
Searching of Training Images with Rich Features Required for Generalization Performance of CNN Models Using Interactive Genetic Algorithms

Guangsheng Shao , [Fusaomi Nagata](#) ^{*} , Shiori Nakashima , Akimasa Otsuka , Keigo Watanabe , [Maki K. Habib](#)

Posted Date: 20 April 2026

doi: 10.20944/preprints202604.1356.v1

Keywords: interactive genetic algorithms (GAs); convolutional neural network (CNN); training parameters; transfer learning; image usage rate; VGG19; generalization ability



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Searching of Training Images with Rich Features Required for Generalization Performance of CNN Models Using Interactive Genetic Algorithms

Guangsheng Shao¹, Fusaomi Nagata^{1,*}, Shiori Nakashima², Akimasa Otsuka¹, Keigo Watanabe^{3,4} and Maki K. Habib⁵

¹ Graduate School of Engineering, Sanyo-Onoda City University, Yamaguchi 756-0884, Japan

² Department of Mechanical Engineering, Faculty of Engineering, Sanyo-Onoda City University, Yamaguchi 756-0884, Japan

³ International Pacific University (IPU), Tokyo Campus, Ichikawa 272-0001, Japan

⁴ Okayama University, Okayama 700-8530, Japan

⁵ Mechanical Engineering Department, SSE, The American University in Cairo, New Cairo 11835, Egypt

* Correspondence: nagata@rs.socu.ac.jp

Abstract

Selecting training parameters for convolutional neural networks (CNNs) and determining the amount of training data required for reliable generalization remain challenging and often time-consuming tasks, typically relying on manual trial-and-error. While genetic algorithms (GAs) have been applied to hyperparameter tuning, less attention has been given to how the proportion of training data influences generalization performance. In this study, we propose an interactive GA-based framework that simultaneously optimizes key training parameters and the image usage rate, defined as the proportion of training images used during learning. The approach is implemented within a MATLAB-based environment, allowing parameters to be adjusted dynamically during the optimization process. Experimental results on datasets including CIFAR-10 and EuroSat show that the proposed method can achieve classification performance comparable to manually tuned models while using a reduced portion of the available training data. In particular, similar accuracy levels were obtained with image usage rates in the range of approximately 70–95%, suggesting that not all training samples contribute equally to model performance. These findings indicate that incorporating data usage into the optimization process can support more efficient CNN training and provide practical guidance for selecting both training parameters and data subsets in practical applications. .

Keywords: interactive genetic algorithms (GAs); convolutional neural network (CNN); training parameters; transfer learning; image usage rate; VGG19; generalization ability

1. Introduction

In the production facilities for various industrial products and materials, inspection processes are established to verify that the finished products do not contain defects. Although some aspects of product inspection within the inspection process have been automated, there remains a significant reliance on visual inspection by skilled inspectors. In recent years, numerous attempts have been reported to apply deep learning technologies, such as CNNs specialized for image recognition and AI models that fuse CNNs with SVMs as feature extractors, to product defect detection. A highly active research field has emerged, with new techniques being published daily, indicating a continuous evolution towards higher performance [1–3].

The first requirement when training a CNN model for defect detection is the collection of image data for training. The parameters that must be set during training include: for example, maximum number of epochs, mini-batch size, learning rate, number of images used for training to achieve generalization. In the MATLAB application currently under development in our laboratory [4,5],

while it is possible to adjust parameters including these four items when training various types of defect detection models, the current practice involves operators relying on intuition to determine such parameters through trial and error, making it a laborious task.

Genetic Algorithms (GAs) introduced by Holland can simulate the process of natural evolution to find optimal or near-optimal solutions to complex problems [6]. By iteratively applying selection, crossover, and mutation operations, GAs evolve a population of candidate solutions toward better performance. GAs are widely used in optimization, scheduling, and machine learning tasks where it is difficult for traditional gradient-based methods to be applied or to solve them.

As for the design of machine learning models as CNN using GAs, for example, Xie and Yuille proposed an idea of encoding method to represent each network structure in a fixed-length binary code, in which GAs were adopted to efficiently explore the large search space as network structure resulting in demonstrating its performance to find high-quality structures [7]. Kim et al. proposed how to use feature selection via evolutionary algorithms to remove the irrelevant deep features. It is reported that their proposed method could minimize the computational complexity and the amount of overfitting while maintaining a good quality of representation. It is demonstrated that the improvement of the filter representation by performing experiments on three data sets of CIFAR10, metal surface defects, and variation of MNIST and by analyzing the classification performance as well as the variance of the filter [8].

Bakhshi et al. proposed a GA that could efficiently explore a defined space of potentially suitable CNN architectures and simultaneously optimize their hyper parameters, for a given image classification task. The fast automatic optimization model was named fast-CNN and employed to find competitive CNN architectures for an image classification task on CIFAR10 dataset [9]. Sun et al. proposed an automatic CNN architecture design method by using genetic algorithms, to efficiently deal with image classification tasks. The merit of the proposed algorithm seems that domain knowledge of CNNs is not required to users, while they can still obtain a promising CNN architecture for the given images [10]. Lima Mendes et al. proposed the gaCNN, which is a hybrid classification architecture consisting of a CNN and a GA [11]. The gaCNN utilized heterogeneous activation functions to classify images, optimizing its hyper parameters and activation functions automatically, regardless of the analyzed dataset. The results showed that gaCNN was able to identify good architecture. Around the same time, Lee et al. showed the possibility of optimizing network architectures using GA, in which the search space included both network structure configuration and hyper parameters. They used an amyloid brain image dataset, that is used for Alzheimer's disease diagnosis, to verify the performance of their proposed algorithm. It is reported that their proposed algorithm worked better than the Genetic CNN [7] by 11.73% on a given classification task [12]. Josiah et al. introduced that GA and machine learning have a long history of development and use in chemistry. In their review, it is focused on how GA and machine learning have been used in conjunction with chemical simulation techniques to advance understanding of surface chemistry, examining the history, recent work, and overall success of these applications [13]. Domashova et al. developed a software that generates a neural network with the best parameters for solving classification problems, in their paper, the process of population formation demonstrated the choice of the fitness function and parent selection method, and modifications of the crossover and mutation operators are introduced to ensure the operability of the algorithm on variable sized individuals [14].

Ali et al. systematically evaluated the performance of GA optimizer in tuning machine learning hyper parameters while comparing to common other techniques such as grid search, random search, and Bayesian optimization. It is reported that GA slightly outperformed other methods with respect to the optimality due to its general ability to pick any continuous values within the search range [15]. Santoso et al. contributed to the automatic tuning of hyper parameters to train CNN models using genetic algorithms [16]. Their proposed method was evaluated using MNIST dataset. The experiments results showed that using a genetic algorithm for tuning hyper parameters automatically, the accuracy of validation data is 97.02% and the accuracy of training data is 99.77%. Abdelaziz et al. employed

CNNs and CNNs combined with a GA as intelligent models for energy consumption prediction. The GA was utilized to fine-tune some of CNN parameter settings. It is reported that the CNN with the GA outperformed the CNN model with respect to the accuracy and standard error metrics [17]. Also, Rom et al. reviewed six prominent hyper parameter tuning methods such as manual tuning, grid search, random search, Bayesian optimization (BO), particle swarm optimization (PSO), and GA [18]. Their comparative analysis highlighted the strengths, weaknesses, and appropriate application contexts for each method. However, it seems that there was almost no discussion about how to extract the minimum amount of training data necessary to achieve generalization performance.

Caparrini and Arroyo pointed out the problem that traditional approaches such as manual tuning, grid search, and random search become inefficient when dealing with complex, high-dimensional search spaces. To cope with the limitation, they introduced an open-source package named *mloptimizer* that implements GA-based hyper parameter optimization for machine learning models. The package is designed to be easy to use, highly customizable and reproducible [19]. Yilmaz and Kuş employed a GA to optimize activation function, padding, number of filters, kernel size, dropout rate, pooling size, and batch size. The optimized CNN was trained on brain magnetic resonance imaging (MRI) images of multiple sclerosis and validated on Alzheimer's MRI and COVID-19 chest X-ray datasets. It is reported that the results showed substantial improvements across all datasets [20].

This paper aims to develop a system that employs interactive genetic algorithms [21–23] to attempt the automatic adjustment of training parameters, thereby proposing guiding conditions to assist engineers and researchers, in constructing desirable deep learning models such as CNN. By interacting with the user through our developed MATLAB application that provides the functionality of interactive GA, it becomes possible to systematically build user's desired CNN models, e.g., for defect detection of an industrial material or product, trained based on only rich featured images extracted from an original big dataset. The effectiveness and promise of the proposed system is shown through CNN trainings using four different kind of dataset.

This work differs from existing GA-based CNN optimization approaches by explicitly incorporating image usage rate as a controllable parameter within the evolutionary search, enabling the identification of reduced training subsets that retain relevant features for maintaining generalization performance. Unlike conventional hyper parameter tuning, the proposed method considers both training parameters and data usage within a unified framework, providing a practical approach for reducing dataset size while preserving classification accuracy.

2. Genotype and GA Operation

2.1. Genetic Code

In our approach for CNN model design, four parameters such as max epochs, mini-batch size, learning rate, and image extraction (usage) rate are optimized by the interactive GA. Regarding the extraction (usage) rate from an original training dataset, for example, if the value of 100 is set, all the training data are used, also if 80 is set, 80% of the training dataset, extracted randomly, are used for training the deep learning model. These four phenotypes are converted into genotypes, i.e., binary form using 8 bits, as shown in Figure 1. For example, in this experiment, the population consists of 12 individuals ($N=12$), including two elite individuals ($E=2$), all of whom have a gene length of 32 bits ($=8 \text{ bits} \times 4$). Table 1 shows an example of parameters used for the GA search in this experiment, in which values from No. 6 to No. 10 can be interactively manually changed even in GA operation while observing the progress of optimization.

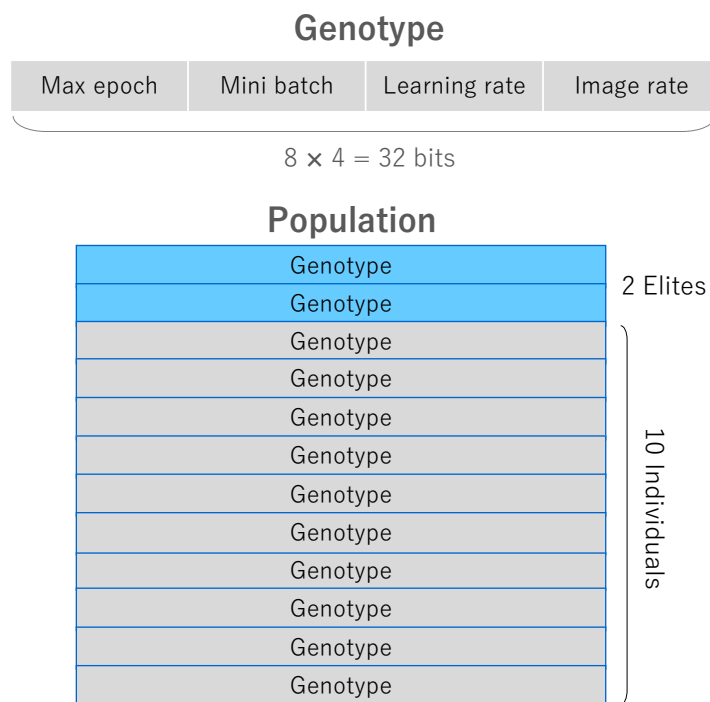


Figure 1. Structure of genotype and population for GA operation.

Table 1. Example of parameters in GA operation, and search spaces for Max epoch, Mini batch size, Learning rate, and Image usage rate.

| | |
|--------------------------|---------------------------------|
| 1. Population size N | 12 |
| 2. Number of elites E | 2 |
| 3. Selection | Tournament selection |
| 4. Crossover | Uniform crossover (rate=50%) |
| 5. Mutation | Random mutation (rate=3.13%) |
| 6. Number of generations | 30 |
| 7. Max epoch | Search range: 20 to 50 |
| 8. Mini batch size | Search range: 4 to 64 |
| 9. Learning rate | Search range: 0.00005 to 0.0005 |
| 10. Image usage rate [%] | Search range: 70 to 90 |

2.2. GA Flow

As shown in Figure 2, to generate the next generation ($N=12$), selection is first performed. This time, tournament selection is employed: three individuals are randomly selected from among those excluding the elite ($E=2$), and the most fit among them are retained for the next generation. This operation is performed $N-E$ times, where N is the total population size and E is the number of elites, generating the same number of individuals for the next generation. Crossover and mutation are two basic operation of GA. Crossover is performed after selection. This time, uniform crossover with a 50% crossover probability was employed. Also, the mutation rate was set to $1/32 = 3.13\%$, taking into account the bit length of the individuals shown in Figure 1.

In addition, the upper and lower bounds for the four search parameters shown in Table 1, along with the number of generations for evolving the population, can be interactively modified even during GA execution. The interactive aspect allows users to adjust parameter search ranges and optimization constraints during runtime, enabling human-in-the-loop guidance that can influence the search process and adapt it based on intermediate results.

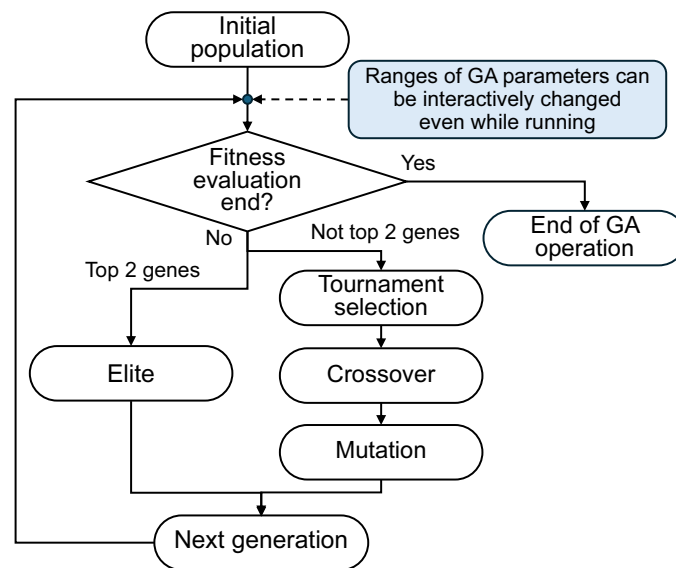


Figure 2. Flow of the proposed interactive GA. In the GA operation, lower and upper values of search spaces can arbitrarily be changed by human.

2.3. GA Parameter

The image extraction rate, one of the four GA parameters to be optimized, is the proportion X [%] of images actually used for training out of all images available for learning. When higher classification accuracy is achieved, that is, when the best individual is updated, the extracted images comprising X [%] used for training at that time are stored for subsequent evaluation. Images extracted and stored through this process can be regarded as the minimum training images required for the designed CNN to acquire sufficient generalization performance on test images.

2.4. Evaluation Function as a Maximization Problem

Fitness F to rank individuals is calculated by the evaluation function given by

$$F = \gamma A_c + (1 - \gamma) \frac{ImageRate_min}{ImageUsageRate} 100 \quad (1)$$

where A_c [%] and $ImageUsageRate$ are the classification accuracy obtained through a training process using an individual and the image usage rate at that time, respectively. $ImageRate_min$ is the lower limit value in searching the image usage rate, e.g., 70 [%] in Table 1. Also, γ is the weight coefficient for A_c . Individuals with larger F can survive to the next generation through the GA operation.

The formulation balances classification accuracy and data usage, enabling exploration of trade-offs between performance and training data size. The underlying assumption is that images contributing most to classification accuracy contain more discriminative features. By incorporating the image usage rate into the fitness function, the GA implicitly promotes the selection of subsets that retain relevant information while reducing redundancy. This process can be interpreted as an evolutionary sampling mechanism in the training data space that supports the identification of effective training subsets for maintaining generalization performance.

3. Training and Evaluation of CNN (sssNet)

3.1. CNN (sssNet) Built by Transfer Learning

Transfer Learning is a method that leverages knowledge obtained from a source domain to assist learning in a target domain, and it is particularly effective when the target data are limited [25]. Unlike traditional machine learning, transfer learning does not require that the training and testing data come from the same distribution, thus it can significantly improve the generalization performance of models.

In recent years, transfer learning in deep learning is commonly implemented by using pre-trained models followed by fine-tuning, such as convolutional neural networks pre-trained on ImageNet. Transfer learning based on a pre-trained CNN model can significantly reduce training time to build a new role-playing CNN one compared to that trained from scratch.

3.2. In Case of Parameter Tuning by Manual Setting by B4 Students

Here, seven final-year undergraduate students from our laboratory each manually trained a CNN model and evaluated it using test images, while adjusting parameters through trial and error. Images extracted from the CIFAR10 dataset are used for evaluation. CIFAR10 dataset comprises 50,000 training images and 10,000 test images across ten classes, as shown in Figure 3, including cat, bird, and airplane, with a resolution of $32 \times 32 \times 3$. In the experiment, 500 images per one category (500 images \times 10 categories = totally 5,000 images) were extracted for training, and 300 images per category (300 images \times 10 categories = totally 3,000 images) for testing.

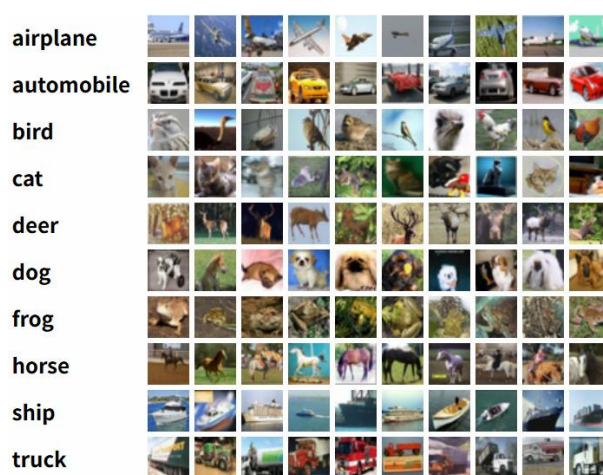


Figure 3. CIFAR-10 dataset [24] used for evaluation.

CNN models named *sssNet* are designed based on transfer learning technique. This time, VGG19 was employed as the backbone of the transfer learning, which had demonstrated high classification performance in our previous applications. The fully connected layers of VGG19 were replaced from one designed for 1000-class classification to one for 10-class classification [26], as shown in Figure 4.

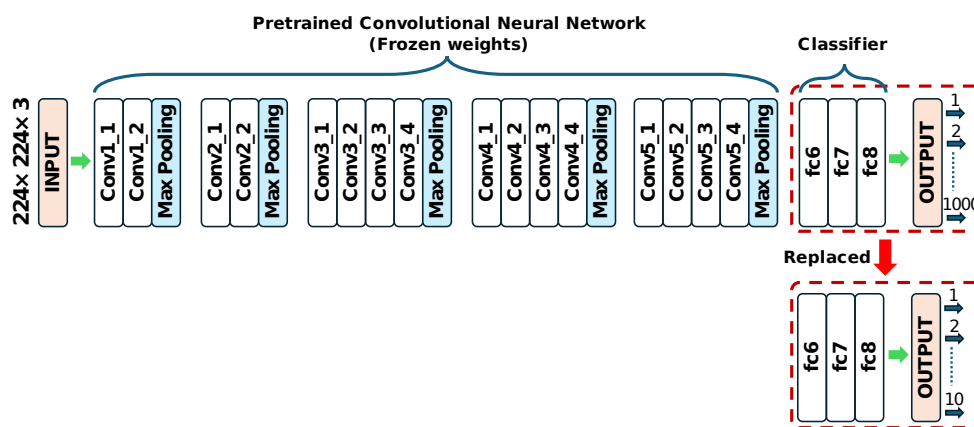


Figure 4. CNN model designed by transfer learning based on VGG19, which can cope with the domain for 10 classes shown in Figure 3.

Table 2 shows the classification accuracies [%] on 3,000 test images predicted by each of the five *sssNet* models obtained from the five training sessions (from trial 1 to 5) conducted by seven students. Note that in this case, each training was conducted over 40 to 100 epochs to achieve sufficient accuracy on the test images.

Table 2. Classification accuracies [%] tried by seven students.

| Trial | 1st | 2nd | 3rd | 4th | 5th |
|-----------|-------|-------|-------|-------|-------|
| Student 1 | 81.13 | 80.90 | 82.93 | 80.93 | 79.90 |
| Student 2 | 81.03 | 81.30 | 79.93 | 78.27 | 81.73 |
| Student 3 | 82.17 | 81.07 | 80.43 | 78.57 | 82.27 |
| Student 4 | 81.97 | 80.80 | 82.30 | 78.53 | 81.97 |
| Student 5 | 79.63 | 81.53 | 80.63 | 80.93 | 78.90 |
| Student 6 | 82.93 | 80.23 | 80.60 | 80.46 | 80.90 |
| Student 7 | 82.10 | 80.23 | 80.67 | 81.20 | 73.13 |

3.3. In Case of Automatic Tuning Using the Proposed Interactive GA

Table 3 shows results of five trial tests, in which classification accuracy achieved by each best individual when automatically training the CNN using the proposed method based on GA. These results confirm that using GA enables the automatic acquisition of parameters capable of achieving high accuracy. Furthermore, as indicated by the extracted rate of images used in CNN training, it was confirmed that equivalent accuracy to that achieved using the entire dataset could be obtained without using all the images in the dataset. Note that in the trial tests, the weight coefficient γ was set to 1.

The results in Table 3 indicate that the proposed method is able to identify subsets of images that contribute more effectively to classification performance. This observation suggests that images contributing more to classification accuracy tend to contain more discriminative features. By incorporating the image usage rate into the optimization process, the GA promotes the selection of subsets that retain relevant information while reducing redundancy. This behavior supports interpreting the proposed method as a data-driven sampling mechanism for identifying effective training subsets that maintain generalization performance.

Table 3. Best results obtained using the proposed GA-based optimization approach.

| Trial | 1st | 2nd | 3rd | 4th | 5th |
|-----------------------------|--------|--------|--------|--------|--------|
| Generation | 9 | 3 | 18 | 27 | 2 |
| Accuracy [%] | 82.1 | 82.2 | 82.6 | 82.8 | 82.5 |
| Image usage rate [%] | 86 | 93 | 95 | 97 | 90 |
| Mini batch size | 22 | 23 | 19 | 24 | 6 |
| Max Epoch | 39 | 33 | 42 | 48 | 31 |
| Learning rate | 0.0003 | 0.0004 | 0.0004 | 0.0004 | 0.0001 |

4. Additional Experiments

Three additional datasets from different domains were used to further evaluate the proposed method.

4.1. EuroSat

The first image dataset is EuroSat [27], as shown in Figure 5, which consists of aerial views of ten types of terrains captured by artificial satellites, including Forest, River, and Highway, with a resolution of $64 \times 64 \times 3$. Just like CIFAR-10, in the experiment, 500 images per one category (500 images \times 10 categories = 5,000 images) were extracted for training, and 300 images per one category (300 images \times 10 categories = 3,000 images) for testing.

Similarly, the training images in EuroSat were used for training a CNN model, in which the conventional manual setting was applied, and the trained model was evaluated using test images. As can be seen from Table 4, average classification accuracy across several trials by a B4 student was approximately 96%.

On the other hand, when trained using the proposed GA as with CIFAR-10, an accuracy of 96.8% was achieved even with a learning rate of 0.0005, 7 epochs, a mini-batch size of 28, and an image usage rate of 80%. An accuracy of 96.4% was obtained even with a significantly lower image utilization rate of 73%. These results are consistent with those obtained on CIFAR10, further supporting the effectiveness of the proposed approach across different image domains.

Table 4. Manual training results by a B4 student and GA training results by the proposed method.

| Trial | Manual 1st | Manual 2nd | GA 1st | GA 2nd |
|-----------------------------|------------|------------|--------|--------|
| Accuracy [%] | 96.3 | 96.6 | 96.4 | 96.8 |
| Image usage rate [%] | 100 | 100 | 73 | 80 |
| Mini-batch size | 32 | 32 | 9 | 28 |
| Max epoch | 100 | 62 | 10 | 7 |
| Learning rate | 0.0005 | 0.0005 | 0.0003 | 0.0005 |

4.2. Dataset Consisting of Images Labeled with Capping, Noise, and Voice

The second dataset originates from B4 students' graduation study in our laboratory, capturing the waveform of sounds emitted by a robotic arm during robotic capping task. As shown in Figure 6, this image dataset consists of three categories: Capping, Noise, and Voice, with a resolution of $469 \times 469 \times 3$. Each category contains 300 training images ($300 \text{ images} \times 3 \text{ categories} = 900 \text{ images}$) and 50 test images ($50 \text{ images} \times 3 \text{ categories} = 150 \text{ images}$). The role of a CNN model trained using this image dataset is the distinction between normal capping sound and other sounds such as noise and voice.

Unlike previous datasets, this dataset achieved 100% accuracy when evaluated on test images after pre-trained using traditional manual methods. Therefore, the objective of the research has shifted to maintaining 100% accuracy while minimizing the amount of training data used as much as possible.

After training the GA using the proposed method on this dataset, it achieved 100% accuracy multiple times when using only a subset of the dataset. Therefore, based on the existing results, we interactively reduced the upper limit of the dataset extraction volume. As shown in Table 5, even when using less than half of the dataset, results with 100% accuracy can still be achieved.

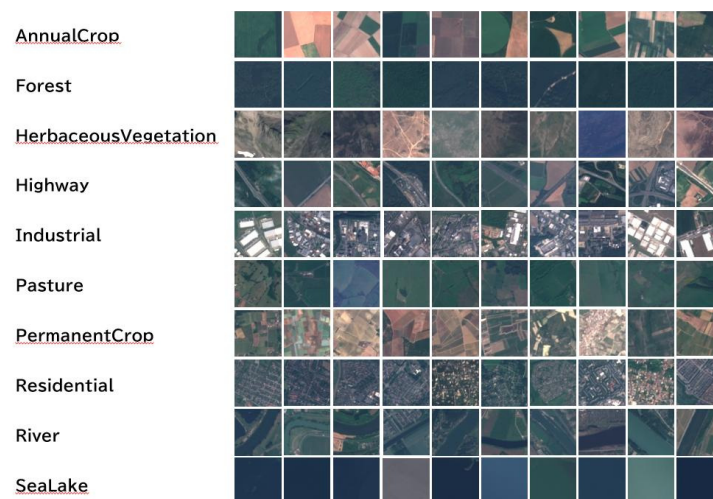


Figure 5. EuroSat dataset [27] used for evaluation.

Table 5. Results of GA Training.

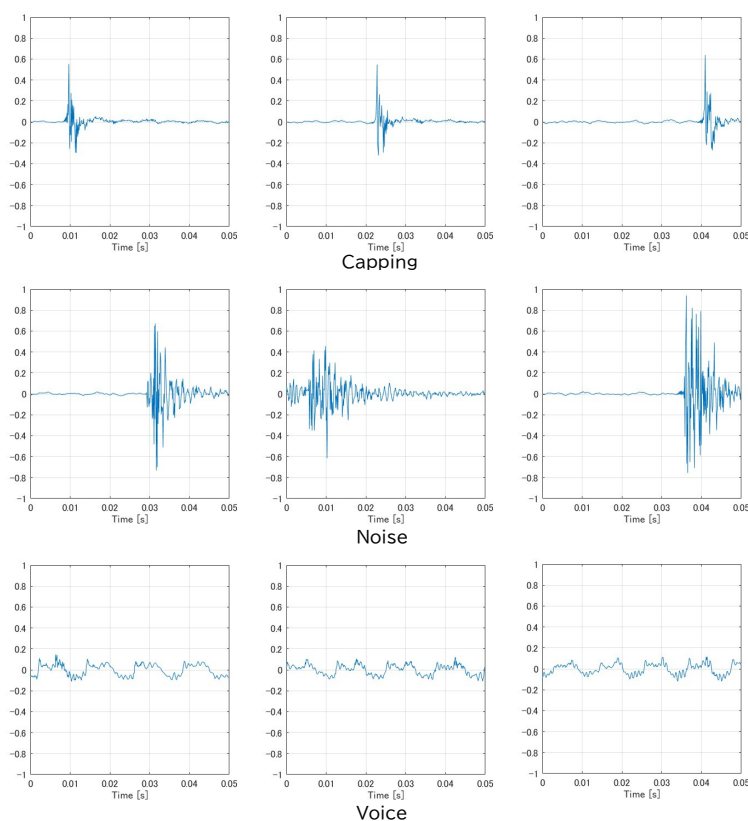
| Trial | 1st | 2nd | 3rd | 4th |
|-----------------------------|---------|---------|---------|--------|
| Accuracy [%] | 100 | 100 | 100 | 100 |
| Image usage rate [%] | 50 | 51 | 44 | 43 |
| Mini-batch size | 20 | 57 | 58 | 16 |
| Max epoch | 30 | 34 | 45 | 23 |
| Learning rate | 0.00006 | 0.00023 | 0.00005 | 0.0008 |

4.3. Another Dataset Consisting of Time-Series Sound Block Data

In this subsection, a sound block dataset obtained from a CNC machine tool is evaluated to extract rich-featured images. Figure 7 shows an example of extraction of one-line time-series sound blocks (SBs) from a WAV file, which is recorded from a general milling machine using a normal and a damaged end mills [28]. In this experiment, SB data are sampled with 0.01 [s], so that the length of a SB data file is $44,100 \times 0.01 = 441$. By normalizing a SB into [0,1] and arranging it to row direction 441 times, a BMP image with the resolution of 441×441 is easily generated as shown in Figure 8. Note that in giving BMP images for training to the input layer of the CNN model shown in Figure 4, the resolution is downsized to 224×224 .

Number of normal and anomaly SB data files for training is 500 each. Also, an equal number of test SB to check the generalization ability is prepared for both normal and anomaly categories. Table 6 shows the results of five trials.

While the proposed GA-based approach demonstrates consistent performance across different datasets, the present study focuses primarily on comparison with manual parameter tuning. A more comprehensive evaluation against other hyper parameter optimization techniques, such as Bayesian

**Figure 6.** Image dataset consisting of three classes, Capping, Noise, and Voice .

optimization, random search, and grid search, remains an important direction for future work to further assess relative efficiency and computational cost.

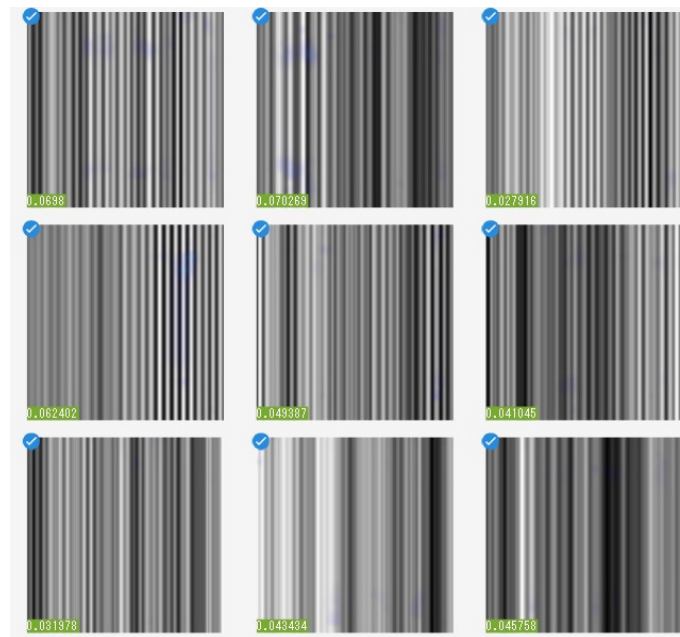


Figure 8. Examples of BMP images expanded from one-line SB data.

Table 6. Results of GA Training for sound block (SB) dataset.

| Trial | 1st | 2nd | 3rd | 4th | 5th |
|----------------------|---------|---------|---------|---------|---------|
| γ in Eq. (1) | 1 | 0.8 | 0.5 | 0.3 | 0.1 |
| Accuracy [%] | 97.3 | 97.3 | 97.6 | 97.0 | 96.8 |
| Image usage rate [%] | 79 | 70 | 60 | 50 | 45 |
| Mini-batch size | 40 | 40 | 33 | 57 | 32 |
| Max epoch | 32 | 57 | 36 | 50 | 81 |
| Learning rate | 0.00038 | 0.00048 | 0.00045 | 0.00026 | 0.00049 |

5. Conclusions

This study aimed to develop a system that proposes key learning conditions to serve as guidelines for engineers and researchers constructing desirable CNN models using interactive genetic algorithms

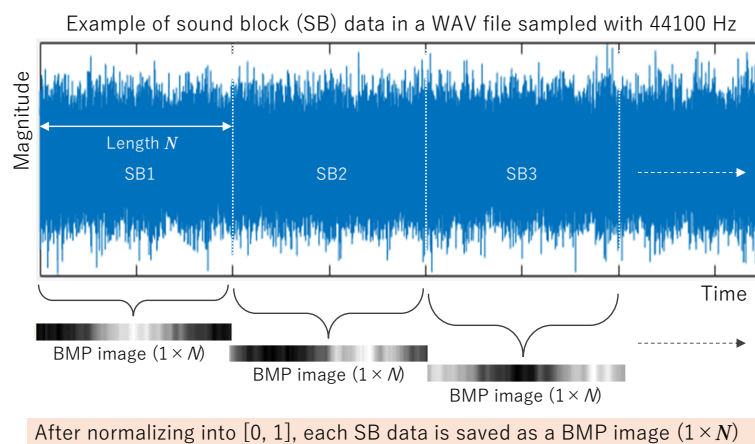


Figure 7. One line BMP converted from one SB [28].

(GA). Through the proposed framework, it becomes possible to construct a target CNN model by automatically adjusting training parameters within defined search ranges through user interaction. In particular, the method enables systematic exploration of the amount of training data required to achieve the desired generalization performance.

Experimental results obtained from four datasets, including CIFAR10, EuroSat, robotic task sound data, and time-series SB data, demonstrate that training subsets extracted through the proposed GA-based approach can achieve generalization performance comparable to that obtained using the full dataset. This indicates that effective training can be achieved using reduced data while preserving classification accuracy.

Although the proposed method demonstrates consistent performance across multiple datasets, the evaluation remains limited to selected domains. The generalization capability across significantly different domains or real industrial scenarios requires further investigation. In addition, while the GA-based optimization improves parameter selection, it introduces additional computational overhead due to iterative population evaluation, and future work should analyze the trade-off between optimization cost and performance gain.

Overall, the proposed method shows potential applicability to industrial defect detection and related tasks, subject to further validation on real-world datasets.

Author Contributions: Conceptualization, F.N., K.W. and M.K.H.; Methodology, F.N., A.O. and G.S.; Software, F.N. and G.S.; Validation, F.N., G.S. and S.N.; Formal analysis, F.N., G.S., S.N. and A.O.; Investigation, F.N., G.S. and S.N.; Resources, F.N., G.S. and S.N.; Data curation, G.S. and S.N.; Writing—original draft preparation, F.N.; Writing—review and editing, F.N., M.K.H. and K.W.; Visualization, F.N.; Supervision, F.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received the external funding as JSPS KAKENHI Grant Number JP25K07532.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Number JP25K07532

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

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-----|------------------------------|
| GA | Genetic Algorithm |
| CNN | Convolutional Neural Network |
| BO | Bayesian Optimization |
| PSO | Particle Swarm Optimization |

References

1. Karakas, B.; Kulluk, S. A Hybrid CNN-SVM Algorithm for Detecting Manufacturing Defects, *IEEE Access* **2025**, *13*, 192173–192188. doi: 10.1109/ACCESS.2025.3630968.
2. Wang, Q.; Wang, M.; Sun, J.; Chen, D.; Shi, P. Review of Surface-Defect Detection Methods for Industrial Products Based on Machine Vision, *IEEE Access* **2025**, *10*, 90668–90697. doi: 10.1109/ACCESS.2025.3571297.
3. Qiao, Q.; Hu, H.; Ahmad, A.; Wang, K. A Review of Metal Surface Defect Detection Technologies in Industrial Applications, *IEEE Access* **2025**, *13*, 48380–48400. doi: 10.1109/ACCESS.2025.3544578.
4. Nagata, F.; Nakashima, K.; Miki, K.; Arima, K.; Shimizu, T.; Watanabe, K.; Habib, M.K. Design and Evaluation Support System for Convolutional Neural Network, Support Vector Machine and Convolutional Autoencoder, *Measurements and Instrumentation for Machine Vision*, CRC Press, Taylor & Francis Group: Boca Raton, FL, USA, 2024; pp. 66–82. <https://doi.org/10.1201/9781003343783>.
5. Shimizu, T.; Nagata, F.; Habib, M.K.; Armina, K.; Otsuka, A.; Watanabe, K. Advanced Defect Detection in Wrap Film Products: A Hybrid Approach with Convolutional Neural Networks and One-Class Support Vector Machines with Variational Autoencoder-Derived Covariance Vectors, *Machines* **2024**, *12*(9), 1–20. doi: 10.3390/machines12090603.

6. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The MIT Press, 1992. <https://doi.org/10.7551/mitpress/1090.001.0001>
7. Xie, L.; Yuille, A. Genetic CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1388–1397. doi: 10.1109/ICCV.2017.154
8. Kim, J.; Lee, M.; Choi, J.; Seo, K. GA-based Filter Selection for Representation in Convolutional Neural Networks. *Computer Vision – ECCV 2018 Workshops. Lecture Notes in Computer Science*, 2018, 11132, 609–618, Springer, https://doi.org/10.1007/978-3-030-11018-5_48.
9. Bakhshi, A.; Noman, N.; Chen, Z.; Zamani, M.; Chalup, S.K. Fast Automatic Optimization of CNN Architectures for Image Classification Using Genetic Algorithm. In Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC), 2019; pp. 1283–1290. doi:10.1109/CEC.2019.8790197.
10. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G.; Lv J. Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification, *IEEE Transactions on Cybernetics*, 2020, 50(9), 3840–3854. doi: 10.1109/TCYB.2020.2983860.
11. Mendes, R.D.L.; Alves, A.H.D.S.; Gomes, M.D.S.; P. Luiz Lima Bertarini, P.L.L.; Amaral, L.R.D. gaCNN: Composing CNNs and GAs to Build an Optimized Hybrid Classification Architecture, In Proceedings of 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June 2021; pp. 79–86. doi: 10.1109/CEC45853.2021.9504850.
12. Lee, S.; Kim, J.; Kang, H.; Kang, D.Y.; Park, J. Genetic Algorithm Based Deep Learning Neural Network Structure and Hyperparameter Optimization. *Applied Science*, 2021, 11, 744. <https://doi.org/10.3390/app11020744>
13. Josiah R.; Julia R.S.; Bursten C.R. Genetic Algorithms and Machine Learning for Predicting Surface Composition, Structure, and Chemistry: A Historical Perspective and Assessment, *Chemistry of Materials*, 2021, 33(17), 6589–6615. <https://pubs.acs.org/doi/10.1021/acs.chemmater.1c00538>
14. Domashova, J.V., Emtseva, S.S., Fail, V.S., Gridin, A.S. Selecting an Optimal Architecture of Neural Network Using Genetic Algorithm, *Procedia Computer Science*, 2021, 190, 263–273. <https://doi.org/10.1016/j.procs.2021.06.036>.
15. Ali, A.; Jayaraman, R.; Azar, E.; Sleptchenko, A. A Systematic Assessment of Genetic Algorithm (GA) in Optimizing Machine Learning Model: A Case Study from Building Science, In Proceedings of 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Kuala Lumpur, Malaysia, 2022; pp. 384–389. doi: 10.1109/IEEM55944.2022.9989719.
16. Santoso, F.Y.; Sediyo, E.; Purnomo, H.D. Genetic Algorithm for Convolutional Neural Network Hyperparameter Tuning, In Proceedings of 2023 3rd International Conference on Computing and Information Technology (ICCIIT), Tabuk, Saudi Arabia, 13–14 September 2023; pp. 232–236. doi: 10.1109/ICCIIT58132.2023.10273928.
17. Abdelaziz, A.; Santos, V.; Dias, M.S. Convolutional Neural Network with Genetic Algorithm for Predicting Energy Consumption in Public Buildings, *IEEE Access*, 2023, 11, 64049–64069. doi: 10.1109/ACCESS.2023.3284470.
18. Rom, A.R.M.; Ibrahim, S.; Fadzil, A.F.A.; Mangshor, N.N.A.; Ghani, N.A.M. A Review of Hyperparameter Tuning Methods in Machine Learning, In Proceedings of 2025 6th International Conference on Artificial Intelligence and Data Sciences (AiDAS), 2025; pp. 1–6. doi: 10.1109/AiDAS67696.2025.11213530.
19. Caparrini, A.; Arroyo, J. mloptimizer: Genetic Algorithm-based Hyperparameter Optimization for Machine Learning Models in Python, *SoftwareX*, 2026, 34, 102567. <https://doi.org/10.1016/j.softx.2026.102567>.
20. Yilmaz, A.; Kuş, İ. General CNN Model for Biomedical Image Classification via Genetic Algorithm-based Hyperparameter Optimization, *Ain Shams Engineering Journal*, 2026, 17(1), 103891. <https://doi.org/10.1016/j.asej.2025.103891>.
21. Takagi, H. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation, In Proceedings of the IEEE, 2001, 89(9), 1275–1296. doi: 10.1109/5.949485.
22. Brintrup, A.M.; Ramsden, J.; Takagi, H.; Tiwari, A. Ergonomic Chair Design by Fusing Qualitative and Quantitative Criteria Using Interactive Genetic Algorithms, *IEEE Transactions on Evolutionary Computation*, 2008, 12(3), 343–354. doi: 10.1109/TEVC.2007.904343.
23. Katoch, S.; Chauhan, S.S.; Kumar, V. A Review on Genetic Algorithm: Past, Present, and Future, *Multimedia Tools and Applications*, 2021, 80(5), 8091–8126. <https://doi.org/10.1007/s11042-020-10139-6>.
24. CIFAR-10
<https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 25 February 2026)

25. Pan, S.J.; Yang, Q. A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, 2009, 22(10), 1345–1359. doi: 10.1109/TKDE.2009.191.
26. K. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-scale Image Recognition, In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 7-9, 2015; pp. 1–14.
27. EuroSat
<https://www.kaggle.com/datasets/ryanholbrook/eurosat/data> (accessed on 25 February 2026)
28. Nagata, F.; Morimoto, T.; Watanabe, K.; Habib, M.K. Design of Identification System Based on Machine Tools' Sounds Using Neural Networks. *Designs* 2025, 9, 121. <https://doi.org/10.3390/designs9050121>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.