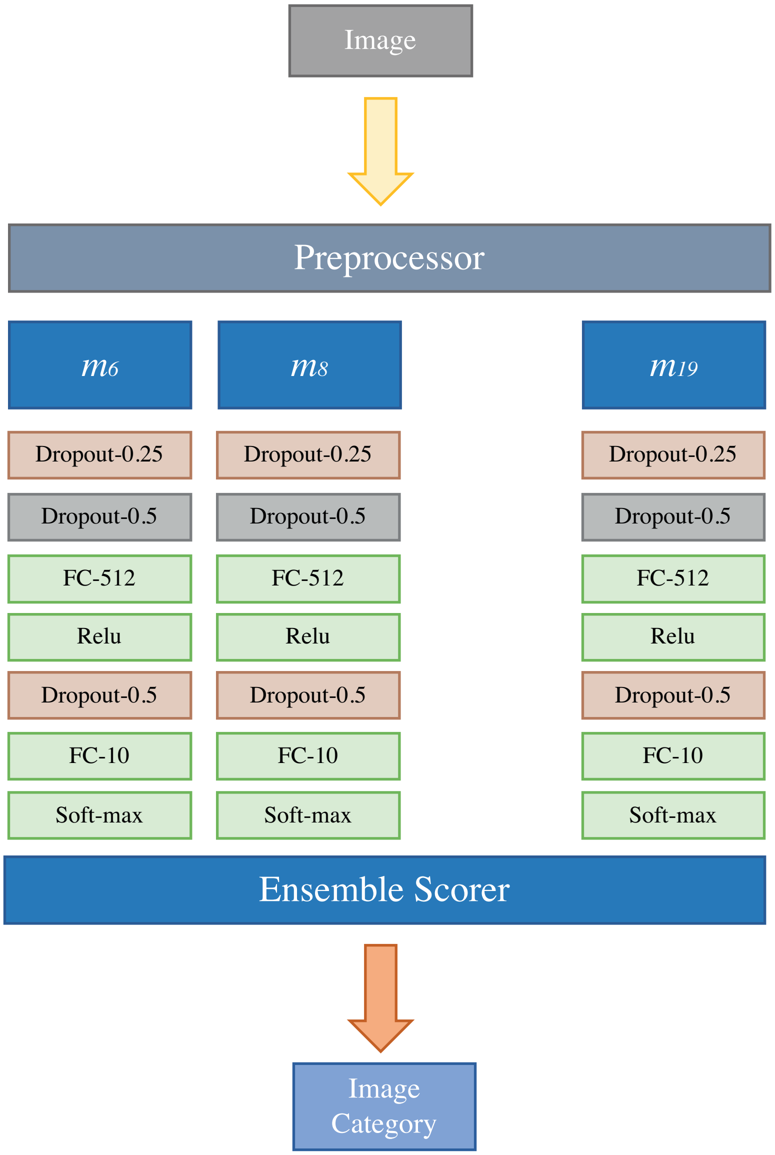


Figure 4. The overall architecture of the proposed method.

probability vectors, which includes the likelihood probabilities of each class y_s for a model m_t are $P_{m_t}(x)$.

$$P_{m_t}(x) = [y_0, y_1, \ldots, y_S] \quad (2)$$

$$P_E(x) = \sum_{t=1}^{T} P_{m_t}(x)$$

$$y_{\text{ensemble}} = \operatorname{argmax}(P_E(x))$$

3.2.2. Ensemble scoring via majority voting with ensemble elimination

The ensemble consists of models with different generalization performance, the generalization ability of the small models are higher than more complex ones, thus eliminating some of the models from the ensemble considering their performances could improve the overall performance of the ensemble. Based on this idea, as a second approach, we remove the models whose classification accuracies are lower than a threshold value. To determine model performances, firstly we create a Gaussian noise added variant of the actual training set and measure each model's classification accuracy on it. Then, we adopt the mean classification accuracy of the models as a threshold and eliminate the ones whose performance worse than the threshold. Since we evaluate models on a similar but noisy set, this approach allows us to select robust models for the final ensemble. The details of the $y_{ensemble}$ calculation with this procedure are shown in Eq. 3 where the $score_i$ is classification accuracy of the model i in noisy reference dataset, and the mean accuracy of the ensemble on the reference dataset is thr.

$$P_E(x) = \sum_{t=1}^{T} \begin{cases} P_{m_t}(x) & \text{score}_i \geq thr \\ 0 & \text{score}_i < thr \end{cases} \qquad (3)$$

$$y_{\text{ensemble}} = \text{argmax}\,(P_E(x))$$

3.2.3. Ensemble scoring via neural network weighting

Artificial neural networks (ANN) can be utilized to optimize weights to produce the best linear neural network combination for an ensemble [30]. In this technique, which is shown in Figure 5, the ensemble weights are considered as an artificial neuron's weights and optimized with the standard backpropagation method.

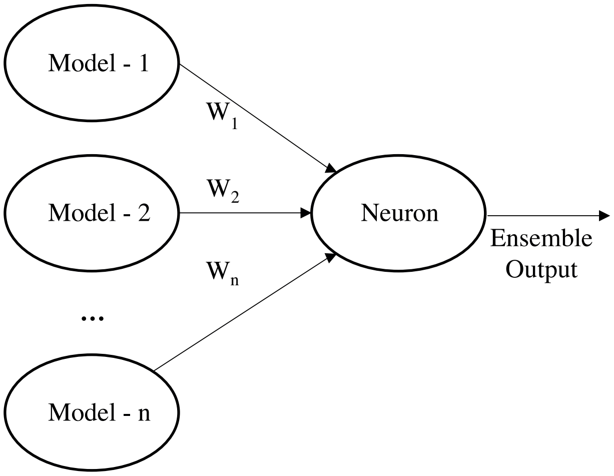


Figure 5. Ensemble scoring via NN

For the third ensemble scoring approach, we adopt a similar neural weighting technique with a couple of modifications. Instead, employing a single scalar weight value for each ensemble item, we use weight vectors. More specifically, we define a weight value for each models' corresponding probability output of each class. By considering some of the ensemble items are good at predicting only specific classes, this method allows us to control ensemble weights with respect to classes. The relation between neuron weights w_t, output probability vector of each ensemble item P_t, and ensemble's final prediction is shown in Eq. 4 where the t denotes the id of a model in the ensemble.

$$y_{\text{ensemble}} = \operatorname{argmax}\left(\sum_{t=1}^{T} w_t P_t\right) \qquad (4)$$

When an ensemble with 8 different models and a sample dataset with 10 classes are considered, this approach requires 10 different input layers each of which corresponds to output probabilities for each class. The input vector length of the input layers is equal to the model count. The layer connections and the details of the modified neural network architecture are shown in Figure 6 and Figure 7, respectively.

The neural network which is used for ensemble weighting requires an additional training set for weight optimization. Hence, we utilize the Gaussian noise added variant of the actual training set, as we do for Ensemble Elimination, and limit the training iteration count to 5 to avoid memorization.

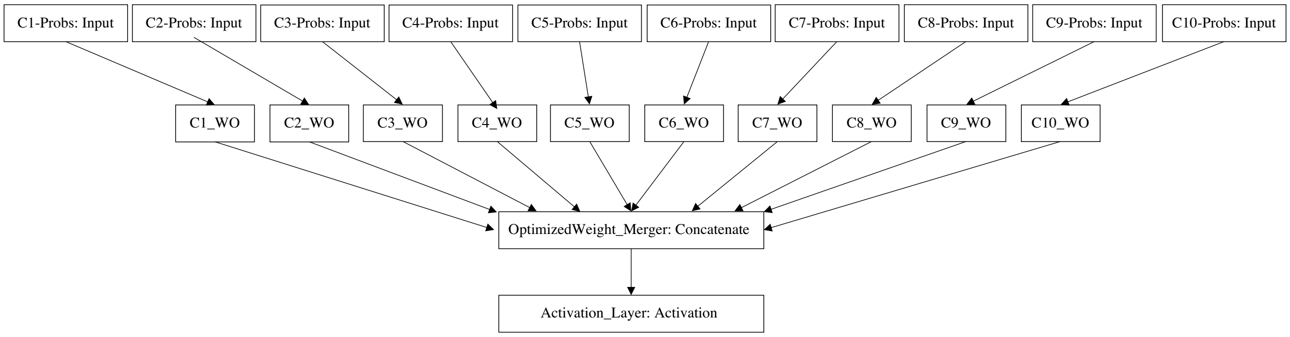


Figure 6. Modified neural network architecture.

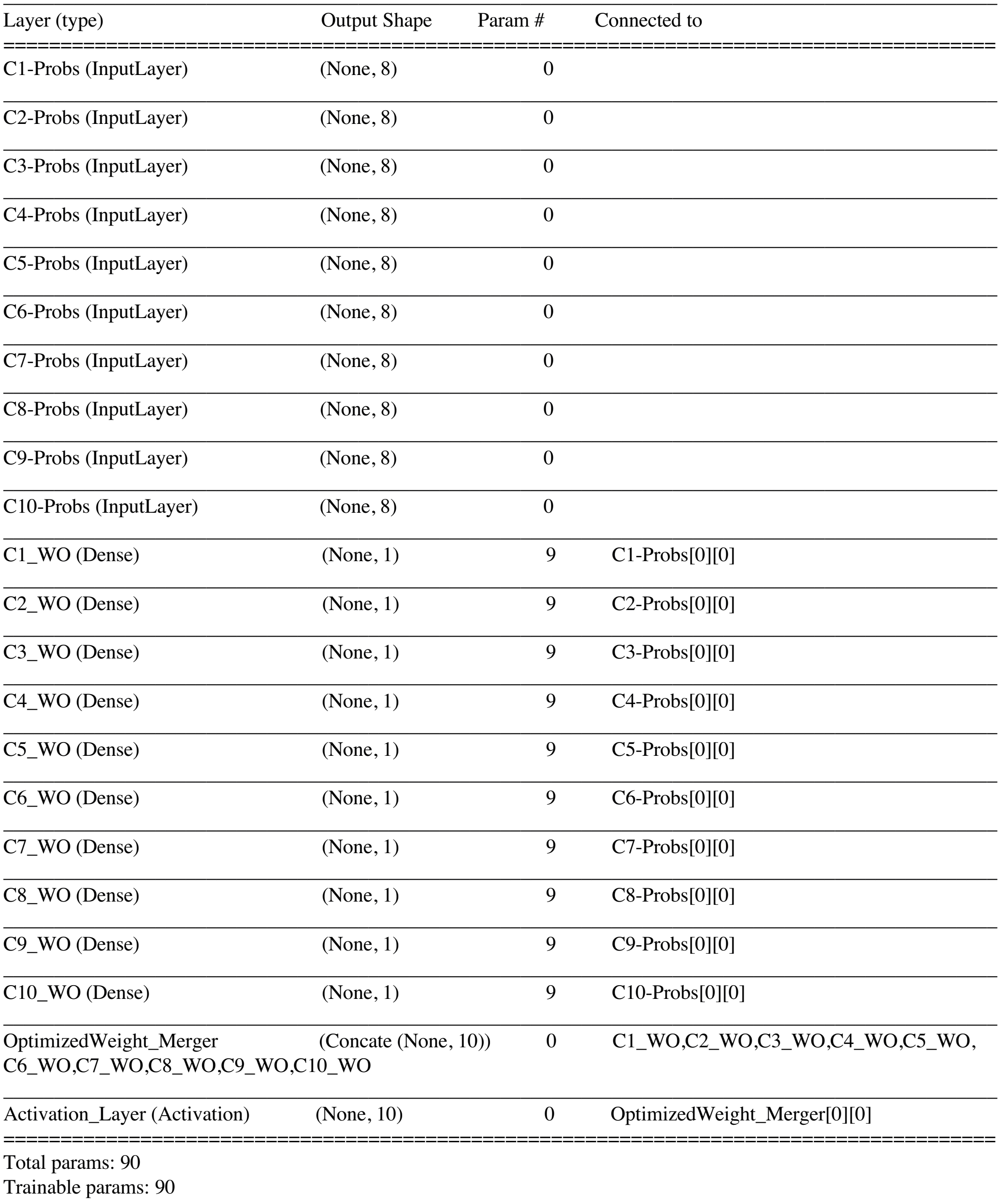
4. Experiments

To evaluate the performance of the proposed approach on a small dataset, we took the advantages of Cifar10 dataset which is frequently used in computer vision studies of deep learning literature. The original Cifar10 dataset includes 32x32 RGB images in balanced 10 different classes, 50000 of which are in training and 10000 are utilized for test sets. Although the Cifar10 dataset contains relatively fewer images compared to other general-purpose image datasets such as ImageNet, it is still a large set for a 10 class classification task and slightly affected by overfitting effects. Therefore, to reveal overfit effects, we created and used different-sized subsets of Cifar10 dataset by applying stratified random sampling. We decided subset sizes by measuring over fit effects with a pre-trained VGG16 model and accepted the sub (balanced) datasets, which cause more than 15 % training-test accuracy difference.

We conducted all experiments on GPU instances that are running on the Google Colab system and to minimize the nondeterministic behavior of the GPU job scheduler, we repeated training and test procedures 40 times and calculated mean classification accuracies. Besides ensemble-based ones, we also conducted experiments with individual neural networks with the following configurations to understand the effect of ensembling on this particular small dataset:

- Only the largest model: the model that includes the whole layers of VGG16 network.

- Only the most successful model: a model that is chosen from a heterogeneous ensemble of different sized models. The classification accuracy on Gaussian noise added variant of the training set was considered as success criteria.



```
________________________________________________________________________________________
Layer (type)                       Output Shape         Param #     Connected to
========================================================================================
C1-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C2-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C3-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C4-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C5-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C6-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C7-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C8-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C9-Probs (InputLayer)              (None, 8)                  0
________________________________________________________________________________________
C10-Probs (InputLayer)             (None, 8)                  0
________________________________________________________________________________________
C1_WO (Dense)                      (None, 1)                  9       C1-Probs[0][0]
________________________________________________________________________________________
C2_WO (Dense)                      (None, 1)                  9       C2-Probs[0][0]
________________________________________________________________________________________
C3_WO (Dense)                      (None, 1)                  9       C3-Probs[0][0]
________________________________________________________________________________________
C4_WO (Dense)                      (None, 1)                  9       C4-Probs[0][0]
________________________________________________________________________________________
C5_WO (Dense)                      (None, 1)                  9       C5-Probs[0][0]
________________________________________________________________________________________
C6_WO (Dense)                      (None, 1)                  9       C6-Probs[0][0]
________________________________________________________________________________________
C7_WO (Dense)                      (None, 1)                  9       C7-Probs[0][0]
________________________________________________________________________________________
C8_WO (Dense)                      (None, 1)                  9       C8-Probs[0][0]
________________________________________________________________________________________
C9_WO (Dense)                      (None, 1)                  9       C9-Probs[0][0]
________________________________________________________________________________________
C10_WO (Dense)                     (None, 1)                  9       C10-Probs[0][0]
________________________________________________________________________________________
OptimizedWeight_Merger             (Concate (None, 10))       0       C1_WO,C2_WO,C3_WO,C4_WO,C5_WO,
C6_WO,C7_WO,C8_WO,C9_WO,C10_WO
________________________________________________________________________________________
Activation_Layer (Activation)      (None, 10)                 0       OptimizedWeight_Merger[0][0]
========================================================================================
Total params: 90
Trainable params: 90
```

Figure 7. Layers of the modified NN architecture.

Heterogeneous neural network ensembles, which are created by transfer learning is one of the outcomes of this study. Therefore, we also compared the performances of homogeneous and heterogeneous CNN ensembles. The classification performances of the proposed method for various ensemble scoring approaches and the comparison of heterogeneous/homogeneous ensembles are shown in Table 1 and Table 2, respectively.

Table 1. Mean accuracy, maximum - minimum accuracy difference and standard deviation results for 40 different training-test operations.

		Only the largest model	Only the most successful model	Majority voting for ensemble	Majority voting for eliminated ensemble	Ensemble weighting with NN
750 Train 150 Test	Mean Accuracy	72.29 %	73.98 %	76.11 %	75.97 %	**76.15 %**
	Max – Min Accuracy difference	8.67 %	8.67 %	7.33 %	4.67 %	6.67 %
	Standard Deviation of accuracy	0.0417	0.0500	0.0300	0.0250	0.0267
3000 Train 600 Test	Mean Accuracy	81.18 %	81.07 %	83.22 %	**83.41 %**	83.36 %
	Max – Min Accuracy difference	1.92 %	2.08 %	1.34 %	0.98 %	1.26 %
	Standard Deviation of accuracy	0.0098	0.0120	0.0061	0.0063	0.0057

Table 2. Heterogeneous vs. homogeneous ensembles for 40 different training-test operations.

		Majority voting for ensemble	Majority voting for eliminated ensemble	Ensemble weighting with NN
Heterogeneous Ensemble 750 Train 150 Test	Mean Accuracy	76.11 %	75.97 %	**76.15 %**
	Max – Min Accuracy Difference	7.33 %	4.67 %	6.67 %
	Standard Deviation of Accuracy	0.0400	0.0667	0.0333
Homogeneous Ensemble 750 Train 150 Test	Mean Accuracy	75.71 %	**75.40 %**	75.69 %
	Max – Min Accuracy Difference	1.34 %	0.98 %	1.26 %
	Standard Deviation of Accuracy	0.0090	0.0136	0.0077
P Value Hom. vs. Het. Ensembles	Unpaired t-test	0.05	0.05	0.03
	Paired t-test	0.03	0.06	0.02

5. Performance evaluation of the ensemble scoring methods for different data set sizes

We investigated the performances of the aforementioned ensemble scoring methods for different dataset sizes via 5-folds cross-validation. According to the results, which are shown in Table 3, the contribution of the suggested scoring methods are not statistically significant (p values are greater than 0.05). However, when the training item count exceeds 3000, suggested "majority voting for eliminated ensemble" and "ensemble scoring via neural network weighting" produce better results than other approaches. Based on these results we think that one may select a different ensemble selection method depending on the dataset sizes. To reveal the contribution of different but similar CNN network architectures, we also tested the VGG19 based weight transfer instead of VGG16. To this end, we created an ensemble that consists VGG19 based subnetworks and obtained the ensemble's final outputs by employing "majority voting for eliminated ensemble."

Table 3. Cross-validation results of ensemble scoring methods on different-sized datasets. To express statistical significance, the P values of the paired t-tests are also shown for suggested scoring methods.

	750 Train **150 Test**	**1500 Train** **300 Test**	**3000 Train** **600 Test**	**6000 Train** **1200 Test**
Only the largest model	73.02 %	78.35 %	82.76 %	87.01 %
Only the most successful model	71.59 %	78.11 %	82.48 %	86.21 %
Majority voting for ensemble	74.70 %	80.02 %	83.73 %	87.23 %
Majority voting for eliminated ensemble	74.88 % (p-val. $>$ 0.05)	80.37 % (p-val. $>$ 0.05)	84.01 % (p-val. $>$ 0.05)	**87.86 %** **(p-val. = 0.02)**
Ensemble weighting with NN	74.62 % (p-val. $>$ 0.05)	80.37 % (p-val. $>$ 0.05)	**84.34 %** **(p-val. = 0.01)**	**87.96 %** **(p-val. = 0.02)**
Majority voting for eliminated ensemble (VGG-19)	73.77 %	80.24 %	82.73 %	86.14 %

The conventional machine learning methods still favorable for small datasets due to the performance degradation of deep learning variants on limited data. In addition to deep learning techniques, we conducted experiments on the same datasets with traditional techniques such that histogram of oriented gradients (HOG) [23] and local binary pattern (LBP) [24]. Those methods are commonly used as feature descriptors for pattern and scene recognition tasks [20–22] in the literature. We extracted HOG and LBP based features from our dataset and classified the images via Support vector machines (SVM).

For HOG features, we used 4x4 windows and for LBP features, we employed 4-pixel neighborhoods. To determine HOG, LBP and SVM parameters, we applied 5-folds cross-validation and selected the best parameter combinations. HOG and LBP based classification results are shown in Table 4. Well-known conventional texture/scene classification methods, HOG + SVM and LBP + SVM, are applied to the same randomly sampled (750 train / 150 test) Cifar10 subdataset, the 5 folds cross-validated and parameter optimized accuracy rates are below 45%.

6. Conclusion

In this study, we propose a new method that combines transfer and ensemble learning techniques to improve the classification performance of CNN models on small datasets. Our method directly focuses on small datasets and differs from other studies in generating transfer learning based ensembles, creating submodels, and calculating

Table 4. Cross-validated and parameter optimized classification results of conventional texture classification methods on subsampled (750 train 150 test) Cifar10 dataset.

	Standard Deviation of Classification Accuracy	Mean Classification Accuracy
Histogram of Oriented Gradients (HOG) + SVM	0.011	43.10 %
Local Binary Pattern (LBP) + SVM	0.017	21.10 %

ensemble score.

In this method, first, we create subnetworks by dividing an existing pre-trained VGG16 model into different sized submodels. Then, we combine the results of models in the ensemble by employing three different ensemble scoring methods. Besides introducing this new ensemble generation technique, we also propose new ensemble scoring approaches. We take the advantages of Cifar-10 dataset for performance evaluation. Although Cifar-10 dataset is a small one especially when it is compared with ImageNet, it is still a large dataset and does not considerably suffer from overfit effects. Thus, performance comparison tests of the different methods are conducted on randomly sampled 2 different Cifar-10 subdatasets. These 2 subsets are balanced and consist of 750 training / 150 test and 3000 training / 600 test images, respectively. Sizes of the generated subsets are decided empirically by checking the overfit effects and accuracy decreases. Performance results of the methods are shown in Table 1 and Table 2.

Conducted experiments revealed that the proposed method performs better classification accuracies than the classical one-model transfer learning approach. For ensemble-based transfer learning approaches, the proposed heterogeneous ensemble method exposes better results than the ensemble consisting of homogeneous models. Also performed t-tests validate the statistical significance of the contribution of our heterogeneous ensemble approach by providing p-values lower than 0.05.

Some of the existing studies in the literature [5,7,9] use an ensemble of different pre-trained network architectures to classify images. Unlike other studies, our model pruning-based heterogeneous ensemble creation approach requires only one pretrained transfer learning source to create ensembles. We also compared single and multiple source network-based ensembles. To this end, first, we created an ensemble that incorporates models from pretrained (and fine-tuned) VGG16, ResNet50, DenseNet121, MobileNetV2 then compared the cross-validation results with our single source model based ensemble approach. The average 5-folds cross-validation accuracy of the ensemble, which includes 4 different models is 72.62%, whereas our single source model based ensemble approach reaches 74.70% accuracy. According to these results, one might say that our ensemble approach produces better results than the ensemble of 4 different source models; however, it's expected that to get higher accuracies by using larger or more sophisticated pretrained model ensembles.

On the other hand, the contributions of the proposed ensemble scoring methods do not seem statistically significant (p values greater than 0.05) on very small datasets (750 training samples, 10 classes). However, for larger sets, they become more profitable and statistically significant (p values are lower than 0.05). The cross-validation results in Table 3 show that "majority voting for eliminated ensemble" and "Ensemble weighting with NN" approaches deliver better results than "direct majority voting" method, which is frequently used in literature and those two methods could be employed interchangeably.

Also, the deep learning methods still perform better on our subsampled Cifar-10 datasets than the conventional machine learning techniques. Therefore, we think that the proposed method would be adopted for the deep learning based image classification tasks on small datasets.

References

[1] McLaughlin N, Del Rincon JM, Miller P. Data-augmentation for reducing dataset bias in person re-identification. In 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2015: 1-6.

[2] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing systems, 2012; (25): 1097-1105.

[3] Dieleman S, Willett K, Dambre J. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly Notices of the Royal Astronomical Society, 2015; 450 (2): 1441-1459.

[4] Szegedy C, Liu W, Jia Y,Sermanet P, Reed S et. al. Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 1-9.

[5] Shin HC, Roth HR, Gao M, Lu L, Xu Z et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN architectures, dataset characteristics and transfer Learning. IEEE Transactions on Medical Imaging 2016; 35 (5): 1285-1298. doi: 10.1109/TMI.2016.2528162

[6] Razavian A, Azizpour H, Sullivan J, Carlsson S. CNN features off-the-shelf: An astounding baseline for recognition. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014: 512-519. doi: 10.1109/CVPRW.2014.131

[7] Kandaswamy C, Silva LM, Alexandre LA, Santos JM. Deep transfer learning ensemble for classification. Advances in Computational Intelligence. IWANN 2015. Lecture Notes in Computer Science, Springer, Cham, 2015; 9094: 335-348. doi: 10.1007/978-3-319-19258-1_29

[8] George D, Shen H, Huerta EA. Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO. ArXiv abs/1706.07446, 2017. doi: 10.1103/PhysRevD.97.101501

[9] Yu Y, Lin H, Meng J, Wei X, Guo H et al. Deep transfer learning for modality classification of medical images. Information, 2017; 8 (3): 91. doi: 10.3390/info8030091

[10] Maji D, Santara A, Mitra P, Sheet D, Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images. ArXiv abs/1603.04833, 2016.

[11] Yan Z, Zhang H, Piramuthu R, Jagadeesh V, DeCoste D et al. HD-CNN: Hierarchical deep convolutional neural network for large scale visual recognition. IEEE International Conference on Computer Vision (ICCV), 2015: 2740-2748. doi: 10.1109/ICCV.2015.314

[12] Pouyanfar S, Chen S. Semantic event detection using ensemble deep learning. IEEE International Symposium on Multimedia (ISM), 2016: 203-208. doi: 10.1109/ISM.2016.0048.

[13] Ng HW, Nguyen VD, Vonikakis V, Winkler S. Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning. ICMI '15 Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 2015: 443-449. doi: 10.1145/2818346.2830593

[14] Korzh O, Joaristi M, Serra E. Convolutional neural network ensemble fine-tuning for extended transfer learning. Big Data – BigData 2018 Lecture Notes in Computer Science, Springer, Cham 2018; 10968: 110-123. doi: 10.1007/978-3-319-94301-5_9

[15] Nguyen L, Lin D,Lin Z, Cao J. Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. IEEE International Symposium on Circuits and Systems (ISCAS) 2018; 1-5. doi: 10.1109/ISCAS.2018.8351550

[16] Cruz RMO, Sabourin R, Cavalcanti GDC, Ren TI. META-DES: A dynamic ensemble selection framework using meta-learning. Pattern Recognition, 2015; 48 (5): 1925–1935. doi: 10.1016/j.patcog.2014.12.003

[17] Baker B, Gupta O, Naik N, Raska R. Designing neural network architectures using reinforcement learning. ICLR'17: 5th International Conference on Learning Representations, 2017.

[18] Miikkulainen R, Liang J, Meyerson E, Rawal A, Fink D et al. Evolving deep neural networks. ArXiv abs/1707.07012v3, 2017.

[19] Xie L, Yuille A. Genetic CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 1388-1397. doi: 10.1109/ICCV.2017.154

[20] Song M, Guo P. Combining Local Binary Patterns for Scene Recognition. Journal of Software, 2014; 9: 203-210. doi: 10.4304/jsw.9.1.203-210

[21] Hu J, Guo P. Spatial local binary patterns for scene image classification. Technologies of Information and Telecommunications (SETIT), 2012: 326-330. doi: 10.1109/SETIT.2012.6481936

[22] Shakerdonyavi M, Shanbehzadeh J. Sarrafzadeh A. Large-Scale image retrieval using local binary patterns and iterative quantization. 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2015: 1-5. doi: 10.1109/DICTA.2015.7371276

[23] Dalal N, Triggs B. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005; 1: 886-893. doi: 10.1109/CVPR.2005.177

[24] Ojala T, Pietikäinen M, Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Proceedings of 12th International Conference on Pattern Recognition, 1994; 1: 582-585. doi: 10.1109/ICPR.1994.576366

[25] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv abs/1409.1556, 2015.

[26] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 770-778. doi: 10.1109/CVPR.2016.90

[27] Szegedy C, Vanhoucke V, Ioffe S. Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 2818-2826.

[28] Krizhevsky A,Ilya S, Hinton G. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 2012; 25: 1097-1105.

[29] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S et al. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015: 1-9. doi:10.1109/CVPR.2015.7298594

[30] Hashem S. Optimal linear combinations of neural networks. Neural networks : the official journal of the International Neural Network Society, 1997; 10 (4): 599-614. doi:10.1016/S0893-6080(96)00098-6

[31] Taherkhani A, Cosma G, McGinnity TM. AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. Neurocomputing, 2020; 404: 351-366. doi: 10.1016/j.neucom.2020.03.064

[32] He Z, Shao H, Zhong X, Zhao X. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. Knowledge-Based Systems, 2020; 207: 106396. doi: 10.1016/j.knosys.2020.106396

[33] Swati ZNK, Zhao Q, Kabir M, Ali F, Ali Z et al. Brain tumor classification for MR images using transfer learning and fine-tuning. Computerized Medical Imaging and Graphics 2019; 75: 34-46. doi: 10.1016/j.compmedimag.2019.05.001

[34] Moon WK, Lee Y, Ke H, Lee SH, Huang C et al. Computer-aided diagnosis of breast ultrasound images using ensemble learning from convolutional neural networks. Computer methods and programs in biomedicine, 2020, 190, 105361. doi: 10.1016/j.cmpb.2020.105361

[35] Sahinbas K, Catak FO. Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images. Data Science for COVID-19, 2021: 451–466. doi:10.1016/B978-0-12-824536-1.00003-4

[36] Li Y, Li K, Liu X, Wang Y, Zhang L. Lithium-ion battery capacity estimation—A pruned convolutional neural network approach assisted with transfer learning. Applied Energy 2021; 285: 116410. doi: 10.1016/j.apenergy.2020.116410

[37] Cooney C, Folli R, Coyle D. Optimizing layers improves CNN generalization and transfer learning for imagined speech decoding from EEG. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) 2019: 1311-1316. doi: 10.1109/SMC.2019.8914246

[38] Chandrarathne G, Thanikasalam K, Pinidiyaarachchi A. A Comprehensive Study on Deep Image Classification with Small Datasets. In: Zakaria Z, Ahmad R. (eds) Advances in Electronics Engineering. Lecture Notes in Electrical Engineering, Springer, Singapore 2020; 619: 93-106.