Computer Science and Information Systems 20(4):1503–1517 https://doi.org/10.2298/CSIS230223056A

# Ensemble of Top3 Prediction with Image Pixel Interval Method Using Deep Learning *

Abdulaziz Anorboev[1], Javokhir Musaev[1], Sarvinoz Anorboeva[1], Jeongkyu Hong[2]**, Yeong-Seok Seo[1], Ngoc Thanh Nguyen[3,4], and Dosam Hwang[1]

[1] Department of Computer Engineering, Yeungnam University
38541 Gyeongsan, South Korea
{abdulaziz.anorboyev, javokhirmuso, sarvinozanorboeva, dosamhwang}@gmail.com
[2] School of Electrical and Computer Engineering, University of Seoul,
Seoul, 02504, South Korea
jhong0301@uos.ac.kr
[3] Faculty of Information and Communication Technology, Wroclaw University of Science and Technology
50-370 Wroclaw, Poland
ngoc-thanh.nguyen@pwr.edu.pl
[4] Faculty of Information Technology, Nguyen Tat Thanh University
Ho Chi Minh 70000, Vietnam

**Abstract.** Computer vision (CV) has been successfully used in picture categorization applications in various fields, including medicine, production quality control, and transportation systems. CV models use an excessive number of photos to train potential models. Considering that image acquisition is typically expensive and time-consuming, in this study, we provide a multistep strategy to improve image categorization accuracy with less data. In the first stage, we constructed numerous datasets from a single dataset. Given that an image has pixels with values ranging from 0 to 255, the images were separated into pixel intervals based on the type of dataset. The pixel interval was split into two portions when the dataset was grayscale and five portions when it was composed of RGB images. Next, we trained the model using both the original and newly constructed datasets. Each image in the training process showed a non-identical prediction space, and we suggested using the top-three prediction probability ensemble technique. The top three predictions for the newly created images were combined with the corresponding probability for the original image. The results showed that learning patterns from each interval of pixels and ensembling the top three predictions significantly improve the performance and accuracy, and this strategy can be used with any model.

**Keywords:** Classification probability, model optimization, ensemble learning.

## 1. Introduction

With the advent of neural networks (NNs) and the advancements in deep learning (DL), computer vision (CV) is no longer limited to science-based fundamental experiments and is now increasingly being applied to practical applications. The automotive sector, IoT,

* This is an extended version of our INISTA 2022 conference paper [26]
** Corresponding author

healthcare, and finance are only a few among the industries that are increasingly using CV products as their main source of supply. In addition, there is a sharp increase in the demand for high- accuracy models. To satisfy this requirement, numerous studies have been conducted on DL and CV. Future advancements in this sector will be affected by a slight improvement in model accuracy. The evolution of DL has led to the development of numerous models and methods for each DL subfield to address various issues. To address the issues and close the gaps in literature, numerous algorithms and tools have been proposed. The first is ensemble learning, which enables DL models to exchange knowledge using ensemble approaches.

Numerous researchers [1] - [7] have demonstrated the advantages of DL ensembles. The models, data, and methods that enable the leveraging of the knowledge of ensemble units have been approached in various ways. The outcomes of the ensemble models have been presented in [8] - [20], among other domains, including medical, financial risk analysis, and oil price prediction . Improvements in the DL models can aid their evolution in the future.

To resolve DL classification issues, we present an image pixel interval power (IPIP) ensemble technique. We propose two sub-methods (IPDR and IPMR ), described by IPIP, to create new datasets from the original dataset utilized for DL classification tasks. We replicated the original dataset using the IPDR and replaced any pixel having value 127 or less with zero. Consequently, we obtained a second dataset for training. We duplicated the initial dataset to create the third dataset; however, this time, all pixel values higher than 127 were changed to zero. In the second stage, we trained three datasets using three models and their ensemble prediction outcomes. The outcomes of this strategy were favorable. The accuracy of the model improved from 98.84% to 99.11%. Similar to IPDR, IPMR uses more than two intervals to extract datasets from the original dataset and applies them to training.

This paper is an extended version of our conference paper [26]. The new methods consist of the following elements: the Sole-Top3 ensemble and Sum-Top3 ensemble are two unique ensemble methods for enhancing the precision of image classification models. This approach benefits from the fact that many image-classification methods can generate precise predictions for the top three classes of images. Each submodel that we trained was tailored to a certain subset of the classes. The top three prediction probabilities from each submodel were then combined to create one prediction for the main model. We tested our method on a number of image classification datasets and verified that it consistently outperformed the competing ensemble methods. The proposed method makes two key contributions to the literature. First, it can effectively combine the predictions of multiple models even when the models are trained on different datasets. Second, it can improve the accuracy of the image classification models without significantly increasing the training time or computational cost.

An improvement in the accuracy to nearly 100% is an important factor to highlight. It can be challenging to improve the results of various DL ensemble models once the accuracy has reached close to 100%. The prediction scope of an ensemble typically overlaps with that of the main model and does not permit improved accuracy. Using our approach, we partially resolve this issue and improve the outcomes of our models. This model is not able to interfere with the primary model's training ¬because it was developed independently and incorporates nearly all its knowledge, which is another appealing aspect.

This section provided background information on ensemble learning and an overview of the subject matter. The remainder of this paper is organized as follows. Information on relevant studies and issues related to ensemble learning is provided in Section 2. The challenges described in Section 2 are addressed in Section 3, which offers extensive justification for the proposed methodology. Detailed information about the dataset, base method, training configuration, evaluation metrics, experimental findings, and discussions is included in Section 4, along with the experiments and findings of the study. Section 5 concludes the paper with a few observations of the research as a whole, and suggests areas for further investigation.

## 2. Literature Review

Deep neural networks include multiple nonlinear hidden layers, leading to expressive models that learn complex relationships between inputs and outputs. With limited training data, many complex relationships involve noise, resulting in overfitting. The transfer learning of image pixel values and prediction probability-based methods have shown advantages in numerous research areas, such as image classification [21], natural language processing [22], speech recognition, and remote sensing. In recent years, several machine-learning [23], [24] and deep-learning models [29], including convolutional neural networks [25] and recurrent neural networks, have been evaluated for image classification tasks.

To address the limitations of conventional classification approaches, [26] we pro¬pose a novel ensemble learning paradigm with imbalanced data, which comprises three phases: data preprocessing , base classifier training, and an output ensemble. CNNs and deep residual networks are used as individual classifiers and random subspaces, respectively. To diversify the ensemble system in a simple and effective manner to further improve classification accuracy, ensemble and transfer learning were evaluated, as specified in [28], to transfer the learned weights from one individual classifier to another (i.e., CNNs). The generalization of the existing semi-supervised ensembles can be strongly affected by incorrect label esti¬mates produced by ensemble algorithms to train supervised base learners. [27] proposed cluster-based boosting, which is a multiclass classification algorithm with cluster regularization. In contrast to existing algorithms, cluster-based boosters and their base learners jointly perform cluster-based semi-supervised optimization.

Clustering ensembles and their applications in different disciplines have been the focus of numerous studies [30,31,32,33]. The clustering ensemble investigates the joint success of clustering models; however, it generates a large number of clustering partitions and combines them to obtain a better consensus function. Soares et al. [34] investigated the labeling of data in dense locations with overlapping classes and suggested a cluster-based boosting technique with cluster regularization for multiclass classification. Numerous studies have been conducted on semi-supervised clustering methods. One of them is [35], which uses the random subspace technique and constraint propagation approach for incremental semi-supervised clustering for high-dimensional data clustering. Furthermore, this method selects incremental ensemble members using newly proposed local and global cost functions.

The advantage of using the DL ensemble employing faster RCNN, TOOD, YOLOX , and cascade with Swin-Transformers in detecting intestinal parasite illnesses was presented by [36] and achieved better performance than each technique when used separately. Another recent study [37] integrated convolutional and bidirectional LSTM in a customized deep learning ensemble model for the time-series forecasting of COVID-19. The results obtained for several measures were remarkable. [38] improved COVID detection using CT scan images by combining the retrieved features from the CNN and ML algorithms. [39] used a DL ensemble comprising robustly optimized bidirectional encoder representations from transformers approach (RoBERTa), long short-term memory (LSTM), bidirectional long short-term memory (BiLSTM), and a gated recurrent unit (GRU) in sentiment analysis. The model outperformed state-of-the-art models in terms of performance.

During image preprocessing, visual data is captured as pixel arrays. Convolutions, which is the de facto deep learning operator for computer vision, are then used to process these pixel arrays. Convolutions attempt to relate to spatially distant concepts by explicitly modeling all concepts across all images, regardless of content, and taking into account all image pixels equally, regardless of relevance. The majority of the research under review left missing the ensemble of the top three prediction outcomes and knowledge of the image pixel interval variance, which could be useful for improving the classification accuracy of CNN models. To close this gap, we conduct this study to examine the outcomes of the aforementioned approach.

## 3. Proposed Methodology

### 3.1. Image Pixel Interval

In data pre-processing, we applied a method comprising two sub-methods that studies image pixel variance. The main contribution of our method is the use of datasets copied from the original dataset, except certain interval pixel values . The difference in the number of intervals enabled us to create an initial double representation of the main dataset and multiple representations of the main dataset. The IPDR is a simple double representation of the main dataset. For IPDR, we created two zero arrays of the same size as the main dataset. We used the MNIST and CIFAR-10 dataset in our experiments. For the MNIST dataset, we created two arrays of size 60000 x $32 \times 32 \times 1$, all filled with zeros. For the first subdataset, we selected pixel values from the main dataset that belonged only to the [0:127] interval, copied and pasted them at the same position in the image, and all other image pixels in the dataset were changed to zero. The second subdataset was built using the same method as the first subdataset, except that the pixel value interval for the second subdataset was [128:255]. All values higher than 127 were copied and pasted at appropriate positions in the image, and all other image pixels in the dataset were changed to zero (figure 1).

For the IPMR, we applied the same method as previously described; however, instead of two intervals, we used multiple intervals of 50 (i.e., [0:50], [51:100], [101:150], [151:200], and [201:255]) for the CIFAR-10 dataset, as shown in figure 2. Number of intervals depends on the type of dataset. During our experiments, we observed that to achieve high accuracy in training, the RGB channel images should be divided into five parts.

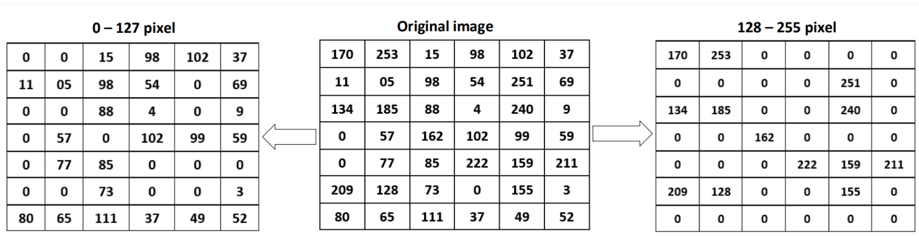


**Fig. 1.** Image pixel interval process for MNIST dataset

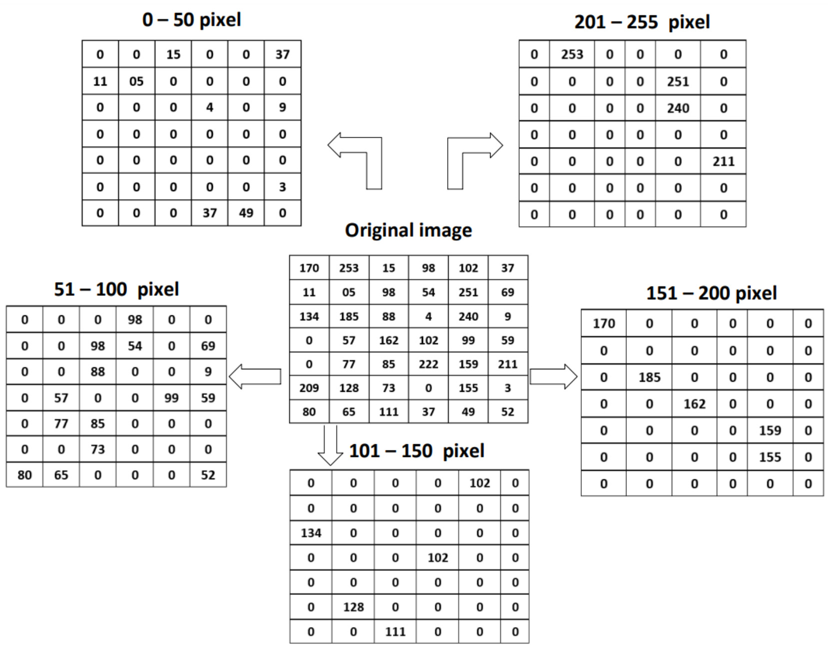


**Fig. 2.** Image pixel interval process for Cifar 10 dataset

### 3.2. Transfer Learning

Two model architectures with different numbers of parameters were used during the training process. The first model included 259,658 trainable parameters, as shown in Figure 3. The other proposed model had 165,514 trainable parameters.

We selected a model with a small number of parameters to avoid the overuse of time and power during IPIP implementation. However, the important aspect of our study was that we could achieve the same outcomes using a decreased-size model. The architecture for the regular model comprised three convolutional layers with 32, 64, and 128 filters

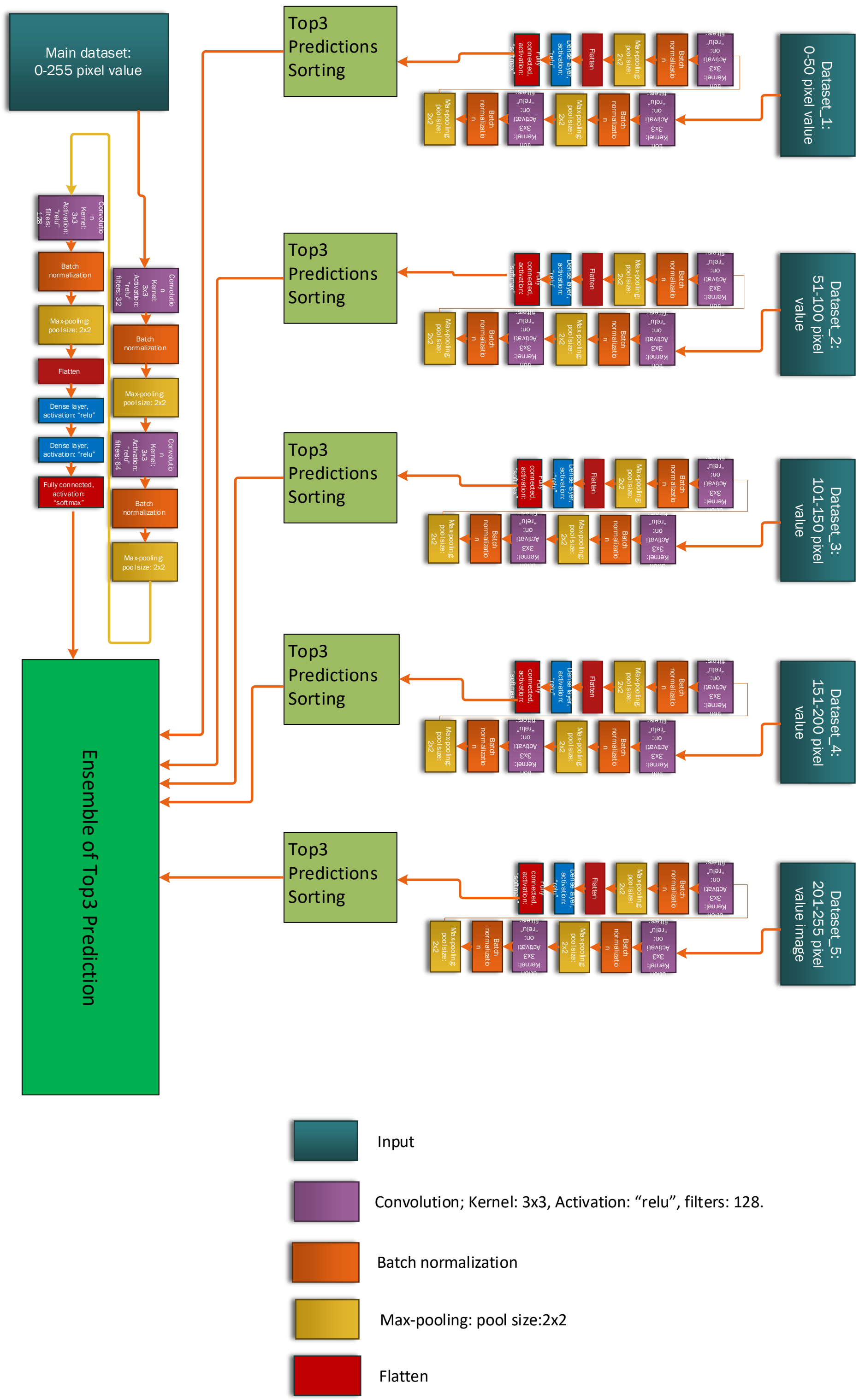


**Fig. 3.** Proposed model

and four dense layers with 256, 256, 128, and the number of class nodes for each dataset. In addition, we used max pooling with a $2 \times 2$ filter and batch normalization layers in both model architectures. The filter of size $3 \times 3$ was used for the convolutional layers in both the models. The architecture of the proposed model, as presented in Figure 3, comprises three convolutional layers with 32, 64, and 128 filters, and one fully connected layer in Part 1. The entire model in Part 1 was replicated by excluding the output layer from the new sequential model. We then iterated each layer in a new sequential model and set it as nontrainable. This froze the weights and other trainable parameters in each layer so that they would not be trained or updated. We added new fully connected layers and an output layer with a softmax activation function. The model was identical to the original model except that the number of trainable parameters decreased.

### 3.3. Prediction Probability Ensemble

In this section, we propose the top-three prediction probability ensemble. Each training dataset in our instance—three datasets each for the MNIST and Fashion MNIST datasets and six datasets for the CIFAR-10 dataset—represents a different prediction space. To create the main models that correspond to the prediction 5 probabilities, we ensembled the top three prediction probabilities of each image trained in the sub-model. The Sole-Top3 ensemble and Sum-Top3 ensemble were both used to test the ensemble method. The top three prediction probabilities from each submodel were combined with the corresponding prediction probabilities from the main model in the Sole-Top3 ensemble procedure. The top three prediction probabilities from each submodel were combined into the corresponding prediction probabilities for the main model in the Sum-Top3 ensemble. Equation 1 shows method of sorting prediction probabilities for each trained model. Here, k is the number of ensemble model indices, and k=1–5. Equation 2 shows the ensemble of each prediction probability from the submodels to the main model. Here, i is the number of the image in the datasets, that is, i= 0,1,2...m and j is the top three maximum probability indices, where j= 0,1,2.

$$b_k = np.argsort(A_k, axis = 0).sort(reverse = True). \tag{1}$$

$$A[b_k[:, i][j], i] += A_k[b_k[:, i][j], i]. \tag{2}$$

## 4. Experiments and Results

### 4.1. Datasets

We used two widely used grayscale and RGB color datasets in our proposed method: CIFAR-10, Fashion MNIST, and MNIST shown in Fig. 4, 5(a), and 5(b), respectively. The MNIST database of handwritten digits contains a training set of 60000 examples and a test set of 10000 examples. It is a subset of a larger set available from NIST . The digits were size normalized and centered on a fixed-size image.

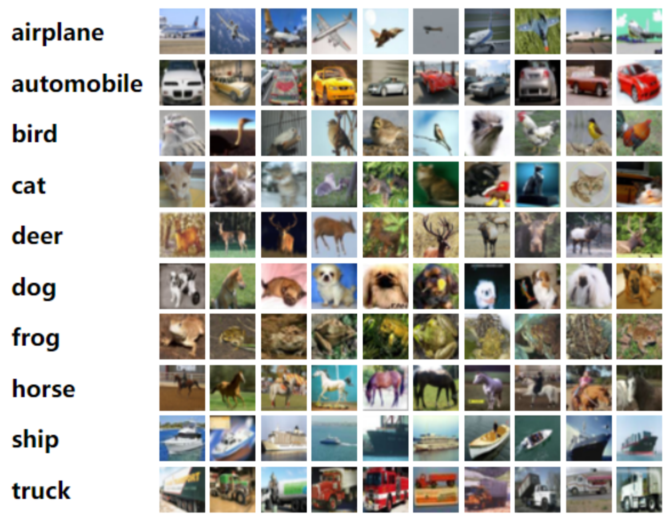


**Fig. 4.** Cifar 10 dataset

This is a database for people who want to test learning techniques and pattern 2recognition methods on real-world data while spending minimal effort on preprocessing and 3formatting. It contains 60000 training sets and 10000 test sets. The MNIST dataset was originally selected and tested by Chris Burges and Corinna Cortes. They used bounding box 5 normalization and centering. Yann LeCun's version used center of mass6 to center within a larger window. This is an extremely light-task DL model used to train the dataset. The 7CIFAR-10 dataset consists of 60000 color images of size 32 x 32 in 10 classes, with 6000 images8 per class. It includes 50000 training images and 10000 test images. The dataset was 9 divided into five training batches and one test batch, each containing 10000 images. The 10test batch contained exactly 1000 randomly selected images from each class. The training 11batches contained more images from one class than from another. The 12training batches contained exactly 5000 images from each class.

### 4.2. Experimental Setup

Python 3.6.12 and TensorFlow 2.1.0 were used to build the model architecture for the proposed method. For the experiments, we used a 12 GB Nvidia GeForce RTX 3080 Ti 16 with CUDA 10.2, on a computer with an AMD Ryzen 7 3700X and 64 GB of RAM. For training, the weight of the model was randomly initialized and trained for certain epochs. To train the model, we used the Adam optimizer with a default learning rate of 0.001 and 19 sparse categorical loss function. Each model was trained for 15 epochs. Results 20 from the 15th epoch were used to build a new ensemble and were evaluated using metrics addressed in this study.

### 4.3. Evaluation Metrics

For this study, we selected two metrics that meaningfully explained the method's achievements in different datasets. The accuracy is the ratio of true predictions to the total number

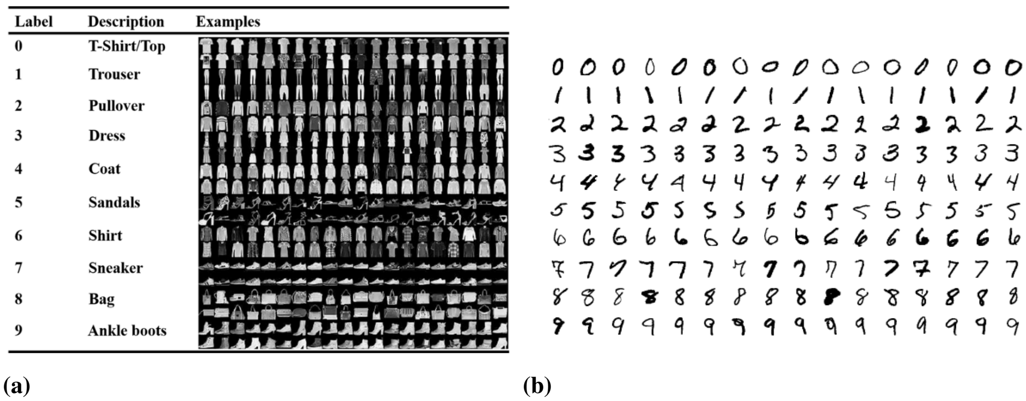


**Fig. 5.** Fashion MNIST (a) and MNIST (b) datasets

of cases used to evaluate the model. Equation 3 shows the accuracy of the calculations. The next evaluation metric was the UTP , which is the percentage of unique predictions for each model with respect to another model. In Equation 4, UTP(X, Y) determines the unique true predictions of Model X with respect to those of Model Y. These metrics explain why the proposed model achieved better results than the main model, for which only the main dataset was trained. The indices of the true predicted images were different for each model but had the same accuracy. This enabled the ensemble to achieve better results.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3}$$

$$UTP(X, Y) = X - X \cap Y \tag{4}$$

### 4.4. Experimental Results and Discussions

The main motivation for this study was to utilize knowledge from pixel-level variance by changing the data representation and achieve better accuracy metrics for the classification task. After changing the data representation, we proposed an ensemble of the outcomes of each subdataset from the top three predictions to the corresponding three predictions of the main model, as shown in fig. 6. Classic training was used as the baseline method for the experimental evaluation. Classic training was selected because of the difficulty in finding alternative methods that could be used to compare the results. Most DL ensemble models focus on model architectures and data representations by image-level preprocessing rather than on researching pixel-level variances. The main objective of this method is to apply it simultaneously to various combinations with many other DL ensembles. The training dataset was prepared by dividing it into different parts depending on the image. CIFAR-10 was divided into six types. The first copy of the dataset included all pixel values in the images from 0 to 255. The second dataset had pixel values between 0 and 50, and all other pixel values were set to zero. In all the other datasets, a certain interval

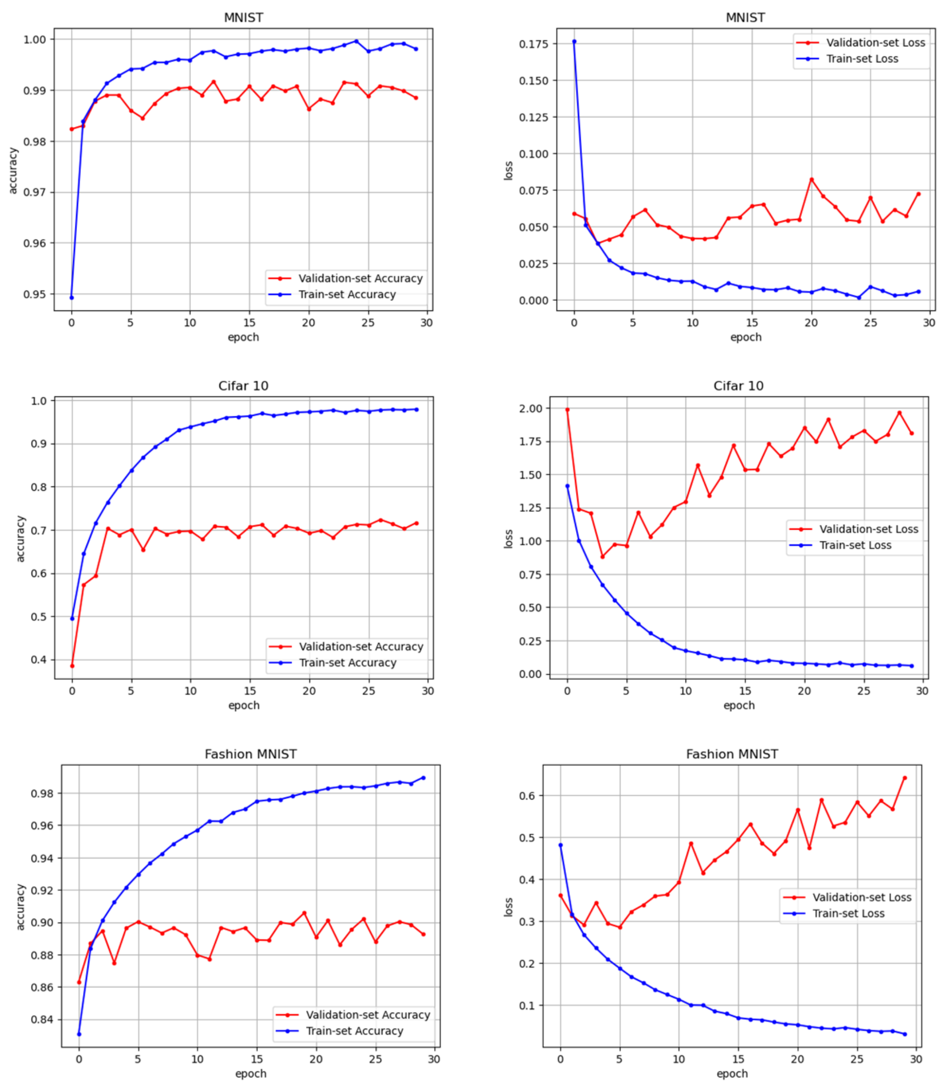


**Fig. 6.** Experimental results of accuracy and loss metrics using proposed model on the validation set and training set of the datasets: MNIST, Cifar10 and Fashion MNIST

of pixel values was maintained by ignoring the other values at intervals of 50, [51:100], [101:150], [151:200], and [201:255]. The primary purpose of the pixel interval method is to study and classify image pixels individually. After splitting the images, each dataset was individually trained using the specified CNN models described in Section 3. The experiments were conducted using the same method as for the MNIST dataset, although the data preprocessing was different. Because the MNIST dataset is grayscale, the image interval splitting of dataset consisted of two equal parts. The first part included pixel values from 0 to 127 and excluded all other values by equaling them to zero. The second part had pixel values between 128 and 255. All datasets were trained using the proposed fine-tuned model with 165,514 trainable parameters, whereas the original model used 259,658 trainable parameters. There was a reduction of approximately 100,000 parameters that saved computational power and memory and also helped to increase the training speed. The numbers were more visible when the model epochs were set to 100: 16,551,400 and (1) 25,965,800. By applying the proposed model, we achieved output results of 73.58% and 99.07% of the original model, for the CIFAR-10 and MNIST datasets, respectively (Table 1).

**Table 1.** Test set accuracy for MNIST, Fashion MNIST and CIFAR-10 datasets and proposed methods

| Dataset | Method | Accuracy Score |
|---|---|---|
| MNIST | IPIP | 98.90 |
| | Sole-Top3 | 99.01 |
| | Sum-Top3 | 99.07 |
| Cifar 10 | IPIP | 73.38 |
| | Sole-Top3 | 73.45 |
| | Sum-Top3 | 73.58 |
| Fashion MNIST | IPIP | 89.46 |

## 5. Conclusion

In this study, we proposed an enhanced version of the layer dropout technique and transfer learning to regularize deep neural networks. By applying transfer learning, we extracted features from the images before feeding them into the model. A fine-tuned convolutional neural network was applied by placing dropout operations before each convolutional layer. In addition to image classification, we tested the proposed method on a pixel-interval dataset. The proposed method could generalize deep neural networks with lower gradient descent and faster convergence during training and achieved promising results on classification tasks. Deactivation of the basic components of the neural network models discussed in this study are based on randomly setting a particular component to zero. The main disadvantage of this technique is the risk of losing useful information in the image. In future research, we will focus on further improving the pixel-interval image pre-processing method and we plan to combine with the analysis of user opinions in social media to improve the accuracy. We will use consensus methods for integrating the user opinions yielding a common opinion which can be taken into account for determining the prediction. Consensus methods have been proved to be very useful in many integration tasks and should be also useful in integrating user opinions [40, 41, 42].

**Acknowledgments.** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1074641).

## References

1. S. Vega-Pons and J. Ruiz-Shulcloper, “A survey of clustering ensemble algorithms,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 25, no. 3, pp. 337–372, May 2011.
2. Zhou, ZH. (2009). Ensemble Learning. In: Li, S.Z., Jain, A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA.
3. Lappalainen, H., Miskin, J.W. (2000). Ensemble Learning. In: Girolami, M. (eds) Advances in Independent Component Analysis. Perspectives in Neural Computing. Springer, London.
4. T. Alqurashi and W. Wang, “Clustering ensemble method,” International Journal of Machine Learning and Cybernetics, vol. 10, no. 6, pp. 1227–1246, Jun. 2019.
5. A. Krogh, “Neural Network Ensembles, Cross Validation, and Active Learning.”
6. O. Sagi and L. Rokach, “Ensemble learning: A survey,” Wiley Interdisciplinary Re¬views: Data Mining and Knowledge Discovery, vol. 8, no. 4. Wiley-Blackwell, Jul. 01, 2018.
7. X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” Frontiers of Computer Science, vol. 14, no. 2. Higher Education Press, pp. 241–258, Apr. 01, 2020.
8. G. I. Webb and Z. Zheng, ”Multistrategy ensemble learning: reducing error by combining ensemble learning techniques,” in IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 8, pp. 980-991, Aug. 2004.
9. K. Faceli, A. de Carvalho, M. Carlos, and P. de Souto, “Multi-objective clustering ensemble. Classical Weightless Neural Systems View project Feature Extraction and Se-lection Analysis in Biological Sequences View project SEE PROFILE,” 2007. [Online]. Available: https://www.researchgate.net/publication/220515994
10. H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, “A survey on ensemble learning for data stream classification,” ACM Computing Surveys, vol. 50, no. 2. Asso-ciation for Computing Machinery, Mar. 01, 2017.
11. S. Qummar et al., “A Deep Learning Ensemble Approach for Diabetic Retinopathy De-tection,” IEEE Access, vol. 7, pp. 150530–150539, 2019.
12. D. P. Gaikwad and R. C. Thool, “Intrusion detection system using bagging ensemble method of machine learning,” in Proceedings - 1st International Conference on Com-puting, Communication, Control and Automation, ICCUBEA 2015, Jul. 2015, pp. 291–295.
13. S. Hamori, M. Kawai, T. Kume, Y. Murakami, and C. Watanabe, “Ensemble Learning or Deep Learning? Application to Default Risk Analysis,” Journal of Risk and Financial Management, vol. 11, no. 1, p. 12, Mar. 2018.
14. Y. Zhao, J. Li, and L. Yu, “A deep learning ensemble approach for crude oil price fore-casting,” Energy Economics, vol. 66, pp. 9–16, Aug. 2017.
15. Z. Yu et al., “Incremental Semi-Supervised Clustering Ensemble for High Dimensional Data Clustering,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 3, pp. 701–714, Mar. 2016.
16. H. Sarmadi, A. Entezami, B. Saeedi Razavi, and K. V. Yuen, “Ensemble learning-based structural health monitoring by Mahalanobis distance metrics,” Structural Control and Health Monitoring, vol. 28, no. 2, Feb. 2021.

17. L. Yu, S. Wang, and K. K. Lai, “Credit risk assessment with a multistage neural network ensemble learning approach,” Expert Systems with Applications, vol. 34, no. 2, pp. 1434–1444, Feb. 2008.
18. Y. Xiao, J. Wu, Z. Lin, and X. Zhao, “A deep learning-based multi-model ensemble method for cancer prediction,” Computer Methods and Programs in Biomedicine, vol. 153, pp. 1–9, Jan. 2018.
19. T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, “The ensemble deep learning model for novel COVID-19 on CT images,” Applied Soft Computing, vol. 98, Jan. 2021.
20. A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez, “Multi-step forecasting for big data time series based on ensemble learning,” Knowledge-Based Systems, vol. 163, pp. 830–841, Jan. 2019.
21. D. Müller, I. Soto-Rey, and F. Kramer, “An Analysis on Ensemble Learning optimized Medical Image Classification with Deep Convolutional Neural Networks.”
22. B. T. Pham et al., “Ensemble modeling of landslide susceptibility using random subspace learner and different decision tree classifiers,” Geocarto International, 2020.
23. D. P. Gaikwad and R. C. Thool, “Intrusion detection system using bagging ensemble method of machine learning,” in Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015, Jul. 2015, pp. 291–295.
24. I. Tolstikhin et al., “MLP-Mixer: An all-MLP Architecture for Vision,” May 2021.
25. A. Krogh, “Neural Network Ensembles, Cross Validation, and Active Learning.”
26. A. Anorboev, M. Javokhir, J. Hong, N. T. Nguyen and D. Hwang, ”Input Image Pixel Interval method for Classification Using Transfer Learning,” 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 2022.
27. R. G. F. Soares, H. Chen, and X. Yao, “A Cluster-Based Semisupervised Ensemble for Multiclass Classification,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 1, no. 6, pp. 408–420, Dec. 2017.
28. Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, “Deep Learning Ensemble for Hyperspectral Image Classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.
29. F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” Oct. 2016.
30. K. Faceli, A. de Carvalho, M. Carlos, and P. de Souto, “Multi-objective clustering ensemble. Classical Weightless Neural Systems View project Feature Extraction and Selection Analysis in Biological Sequences View project SEE PROFILE,” 2007. [Online]. Available: https://www.researchgate.net/publication/220515994
31. T. Alqurashi and W. Wang, “Clustering ensemble method,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 1227–1246, Jun. 2019, doi: 10.1007/s13042-017-0756-7.
32. S. olah Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, “Clustering ensemble selection considering quality and diversity,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1311–1340, Aug. 2019, doi: 10.1007/s10462-018-9642-2.
33. S. Vega-Pons and J. Ruiz-Shulcloper, “A survey of clustering ensemble algorithms,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337–372, May 2011, doi: 10.1142/S0218001411008683.
34. R. G. F. Soares, H. Chen, and X. Yao, “A Cluster-Based Semisupervised Ensemble for Multiclass Classification,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 6, pp. 408–420, Dec. 2017, doi: 10.1109/TETCI.2017.2743219.
35. Z. Yu *et al*., “Incremental Semi-Supervised Clustering Ensemble for High Dimensional Data Clustering,” *IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 3, pp. 701–714, Mar. 2016, doi: 10.1109/TKDE.2015.2499200*.
36. J. Ruiz-Santaquiteria, A. Pedraza, N. Vallez, and A. Velasco, “Parasitic Egg Detection with a Deep Learning Ensemble,” IEEE Xplore, Oct. 01, 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9897858 (accessed Dec. 21, 2022).

37. S. Shastri, K. Singh, M. Deswal, S. Kumar, and V. Mansotra, “CoBiD-net: a tailored deep learning ensemble model for time series forecasting of covid-19,” Spatial Information Research, Jun. 2021, doi: 10.1007/s41324-021-00408-3.
38. Md. R. Islam and Md. Nahiduzzaman, “Complex features extraction with deep learning model for the detection of COVID19 from CT scan images using ensemble based machine learning approach,” Expert Systems with Applications, vol. 195, p. 116554, Jun. 2022, doi: 10.1016/j.eswa.2022.116554.
39. K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, “Sentiment Analysis With Ensemble Hybrid Deep Learning Model,” IEEE Access, vol. 10, pp. 103694–103704, 2022, doi: 10.1109/access.2022.3210182.
40. Nguyen N.T. (2006): Conflicts of Ontologies – Classification and Consensus-based Methods for Resolving. In: Proceedings of KES 2006, Lecture Notes in Artificial Intelligence 4252, 267-274
41. Nguyen N.T., Sobecki J. (2003): Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems. Journal of Universal Access in the Information Society 2(4), 342-358
42. Katarzyniak R., Nguyen N.T. (2000): Reconciling inconsistent profiles of agents’ knowledge states in distributed multi-agent systems using consensus methods. System Science 26(4), 93-119

**Abdulaziz Anorboev** is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Yeungnam University, South Korea. His research interests includes deep learning, computer vision and pattern recognition.

**Javokhir Musaev** is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Yeungnam University, South Korea. He has authored two conference papers and one patent. His research interests include computer vision and pattern recognition using deep learning techniques.

**Sarvinoz Anorboeva** is currently pursuing the M.S. degree with the Department of Computer Engineering, Yeungnam University, South Korea. Her research interests includes artificial intelligence, machine learning and computer vision.

**Dosam Hwang** received the Ph.D. degree in Kyoto University, Kyoto, Japan. He is a full professor of the Department of Computer Engineering at Yeungnam University in the Republic of Korea, whose research interests mainly include Natural Language Processing, Ontology, Knowledge Engineering, Information Retrieval, and Machine translation.

**Yeong-Seok Seo** was an Assistant Professor (Tenure Track) with the Department of Computer Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, Republic of Korea from September 2016 to August 2022, he. Since September 2022, he has been an Associate Professor (Tenure Track) with the same university. His research interests include software engineering, artificial intelligence, Internet of Things, and big data analysis.

**Ngoc Thanh Nguyen** (Senior Member, IEEE) is currently a Full Professor with the Wroclaw University of Science and Technology, and the Head of Information Systems Department, Faculty of Computer Science and Management. He is the author or coauthor of five monographs and more than 350 journal and conference papers. He has given 22 plenary and keynote speeches for international conferences, and more than 40 invited lectures in many countries. His research interests include collective intelligence, knowledge integration methods, inconsistent knowledge processing, and multi-agent systems.

**Jeongkyu Hong** received his PhD in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2017. From 2018 to 2023, he served as an Associate Professor in the Department of Computer Engineering at Yeungnam University, South Korea. Since 2023, he has been an Associate Professor at the School of Electrical and Computer Engineering, University of Seoul. His current research interests lie in the design of reliable and low-power embedded systems and software.

*Received: February 23, 2023; Accepted: July 20, 2023.*