ISSN: 2508-7894 © 2022 KODISA & KAIA.
KJAI website: http://acoms.kisti.re.kr/kjai
doi: http://dx.doi.org/10.24225/kjai.2023.11.1.17

Efficient Large Dataset Construction using Image Smoothing and Image Size Reduction

Jaemin HWANG[1], Sac LEE[2], Hyunwoo LEE[3], Seyun PARK[4], Jiyoung LIM[5]

Received: Febrary 26, 2023. Revised: Febrary 28, 2023. Accepted: March 04, 2023

Abstract

With the continuous growth in the amount of data collected and analyzed, deep learning has become increasingly popular for extracting meaningful insights from various fields. However, hardware limitations pose a challenge for achieving meaningful results with limited data. To address this challenge, this paper proposes an algorithm that leverages the characteristics of convolutional neural networks (CNNs) to reduce the size of image datasets by 20% through smoothing and shrinking the size of images using color elements. The proposed algorithm reduces the learning time and, as a result, the computational load on hardware. The experiments conducted in this study show that the proposed method achieves effective learning with similar or slightly higher accuracy than the original dataset while reducing computational and time costs. This color-centric dataset construction method using image smoothing techniques can lead to more efficient learning on CNNs. This method can be applied in various applications, such as image classification and recognition, and can contribute to more efficient and cost-effective deep learning. This paper presents a promising approach to reducing the computational load and time costs associated with deep learning and provides meaningful results with limited data, enabling them to apply deep learning to a broader range of applications.

Keywords : CNN, Gaussian Filter, Median Filter, Image Smoothing, Image Size Reduction, Dataset Construction

Major Classification Code: C45, C81, C88, L86, O33

1. Introduction

Convolutional Neural Networks (CNN) are a powerful technique that enables machines to acquire prior knowledge and detect objects in images, starting with the classification of unclassified images (LeCun, Bottou, Bengio, & Haffner, 1998). To improve the accuracy of object detection, it is essential to use good models and techniques that prevent overfitting and learn from large datasets. However, until the release of large datasets such as ImageNet (Deng, Dong,

* This research was supported by UISP (University Innovation Support Project) of Korean Bible University in 2022.

1 First Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea.
Email: nacer6617@bible.ac.kr

2 Second Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea.
Email: issac0122@bible.ac.kr

3 Third Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea.
Email: bible127@bible.ac.kr

4 Fourth Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea.
Email: seyun298@bible.ac.kr

5 Corresponding Author. Professor, Dept. of Computer Software, Korean Bible University, Korea. Email: jylim@bible.ac.kr

Socher, Li, Li, & Fei-Fei, 2009), datasets were relatively small in size, including NORB (LeCun, Huang, & Bottou, 2004), Caltech-101/256 (Fei-Fei, Fergus, & Perona, 2004), CIFAR-10/100 (Krizhevsky & Hinton, 2009), and MNIST. Since the image size was small, it was not challenging for machines to learn and recognize objects.

Some researchers conducted an experiment that allowed machines to learn MNIST, a dataset of handwritten Arabic numbers, to determine the extent to which they could be classified. The results showed that the most error-prone performance differed only by 0.3% from that of human classification. This experiment demonstrated that there was little difference between human and machine classification, and machines could judge and classify with high accuracy (Ciregan, Meier, & Schmidhuber, 2012).

However, some researchers argued that the results of this experiment were limited to specific datasets and that a large volume of training data was necessary to recognize practical objects. They argued that a large volume of the training dataset was needed to recognize a practical object (Krizhevsky, Sutskever, & Hinton, 2017).

While learning from a large dataset can improve the detection and recognition rates of machines, research has been undertaken to find efficient learning methods with limited data (Kim, & Chung, 2022). However, research has been studied to find efficient learning methods with limited data (Al-Jarrah, Yoo, Muhaidat, Karagiannidis, & Taha, 2015; Li, Zhou, Chen, & Li, 2017; Thompson, Greenewald, Lee, & Manso, 2020) since learning from a large volume of data is not a solution to all problems. Image augmentation techniques are also used to achieve higher effects with limited datasets in the field of imaging (Bloice, Stocker, & Holzinger, 2017).

In this paper, we propose an effective dataset construction method for efficient learning using image smoothing and image size reduction. The paper presents several contributions, as follows:

[1]. Dataset reduction: The authors reduced the amount of dataset by applying image smoothing techniques like Median filter and Gaussian filter.

[2]. Reduced learning time: Using this reduced dataset helps machines learn faster as there is less data to process.

[3]. Reduced computational load: By reducing the amount of data, the amount of computation needed is also reduced, which can lead to faster and more efficient training.

[4]. Similar or slightly higher accuracy rate: The authors have reported that the accuracy rate of their reduced dataset is comparable or slightly higher than that of the original dataset, indicating that the reduction process does not negatively impact the quality of the dataset.

Overall, the contributions of this paper highlight the potential benefits of data reduction techniques in machine learning, which can lead to more efficient and effective training without sacrificing the quality of the results.

The rest of this paper is organized as follows. Section 2 overviews the related works. Section 3 introduces our proposed dataset construction method. Section 4 presents the experiments and analysis of our dataset. Finally, Section 5 concludes the paper with some remarks and possible future directions.

2. Related Works

2.1. Classification based on Superficial Patterns

In general, we perceive that when a CNN learns an object, it grasps the shape and outline of the object, proceeds with learning, and recognizes it. However, Geirhos' experiment with 97 observers showed the opposite result (Geirhos, Rubisch, Michaelis, Bethge, Wichmann, & Brendel, 2018). CNN models (AlexNet, VGG (Simonyan, & Zisserman, 2014), GoogleNet(Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, & Rabinovich, 2015), ResNet(He, Zhang, Ren, & Sun, 2016)) and most observers identified cats when judging cat images, but not all CNN models could be identified by leaving only the outline or silhouette. In other words, when machines learn an object, they do not understand the shape of objects like humans, as confirmed by experimental results. Figure 1 provides more detailed view.

To confirm the results of machine learning, they conducted an experiment where they presented the machine with an image of an elephant's texture. As shown in Figure 1(a), the machine identified it as an 'Indian elephant,' with a prediction rate of 81.4% in Top1. They also presented the machine with an image of a spotted cat after it learned a dataset with object images, and as shown in Figure 1(b), the machine identified it as a 'tabby cat,' with a prediction rate of 71.1% in Top1.

Looking at the results of (a) and (b) in Figure 1, we might think that the machine recognizes and understands the object by grasping its characteristics through a complex process. However, Figure 1(c) shows an inexplicable result. This figure shows that when machines learn using a dataset with mixed textures and shapes of objects, the machine identified an image of a mixture of cat shapes and elephant textures as an 'Indian Elephant' with a prediction rate of 63.9%.

Based on the previous view that machines learn similarly to humans, we would expect the Top-1 to have a "tabby cat," but we can see that there are no types of cats from Top-1 to Top-3 and that there are 'Indian Elephant,' 'indri' and 'black swan.'

Based on the above results, it can be observed that while humans tend to learn the shape of an object as a primary characteristic to recognize it, the process of machine learning differs significantly. Geirhos' experiment highlighted that CNN models rely on superficial patterns, such as texture, to learn, judge, and classify objects. This indicates a fundamental difference in the learning approach between humans and machines. The results suggest that current machine learning models have limitations, and more work is required to develop models that can recognize and understand objects in a way that is similar to how humans do.

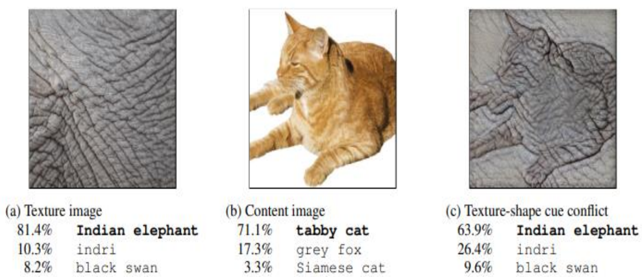

Figure 1: Texture-Shape Cue Conflict

2.2. Color Mixing using Filters

To determine the new pixel value of an output image, a spatial region operation that considers not only the pixel itself but also the neighboring pixels around it is used, known as pixel group processing. This technique includes various methods such as smoothing, sharpening, edge detection, and noise reduction. Smoothing, in particular, is a linear spatial filter used to express images smoothly or reduce noise.

The median filter, one of the image smoothing techniques, is a low-pass filtering (LPF) method that replaces the value of the central pixel with the median of neighboring pixels. It is especially useful for reducing 'salt-and-pepper noise,' which is a type of dot noise added to an image that looks like salt or pepper (Brownrigg, 1984). The median filter effectively reduces such noise by replacing the central pixel value with the median value of its neighboring pixels.

On the other hand, the Gaussian filter is a technique that generates a filter mask by approximating a Gaussian distribution function. Although the Gaussian distribution depends on the mean and standard deviation, the Gaussian filter in image processing mainly uses a Gaussian distribution function with a mean of zero (Deng & Cahill, 1993). To generate a two-dimensional filter mask matrix that follows the Gaussian distribution, it approximates the two-dimensional Gaussian distribution function. The Gaussian filter assigns larger weights to pixels that are closer to each other and smaller weights to pixels that are farther away. This is because neighboring pixels tend to have similar values in images with slowly changing spatial characteristics.

In this paper, we use the Median and Gaussian filters to smooth the image and determine the color degree of each pixel. We also examine how machines learn pixel values.

3. Proposed Dataset Construction Method

The purpose of this paper is to develop a dataset that utilizes color elements as superficial patterns, given CNNs' tendency to prioritize texture recognition over object shape.

3.1. Smoothing

When people perceive an object, they typically focus on its shape and store it in their memory based on their understanding of it. In other words, the outline of an object plays a crucial role in recognizing it. However, this paper aims to construct a dataset where color elements are more prominent than the shape of an object. This is based on the fact that CNNs learn based on textures rather than the contours of objects. By learning a dataset constructed in this manner, we can reduce the computational volume and time required for learning.

To achieve this, we use a method of mixing colors at regular intervals with adjacent pixels, which emphasizes the color elements over the shape of the object. As shown in (b) of Figure 2, we mix each color of 2 pixels x 2 pixels into one color and apply this to every pixel in the image to reconstruct it into a single image. We construct a dataset using the Median and Gaussian filters, which are low-frequency filtering techniques that blur edges with many high-frequency components, among various image smoothing techniques.

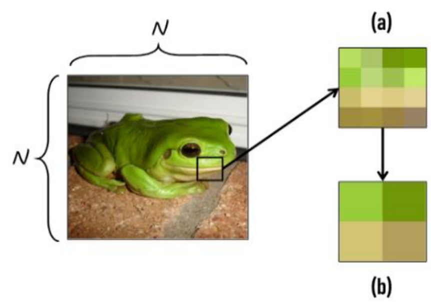

Figure 2: Structural diagram of our proposed dataset construction methods using image smoothing

3.2. Image Size Reduction

In this paper, we aim to emphasize color elements as superficial patterns. However, as discussed in Chapter 1, we also consider the relationship between image size and classification accuracy. Therefore, we employ a smoothing technique to reduce the size of the image, as shown in Figure 3(a). The original image size is N in both horizontal and vertical dimensions, while the smoothed image size is N/M as shown in Figure 3(b). The reduction in image size follows the following rule:

$$N \rightarrow \frac{N}{M} \begin{cases} N \ (N > M) \\ M \ (M > 0, M \ \% \ 2 = 0) \end{cases} \quad (1)$$

N must be greater than M, M must be greater than 0, and must be a multiple of 2. We construct a dataset by setting N and M that fit these rules.

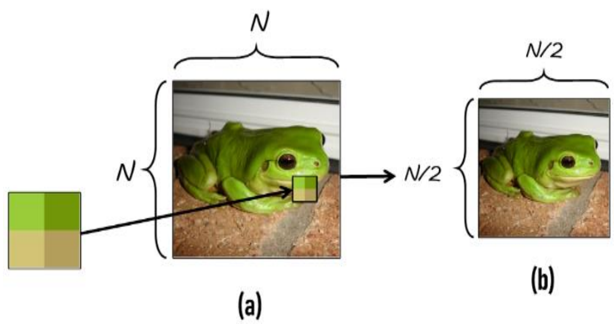

Figure 3: Structural diagram of our proposed learning methods using image size reduction

3.3. Selection of Learning Model

We will use MobileNet (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, & Adam, 2017) and ResNet among CNNs to efficiently learn datasets built by our proposed method.

MobileNet is useful for mobile or embedded systems with relatively small memory using Depthwise Separable Convolution, making the model lighter. It demonstrates the efficient performance of memory in a constrained environment.

ResNet is a learning method for solving learning difficulties caused by overfitting and gradient loss in the learning process, although performance improves as neural networks deepen. Therefore, ResNet uses a Residual Learning Framework that utilizes residuals to facilitate learning even if the neural network structure is deepened. The core technologies of ResNet are: Direct mapping of existing data requires no additional parameters and complex multiplication with Identity Short Connection, which crosses one or more layers through the display.

Mobile Net is similar to the purpose of this paper to show the same or higher performance while reducing operations. The core technology of ResNet could implement the efficient learning method that this paper wants.

4. Experiments and Analysis

4.1. Basic Dataset

To verify that the datasets constructed through our proposed method is more efficient than conventional methods, we would like to describe the results learned in the following environments.

4.1.1. Dataset Construction

We conducted experiments using the ImageNet dataset, which is an extensive image database constructed based on the WordNet hierarchy, consisting of hundreds or thousands of images for each node in the hierarchy representing objects such as beds, chairs, clocks, and keyboards. To study a subset of the dataset, we randomly selected ten objects and created a dataset of 750 learning images and 250 test images.

To apply smoothing to the constructed dataset, we used median filters and Gaussian filters, with a sigma value ranging from 3 to 7. However, we observed that when the sigma value was set to 3, there was little difference between the filtered and original images. Similarly, there was minimal distinction between the results obtained when the sigma values were set to 5 and 7. As a result, we conducted four experiments simultaneously by applying a sigma value of 5.

To compare the performance of using a newly constructed dataset with smoothing and reduced size against the original dataset, experiments were conducted in the environment outlined in Table 1. The number of learning repetitions, or Epoch, was set to 100.

As a result of the experiment, we obtained the loss value, which is the loss value of the learning data, Accuracy, the accuracy of the learning data, and valAccuracy, the accuracy of the verification data. Among them, to compare performance, this value was used because valAccuracy means a substantial judgment rate.

Table 1: The experimental environment and Training Model parameters

OS	Windows
Python Version	3.8.14
GPU Count	1
GPU Type	Tesla T4
Framework	Google Collaboratory
Optimizer	Adam
Learning Late	0.001
Epochs	100

4.1.2. Performance Evaluation

MobileNet Model : Figure 4 shows the results of learning a dataset with original size before smoothing. As the learning progresses, the verification accuracy (valAccuracy) increases, but converges to 0.4 (40%).

Figure 5 shows the results of learning our dataset with reduced image size after smoothing application. The accuracy was close to 1 before smoothing. The verification accuracy is about 0.45 (45%) which is an increase of about 5% compared to when learning the dataset before smoothing.

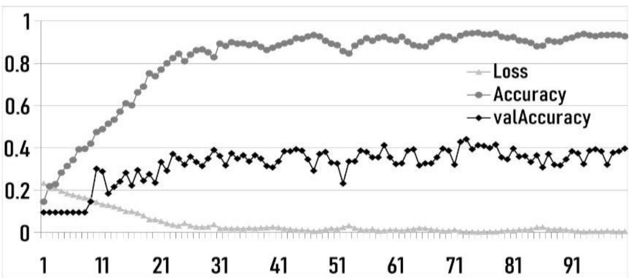

Figure 4: MobileNet Learning Results Before Smoothing

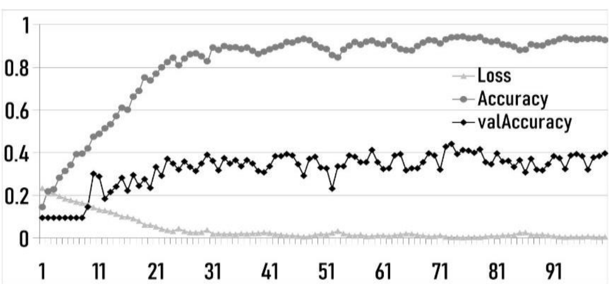

Figure 5: MobileNet Learning Results After Smoothing (Gaussian Filter, σ=5)

ResNet Model: Figure 6 shows the results of learning the dataset before smoothing, as in Figure 4. As the learning progressed, the verification accuracy showed an increasing trend, but it converges at 0.4 (40%).

Figure 7 shows the results of using the dataset with smoothing for learning. As with the results learned using MobileNet, the verification accuracy was approximately 0.45 (45%) which was approximately 5% higher than when the datasets before smoothing were used for learning.

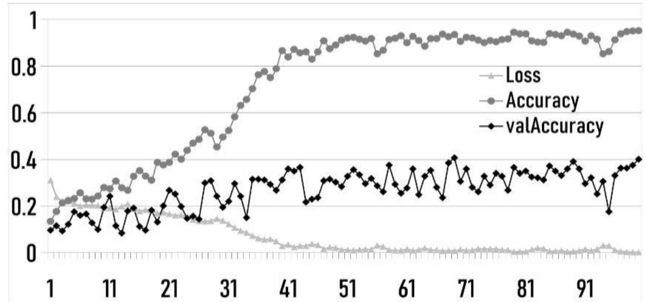

Figure 6: ResNet-50 Learning Results Before Smoothing

4.1.3. Limits

Our findings reveal that the verification accuracy did not exceed 50% in the results obtained by learning datasets constructed using images after smoothing, as illustrated in Figures 3 and 6. Similarly, the accuracy of the datasets before smoothing, as presented in Figures 4 and 7, did not exceed 50%. While the accuracy of the training data approached 1, we determined that the results were unreliable due to inaccurate judgment.

Although various factors can contribute to unreliable results, we concluded that the dataset we constructed contained data that was difficult to learn and assess accurately. As a result, even though the accuracy of the training data was high, we could not rely on the results of the dataset for making accurate predictions.

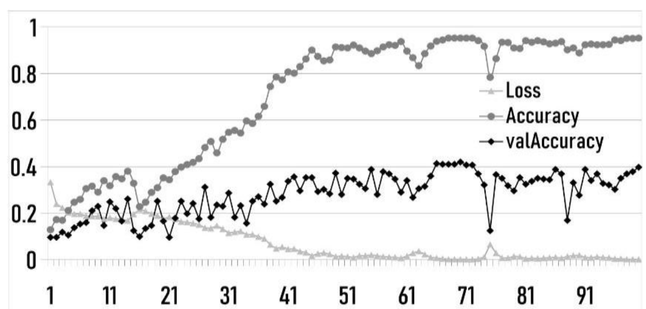

Figure 7: ResNet-50 Learning Results After Smoothing (Median Filter, σ=5)

4.2. Dataset without Background

4.2.1. Dataset Construction

After analyzing the results obtained from our initial experiments, we determined that the dataset was constructed using data that was challenging for the machine to learn. Therefore, we rebuilt the dataset using a different approach.

The left image in Figure 8 depicts one of the datasets constructed in the previous experiments, which consists of the object (a piano) and the background. However, since our goal was to build and experiment with images learned by

CNN, we constructed a new dataset where only the object's shape exists by erasing the background, as shown in the right image of Figure 8. We conducted our experiments using the environment outlined in Table 2, and we increased the number of epochs to 150 times to ensure sufficient learning time.

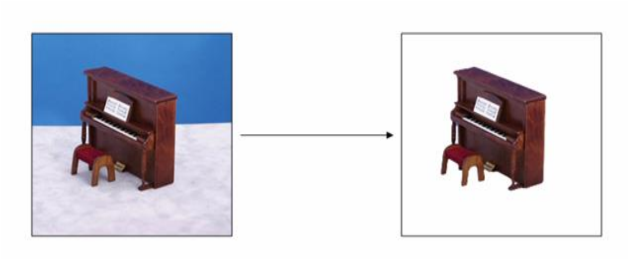

Figure 8: Change of Datasets

Table 2: The experimental environment and Training Model parameters

OS	Windows
Python Version	3.8.15
GPU Count	1
GPU Type	Tesla T4
Framework	Google Colaboratory
Optimizer	Adam
Learning Late	0.001
Epochs	100

4.2.2. Performance Evaluation

MobileNet Model: Figure 9 shows the result of learning the new dataset, which revealed a verification accuracy of around 0.68 (68%), indicating an increase of more than 20% compared to before background erasing.

Based on this improvement, we further applied smoothing and image size reduction to the background-erased dataset and experimented with it. The learning results of the dataset are shown in Figure 10, with a verification accuracy of around 0.68 (68%), similar to the results before applying smoothing and image reduction.

Table 3 shows the learning results before and after applying smoothing and image size reduction to the dataset. Although accuracy and verification accuracy seemed to have little difference before and after, the Time Average per 1 Epoch reduced by about 11% from 6.13 seconds to 5.42 seconds. The size of the dataset also decreased with smoothing and reducing image size, resulting in a decrease in the amount of computation required for learning. We found that the learning time was reduced by about 11%, and the size of the dataset used for learning was reduced by 26.3% from 118,217,420 bytes to 87,080,740 bytes.

ResNet Model: Figure 11 shows the result of learning a dataset consisting of images with only objects left by erasing the background. As the learning progressed, the verification accuracy increased, showing a result of about 0.66 (66%). The newly constructed dataset was used for learning because the verification accuracy increased by more than 0.2 (20%) compared to before the background was erased.

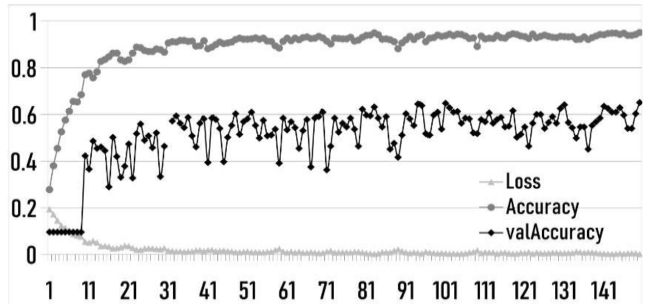

Figure 9: MobileNet Learning Results Before Smoothing

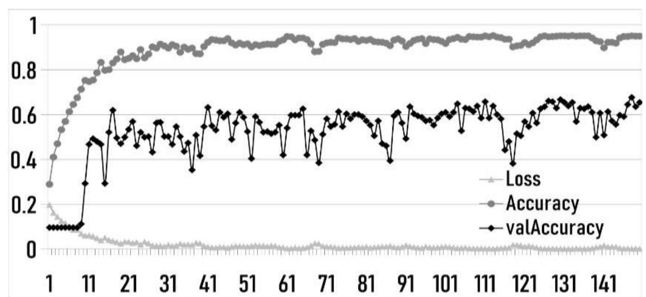

Figure 10: MobileNet Learning Results After Smoothing (Gaussian Filter, σ=5)

Table 3: Comparison Before and After smoothing

Method	Accuracy	Val Accuracy	Time Average
Before Smoothing	0.9986	0.6833	6.13
After Smoothing	0.9971	0.6867	5.42

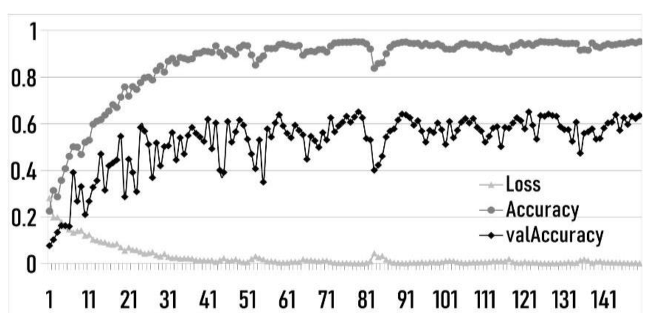

Figure 11: ResNet-50 Learning Results Before Smoothing

Figures 12 and 13 show the results of using a dataset constructed by applying smoothing and image size reduction to a dataset that erased the background for learning. As the learning progressed, the verification accuracy increased to 0.66 (66%) or more.

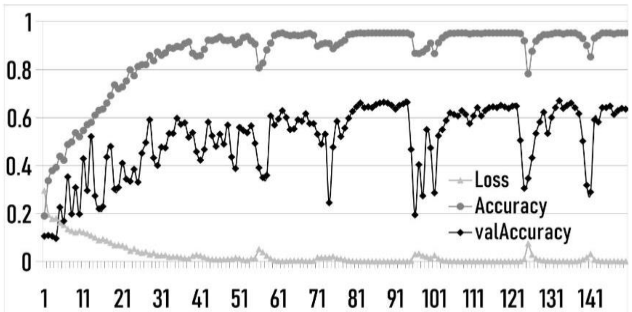

Figure 12: ResNet-50 Learning Results After Smoothing (Gaussian Filter, σ=5)

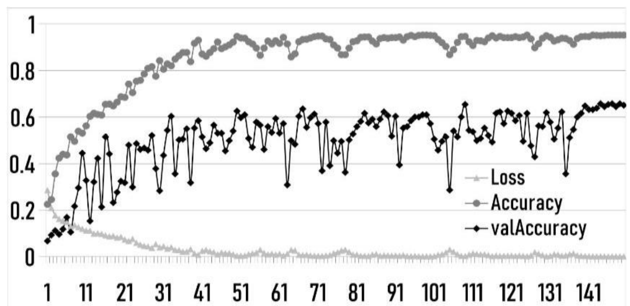

Figure 13: ResNet-50 Learning Results After Smoothing (Gaussian Filter, σ=5)

Table 4 presents the results of learning the dataset before and after applying image smoothing and size reduction. The results indicate that there is little difference in verification accuracy between the original dataset and the smoothed dataset, particularly for MobileNet. However, there is a noticeable difference in the time required to complete one epoch. When smoothing was applied using a Gaussian filter with σ=5, it took 13.34 seconds, which is 0.67 seconds (5%) less than the 14.01 seconds required for learning the original dataset. When the median filter with σ=5 was used, it decreased to 10.13 seconds, which is about 27.6% less than the original dataset.

Additionally, the size of the dataset was reduced through image smoothing. The Gaussian filter with σ=5 resulted in a decrease of approximately 26.3% in the dataset size, while the median filter with σ=5 led to a more significant decrease of about 37.4% (44,280,080 bytes) to a dataset size of 73,937,340 bytes.

In summary, Table 4 demonstrates the effectiveness of image smoothing and size reduction techniques in reducing the dataset size and improving the efficiency of the learning process without compromising verification accuracy.

Table 4: Comparison Before and After Smoothing

Method	Accuracy	Val Accuracy	Time Average
Before Smoothing	1	0.6667	14.01
After Smoothing	1	0.6667	10.13
	1	0.6833	13.34

5. Conclusions

The paper is based on the premise that when a Convolutional Neural Network (CNN) learns an object, it learns based on superficial patterns rather than the object's shape. To investigate this, the authors applied image smoothing techniques to enhance the color in a superficial pattern and reduce the image size, resulting in a color-focused dataset that can be learned more efficiently by the CNN.

The main contributions of the paper are as follows. First, the results of the study show that the proposed image smoothing method can achieve similar or slightly higher verification accuracy rates compared to the dataset before smoothing. Furthermore, the method reduced the dataset size by as much as 37.4% and decreased the time taken to complete one epoch by 5% to 27.6%.

Secondly, the proposed method can be useful in a capacity-constrained environment where a larger number of images cannot be used due to resource limitations. By constructing a dataset with fewer but more efficient images, the proposed method allows for more efficient learning.

Based on the results of this study, we suggest that future research could focus on increasing the judgment rate while further utilizing the element of color.

In conclusion, this paper presents a method for constructing a color-focused dataset using image smoothing techniques that can lead to more efficient learning in CNNs. The results show promising benefits for a capacity-constrained environment and provide avenues for further research.

References

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient Machine Learning for Big Data: A Review. *Big Data Research*, *2*(3), 87–93.

Bloice, M.D., Stocker, C., & Holzinger, A. (2017). Augmentor: An Image Augmentation Library for Machine Learning. *The Journal of Open Source Software*, *2*(19), 432.

Brownrigg, D. R. K. (1984). The weighted median filter. *Communications of the ACM*, *27*(8), 807–818.

Ciregan, D., Meier, U., & Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*. 3642-3649

Deng, G., & Cahill, L. W. (1993, October). An adaptive Gaussian filter for noise reduction and edge detection. In *1993 IEEE conference record nuclear science symposium and medical imaging conference,* 1615-1619.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition,* 248-255.

Fei-Fei, L., Fergus, R., & Perona, P. (2004, June). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178-178.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 770-778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Kim, Kyung-A., & Chung, M.-A. (2022). Analysis of the Status of Artificial Medical Intelligence Technology Based on Big Data. *Korean Journal of Artificial Intelligence*, *10*(2), 13-18.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Retrieved May 07, 2023 from https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

LeCun, Y., Huang, F. J., & Bottou, L. (2004, June). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 2, II-104.

Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *ArXiv:1707.09835*

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.

Thompson, N. C., Greenewald, K., Lee, K., & Manso, G. F. (2020). The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*.