Deep Convolutional Networks for Human Sketches by means of the Evolutionary Deep Learning

Saya Fujino
College of Engineering,
Osaka Prefecture University,
1-1 Gakuencho, Nakaku, Sakai,
Osaka 599-8531, Japan
Email: fujino@ss.cs.osakafu-u.ac.jp

Naoki Mori
Graduate School of Engineering,
Osaka Prefecture University,
1-1 Gakuencho, Nakaku, Sakai,
Osaka 599-8531, Japan
Email: mori@cs.osakafu-u.ac.jp

Keinosuke Matsumoto
Graduate School of Engineering,
Osaka Prefecture University,
1-1 Gakuencho, Nakaku, Sakai,
Osaka 599-8531, Japan
Email: matsu@cs.osakafu-u.ac.jp

***Abstract*—Recognition of human sketches is one of the most interesting and difficult issues in image recognition. Recently, deep convolutional neural networks (DCNNs) have been successfully applied to various image recognition tasks. Though the DCNN is a very powerful method, the high computational effort required to tune its hyperparameters represents a critical problem. In this paper, we propose a novel method called evolutionary deep learning (evoDL) that uses a genetic algorithm in order to obtain effective deep learning networks. The generalization ability of the network structure obtained using the proposed method is confirmed by a computer experiment.**

I. Introduction

Categorizing human sketches using a computer is one of the most important topics in the image recognition field. In fact, several studies have analyzed picture information using computer algorithms. However, three common problems arise in these studies:

- It is not sufficient to collect freehand drawings from different people, especially children and people who are not adept at drawing.
- Several researchers have analyzed the features of existing pictures, such as color histograms and picture composition, but they ignore the drawing mode.
- An important feature of pictures is the object class. The study presented in [1] focuses on sketch recognition and handwritten character recognition to predict a single object class. However, this type of approach cannot be applied to freehand drawings because they may be composed of various object classes such as animals, characters, or symbols.

One of the most powerful approaches for sketch recognition include the use of deep convolutional neural networks (DCNNs) [2]. We introduced a DCNN model as a sketch recognition module into the proposed interactive picture book [3]. However, tuning the parameters of the corresponding convolutional neural networks (CNNs) requires considerable effort. In this paper, we propose a novel framework for evolving deep learning (DL) architectures. The proposed method, called evolutionary deep learning (evoDL), is expected to be capable of finding novel architectures for DL. As the first step, we focus on the evolution of hyperparameters in a DCNN. A genetic algorithm (GA) [4], which is a search-and-optimization algorithm based on mechanisms of natural evolution, is adopted as the evolutionary part of evoDL. Next, we define AlexNet [2] as the basic structure of the CNN and apply evoDL to optimize both the activation functions and the tuning of parameters. Finally, we demonstrate the generalization ability of the network structure by applying the evolved DCNN obtained by evoDL to another type of sketch recognition problem using a different dataset.

978-1-5090-4917-2/17$31.00

II. Related Works

In the research presented in [1], a large-scale exploration of human sketches was performed. The aim was to compare human performance with computational recognition methods. To perform that comparison, the researchers used a dataset of 20,000 unique sketches that were evenly distributed over 250 object categories. A key point of this dataset is that it consisted of non-expert human sketches. Then, a study was performed to determine the average recognition rate of humans over this dataset. Regarding the computational part, the identification process used a method similar to SIFT (scale-invariant feature transform) for feature extraction, a bag-of-words model for sketch representation, and multi-class support vector machines (SVMs) for classification. The researchers used a state-of-the-art machine learning technique called parallel SVMs. Bertla et al. also proposed sketch recognition using a multi-class SVM classifier based on a binary decision tree [5]. However, there no research has been reported on sketch recognition for an interactive picture book using a DCNN model. The research of Sketch-a-net [6] has been reported to compare recognition ability between humans and DCNNs. In fact, Sketch-a-net using a multi-scale DCNN can beat humans at sketch recognition. Research on combining GA with CNN has also been reported [7], but this study did not consider directly the selection of the CNN hyperparameters.

III. DCNN

In this section we present the used DCNN model.

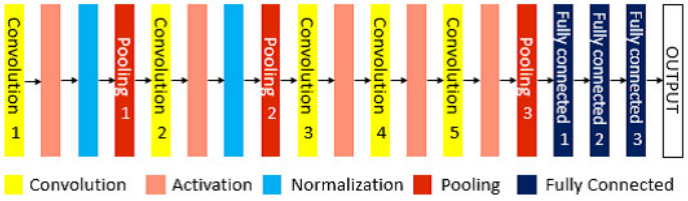

Fig. 1. The architecture of AlexNet

TABLE I
THE GENE OF INDIVIDUALS IN EVODL

Design Variables	Allele
The number of filter (NF)	16, 32, 64
Filter size (FS)	3, 5, 7, 9, 11
Pooling size (PS)	3, 5, 7
Nodes in fully connected layer 1 (NL1)	512, 1024, 2048, 4096
Nodes in fully connected layer 2 (NL2)	128, 256, 512, 1024
Batch size (BS)	10, 20, 30
Activate function with ReLU (Re)	1 (use), 0 (not use)

A. CNN Outline

A CNN is among the DL methods most widely used in the image recognition field. In the ImageNet LSVRC contest, it was reported that using a DCNN model shows a remarkable improvement of performance compared to conventional image recognition approaches [2]. Currently, using a DCNN is admitted as one of the most powerful methods for image recognition. The basic structure of a DCNN contains one or more convolutional layers, pooling layers, and fully-connected layers. The fully-connected layers are located after the convolutional and pooling layers for fine-tuning.

Moreover, the DCNN has the advantage of using a small number of learnable parameters. The convolutional layers are its core building blocks with parameters from a set of learnable filters. An input image is passed through the convolutional layers, and dot products are computed between the entries of the filter and the image at any given position. Pooling layers are also a component of the CNN architecture. We used max-pooling that operates independently at every depth slice and resizes the input using the *max* operation.

B. AlexNet

We now present the architecture of AlexNet [2], which is shown in Fig. 1.

AlexNet is one of the most popular architectures for a DCNN. It contains eight layers: the first five are convolutional and the remaining three are fully-connected. Using a *softmax* function, the last fully-connected layer produces an output consisting of four class labels. The response-normalization layers follow the first and second convolutional layers. The pooling layers follow both the response-normalization layers and the fifth convolutional layer. To reduce overfitting of the data during the training stage, we use the dropout technique [8][9] in the first two fully connected layers. Moreover, we rely on ReLU nonlinearity, which is applied to the output of every convolutional and fully connected layer.

IV. EVODL FOR DCNNS

As stated before, DCNNs are suitable for image recognition. However, setting adequate structures and hyperparameters of DCNNs require great effort. Moreover, the performance of a DCNN is strongly dependent on the network characteristics, so finding the adequate structure and hyperparameters are very important issues. Though grid search solve these issues, a more efficient approach is required given the large hyperparameters space.

Therefore, we propose evoDL as a novel evolutionary approach for finding a finely tuned DL model. The evoDL approach can be applied to any kind of DL model. In this paper, we apply evoDL to tune the hyperparameters of a DCNN, which uses a GA as optimization method.

To find the finely tuned DCNN model, the following characteristics are considered:

Structure: The basic structure of the DCNN such as number and type of layers, type of activation function, and methods of fine tuning.
Hyperparameters (integer): Hyperparameters represented by integer values such as number of filters and shape of filter, max pooling, and convolutional layers.
Hyperparameters (real number): Hyperparameters represented by real numbers such as weight values in the filter and convolutional layers.

In this study, we only focus on hyperparameters represented by integer or Boolean values such as the number of filters, the filter and the max pooling sizes, by using an activation function. In addition, we use the conventional GA framework except for the representation of the genotype. Moreover, uniform crossover, uniform transition type mutation, and tournament selection are applied in this order to the population of the GA.

A. Representation

In evoDL, solutions are represented as individuals and each individual has a chromosome. A symbol, which makes up the chromosome, is known as a gene. The position of a gene in the chromosome is known as a locus, and possible genes in the different loci are called alleles.

In this study, we set a unique number of alleles for each locus because the degree of contribution to the network quality depends on each design variable, here represented by specific genes. Table I shows the representation details.

Fig. 2 shows an example of the individuals chromosome. Each string in the locus is related to the strings in Table I. The basic structure of the CNN is fixed according to AlexNet (see Section III-B). In Fig. 2, the seven genes labeled as "Re" represent the flag of the ReLU functions, which are set just after the five convolutional layers and two fully-connected

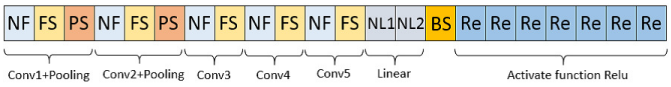

Fig. 2. An example of genotype

layers. If gene Re is 1, the corresponding ReLU function is set after the corresponding layer. Although AlexNet does not use ReLU functions in fully connected layers, our system can consider ReLU functions in these layers.

B. Fitness Function

In this study, fitness function $F(s)$ of the GA is defined as the sum of the accuracy in a k-fold cross-validation using training data, where s is an individual related to a specific CNN.

In evoDL, fitness function $F(s)$ of this problem is represented as follows:

$$F(s) = \sum_{i=1}^{k} f_i^{\mathrm{acc}}(s), \tag{1}$$

where s denotes an individual and $f_i^{\mathrm{acc}}(s)$ is the accuracy function of a k-fold cross-validation after i epochs.

C. Speed-Up Methods

To consider the GA-based CNN evolution, we need to use speed-up methods because the training of the CNN takes a long time. In this study, we introduce two types of speed-up methods.

The first method, called "genotype and fitness memory," is applied to the GA. Once the fitness of an individual is calculated, the relation between the genotype and the fitness value is recorded using this method. If the same genotype appears in the GA, the fitness value is retrieved from memory instead of being recalculated. Although the cost of saving the genotype and fitness value is not negligible, this method reduces the total simulation time given the heavy CNN training time.

The second method involves introducing a cutoff point into the CNN training stage. In this study, we stop the CNN training process when the improved quantity of the accuracy rate of 10 epochs is less than 1% compared to that of 1 epoch. Thus, we remove the useless training time of lethal individuals using this method.

D. Algorithm

Fig. 3 shows the process for creating a generation in the GA. The GA has three stages, namely, crossover, mutation, and survival selection.

V. Experiments

In this section, we show the two experimental results from evolving a DCNN model using evoDL as well as the generalization ability of the obtained network structure. The main purpose of this experiment was to confirm the validity of evoDL.

Algorithm 1 GENERATION-GA

Create random population
while $|P_i'| < popsize$ **do**
  Select parents
  Apply crossover to them
  Add children to P_i'
for all individual $\in P_i'$ **do**
  Apply mutation to it
while $|P_{i+1}| < popsize - 1$ **do**
  Apply survival selection to P_i'
  Add individual to P_{i+1}
Add elite to P_{i+1}

Fig. 3. One generation of GA. $popsize$ is the number of individuals in a population, and P_i is a population in generation i.

A. Target Problem

We used bitonal sketch images as the recognition problem for the DCNN. In the first experiment, the DCNN model was evolved for the relatively simple task of recognizing four classes (i.e., *face, animal, character, and symbol*) from bitonal sketch images having a size of 28×28 pixels.

Each class contained 250 images, so the total amount of data was $250 \times 4 = 1,000$ images. We divided the 1,000 images into two groups as follows:

Training data in evolution Each class contained 225 images, hence the total amount of training data was equal to 900 images. In the GA, the fitness value in subsection IV-B was calculated from the results of a threefold cross-validation after 100 epochs. Each individual was translated into the corresponding CNN model and trained using 600 images with 100 epochs. After this, we obtained the CNN accuracy and loss rates from 300 images, which were the remainder of the $900 - 600$ images. Because of the threefold cross-validation, we repeated this process three times and obtained the fitness value using eq. (1).
Test data In our test data, each data class contained 25 images. Since there were four classes, $25 \times 4 = 100$ images were used to validate the obtained CNN model defined by a unique individual in the final generation of the GA.

B. Experimental Conditions

Table II shows the setting of the GA used for evoDL.

We use simple mutation that changes a gene to a random allele because target problem is not very complex.

1) Results and Discussion: The training data accuracy reached 1.0 in approximately 100 epochs and then fluctuated while keeping a high accuracy rate. The test data accuracy reached 0.98 in approximately 30 epochs and then stabilized around 0.95. However, after 500 epochs, the test data accuracy declined below 0.95 because of overfitting. Since the fitness in the GA was evaluated after 100 epochs, only individuals with

TABLE II
SETTING OF GA

generation size	20
population size	20
chromosome length	22
crossover type	uniform
crossover rate	1.0
mutation rate of each locus	$\frac{1}{L}$ (L is chromosome length)
selection	tournament selection
tournament size	3

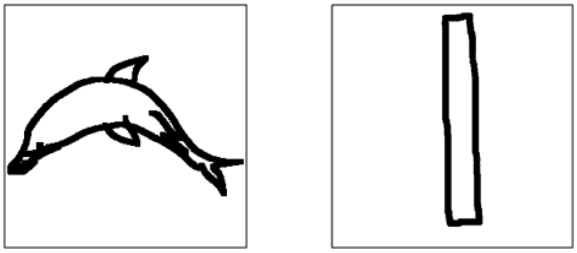

Fig. 4. Images leading to identification failure in test

CNN hyperparameters that finished training before 100 epochs could survive. In this experiment, because the dataset was not so large, our system easily found suitable individuals for the CNN. In addition, we tried another fitness function using a negative value of the total losses instead of the total accuracy fitness in eq. (1) and also obtained high accuracy. The training loss became almost 0.0 in approximately 30 epochs, increasing occasionally. On the other hand, the test loss increased with oscillation. In fact, given the small size of the test dataset, overfitting caused that oscillation.

The training data accuracy reached 1.0 in approximately 100 epochs and then fluctuated while keeping a high accuracy rate. The test data accuracy again reached 0.98 in approximately 30 epochs and then stabilized around 0.95. As with the other fitness function, after 500 epochs, the test accuracy declined below 0.95 because of overfitting. Since the fitness in GA was evaluated after 100 epochs, only individuals with CNN hyperparameters that finished training before 100 epochs could survive.

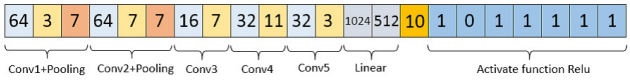

Fig. 5. The genotype of best individuals in test data

Fig. 4 shows the two images in the test dataset that the CNN model with a 0.98 test data accuracy failed to recognize. The left sketch represents a dolphin that belongs to the animal class, whereas the right sketch represents a rectangle that belongs to the symbol class. The CNN misclassified the dolphin as belonging to the face class and the rectangle as belonging to the character class. The dolphin might be hard to recognize since there were few water-creature images in our dataset. On the other hand, since the difference between the rectangle and some characters was very small, the misclassification can be reasonable.

Fig. 5 shows the genotype of elite individuals who performed best reaching a test accuracy of 0.98. In these individuals, ReLU functions were used in every except the second convolutional layer. This result suggests that the ReLU function is important to obtain an effective CNN. On the other hand, in the obtained network the number of filters and nodes in the fully connected layer were both smaller than those of AlexNet. This is possibly because our network was evolved to recognize bitonal images, whereas AlexNet was tuned for RGB images.

In this experiment, because the dataset was not very large, evoDL easily found suitable individuals for the DCNN. We tried another fitness function using a negative value of the total losses instead of the total accuracy fitness in eq. (1) and also obtained high accuracy. In our experiments, evoDL easily found a suitable DCNN structure since our dataset is not very large and the recognition task is relatively simple.

However, if the structure we obtained is suitable for the proposed sketch recognition task, then it might be generalized for use in another recognition task. To confirm this, we did additional experiments as follows.

The DCNN represented by Fig. 5 is applied to a different sketch recognition task that contains 250 object categories, as presented in [1]. In this dataset, each category contains 80 original sketch images drawn by people. We divide this dataset and use 70 images for training and the remaining 10 images for test. After 2,000 epoch training with 64×64-pixel images, the DCNN shows an accuracy of 0.95 for the training data and 0.59 for the test data, whereas the accuracy for the test data in [1] is 0.56, in spite of using higher-resolution 256×256-pixel images. Therefore, the DCNN obtained using the proposed evoDL shows higher performance even when using a smaller-size images ($256 \times 256 \rightarrow 64 \times 64$). This result confirms the generalization ability of evoDL outcomes.

VI. CONCLUSION

In this study, we proposed a novel method, the evoDL, to obtain DCNN models for human sketch recognition from a GA. Using AlexNet as the basic DCNN model, evoDL was used to tune the hyperparameters of the DCNN. We also demonstrate the generalization ability of the evoDL outcomes by using a different dataset from that initially used for the DCNN evolution. Important future works include:

- Increasing the amount of data in order to observe the search dynamics of ensemble classifiers (ECs) in detail.
- Tuning the real number hyperparameters by using covariance matrix adaptation evolution strategy (CMA-ES)[10].
- Extending the ECs genotype to represent complex network topologies.
- Analyzing the search-space landscape using various fitness functions.

- Introducing the transfer learning concept into evoDL.

ACKNOWLEDGEMENT

This work was partially supported by the JSPS KAKENHI Grant, Grant-in-Aid for Scientific Research(C), 26330282.

REFERENCES

[1] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1106–1114. [Online]. Available: http://books.nips.cc/papers/files/nips25/NIPS2012_0534.pdf

[3] S. Fujino, T. Hasegawa, M. Ueno, N. Mori, and K. Matsumoto, "The convolutional neural network model based on an evolutionary approach for interactive picture book," in *Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings*. Springer, 2017, pp. 103–116.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[5] K. M. P. Bertola, N. Mori, "Sketch recognition for interactive multi-agent system," *Proc. of SCI'14*, no. 334–4, 2014.

[6] Y. Yang and T. M. Hospedales, "Deep neural networks for sketch recognition," *CoRR*, vol. abs/1501.07873, 2015. [Online]. Available: http://arxiv.org/abs/1501.07873

[7] P. Y. Y. Zhining, "The genetic convolutional neural network model based on random sample," pp. 317–326, 2015.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–158, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[9] S. Wang and C. Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, S. Dasgupta and D. Mcallester, Eds., vol. 28, no. 2. JMLR Workshop and Conference Proceedings, May 2013, pp. 118–126. [Online]. Available: http://jmlr.csail.mit.edu/proceedings/papers/v28/wang13a.pdf

[10] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.