electronics

Article

A Novel on Conditional Min Pooling and Restructured Convolutional Neural Network

Jun Park [1], Jun-Yeong Kim [1], Jun-Ho Huh [2], Han-Sung Lee [3], Se-Hoon Jung [3,*] and Chun-Bo Sim [4,*]

[1] Interdisciplinary Program in IT-Bio Convergence System (BK 21 Plus), Sunchon National University, Suncheon 57922, Korea; j_park@scnu.ac.kr (J.P.); jy_kim@scnu.ac.kr (J.-Y.K.)

[2] Department of Data Science, (National) Korea Maritime and Ocean University, Busan 49112, Korea; 72networks@kmou.ac.kr

[3] School of Creative Convergence, Andong National University, Andong 36729, Korea; mohan@anu.ac.kr

[4] School of ICT Convergence Engineering, Sunchon National University, Suncheon 57922, Korea

* Correspondence: jungsh@anu.ac.kr (S.-H.J.); cbsim@scnu.ac.kr (C.-B.S.)

Abstract: There is no doubt that CNN has made remarkable technological developments as the core technology of computer vision, but the pooling technique used for CNN has its own issues. This study set out to solve the issues of the pooling technique by proposing conditional min pooling and a restructured convolutional neural network that improved the pooling structure to ensure efficient use of the conditional min pooling. Some Caltech 101 and crawling data were used to test the performance of the conditional min pooling and restructured convolutional neural network. The pooling performance test based on Caltech 101 increased in accuracy by 0.16~0.52% and decreased in loss by 19.98~28.71% compared with the old pooling technique. The restructured convolutional neural network did not have a big improvement in performance compared to the old algorithm, but it provided significant outcomes with similar performance results to the algorithm. This paper presents the results that the loss rate was reduced rather than the accuracy rate, and this result was achieved without the improvement of convolution.

Keywords: artificial intelligence; convolutional neural network; conditional min pooling; deep learning; pooling

Citation: Park, J.; Kim, J.-Y.; Huh, J.-H.; Lee, H.-S.; Jung, S.-H.; Sim, C.-B. A Novel on Conditional Min Pooling and Restructured Convolutional Neural Network. *Electronics* **2021**, *10*, 2407. https://doi.org/10.3390/electronics10192407

Academic Editor: Juan M. Corchado

Received: 6 August 2021
Accepted: 10 September 2021
Published: 2 October 2021

1. Introduction

Before deep learning, computer vision technologies used rule-based approaches that needed to find the characteristics of an object itself for programming. At the ILSVRC 2012 event held by ImageNet, deep learning-based AlexNet [1] showed an overwhelmingly greater performance than the rule-based approaches. Since then, there has been a shift in the direction of computer vision technologies from rule-based to deep-learning-based approaches. Deep learning not only boasts higher performance than old rule-based approaches, but it is also capable of learning for itself by finding important patterns and rules in large data without a need to program the characteristics of an object one by one. Deep learning received further attention as it became possible to do vision testing, which was not used in old rule-based approaches. Computer vision has advanced its performance through deep learning and developed sophisticated technologies such as object detection and segmentation [2,3].

The development of portable electronic devices with a built-in camera and the Internet has spread social media (SNS) and opened easy access to videos, which has created an environment where high quality video data is easily obtainable. This environment has accommodated the characteristics of deep learning and makes use of large volumes of data, promoting the massive development of deep learning. Advanced deep learning has been applied to a variety of industries, including manufacturing, medicine, fashion, and agriculture, and has made various achievements [4–9]. There is intensive research on deep learning as a core technology of unmanned systems, such as autonomous driving vehicles, autonomous flight drones, and smart factories [10–12].

Computer vision technologies are essential to computers or robots examining the current state based on functions such as human eyes, making a judgment for the next move by recognizing an object or figuring out a situation. Since computer vision is a core technology, it can cause huge accidents and loss of lives from just a small error, which is why computer vision requires a more precise performance [13–18].

A convolutional neural network (CNN), a type of deep learning that is mainly investigated in computer vision, can cause such an accident; thus, it has problems [19–26]. Pooling, a CNN structure, plays an important role in reducing the amount of computation and preventing overfitting by reducing the size of a feature map [5,27–31]. However, it can raise the following issues according to techniques: first, min pooling extracts a minimum value with a feature value on a feature map. When there is 0 within the same space, a feature itself can disappear, and noise can be detected as a characteristic; second, average pooling calculates the means between positive and negative features in the same space, thus offsetting features completely or making them blurry; third, max pooling is used most as it can extract clear features by choosing only the most powerful ones in the same space [32,33]. It can, however, make small or minute features disappear by extracting only the most powerful ones and is prone to overfitting by having the most powerful features [34–38].

The present study proposed conditional min pooling (CMP) and a restructured pooling structure for its efficient utilization to solve the problems of pooling techniques. This paper is organized as follows: Section 2 introduces pooling techniques and pooling structure to move the research forward; Section 3 offers explanations about proposed CMP and restructured pooling structure; Section 4 assesses the proposed technique in performance along with the old approaches; and Section 5 reaches conclusions.

2. Related Work

2.1. Pooling Method

Min et al. [39] adopts the Window method used by old pooling techniques to extract a feature map and then extract feature values based on probabilities without any particular conditions. It calculates the probabilities of feature values by dividing the entire feature value with feature values within a window and normalizing them. Feature values are randomly extracted according to their probabilities to extract the values of a feature map. Feature values that have more overlapping values are more likely to be extracted, which leads to a greater probability that meaningful values are extracted. Ian et al.'s [40] research employs a random approach to extract characteristics of a window. The test results based on CIFAR-100 and SVHN (Street View House Numbers) show reduced errors and increased accuracy. Zenglin et al. [41] solves such issues as characteristics being offset and reduced in volume when there have negative and positive features in average pooling. In an operational manner, the largest feature value in the Window is started with 1 and ranked in turn for feature values. Based on this rank, it calculates an average of feature values in Ranks 1~4 and extracts it as a final feature value.

2.2. CNN Model

A CNN model has received ongoing research efforts as a core technology of computer vision. AlexNet marks the beginning of CNN development. Armed with high performance, AlexNet played a leading role in a shift from the old rule-based approaches to deep learning-based approaches in the development of computer vision. AlexNet was the first model that used a GPU. As it used two GPUs in parallel, it divided one into two parts to process a layer in parallel.

Meanwhile, ResNet [42] solved one of CNN issues, which involved a vanishing gradient despite the improved performance according to deeper layers. It succeeded in training a model with a total of 156 layers. A residual block adds a skip connection structure to add input values to output values. It keeps the old learning information by calculating the residuals of input and output values, enabling additional learning.

DenseNet [43] added a new concept of dense connectivity to ResNet. Dense connectivity connects a layer of the former half to a layer of the latter half, enabling additional learning in the long run. Unlike ResNet, it adds a characteristic as a channel instead of residual-based learning. DenseNet was organized to keep the characteristics of the former half over the long term and process the back propagation of errors more efficiently.

3. Design of Proposed CMP

In this study, the design of a proposed pooling structure was depicted for the CMP proposed and its efficient utilization. Figure 1 shows the overall block diagram of the proposed pooling structure.

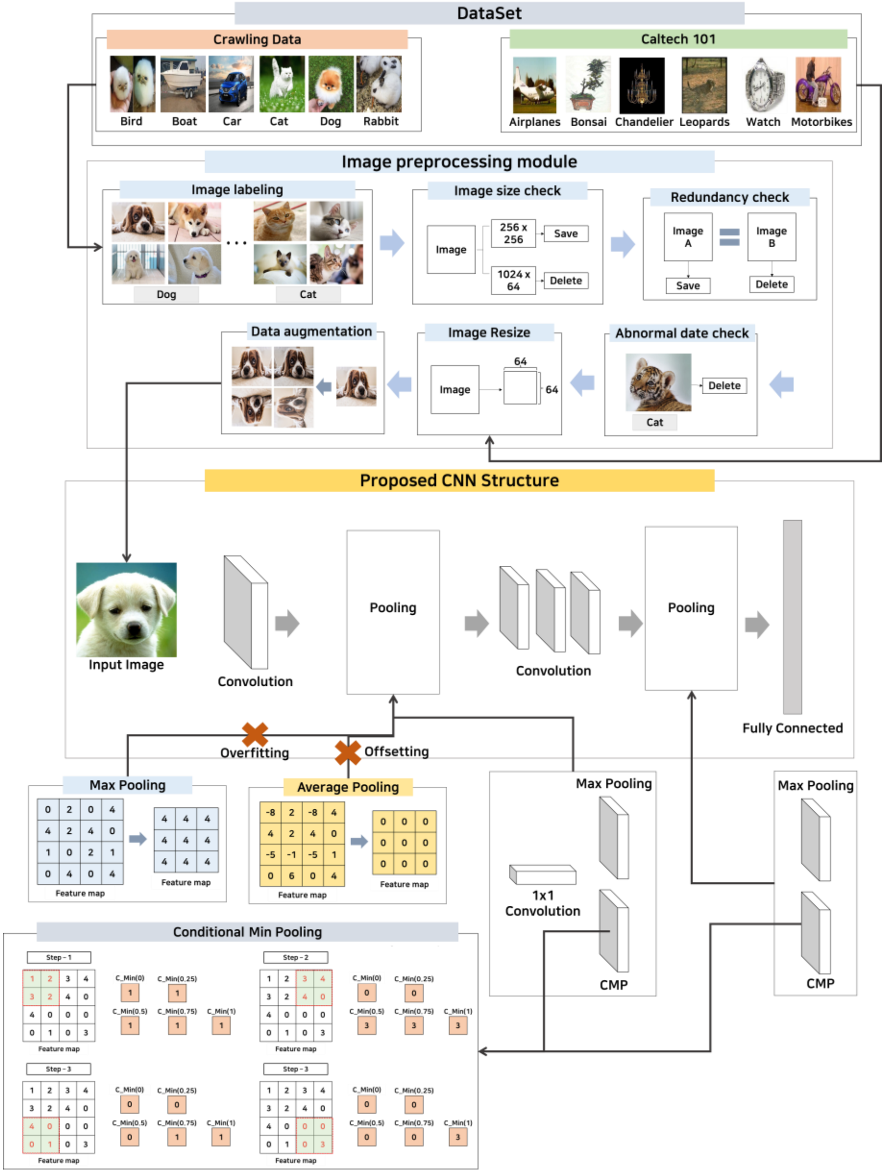


Figure 1. Overall structure of proposed pooling structure.

The dataset is crawling data and Caltech 101. Crawling data go through the entire process of the image preprocessing module, and Caltech 101 data go through image resize and data augmentation. The data passed through the image preprocessing module is delivered to the CNN input. The CNN that receives the image extracts a feature map using convolution and pooling to proceed with classification. Pooling uses the proposed pooling structure and CMP because it can cause overfitting and offsetting when max pooling and average pooling are used.

3.1. Data Pre-Processing Module

For the performance evaluation of a model, the study used Caltech 101 data and image data collected through crawling. There should be a preprocessing process fit for a model to use image data [44–46]. The preprocessing of crawling image data happened in the process of Figure 2. The data preprocessing module followed this order: first, images collected through crawling were labeled around the keywords used in searches; second, collected image data were checked for a horizontal length:vertical length ratio, and images of big ratio differences were removed; third, images of the same size were selected to check image redundancy. The images were then converted on Grayscale and compared in structure. Images of the same structure were removed except for one; fourth, an agreement between images and labels was assessed through manual work. Each image was assessed for objects of two keywords or more used in data collection. Images with such objects were eliminated; fifth, images were converted in the same size for the learning and testing of a CNN model; sixth, data were enlarged through the rotation and distortion of images to increase the amounts of image data.

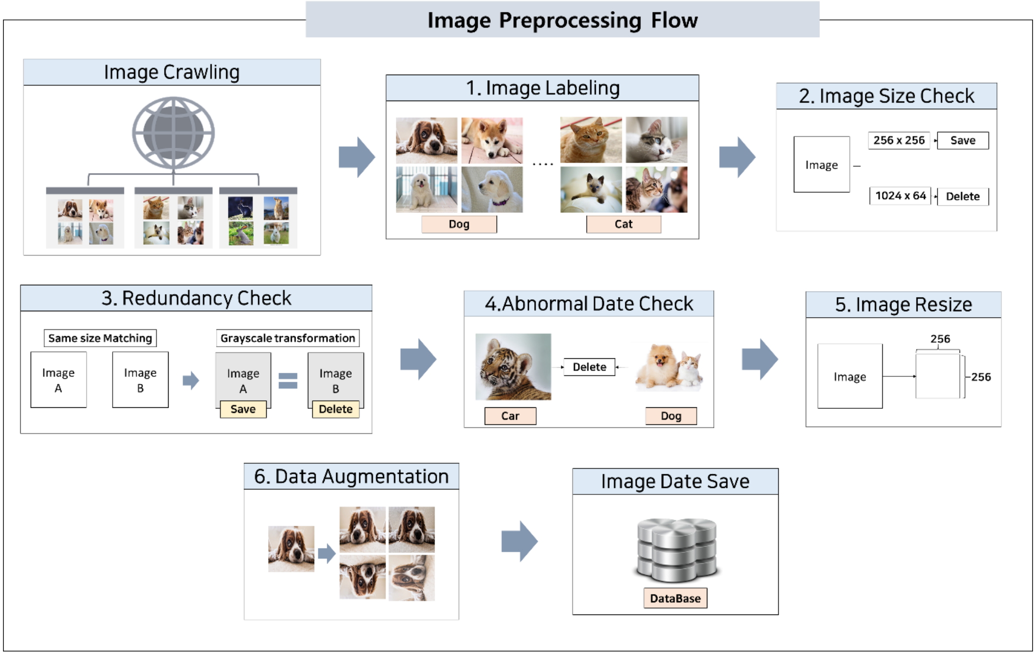


Figure 2. Flow chart of image data pre-processing module.

3.2. Design of CMP

The CMP was designed based on min pooling for its operation. The old min pooling technique extracts a minimum value with a representative value of a window. It can, thus, remove many characteristics when there is a feature value of 0 nearby. In this study, a CMP was proposed that statistically restricted the process of extracting feature values in min pooling.

CMP extracts a minimum value as a feature value just such as min pooling when there is no 0 in a window. When there is a 0 in a window as a feature value, however, it figures out the number of 0s in a window. The number of 0s is subjected to a constraint according to the percentage of 0 tolerance (0~1) given as a hyperparameter. When there are as many 0s in a window as the tolerance percentage, 0 is extracted as a feature value. When there are not as many 0s as the tolerance, the minimum value except for 0 is extracted as a feature value. CMP works in the same way as min pooling when the tolerance percentage of 0 is 0. In case of 0.5, 0 is extracted as a feature value when 0s account for more than half in a window. In the case of 1, 0 is extracted as a feature value when all the feature values of a window is 0. Figure 3 shows how the CMP works when the tolerance percentage are 0, 0.25, 0.5, 0.75, and 1 in a window of 2 × 2 with Stride 2.

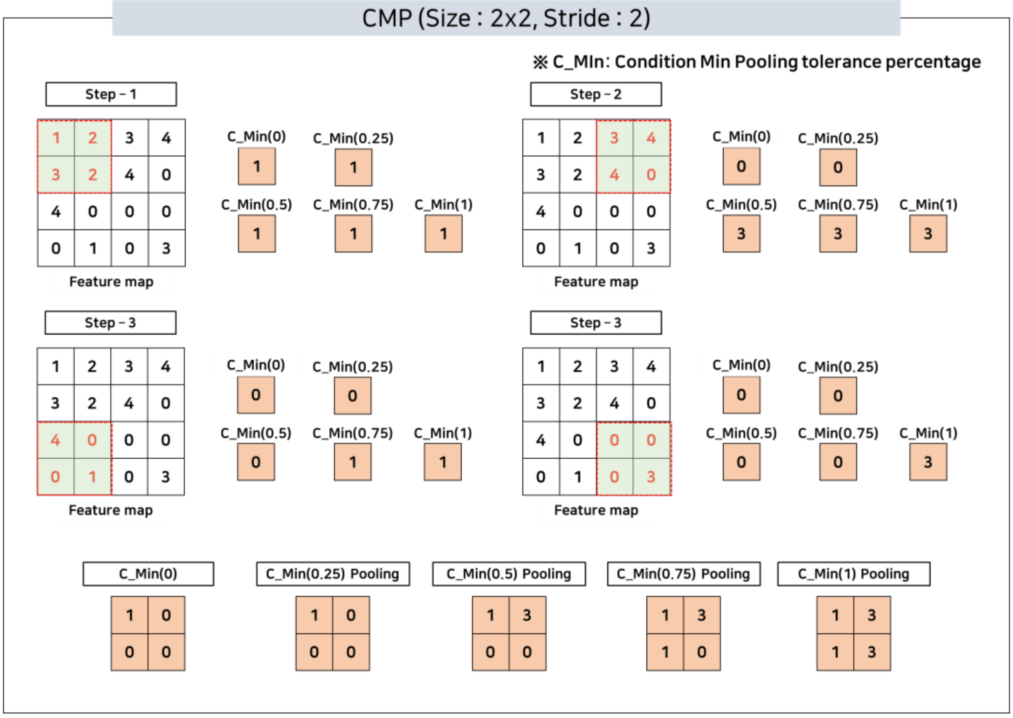


Figure 3. Proposed pooling structure of method.

3.3. Design of Neural Network Structure

The proposed pooling structure has srestructured the pooling structure to ensure the more efficient utilization of CMP. Figure 4 shows the proposed pooling structure, which uses a convolution of 1 × 1 to reduce the number of channels by half before a feature map passes through a restructured pooling layer. The restructured pooling layer was organized in two steps to make use of two pooling techniques with max pooling and CMP applied to be combined in a feature map. After three convolution layers, it passes through max pooling and CMP in a restructured pooling structure once again. Two feature maps identified through pooling layers combine the channels in a feature map without reducing their number and send them to a fully connected layer. The detailed structure of the neural network is shown in Table 1.

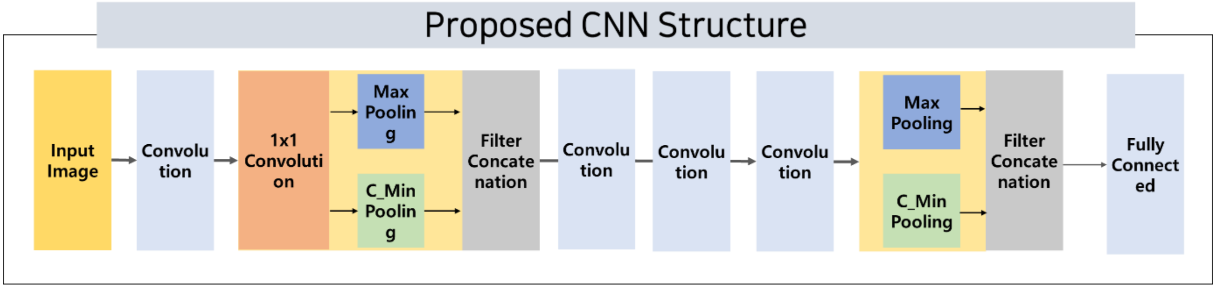


Figure 4. Structure of proposed restructured CNN.

Table 1. Detail structure of proposed restructured CNN.

Parameter	Input Size	Output Size
Input	(64, 64, 3)	-
Convolution	(64, 64, 3)	(64, 64, 64)
Proposed Pooling Structure	(64, 64, 64)	(32, 32, 128)
Convolution	(32, 32, 64)	(32, 32, 128)
Convolution	(32, 32, 128)	(32, 32, 256)
Convolution	(32, 32, 256)	(31, 31, 512)
Proposed Polling Structure	(31, 31, 512)	(15, 15, 1024)
Flatten	(15, 15, 1024)	(230, 400)
FCN	(230, 400)	(1, 1, 7) or (1, 1, 6)
Softmax	(1, 1, 7) or (1, 1, 6)	(1, 1, 7) or (1, 1, 6)

4. Performance Evaluation of CMP

4.1. System Implementation Environment and Performance Evaluation Method

The algorithm proposed in the study was designed, implemented, and assessed in performance in the environment of Table 2.

Table 2. Development environment.

Part	Detail
OS	Windows 10
CPU	Intel Core i7-9700
RAM	32GB
GPU	Geforce RTX 2080 Super
Language	Python 3.6
IDE	Pycharm Community 2020.1.2
Library	Tensorflow 1.14.0, Keras2.3.1

Accuracy rates were used to assess the proposed pooling structure in performance and to compare it with the old models in performance. In performance evaluation, accuracy rates were calculated with the percentage of data that made the right prediction in the entire data. Equation (1) shows the calculation.

$$Accuracy = \frac{Ture\ Pastive + Ture\ Negative}{Ture\ Pastive + Ture\ Negative + False\ Pastive + False\ Negative}. \quad (1)$$

4.2. Data Set

Caltech 101 data and crawling data were used to assess CMP and a restructured neural network in performance. Caltech 101 is public data provided by the University of California, consisting of 9146 images in total and 101 categories. There are huge differences in the amount of data among the categories from the minimum 31 to maximum 800. For model learning, 12 categories were selected which contained 100 images or more. Of them, seven categories were used for their data after five were excluded for similar or black and white images. These seven categories are airplanes, motorbikes, faces, watches, leopards, Bonsai, and chandeliers. Figure 5 shows some data of Caltech 101. Table 3 shows the current organization of data before the application of data augmentation, which was applied to increase the amounts of data by ten times and use them in learning and testing.

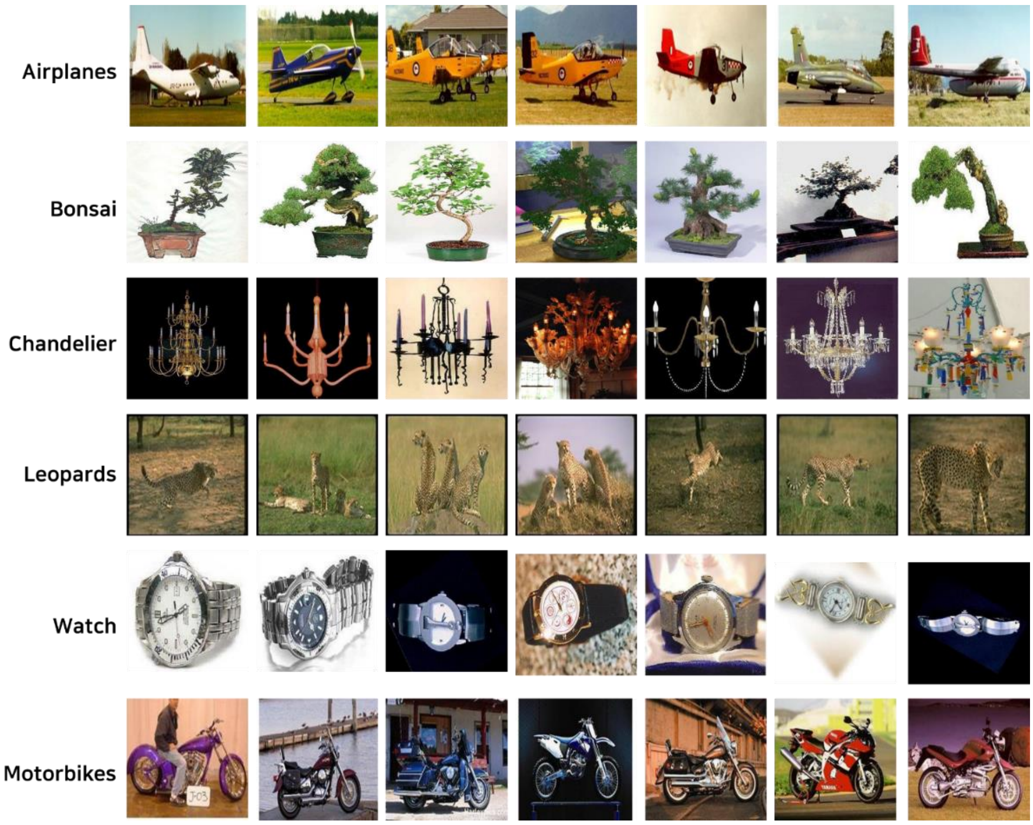


Figure 5. Caltech 101 data set.

Table 3. Configuration of Caltech 101 data set.

Category	Volume	Image Average Size	Image Average Capacity
Airplanes	800	402 × 158	10 KB
Motorbikes	798	263 ×165	9 KB
Faces	435	504 × 333	28 KB
Watch	239	292 × 230	14 KB
Leopards	200	182 × 138	7 KB
Bonsai	128	263 × 281	17 KB
Chandelier	107	269 × 274	15 KB

Image data were collected through crawling based on image searches on Google with Beautifulsoup and ChromeDriver of Python. Collected data were put on the image size, redundancy, and error tests with a preprocessing module to build datasets for learning and testing. Figure 6 shows some of the data collected through crawling, and Table 4 shows the current state of crawling data. There was a total of six labels in the collected data, and they include birds, boats, cars, cats, dogs, and rabbits. Their amounts were increased by ten times through data augmentation for learning and testing after preprocessing.

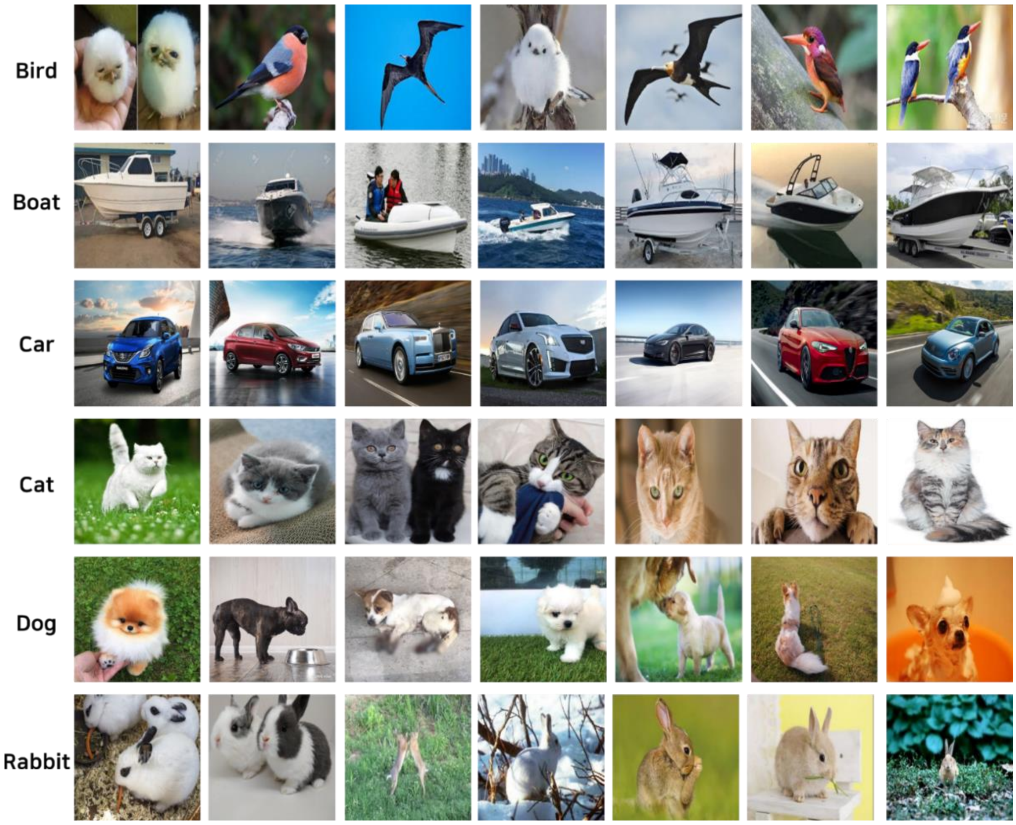


Figure 6. Caltech 101 data set.

Table 4. Configuration of crawling data set.

Category	Data Volume Before Preprocessing	Data Volume After Preprocessing	Image Average Size	Image Average Capacity
Bird	1369	676	349 × 254	26 KB
Boat	1288	557	407 × 284	39 KB
Car	1377	702	566 × 357	82 KB
Cat	1285	786	730 × 553	108 KB
Dog	1229	663	696 × 536	122 KB
Rabbit	1411	500	403 × 302	49 KB

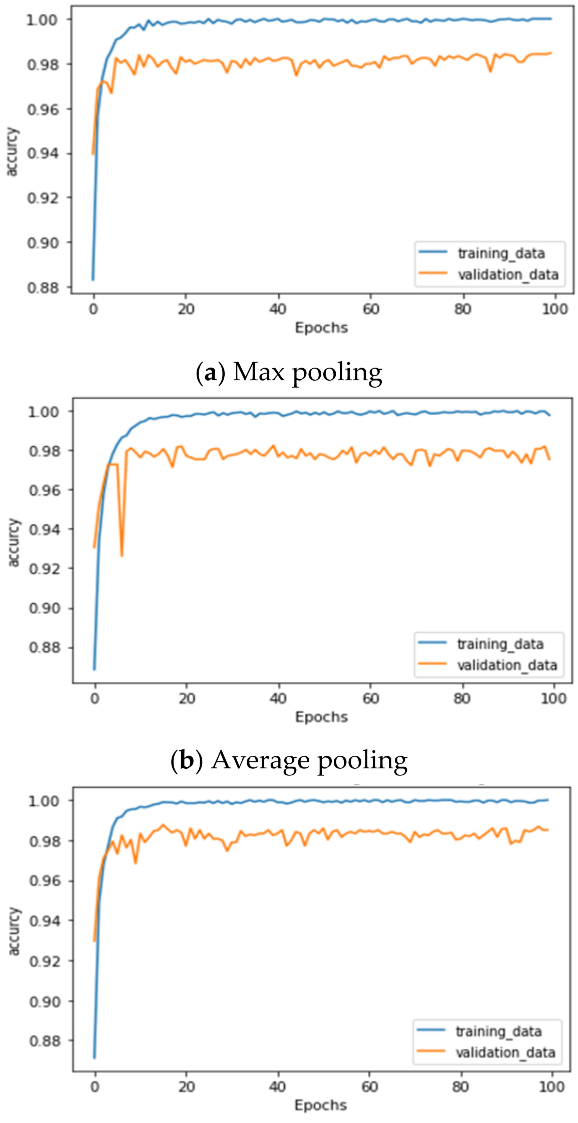
4.3. Performance Evaluation of CMP

CMP was assessed in performance with Caltech 101 and crawling data with the CNN models of the same structure and different pooling techniques. Table 5 shows a model structure for performance assessment. It consists of four convolution layers and two pooling layers. Two models that were only comprised of max and average pooling were compared in performance with a model comprised of CMP and max pooling. There are two reasons behind my using both CMP and max pooling: first, the combination of CMP and max pooling recorded better performance than CMP alone even without any special adjustment to tolerance percentage; second, there is a possibility that noises will be extracted when CMP is used in a shallow neural network, which led to the combination of CMP and max pooling rather than CMP alone. The division for training, validation, and testing of each dataset is as follows. CMP was set to allow 0 when more than half of a window is 0 in case of 0.5 tolerance percentage. In this study, I divided data for learning, validation and testing in the following percentage: 74% of the entire data was used for learning models; 16% for validation to check changes in the model education process; and 10% was for testing to check performance for the last time.

Table 5. Model Structure for CMP Performance Test.

Data Set	Image Size	Network Depth	Convolutional	Polling	Label
Caltech 101	64 × 64	6	4	2	7
Crawling	64 × 64	6	4	2	6

Figure 7 shows the performance results of pooling with Caltech data. On the graph, the x and y axes represent epoch and accuracy rate, respectively. The blue and orange lines represent the accuracy rates of learning and testing data, respectively.



(**a**) Max pooling

(**b**) Average pooling

(**c**) CMP pooling

Figure 7. Pooling performance comparison using Caltech 101 data.

In Figure 7a is a model only comprised of max pooling with a maximum and average accuracy rate of 0.9846% and 0.9802%, respectively. An accuracy rate of 0.98% or lower happened 30 times in total. This model had smaller variations in performance. In the Figure 7b is a model comprised of average pooling with a maximum and average accuracy rate of 0.9824% and 0.9764%, respectively. An accuracy rate of 0.98% or lower happened 81 times in total. This model had bigger variations in performance. In the Figure 7c is a combination of CMP and max pooling with a maximum and average accuracy rate of

0.9877% and 0.9815%, respectively. An accuracy rate of 0.98% or lower happened 21 times in total.

Figure 8 shows the final performance results of the three models. Average pooling recorded the lowest performance based on accuracy rates and loss value. Max pooling and CMP showed similar performance results, but there were differences in loss values between max pooling at 0.1021% and CMP at 0.0817%. A structure using CMP recorded higher performance results than max pooling.

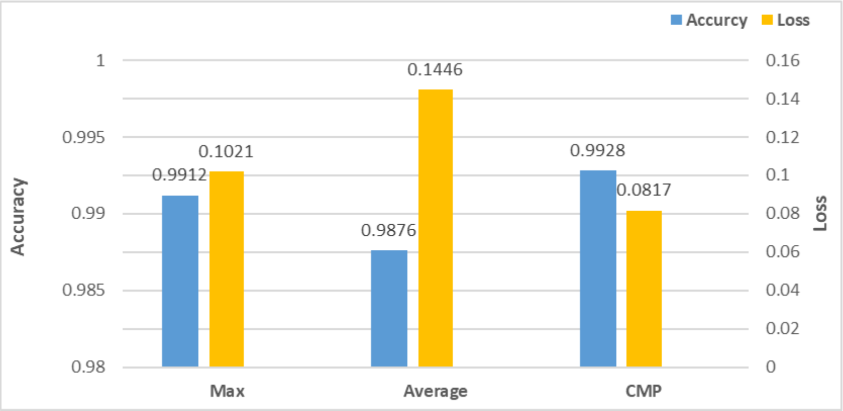
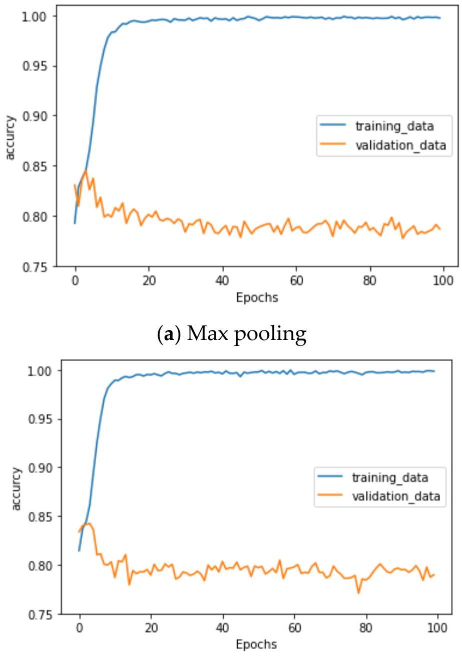


Figure 8. Pooling performance result using Caltech 101 data.

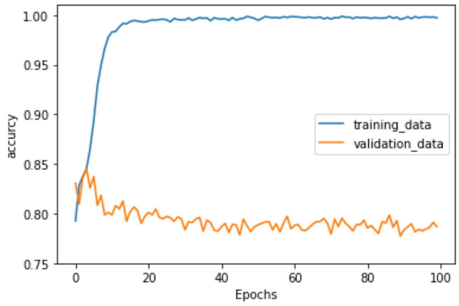
Meanwhile, Figure 9 shows the pooling performance test results with crawling data: 9a shows the test results of only max pooling, whose maximum, average, and minimum accuracy rates were 0.8451%, 0.7934%, and 0.7772%, respectively; 9b shows the performance results of average pooling, whose maximum, average, and minimum accuracy rates were 0.8423%, 0.7963%, and 0.7707%, respectively; 9c was a combination of CMP and max pooling and recorded maximum, average, and minimum accuracy rates of 0.8433%, 0.8062%, and 0.7811%, respectively.



(**a**) Max pooling

(**b**) Average pooling

Figure 9. *Cont.*



(**c**) CMP pooling

Figure 9. Pooling performance comparison using crawling data.

Figure 10 shows the final accuracy rates and loss values of crawling data by pooling type. The model that combined CMP and max pooling recorded the highest accuracy rate at 0.81 and the lowest loss rate at 0.23902. In the performance evaluation test, it recorded the highest performance of the three pooling techniques. The average pooling recorded a little bit of a higher performance result than max pooling, unlike Caltech 101.

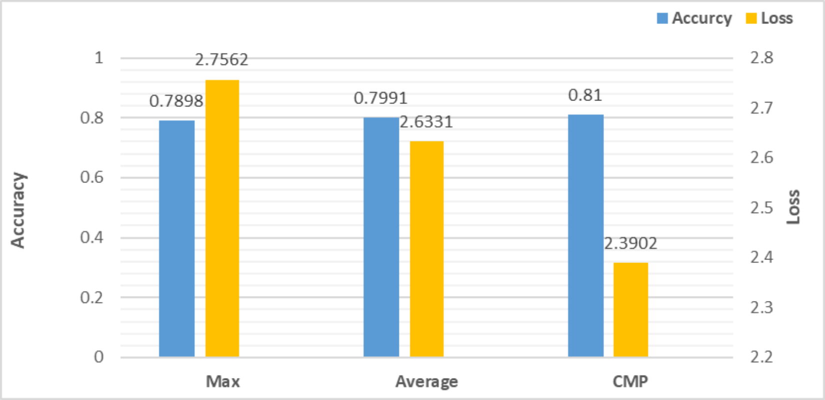
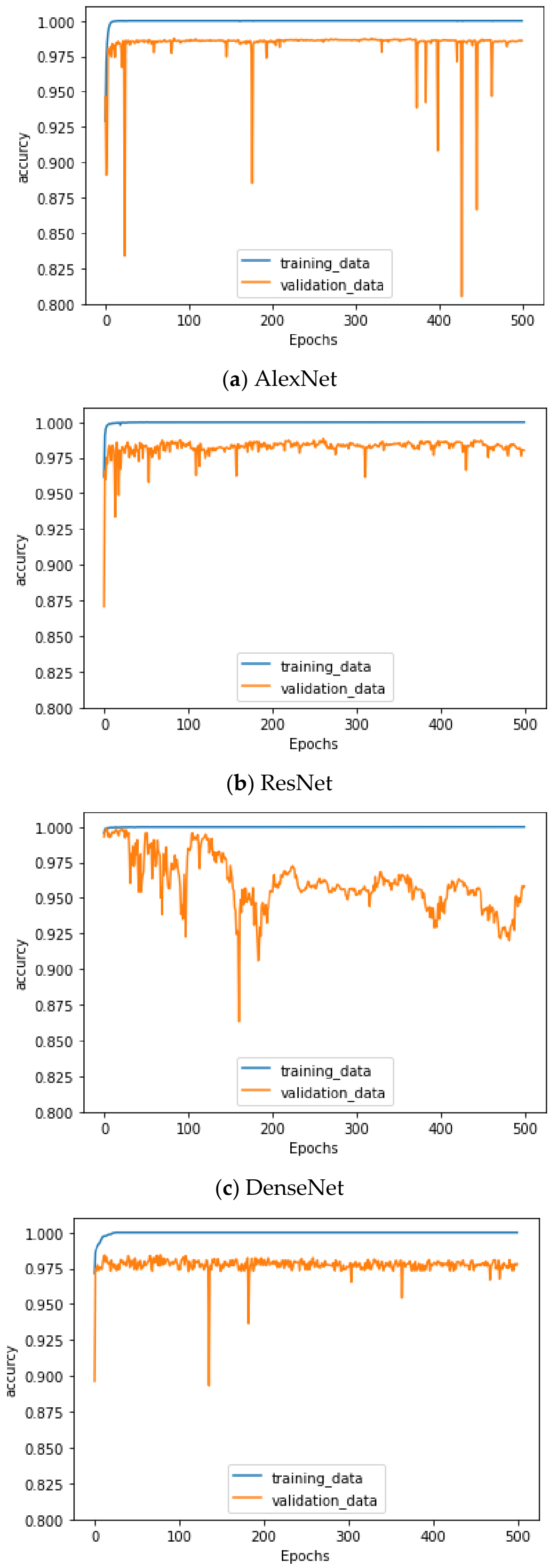


Figure 10. Pooling performance result using crawling data.

4.4. *Performance Evaluation of Restructured Neural Network*

A neural network restructured for the efficient utilization of CMP was assessed in performance along with AlexNet, ResNet, and DenseNet. Caltech 101 and crawling data mentioned earlier was used in the performance test.

Additionally, Figure 11 shows the performance test results by the model with Caltech data: (a) shows the performance test results of AlexNet, which frequently had huge performance drops after keeping its overall performance at a certain level. The maximum, minimum, and average accuracy rates of AlexNet were 0.9876%, 0.8050%, and 0.9837%, respectively; (b) shows the performance results of ResNet, whose maximum, minimum, and average accuracy rates were 0.9884%, 0.8708% at the beginning of learning, and 0.9824%, respectively; (c) shows the performance results of DenseNet, which began learning at the highest accuracy rate of 0.9929% but made a huge drop in performance by failing to achieve performance stability. The maximum, minimum, and average accuracy rates of DenseNet were 0.9989%, 0.8633%, and 0.9605%, respectively; (d) shows the performance results of the proposed pooling structure, whose maximum, minimum, and average accuracy rates were 0.9843%, 0.8932, and 0.9773%, respectively. Figure 12 shows the final performance results of a model that used 10% of Caltech data in a performance test. The models were similar in performance except for DenseNet, but the proposed pooling structure recorded the highest accuracy rate of 0.9813%. AlexNet recorded the lowest loss rate at 0.0491, being followed by a proposed pooling structure at 0.2407.



(**a**) AlexNet

(**b**) ResNet

(**c**) DenseNet

(**d**) Proposed pooling structure

Figure 11. Performance Comparison of Neural Network Models using Caltech 101 Data.

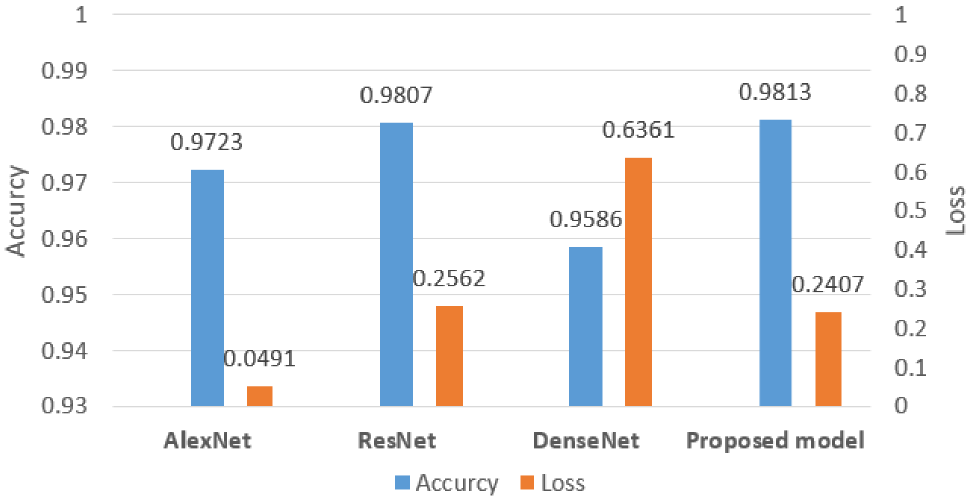
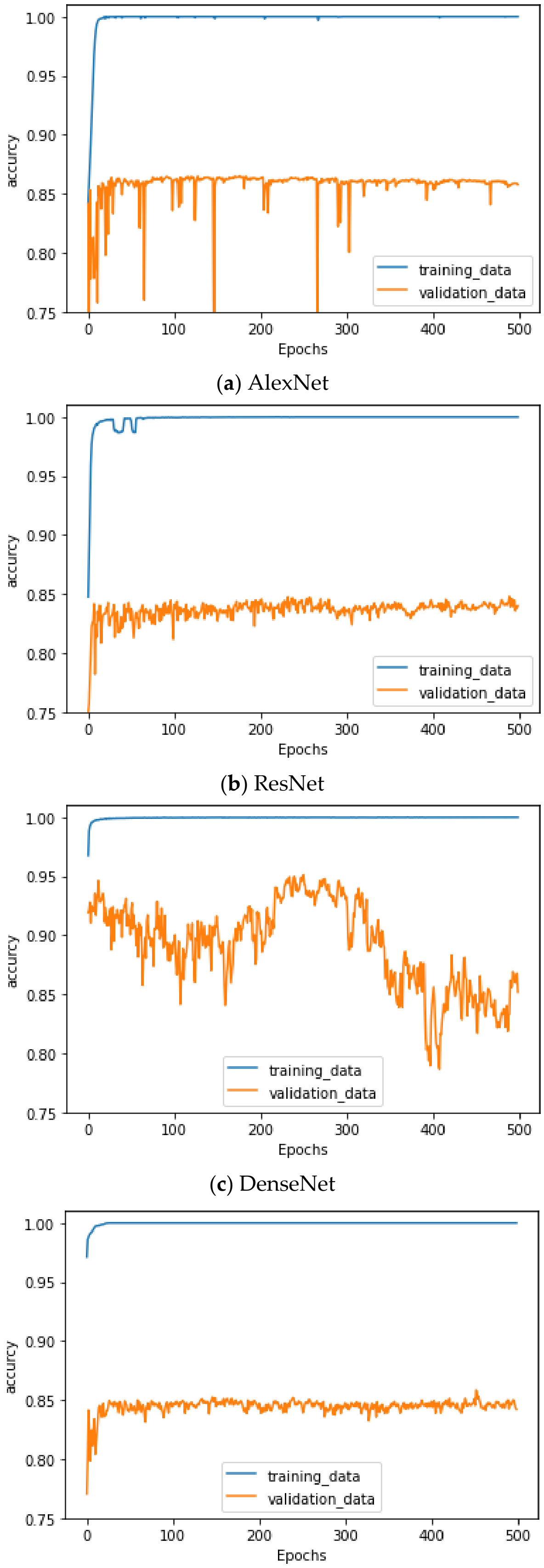


Figure 12. Performance evaluation of neural network models using Caltech 101 data (average accuracy and loss value).

Figure 13 presents the test results with crawling data. The models had overall similar performance results to the test results with Caltech data: (a) shows the test results of AlexNet, whose maximum, average, and minimum accuracy rates were 0.865%, 0.8575%, and 0.7202%, respectively; (b) shows the test results of ResNet, whose maximum, average, and minimum accuracy rates were 0.848%, 0.8363%, and 0.749%, respectively; (c) shows the test results of DenseNet, which recorded the best performance results with the maximum, average and minimum accuracy rates of 0.9511%, 0.8916%, and 0.7866%, respectively; (d) shows the test results of the proposed pooling structure, whose maximum, average, and minimum accuracy rates were 0.8647%, 0.8414%, and 0.7706%, respectively.

Meanwhile, Figure 14 shows the final performance test results of crawling data after learning. DenseNet recorded the highest accuracy rate at 0.8686%, being followed by the proposed pooling structure at 0.8494%. AlexNet recorded the lowest error rate at 0.8454, being followed by the proposed pooling structure at 2.327.



(**a**) AlexNet

(**b**) ResNet

(**c**) DenseNet

(**d**) Proposed pooling structure

Figure 13. Performance comparison of neural network models using crawling data.

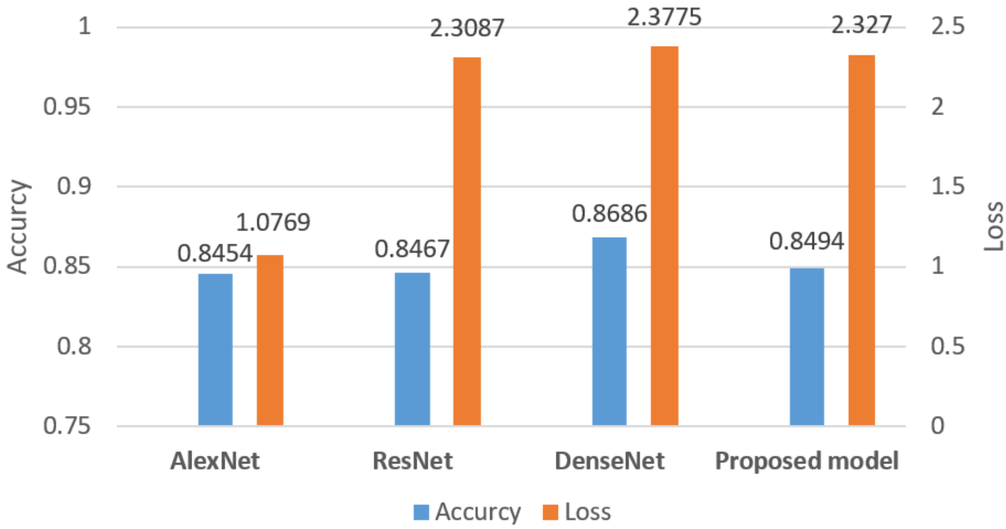


Figure 14. Performance evaluation of neural network models using crawling data (average accuracy and loss value).

5. Conclusions

In an effort to solve the issues of several pooling techniques usually used in the old CNNs such as overfitting and the extinction of features, in this study developed CMP and a restructured pooling structure to promote its efficient utilization and compared them with old techniques.

The CMP structure was designed based on min pooling and solved the issue of feature extinction by designating a tolerance to the feature of 0. The proposed pooling structure organized old pooling composition in two layers and applied different pooling techniques (max pooling and CMP) to have more diverse feature maps than old pooling approaches. CMP and the proposed pooling structure were tested in two forms.

In the first form of research, CMP was compared in performance with max and average pooling. The test results based on Caltech 101 data show that the CMP technique recorded an accuracy rate of 0.9928%, which was higher than old pooling techniques by 0.16~0.52%. Its loss rate was 0.0817, which was lower than old techniques by 19.98~28.71%. In the test with collected images, its accuracy rate was 0.81%, which was higher than old techniques by 1.36~2.56%. Its loss rate was 2.3902, which was lower than old techniques by 9.22~13.28.

In the second form of research, the pooling structure proposed to ensure the efficient utilization of CMP was assessed in performance based on its comparison with AlexNet, ResNet, and DenseNet models. In the final test with the Caltech 101 data, the proposed pooling structure recorded the highest accuracy rate at 0.9813%. AlexNet recorded the lowest error rate at 0.0491, followed by the proposed pooling structure at 0.2393. In the performance test with collected images, DenseNet recorded the highest accuracy rate at 0.8686%, followed by the proposed pooling structure at 0.8494%. AlexNet recorded the lowest error rate at 1.0769, followed by the proposed pooling structure at 2.327.

The first research demonstrated that CMP made an improvement in performance from old pooling techniques even though the improvement was small. The results hold enough significance for the utilization of CMP. The second research assessed the proposed pooling structure in performance and found that it had a relatively outstanding performance even though it was behind the old models. Based on these findings, future studies will transplant various models improved around the old convolution structure [47–59] in the pooling structure of the proposed model.

We will prepare a follow-up study through exiting study and established algorithms to compare, validate, and test the combined model in performance and state to improve its performance using NeMenyi [60] test and Wilcoxon [61] signed rank test will be conducted based on Demšar et al. [62].

Author Contributions: Conceptualization, J.P., J.-Y.K., J.-H.H., H.-S.L., S.-H.J. and C.-B.S.; Data curation, J.P. and J.-H.H.; Formal analysis, J.P., J.-Y.K., H.-S.L., S.-H.J. and C.-B.S.; Funding acquisition, S.-H.J. and C.-B.S.; Investigation, J.P., S.-H.J. and C.-B.S.; Methodology, J.-Y.K., J.-H.H. and H.-S.L.; Project administration, S.-H.J. and C.-B.S.; Resources, J.P., J.-Y.K., S.-H.J. and C.-B.S.; Software, J.P., J.-Y.K., H.-S.L., S.-H.J. and C.-B.S.; Supervision, C.-B.S.; Validation, J.P. and J.-Y.K.; Visualization, J.P., J.-Y.K., J.-H.H., H.-S.L. and S.-H.J.; Writing—original draft, J.P., J.-Y.K., J.-H.H., H.-S.L., S.-H.J. and C.-B.S.; Writing—review and editing, J.-H.H., H.-S.L., S.-H.J. and C.-B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Rural Development Administration(RDA) and Ministry of Science and ICT (MSIT)(421028-3) and This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2020-0-01489) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and 0.5 MB model. *arXiv* **2016**, arXiv:1602.07360.
2. Jun, P.; Jun-Yeong, K.; Se-Hoon, J.; Chun-Bo, S. *Design of CNN Structure Using Conditonal Min Pooling*; Multimedia MITA: Dhaka, Bangladesh, 2020; Volume 23, pp. 119–122.
3. Siri Team. Deep learning for Siri's voice: On-device deep mixture density networks for hybrid unit selection synthesis. *Apple Mach. Learn. J.* **2017**, *1*, 1–36.
4. Sepp, H.; Scmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
5. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
6. Bosch, A.; Zisserman, A.; Munoz, X. Image classification using random forests and ferns. In Proceedings of the IEEE 11th International Conference on Vision, Rio de Janeiro, Brazil, 14–21 October 2007; Volume 1, pp. 1–8.
7. David, G. Lowe Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
8. Deng, L. The MNIST database of handwritten digit imagesfor machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
9. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
10. Gómez-Uribe, C.A.; Hunt, N. The Netflix recommender system. *ACM Trans. Manag. Inf. Syst.* **2015**, *6*, 1–19. [CrossRef]
11. Solyman, A.E. *Introduction To Computer Vision (Computer Vision and Robotics)*; 5-DOF Robotic Arm Manipulator—MSc; Egyptian Atomic Energy Authority: Cairo, Egypt, 2019.
12. Dey, S.; Dutta, A.; Toledo, J.I.; Ghosh, S.K.; Lladós, J.; Pal, U. Signet: Onvolutional siamese network for writer independent offline signature verification. *arXiv* **2014**, arXiv:1707.02131.
13. Ross, G.; Jeff, D.; Trevor, D.L.; Jitendra, M. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2013; pp. 1–21.
14. Elmar, M.; Gregory, H. Adaptive and generic corner detection based on the accelerated segment test. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; pp. 183–196.
15. Girshick, R. Fast r-cnn. In Proceedings of the IEEE Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; IEEE: New York, NY, USA, 2015; pp. 91–99.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unifed, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Vincent, O.R.; Folorunso, O. A descriptive algorithm for sobel image edge detection. *Inf. Sci. IT Educ. Conf.* **2009**, *2009*, 97–107.
19. Graham, B. Fractional max-pooling. *arXiv* **2014**, arXiv:1412.6071.
20. Zhai, S.; Wu, H.; Kumar, A.; Cheng, Y.; Lu, Y.; Zhang, Z.; Feris, R. S3Pool: Pooling with stochastic spatial sampling. *arXiv* **2016**, arXiv:1611.05138.

21. Satti, P.; Sharma, N.; Garg, B. Min–max average pooling based filter for impulse noise removal. *IEEE Signal Process. Lett.* **2020**, *27*, 1475–1479. [CrossRef]
22. Mohsin, Z.; Alzubaidi, L.S. Convolutional neural network with global average pooling for image classification. In Proceedings of the International Conference on Electrical, Communication, Electronics, Instrumentation and Computing (ICECEIC), Kanchipuram, India, 30–31 January 2019.
23. Kolla, M.; Venugopal, T. Concatenated global average pooled deep convolutional embedded clustering. In Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications(ICDSMLA 2019), Hyderabad, India, 29–30 March 2020; Volume 601, pp. 778–786.
24. Huang, G.; Liu, Z.; Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
25. Antioquia, A.C.; Tan, D.S.; Azcarraga, A.; Cheng, W.H.; Hua, K.L. ZipNet: ZFNet-level Accuracy with 48× Fewer Parameters. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 December 2018; pp. 1–4.
26. Jung, Y. A Study on the Embedded Car License Plate Recognition System Based on Deep Learning(DBN). Master's Thesis, Hongik University, Seoul, Korea, 2017.
27. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2014**, *115*, 211–252. [CrossRef]
28. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 558–567.
29. Simard, P.Y.; Steinkraus, D.; Platt, J.C. Best Practices for convolutional neural networks applied to visual document analysis practice. *IEEE Comput. Soc. Ashington* **2003**, *3*, 958–963.
30. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multi-scale structural similarity for image quality assessment. In Proceedings of the IEEE Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 9–12 November 2003; pp. 1398–1402.
31. Jain, A.K.; Mao, J.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *IEEE Comput.* **1996**, *29*, 31–44. [CrossRef]
32. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-excitation networks. *arXiv* **2019**, arXiv:1709.01507.
33. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. *NIPS* **2017**, *2017*, 3859–3869.
34. Yoo, J.S.; Lee, K.C. Deep learning based image recognition technology trend. *Inf. Soc.* **2017**, *33*, 17–24.
35. Jerrod, P.; Shakti, K.; Roussy, J. Adaptive attention span in computer vision. *arXiv* **2020**, arXiv:2004.08708.
36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the NIPS'12 25th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 3 December 2012; Volume 1, pp. 1097–1105.
37. Howard, A.G. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.0486.
38. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. *arXiv* **2015**, arXiv:1512.0056.
39. Zeiler, M.D.; Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv* **2013**, arXiv:1301.3557.
40. Ian Val, P.D.R.; Ariel, M.S.; Ruji, P.M. A novel fused random pooling method for convolutional neural network to improve image classification accuracy. In Proceedings of the 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kuala Lumpur, Malaysia, 20–21 December 2019.
41. Zenglin, S.; Yangdong, Y.; Yunpeng, W. Rank-based pooling for deep convolutional neural networks. *ACM Digit. Libr.* **2016**, *83*, 21–31.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. *Comput. Vis. Pattern Recognit.* **2016**, *3*, 1–15.
43. Huang, G.; Liu, Z.; Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. *Conf. Comput. Vis. Pattern Recognit.* **2018**, *5*, 1–9.
44. Cai, B.; Xu, X.; Jia, K.; Qing, C.; Tao, D. DehazeNet: An end-to-end system for single image haze removal. *IEEE Trans. Image Process.* **2016**, *25*, 5187–5198. [CrossRef] [PubMed]
45. Zhu, Q.; Mai, J.; Shao, L. A fast single image haze removal algorithm using color attenuation prior. *IEEE Trans. Image Process.* **2015**, *24*, 3522–3533.
46. Meng, G.; Wang, Y.; Duan, J.; Xiang, S.; Pan, C. Efficient Image dehazing with boundary constraint and contextual regularization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013; pp. 617–624.
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Published as a conference paper at ICLR. *arXiv* **2015**, arXiv:1409.1556v6.
48. Veit, A.; Wilber, M.; Belongie, S. Residual networks behave like ensembles of relatively shallow networks. *Neural Inf. Process. Syst.* **2016**, *27*, 424–433.
49. Christian, S.; Wei, L.; Yangqing, J.; Pierre, S.; Scott, R.; Dragomir, A.; Dumitru, E.; Vincent, V.; Andrew, R. Going deeper with convolutions. ImageNet large-scale visual recognition challenge. *arXiv* **2014**, arXiv:1409.4842v1.
50. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the Computer Vision and Pattern Recognition(CVPR), San Diego, CA, USA, 20–25 June 2005.

51. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
52. Choi, W.; Choi, K.; Park, J. Low cost convolutional neural network accelerator based on Bi-directional filtering and bit-width reduction. *IEEE Access* **2018**, *6*, 14734–14746. [CrossRef]
53. Fu, Z.; Zhang, F.; Yin, Q.; Li, R.; Hu, W.; Li, W. Small sample learning optimization for resnet based sar target recognition. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 2330–2333.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
55. Huang, G.; Liu, S.; van der Maaten, L.; Weinberger, K.Q. CondenseNet: An efficient densenet using learned group convolutions. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2752–2761.
56. Wu, R.; Yan, S.; Shan, Y.; Dang, Q.; Sun, G. Deep image: Scaling up image recognition. *arXiv* **2015**, arXiv:1501.02876.
57. Sepp, H. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Intern. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116.
58. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR, Lauderdale, FL, USA, 11–13 April 2011; Volume 15, pp. 315–323.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
60. Nemenyi, P. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
61. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [CrossRef]
62. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.