An Image Pixel Interval Power (IPIP) Method Using Deep Learning Classification Models

Abdulaziz Anorboev[1], Javokhir Musaev[1], Jeongkyu Hong[1](✉), Ngoc Thanh Nguyen[2], and Dosam Hwang[1](✉)

[1] Yeungnam University, Daegu, Republic of Korea
jhong@yu.ac.kr, dos-amhwang@gmail.com
[2] Wroclaw University of Science and Technology, Wroclaw, Poland
Ngoc-Thanh.Nguyen@pwr.edu.pl

Abstract. The implementation of deep learning (DL) in various fields is becoming common. In addition, demand for higher accuracy models is increasing continuously at the same rates as the growth of other fields of science. Using all DL tools in the development of computer vision (CV) is a fundamental aspect of its future development. Considering all such tools, we conducted research on the effect of data representation on the final classification accuracy and proposed image pixels' double representation (IPDR) and image pixels' multiple representations (IPMR) for skipping certain pixels in the images in a dataset. Because the image pixel values range from 0 to 255, we proposed including all knowledge from different intervals of pixels. With IPDR, we trained the model using a dataset and obtained the prediction probabilities for the classification task. Next, we created two different datasets from an existing dataset. The first dataset took only image pixels of higher than 127, with all other image pixels in the dataset changed to zeros. The second dataset took only image pixels equal to or lower than 127. These two created datasets were trained on the same model architecture and their prediction accuracies for classification were ensembled with the prediction accuracies of the main model. With the IPMR method, we applied the same method as previously described, although instead of two intervals, from 0 to 127, and 127 to 255, we used, multiple intervals of 50 (i.e., [0:50], (50:100], (100:150], (150:200], and (200:255]) for the Cifar10 dataset. The number of intervals depends on the dataset, and applying our method, we achieved 89.46%, 98.90%, and 73, 38% accuracies on the Fashion MNIST, MNIST, and Cifar10 datasets, respectively, whereas their original classification accuracies under classic training were 89.27%, 98.65%, and 71.29%, respectively. As the advantage of using this method, it can be applied to any classification task and gives only extra knowledge on the trained data. As another simplicity of this method, it can be used with other DL ensemble models simultaneously.

Keywords: Image pixel double representation · Image pixel multiple representation · Model ensemble · Prediction scope

N. T. Nguyen et al. (Eds.): ACIIDS 2022, LNAI 13757, pp. 196–208, 2022.
https://doi.org/10.1007/978-3-031-21743-2_16

1 Introduction

Computer Vision (CV) has recently begun a new era with the development of DL and the introduction of neural networks (NNs). CV has moved from basic experiments in science and has started to implement its knowledge on real-life projects. Sectors such as the automobile industry, IoT, medicine, and finance, among many others, are becoming primary consumers of CV products. At the same time, the need for high accuracy models is growing dramatically. To meet this need, many studies are being conducted on DL and CV. Even a small increase in the model accuracy will affect future development of this field.

As the history of DL shows, various algorithms and models in each subfield of DL have been developed to solve various problems. Many algorithms and tools have been proposed to solve such problems and fill in the gaps in this field. One is ensemble learning, which allows DL models to share their knowledge through the use of ensemble methods.

The power of a DL ensemble has been presented by various researchers [1–7]. Different approaches have been applied to the models, data, and algorithms that allow utilizing the knowledge of ensemble units. The results of ensemble models were presented in [8–20], including medicine, financial risk assessment, and oil price prediction, among other fields.

Any positive change in DL models can help with future development. Hence, we proposed the Image Pixel Interval Power (IPIP) Ensemble method for DL classification tasks. We proposed two sub methods (IPDR and IPMR), which describes IPIP to make other datasets out of original dataset that is used for a DL classification task. Using IPDR, we copied the original dataset and replaced all pixels of lower than or equal to 127 with zeros. As a result, we obtained a second dataset for training. To create the third dataset, we copied the original dataset again, and this time, we replaced all pixel values that are higher than 127 with zeros. At the second step, we trained three datasets with three models and their ensemble prediction results. The result of this method was extremely positive. We increased the accuracy of the model, which was 98.84%, to 99.11%. IPMR uses more than two intervals to create datasets out of the original dataset and applies them in training, utilizing the same process as with IPDR. The key point to mention here is the increase in accuracy to close to 100%. In many DL ensemble models, it is difficult to achieve better results while the model has already achieved an accuracy of near 100%. The prediction scope of an ensemble is usually located in the prediction scope of the main model and does not allow increasing the accuracy. However, with our method, we partially solved this problem and obtained better results for our models.

Another attractive feature of this model is that it does not affect the training of the main model because it is trained separately and includes nearly all knowledge of the main model.

The structure of the work consists of the following. Section 1 introduces the paper, giving general information regarding ensemble learning and the content of the study. Section 2 provides information regarding the related studies and problems that have been found in ensemble learning. Section 3 provides solutions to the problems described in Sect. 2 and gives a detailed explanation of proposed methodology. Section 4 presents the experiments and results of our research, including information regarding the dataset, base method, training setup, evaluation metrics, experiment results, and discussions. Finally, Sect. 5 provides some concluding remarks regarding this research and areas of future study.

2 Related Works

An image pixel interval was one of the remaining topics in DL classification and ensemble models. The focus using an image pixel representation is applied in [21], where the authors recommend applied transformers for a computer vision task dividing images into patches. In [21], the authors achieved a state-of-art result in computer vision using one of the last methods in natural language processing. Another study [22] represented images as semantic visual tokens. The low-level features of the images were extracted using a convolutional neural network and used for further training in Visual Transformers [22]. The use of Visual Transformers outperformed the accuracy of ResNet on ImageNet. In addition, in [23], the authors introduced an MLP-mixer that proved there is no need for convolutional neural networks or an attention-based model. The MLP-mixer also used image patches as an input but included two types of layers: one MLP applied independently to each patch and another MLP applied across image patches. The other approach to our method was ensemble learning. We ensembled the prediction probabilities of the models. Reviewing ensemble learning, we found a gap in which most of the models in a DL ensemble used model ensembles to develop their methods. The majority of studies that varied the training data focused on changing the training data samples, batches, or methods of feeding the data into the model, rather than applying insight regarding the image pixels into the models. Surveys [1, 6, 7, 10, 23, 24] on ensemble learning have shown that the main studies have evaluated the effect of model ensembles and the combinations of whole datasets. Reviewing all of the abovementioned research, we found a gap in that none of the studies considered the effect of pixel variance on the model or its insight, which can add more power to the final data classification. This motivated us to study the power of image pixel intervals in DL classification tasks.

3 Proposed Methodology

To address the gap found in the literature review, we proposed the IPIP method, which studies image pixel variance and includes two sub methods: IPDR and IPMR. IPIP is described through the following steps:

- Creating datasets from the original dataset using IPDR or IPMR
- Training the main dataset with the model architecture
- Training the created datasets with a model chosen for these datasets
- Ensembling the prediction probabilities from each model
- Accepting the maximum predictions as the predicted class

IPIP is described using IPDR and IPMR. The main contribution of IPIP is to use datasets copied from the original dataset, leaving certain interval pixel values. The difference in the number of intervals encouraged us to make an initially double representation of the main dataset and multiple representation of the main dataset. IPDR is a simple double representation of the main dataset. With IPDR, we create two zero arrays (dataset_1 and dataset_2) with the same size as the main dataset. In our experiments, we used the MNIST dataset. For this dataset we created two arrays with a size of 60000 × 32 × 32 × 3 all filled with zeros. For dataset_1, we took only pixel values from the main dataset that belongs to the [0:127] interval and copied and pasted them at the same position in dataset_1. Dataset_2 was also built using the same method as dataset_1, except the pixel value interval for dataset_2 was (127:255]. All values higher than 127 were copied and pasted at the appropriate position in dataset_2. The rest of the training process is presented in Fig. 1.

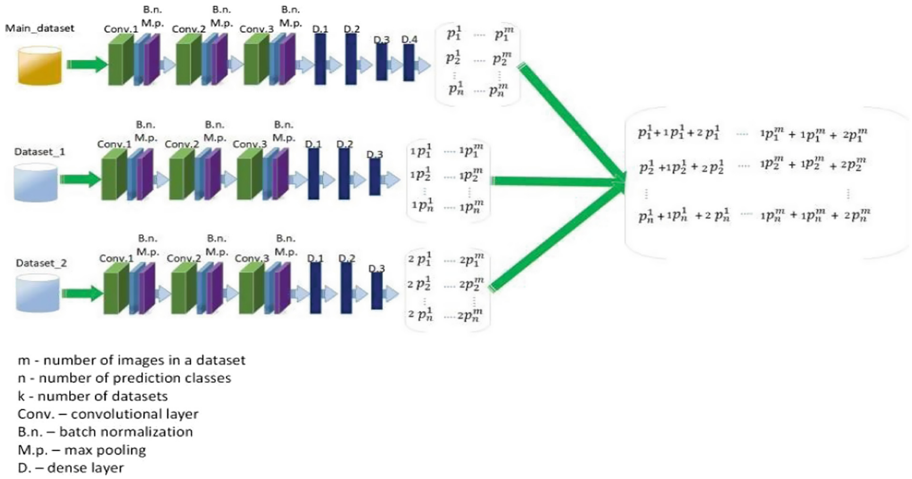

Fig. 1. IPIP: IPDR method architecture. p – prediction probability. The pattern from the right side is ensemble of prediction outcome.

As illustrated in Fig. 1, we started creating the datasets from main dataset. Dataset_1 and dataset_2 are the results of IPDR, which was used for further training of the model architecture shown in Figs. 3 and 4. The main dataset was trained in the architecture proposed in Fig. 3, whereas the other datasets created (dataset_1 and dataset_2) were trained using the model presented in Fig. 4. After training the models, we ensemble the prediction probabilities of each class with the corresponding probabilities from the two other models. Depending on the datasets applied, we use IPDRor IPMR to create datasets from the main dataset and train them using IPIP.IPMR shows the opportunity of creating more datasets and applying its knowledge on the classification task. We present the working scheme of IPIP with IPMR, which can use multiple intervals and n datasets for training, in Fig. 2. Then, the main dataset is trained in the model architecture shown in Fig. 3, and the other datasets are trained using the model architecture shown in Fig. 4. Their classification prediction probabilities were ensembled in the final step to achieve the final classification result. Finding the best number of intervals can be explained through the true prediction scope of each dataset, which adds extra knowledge to final model. We started with IPDR saving resources and decreasing the cost of training.

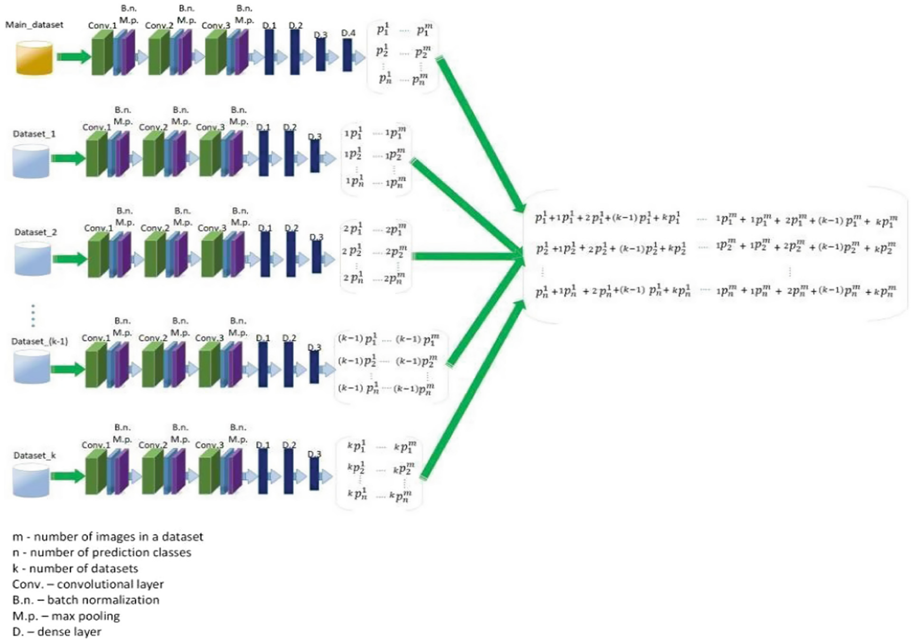

Fig. 2. IPIP: IPMR method architecture. p – prediction probability. The pattern from the right side is ensemble of prediction outcome.

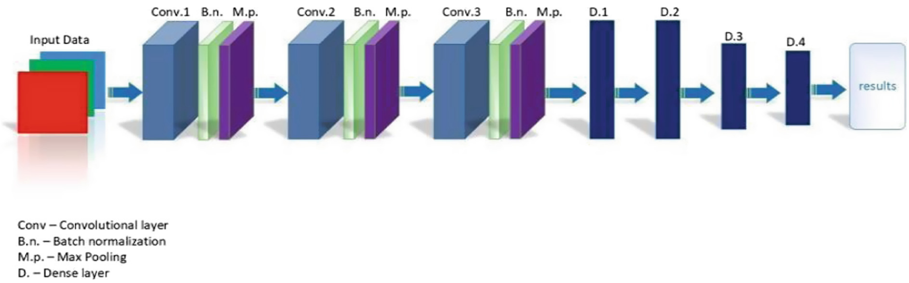

Fig. 3. The model architecture for main dataset.

If IPIP with IPDR does not return the expected result, training can be continued using IPIP with IPMR, where the models number more than two and can represent a difference in the prediction scopes. The goal of our proposed method was to find the optimal way to obtain better insight from the image pixels. This led us to start our training from IPIP with IPDR, whereas IPIP with IPMR requires training the model by spending more time and power.

As the next important point, we used two different model architectures with different numbers of parameters. The main model has 226,122 trainable parameters, as presented in Fig. 3, whereas the model for the other created datasets has 160,330 trainable parameters. We chose a smaller model for the created datasets to avoid an overuse of time

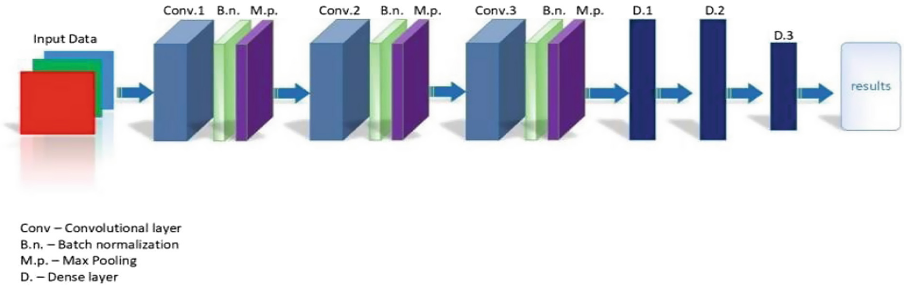

Fig. 4. The model architecture for created datasets.

and power during IPIP implementation. In general, the larger the model is, the higher the results achieved. The architecture for the main model, illustrated in Fig. 3, consists of three convolutional layers with 32, 64, and 128 filters, and four dense layers with 256, 256, 128, and the number of class nodes for each dataset, respectively. In addition, we used max pooling with a 2 × 2 filter and batch normalization layers in both model architectures. The filter used for the convolutional layers in both models had a size of 3 × 3. The architecture of the model, presented in Fig. 4, includes three convolutional layers with 32, 64, and 128 filters, and three dense layers, which have following numbers of nodes: 256, 128, and the number of classes in each dataset. The proof of the effect of our method is shown in the results described in the Unique True Prediction (UTP) metrics introduced in this study. Each model uses different datasets, and as shown in Figs. 1 and 2, classification probabilities from trained models are ensembled to achieve the final classification probabilities. Each model can predict a certain number of images that differ from the prediction of the other models. When we ensemble these predictions, the prediction area of the ensemble model includes some parts of the UTP of the models. In our study, to train the models and ensemble their prediction probabilities, we used different datasets that were created by applying IPIP with IPDR or IPMR.

4 Experiments and Results

In this section, we provide comprehensive information about the experiments and their results. This section also includes test results of the used model and dataset. Its comparison to our base model and the evaluation is the main part of this section.

4.1 Datasets

In our research, we used the Cifar10[1], MNIST, and Fashion MNIST datasets. The first dataset was Cifar10. This dataset is considered to be one of the most popular datasets available and is used in image classification tasks. Furthermore, Cifar10 includes a sufficient number of images to train and obtain reasonable results for classification models. The dataset is a subset of a dataset of 80 million tiny images, which were

[1] https://www.cs.toronto.edu/~kriz/cifar.html.

collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar10 consists of 32 × 32 × 3 sized 10 000 test and 50000 train images that belongs to 10 classes. In our training we did not preprocess the images. For our method we used Cifar10 and created other datasets out of Cifar10 by applying our method. For a simple model similar to our architecture with a small number of parameters, Cifar10 is a suitable choice to train the model and evaluate the results with the requirements of the proposed method.

The next dataset used in our research is the MNIST[2] dataset, which was developed by LeCun, Cortes, and Burges. The dataset consists of handwritten digits that belongs to 10 classes. It has 60,000 training sets and 10,000 test sets 28 × 28 greyscale images which represents digits from 0 to 9. The MNIST dataset was originally selected and experimented with by Burges and Cortes, who used bounding-box normalization and centering. LeCun's version uses centering based on center of mass within in a larger window. Extremely light-task DL models were used to train this dataset.

In addition, this dataset easily achieves a high accuracy with convolutional neural networks. This feature helped us during our experiments to check the effect of the proposed method against high accuracy models.

The next dataset is Fashion MNIST[3], which includes 60,000 training images and 10,000 test images. The dataset consists of Zalando's article images and has a size of 28 × 28. The number of images in the dataset is the same as that of the MNIST dataset, although this dataset includes 10 different classes of fashion.

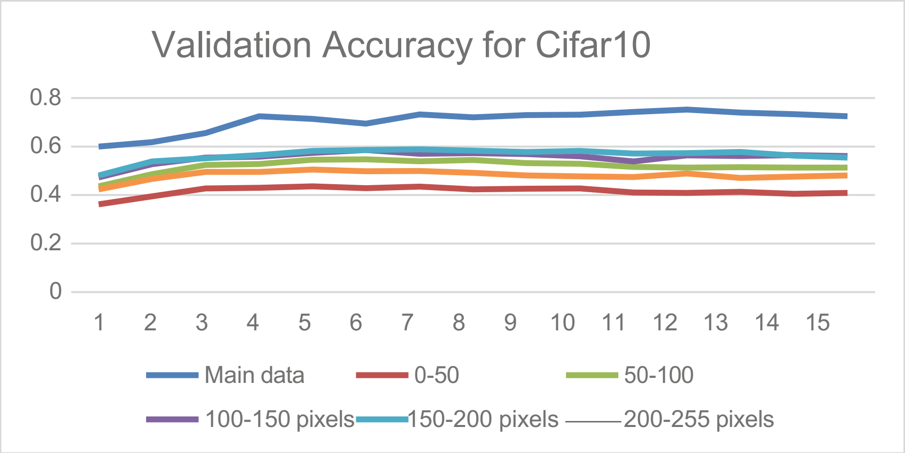

Fig. 5. Validation accuracy of Cifar10 and created datasets with certain pixel intervals.

4.2 Baseline Method

In our study, we used classic training as a baseline method for the experiment evaluations. The reason to choose classic training underlies the difficulty of finding an alternative

[2] http://yann.lecun.com/exdb/mnist/.

[3] https://www.kaggle.com/zalando-research/fashionmnist.

similar method that can be used to compare the results. Most of the DL ensemble models focus on model architectures and data representations through image level preprocessing in not researching the pixel level variances. The main objective of this method is to apply it in various combinations and with many other DL ensembles simultaneously. Another motivation for conducting this research was to utilize the knowledge from the pixel-level variance by changing the data representation and to achieve a better accuracy at the final classification step. As a result, we achieved better results on three different datasets like the MNIST, Fashion MNIST and Cifar10 datasets.

4.3 Training Setup

Python 3.6.12 and TensorFlow 2.1.0 were used to build the model architecture for the proposed method. For our experiments, we used a 12-GB Nvidia Titan-XP with CUDA 10.2 on a computer with an Intel core-i9 11900F CPU and 64 GB of RAM. For our training, we randomly initialized the weight for the model and trained for certain epochs. To train the model, we used an Adam optimizer with a default learning rate of 0.001 and sparse categorical loss function. The model was prepared for training three different datasets. Each model was set to train for 15 epochs. The results from the 15th epoch were used to build a new ensemble and were evaluated using the metrics focused on in this study.

4.4 Evaluation Metrics

We proposed a method that mainly focuses on the accuracy of the model. Many other studies include different metrics such as the F1 score, Recall, IoU, and ROC. For our research, we chose two metrics that meaningfully explains the achievements of the method on the different datasets. The accuracy is the ratio of true predictions to the total number of cases that are used to evaluate the model. Equation 1 shows the calculation of the accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

TP - true predicted positive results
TN - true predicted negative results
FP - false predicted positive results
FN - false predicted negative results

$$UTP(X, Y) = X - X \cap Y \tag{2}$$

UTP - Unique True Prediction
X - Prediction Scope of a model
X Y - Prediction Scope of a model Y

The next evaluation metric is UTP, which identifies the percentage of unique predictions for each model with respect to another model. In Eq. (2), UTP(X, Y) finds unique true predictions of model X with respect to model Y. This metric explains why our proposed model achieved better results than the main model where we trained only main dataset. The index of true predicted images differs for each model, even when they have the same accuracy. This leads the ensemble to achieve better results.

4.5 Experimental Results and Discussions

In this section of our study, we introduced experiment results and their detailed explanations. In addition, this study includes the validation accuracy and UTP as the evaluation metrics. The reason for choosing only these two metrics is because we only focused on an explanation of our study and for clarifying future studies in this field. We calculated the prediction scope and found the unique true prediction of each member of the ensemble with respect to the main model. Figure 5 shows the accuracy of the validation data for the Cifar10 dataset. Figure 6 shows the validation accuracy of the MNIST dataset, and Fig. 7 represents the validation accuracy for the Fashion MNIST dataset. Table 1 shows the test set accuracy of the main and ensemble models for each dataset. Table 2 shows the UTP of the model trained using the dataset created by the authors with respect to the model, which was trained with the original dataset. Because our ensembles have different numbers of ensemble members, our method achieved better results on each of the datasets.

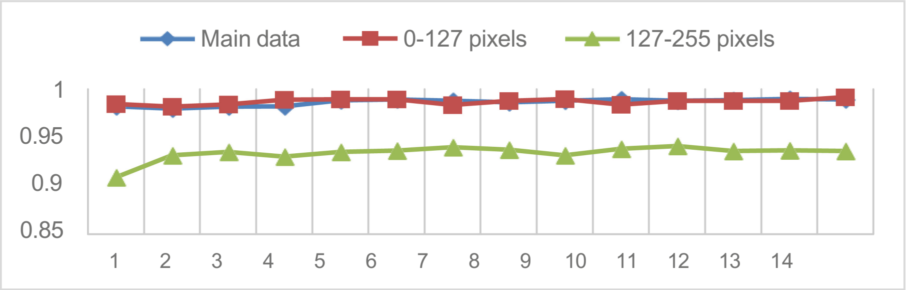

Fig. 6. Validation accuracy of MNIST dataset and created datasets with certain pixel intervals.

As mentioned above, the experiments started from copying the original dataset and ignoring some pixels that do not lie within a certain interval. For the Cifar10 dataset, we applied the same processes described in the methodology part of our study. As the main dataset, we chose the Cifar10 dataset and made five copies of it. Consequently, for the 0–50 interval dataset, we give a value of zero to all pixels that do not belong to the 0–50 pixel value interval. For the second copy of the dataset, we repeated the same processes applied for the first one except the interval was from 50 to 100 pixels. For the third copy, the same process was repeated using a 100–150 pixel value intervals. For the fourth copy of the original dataset, we take only 150–200 pixel values, making all other pixels zero. For the last copy of the dataset, we repeated the same process using 200–255 pixel value intervals. All six datasets were trained using the same model architecture, and their final prediction probabilities were ensembled. Figure 5 shows the validation accuracy of Cifar10 and the five newly created datasets from Cifar10. As the results show, there are no dramatic changes in each of the epochs. Increasing the accuracy of the model becomes a more difficult task when there is no high variance in the accuracies of the different epochs. In our model, Cifar10 achieved an accuracy of approximately 70% during each epoch. For the last epoch, 71.29% accuracy was achieved on the test set, and after applying our method, we obtained an accuracy of 73.38%. The next dataset is the MNIST dataset, which consisted of 28 × 28 greyscale images. From this dataset, we created two other datasets. The first one ignored all pixel values that are higher than 127. For this dataset, we replaced all pixel values that are higher than 127 to zeros. The second dataset was created using higher pixel values than 127.

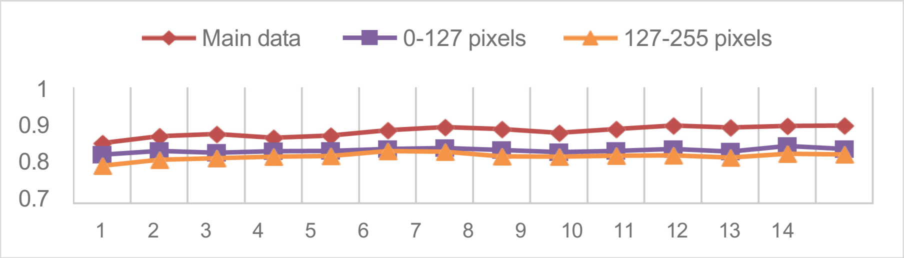

Fig. 7. Validation accuracy of Fashion MNIST dataset and two created datasets that has 0–127 and 127–255 pixel values.

Pixel values of less than 127 were replaced with zeros. The same model architecture was used to train the MNIST and two other created datasets. The validation accuracy of the MNIST dataset and two other datasets, which included pixel values of 0–127 and 127–255, respectively, are presented in Fig. 5. MNIST and 0–127 pixel-valued datasets have a very similar trend on the validation accuracy. The figure shows that the 0–127 interval includes more knowledge of the original dataset than any other part. By contrast, the 127–255 pixel-valued dataset has a lower validation accuracy. However, all three datasets reach an extremely high accuracy. Hence, increasing it was a challenging problem for us. When we ensemble the prediction accuracy of these three models trained on three datasets (MNIST, 0–127 and 127–255 pixel valued), we attained better accuracy than simply training the MNIST dataset on the utilized model architecture. Therefore, we achieved 98.90% accuracy using our method when the original model achieved an accuracy of 98.65%.

As Fig. 7 shows, the last dataset trained using the proposed method was Fashion MNIST. For this dataset, we used the same process as MNIST and created two other datasets out of Fashion MNIST. Table 1 shows the final results of our proposed method where we obtained better results on the three datasets. The results of classic training, using only the original dataset, and training using the model, were lower than the results of our method, which we note as an ensemble model. After applying our method, we

Table 1. Test set accuracy for each of the datasets and models

Datasets	Main model	Ensemble model
Cifar10	0.7129	0.7338
MNIST	0.9865	0.9890
Fashion MNIST	0.8927	0.8946

achieved 89.46% accuracy. The reason for achieving better results can be explained using Table 2.

Table 2. UTP (Secondary Models, Main Model)

Datasets	UTP
Cifar10	0.1097
MNIST	0.0069
Fashion MNIST	0.0353

The UTP of one model with respect to another returns a unique true prediction of first model, which shows a difference in the indexes of the true predicted image in the models. Although one model can achieve a lower accuracy, the other model that achieved a better accuracy cannot truly predict some of the true predicted images of the first model. This explains the increase in the ensemble of our method. Analyzing Table 2, we can see that for each of the datasets created, some extra prediction space can be added to the main model. In a perfect implementation of the method, final predictions can return better results by percent, as presented in Table 2. Considering the error after an ensemble that occurs when the classification probabilities are added, not all of the possible prediction space are added to the final model. This part of our work shows future studies that could be successful for this topic.

5 Conclusion

Utilizing the knowledge of image pixel variance gave our proposed model better results than simply using the whole image. We trained three different datasets and in each of them achieved better results after using the proposed approach. The reason for these increases was the difference in prediction scope of each model. We tried partially to solve the generalization problem of DL by using the variance ensemble of the image pixels. It is important to note that using any knowledge to increase the result of the model will help with the development of the DL field in all studies. In this research, we used the difference from the true prediction as a key metric that explains the growth of our results. In addition, choosing the best interval for the pixel values also adds more knowledge to the model. As presented in the UTP results, some of the models have a

10% different UTP, which provided the chance to increase the model by up to 10%. In our study, we used only a small part of this opportunity. Using all of this knowledge remains as a future area of our study.

The future works should concern in using ontology structures for storing and processing meta-knowledge about images. Such a knowledge base can be very useful in achieving higher accuracy and effectiveness [25–27].

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

References

1. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. Int. J. Pattern Recogn. Artif. Intell. **25**(3), 337–372 (2011)
2. Zhou, Z.H.: Ensemble learning. In: Li, S.Z., Jain, A. (eds.) Encyclopedia of Biometrics, pp. 270–273. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-73003-5_293
3. Lappalainen, H., Miskin, J.W.: Ensemble learning. In: Girolami, M. (ed.) Advances in Independent Component Analysis, pp. 75–92. Springer, London (2000). https://doi.org/10.1007/978-1-4471-0443-8_5
4. Alqurashi, T., Wang, W.: Clustering ensemble method. Int. J. Mach. Learn. Cybern. **10**(6), 1227–1246 (2018). https://doi.org/10.1007/s13042-017-0756-7
5. Krogh, A.: Neural Network Ensembles, Cross Validation, and Active Learning
6. Sagi, O., Rokach, L.: Ensemble learning: a survey. Wiley Interdisc. Rev.: Data Min. Knowl. Discov. **8**(4) (2018)
7. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Front. Comput. Sci. **14**(2), 241–258 (2019). https://doi.org/10.1007/s11704-019-8208-z
8. Webb, G.I., Zheng, Z.: Multistrategy ensemble learning: reducing error by combining ensemble learning techniques. IEEE Trans. Knowl. Data Eng. **16**(8), 980–991 (2004)
9. Faceli, K., de Carvalho, A., Carlos, M., de Souto, P.: Multi-objective clustering ensemble. Classical Weightless Neural Systems View project Feature Extraction and Selection Analysis in Biological Sequences View project See Profile (2007). https://www.researchgate.net/publication/220515994
10. Gomes, H.M., Barddal, J.P., Enembreck, A.F., Bifet, A.: A survey on ensemble learning for data stream classification. ACM Comput. Surv. **50**(2), 1–36 (2017)
11. Qummar, S., et al.: A deep learning ensemble approach for diabetic retinopathy detection. IEEE Access **7**, 150530–150539 (2019)
12. Gaikwad, D.P., Thool, R.C.: Intrusion detection system using bagging ensemble method of machine learning. In: Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015, pp. 291–295 (2015)
13. Hamori, S., Kawai, M., Kume, T., Murakami, Y., Watanabe, C.: Ensemble learning or deep learning? application to default risk analysis. J. Risk Financ. Manag. **11**(1), 12 (2018)
14. Zhao, Y., Li, J., Yu, L.: A deep learning ensemble approach for crude oil price forecasting. Energy Econ. **66**, 9–16 (2017)
15. Yu, Z., et al.: Incremental semi-supervised clustering ensemble for high dimensional data clustering. IEEE Trans. Knowl. Data Eng. **28**(3), 701–714 (2016)
16. Sarmadi, H., Entezami, A., Saeedi Razavi, B., Yuen, K.V.: Ensemble learning-based structural health monitoring by Mahalanobis distance metrics. Struct. Control Health Monit. **28**(2) (2021)

17. Yu, L., Wang, S., Lai, K.K.: Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst. Appl. **34**(2), 1434–1444 (2008)
18. Xiao, Y., Wu, J., Lin, Z., Zhao, X.: A deep learning-based multi-model ensemble method for cancer prediction. Comput. Methods Programs Biomed. **153**, 1–9 (2018)
19. Zhou, T., Lu, H., Yang, Z., Qiu, S., Huo, B., Dong, Y.: The ensemble deep learning model for novel COVID-19 on CT images. Appl. Soft Comput. **98** (2021)
20. Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F.: Multi-step forecasting for big data time series based on ensemble learning. Knowl.-Based Syst. **163**, 830–841 (2019)
21. Dosovitskiy, A., et al.: An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale (2020). http://arxiv.org/abs/2010.11929
22. Wu, B., et al.: Visual Transformers: Token-based Image Representation and Processing for Computer Vision (2020). http://arxiv.org/abs/2006.03677
23. Tolstikhin, I., et al.: MLP-Mixer: An all-MLP Architecture for Vision (2021). http://arxiv.org/abs/2105.01601
24. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. Inf. Fusion **37**, 132–156 (2017)
25. Duong, T.H., Nguyen, N.T., Jo, G.S.: A hybrid method for integrating multiple ontologies. Cybern. Syst. **40**(2), 123–145 (2009)
26. Duong, T.H., Nguyen, N.T., Jo, G.S.: A method for integration of WordNet-based ontologies using distance measures. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008. LNCS (LNAI), vol. 5177, pp. 210–219. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85563-7_31
27. Pietranik, M., Nguyen, N.T.: A multi-atrribute based framework for ontology aligning. Neurocomputing **146**, 276–290 (2014)