

Transfer Learning Layer Selection Using Genetic Algorithm

Satsuki Nagae*, Shin Kawai* and Hajime Nobuhara*

*Department of Intelligent Interaction Technologies,

University of Tsukuba,

Tsukuba-City, Ibaraki, Japan

{nagae, kawai, nobuhara}@cmu.iit.tsukuba.ac.jp

Abstract—The performance of transfer learning in convolutional neural networks depends on the selection of which layers are to be learned again and which are not. Generally, layers selection is performed manually; however, as the number of layers increases, the layers selection process becomes increasingly difficult. Thus, we propose a method to select an effective layers in transfer learning automatically using a genetic algorithm. In the proposed method, a genotype representing which layers' weights are updated or fixed in transfer learning is considered, and we achieve efficient layers selection in the way that a genotype with high validation accuracy is survived during genotype selection. Experiments are performed using the InceptionV3 network that pre-trained ImageNet as source images with transfer learning to CIFAR-100 as target images. Experimental results demonstrate that the test data accuracy in an ensemble of models whose layers are selected by the genetic algorithm is 15% and 12% greater than that of models trained by from-scratch and fine-tuning, respectively. In general transfer learning approach, layers on the output side are selected as adjustable layers; however, it is found that the distribution of the selected layers as an effective adjustable layers obtained by the genetic algorithm extends to the entire network. Transfer learning using a genetic algorithm may successfully capture the characteristics of a convolutional neural network's structure.

Index Terms—Transfer Learning, Genetic Algorithm, Neural Networks, Ensemble Learning

I. INTRODUCTION

Convolutional neural networks (CNNs), which represent a type of machine learning method, can produce high-performance image recognition models; however, CNNs require a large training dataset. In response to this requirement, transfer learning can reduce the number of required learning datasets by reusing pre-trained networks [1]. Reusing a pre-trained network means using the pre-trained layers by the source images as a feature extractor to learn target images. Fine-tuning, where pre-trained weights are used as initial parameters, is a common transfer learning method [2].

The performance of transfer learning depends on which learned layers' weights are fixed or updated. The CNNs extract low-dimensional information, e.g., image colours and edges in the input side layers and high-dimensional information characterising the label in the output side layers [3]. Therefore, a transfer learning method is often applied with fixed input side layers' weights and adjustable output side layers' weights. However, transfer learning that updates the weights of the first convolutional layer and final fully connected layer while fixing

all other layers is the most effective learning technique for Bengali numeral classification (NumtaDB) in the VGG16 [4] architecture pre-trained on the ImageNet [5]. Therefore, it is necessary to update the appropriate layers for each dataset. In recent studies, the number of network layers has increased rapidly, thereby making it increasingly difficult to select layers manually [6]. Thus, we propose a method to automatically select an effective adjustable layers in transfer learning.

PathNet is a layer selection method for transfer learning that uses a genetic algorithm [7]. PathNet uses the genetic algorithm to optimize the transfer learning of modular neural networks by tournament selection. However, PathNet does not support networks with more general structures, and only tournament selection is used to select genotypes. StepwisePathNet is also layer selection method using tournament selection, which extends PathNet to general structure models [8].

In this study, transfer learning using a genetic algorithm in a general structure is performed using several genotype selection methods. Figure 1 shows the position of the proposed method in overall transfer learning. In addition, we demonstrate the advantage of genetic algorithms (providing a variety of models to account for different genotypes) by ensembling the results of multiple models selected by the genetic algorithm. Our proposed method can adapt general structure networks depending on the definition of genotypes.

In our experiments, transfer learning using genetic algorithm

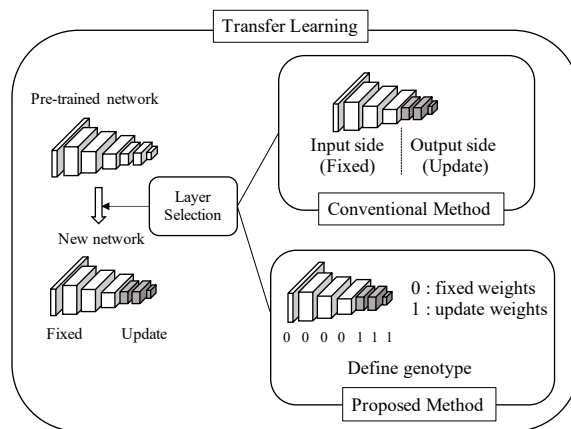


Fig. 1. Comparison of conventional and proposed methods

from InceptionV3 network that pre-trained ImageNet as source images to CIFAR-100 as target images are performed. The results shows test data accuracy in an ensemble of models whose layers are selected by the genetic algorithm is 15% and 12% greater than that of models trained by from-scratch and fine-tuning, respectively. The ensemble high performance may come from the diversity property of genetic algorithm. In addition, the result shows distribution of the selected layers as an effective adjustable layer obtained by the genetic algorithm extends to the entire network. The genetic algorithm seems to capture the structural characteristics of CNNs.

II. PROPOSED METHOD

A. Overview

A genetic algorithm is a solution search optimisation method that imitates the mechanism of biological evolution [9]. In the proposed method, a genotype representing which layers' weights are updated or fixed in transfer learning is considered, and a genotype with high validation accuracy is survived during genotype selection. Figure 2 shows an overview of the proposed method and Algorithm 1 describes how each process works in the proposed method.

B. Initialisation

We defined the genotypes that represent the rules for which layers are fixed weights layers or adjustable layers for the pre-trained model in transfer learning. For each layer of a model performing transfer learning, the genotype is expressed in binary, where 1 and 0 represent adjustable layer and fixed weights layer cases, respectively. For example, the genotype of [1,0,0,1,0] corresponds to a model in which the first and fourth layers are adjustable layers, and the other layers are fixed weights layers. Based on this genotype rule, n genotypes are generated in which one convolutional layer is selected randomly as an adjustable layer. In addition, the fully connected layer of the output layer is always an adjustable layer.

C. Model Training

In the model training stage, models corresponding to each genotype are trained using training data from target images us-

ing transfer learning, and the validation accuracy is determined for all genotypes. Initially, the weights of the convolutional layers are taken as the weights of a pre-trained model, and the weights of the fully connected layer are initialised with a truncated normal and a constant each model training process.

D. Selection

In the selection process, $n/2$ genotypes are selected as parent genotypes for crossover based on the validation accuracy of the model corresponding to each genotype. The selection method is also used when we select parents' genes passed to the next generation. In this study, transfer learning using a genetic algorithm is conducted using four selection methods; i.e., the elite, roulette, tournament, and elite + roulette selection methods.

E. Crossover

Child genotypes are generated from the selected parent genotypes, and a child genotype is inherited as an adjustable layers of the two parent genotypes. Fig. 3 shows an example demonstrating crossover. Child genotype's adjustable layers are the union of two parents genotypes's adjustable layers. Here, set $n/2 + 1$ pairs from $n/2$ parent genotypes to produce $n/2$ child genotypes.

Algorithm 1 Layer selection on transfer learning using genetic algorithm

- 1: **Generate** n initial genotypes
 - 2: **while** Generation < Final Generation **do**
 - 3: **Train** n models corresponding to n genotypes
 - 4: **Select** $n/2$ genotypes based on fitness
 - 5: **Crossover** $n/2$ child genotypes
 - 6: **Select** $n/4$ parent genotypes to next generation
 - 7: **Generate** $n/4$ alien genotypes to next generation
 - 8: Generation \leftarrow Generation+1
 - 9: **end while**
 - 10: **Ensemble** final generation models
-

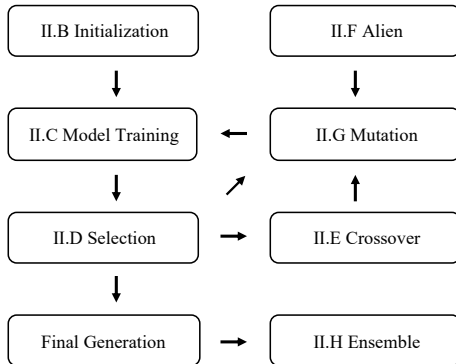


Fig. 2. Overview of proposed method

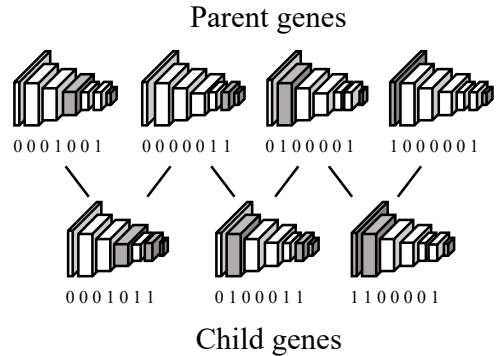


Fig. 3. Crossover process

F. Alien and Inherit Parent Genes

After the crossover process, $n/4$ genotypes are newly initialised. We define these genotypes as alien species to avoid a problem with a genetic algorithm, i.e., the results are highly dependent on the initial parameter values. As a result, particular genotype information is spread in population and genotype diversity is reduced. Here, $n/4$ genotypes are selected from the parent genotypes used for crossover by the selection method used in section II-D selection process and passed to the next generation

G. Mutation

The total number of genotypes is maintained as n in all generations, with the next generation having $n/2$ child genotypes obtained via a crossover, $n/4$ newly initialised alien species genotypes, and $n/4$ inherited parent genotypes. For all genotypes, there is a $p\%$ chance that a mutation operation reverse the array of genotype.

H. Ensemble

Among the genotypes of the final generation, the top m genotypes with the highest accuracy in the corresponding models are trained using target images for five epochs, and the results are ensembled. In the ensemble method, prediction probabilities are averaged for each label of each model, and the label with the highest average prediction probability is set as the prediction label.

III. SELECTION METHODS

A genetic algorithm can control the diversity of genotypes using different genotype selection methods for the proposed method during the crossover. Here, the selection is performed based on the fitness of each genotype, and in the proposed method, the validation accuracy of the model corresponding to the genotype is used. In this study, we compared four selection methods, i.e., the elite, roulette, tournament, and elite + roulette selection methods.

A. Elite Selection

The elite selection method selects a predetermined number of genotypes with the highest fitness. Elite selection ensures that maximum fitness is not reduced between generations. However, the disadvantage of elite selection is that elite genotypes spread too widely in the population, resulting in a loss of genotype diversity.

B. Roulette Selection

The roulette selection method performs random selections by considering the population as roulette. Here, the fitness of each genotype is proportional to the area of the roulette wheel, and the higher the fitness, the higher the probability of being selected. Assuming that the fitness of each genotype is $f(i)$ and the size of the population is N , the probability of each genotype p_i being selected is expressed by

$$p_i = \frac{f(i)}{\sum_{n=1}^N f(n)}. \quad (1)$$

Note that low fitness genotypes are also likely to be selected; thus, only certain genotypes can be prevented from spreading in the population. In addition, high fitness genotype genotypes may not be selected, and the maximum fitness of the population may not increase.

C. Tournament Selection

In tournament selection, random genotypes of a predetermined tournament size are selected, and the genotypes with the highest fitness are selected. Here, we set the tournament size to 2 in our experiments. As with roulette selection, the selection of tournament size may cause less fitness genotypes to be selected.

D. Elite + Roulette Selection

Elite selection is used to select a parent genotype for the crossover, and roulette selection is used to select the parent genotype to be inherited by the next generation. This method is expected to ensure that more appropriate genotypes are passed to the next generation and prevent the same genotype from spreading too much within the population.

IV. EXPERIMENTS

Experiments were performed to compare and evaluate using the InceptionV3 network [10] that pre-trained ImageNet as source images with transfer learning to CIFAR-100 as target images using a genetic algorithm with the four selection methods.

A. Model Architecture

InceptionV3 is an upgraded version of GoogLeNet, which won the ImageNet Large Scale Visual Recognition Challenge in 2014 (ILSVRC2014). InceptionV3 is also a popular model for transfer learning and is implemented in many deep learning libraries. This network comprises 94 convolutional layers and one fully connected layer. In our experiment, InceptionV3 was pre-trained on ImageNet. This massive, general object-recognition dataset contains 1,000 classes and over one million images and is used in the ILSVRCs.

B. Dataset and Augmentation

The CIFAR-100 dataset [11] is used as target images with transfer learning. CIFAR-100 is a 100-class object recognition dataset with 500 train and 100 test images for each class. Here, 500 train data images for each class are randomly divided into 450 train data images and 50 validation data images. The CIFAR-100 image size is 32×32 pixels; thus, we resize the images to 224×224 pixels on ImageNet using the bilinear method. Note that the following data augmentations are applied in all cases:

- Random rotation in $[-15, 15]$ deg
- Width and height shift in $[-10, 10]$ %
- Horizontal flipping

C. Evaluations

Five experiments were performed for each of the four selection methods. The number of initialised genotypes is $n=20$ (up to the fifth generation), and each genotype has a $p=1\%$ mutation probability. Each of the top-five genotypes of the last generation was trained for five epochs, and the results are ensembled. For comparison, we perform the experiments using two conventional methods, i.e., from-scratch and fine-tuning. From-scratch means that all weights in InceptionV3 were initialised randomly with no transfer learning, and fine-tuning is a transfer learning method that uses pre-trained weights as initial parameters. All algorithms were optimised by Adam with the Keras default parameters and executed on a Geforce GTX1080Ti graphics card with a batch size of 16.

V. RESULTS AND DISCUSSION

A. Comparison with Selection Type and Conventional Method

Table I shows the ensembled results of the final five models for each of the four selection methods, and as well as the results obtained by the conventional method. As can be seen, from-scratch and fine-tuning show 98% train accuracy, and 68% and 71% test accuracy, respectively, so conventional methods cause too much adaptation to training images. By comparing the average of the ensemble results of the five models obtained by the genetic algorithm, it can be seen that the elite selection method shows the highest accuracy for both

the train and test data. In contrast, the roulette selection shows the lowest accuracy. The elite selection method can effectively pass the genes of the model adapted to the validation data to the next generation, and roulette selection may pass genes with low fitness values to the next generation.

Figure 4 shows a boxplot of the test accuracy for each learning method listed in Table I. As can be seen, the transfer learning results at the layer selected by the genetic algorithm exceed those of both the from-scratch and fine-tuning methods. In addition, the standard deviations of the from-scratch and fine-tuning results are 2.68 and 2.04, respectively, and the maximum standard deviation of the genetic algorithm's results is 0.68. Therefore, it can be considered that transfer learning by the genetic algorithm is more stable because, by using validation accuracy as fitness, transfer learning can be performed by selecting a layer that is not overly adaptive to the training data. In addition, the genotypes model can capture the generalisation characteristics of images.

B. Fitness Transition

Figure 5 shows the evolution of the maximum fitness in the population for each selection method. Excluding the roulette selection method, it can be seen that the maximum fitness increased with each generation, which confirms that superior genotypes remained. Elite selection steadily increased the maximum fitness. In contrast, elite + roulette selection

TABLE I
RESULTS OF MODEL ACCURACY

Experiment number	train						test					
	Elite	Roulette	Tournament	Elite + Roulette	From-scratch	Fine-tuning	Elite	Roulette	Tournament	Elite + Roulette	From-scratch	Fine-tuning
1	91.4	90.0	90.9	95.8	97.1	98.0	82.9	82.0	82.8	84.2	63.9	73.4
2	95.7	92.3	93.6	92.6	98.2	97.5	84.3	82.9	83.7	83.1	70.2	71.0
3	95.6	91.3	93.0	93.8	98.1	98.0	84.1	82.8	83.6	83.8	69.7	73.1
4	92.6	91.4	94.4	94.8	97.7	97.5	83.3	82.5	84.0	84.0	67.6	68.5
5	93.1	89.6	92.3	90.3	97.9	97.6	83.8	81.7	83.6	82.6	70.1	72.7
Ave.	93.7	90.9	92.9	93.5	97.8	97.7	83.7	82.4	83.5	83.5	68.3	71.7
Std.	1.91	1.09	1.32	2.13	0.44	0.26	0.57	0.51	0.45	0.68	2.68	2.04

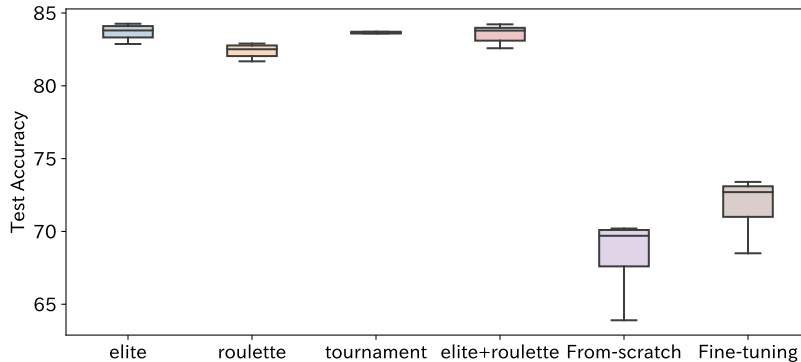


Fig. 4. Boxplot of all method test accuracy

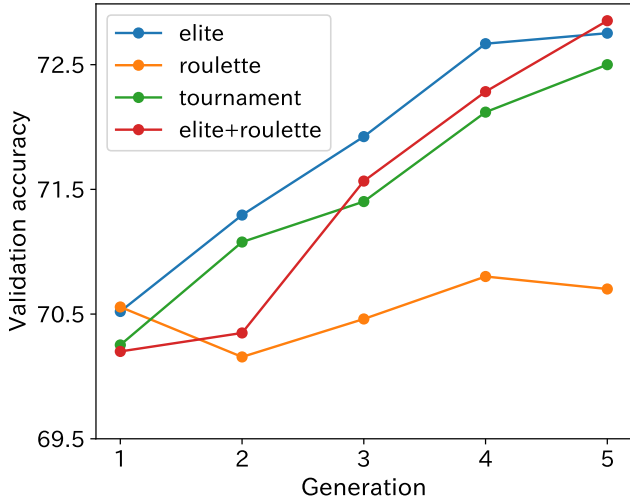


Fig. 5. Maximum fitness in population

is less stable than elite selection; however, the maximum fitness exceeds the elite genotype in the final generation. The maximum fitness of the roulette selection method decreases because genotypes with low fitness have a probability of being selected in the roulette selection method; thus, superior genotypes cannot survive.

C. Model Diversity

Figure 6 shows the evolution of the number of layers selected as adjustable in the genotypes for each generation. As shown, in the first generation, 20 genotypes are initialised and the fully connected layer of each genotype is always selected

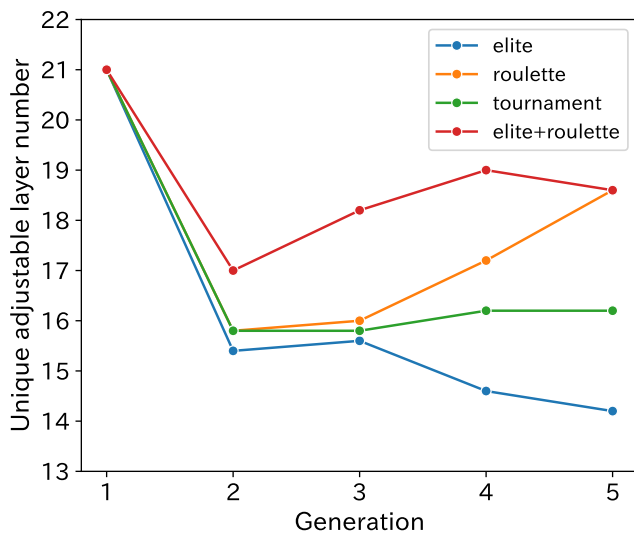


Fig. 6. Number of layers being selected as adjustable

as adjustable. Thus, the total number of adjustable layers is 21 for all selection methods. As can be seen in Fig. 6, the number of adjustable layers decreases with each generation with the elite selection method because the elite genotype information spreads throughout the population, thereby leaving genotypes with higher fitness. In contrast, the roulette selection and elite + roulette selection methods do not reduce the number of adjustable layers significantly over generations. The roulette selection method appears to prevent elite genotype information from spreading too much throughout the population. Figure 7 shows the evolution of the cumulative sum of the unique genotypes. The elite selection method appears to be associated with a lower number of genotypes than other selection methods because the superior genotype information spreads too widely in the population, which results in duplication of the child genotypes obtained by crossover. Thus, as shown in Fig. 6 and Fig. 7, the elite selection can pass superior genotype information to the next generation, whereas the diversity of genotypes is inferior to other selection methods. In addition, the elite + roulette selection method resolves the diversity problem in the elite selection method by passing the various genotypes of the roulette selection to the next generation.

D. Layer Distribution Selected by Genetic Algorithm

Figure 8 shows the distribution of the number of times each layer is selected as adjustable in the genotype with the highest fitness in the final generation of 20 genetic algorithm experiments. Here, the number of InceptionV3 convolutional layers is 94, and each layer number corresponds to each convolutional layer. Note that layers with a number close to 1 are on the input layer side, and layers with a number close to 94 are on the output layer side. It can be seen in Fig. 8 that the distribution of the number of times layers are selected as the

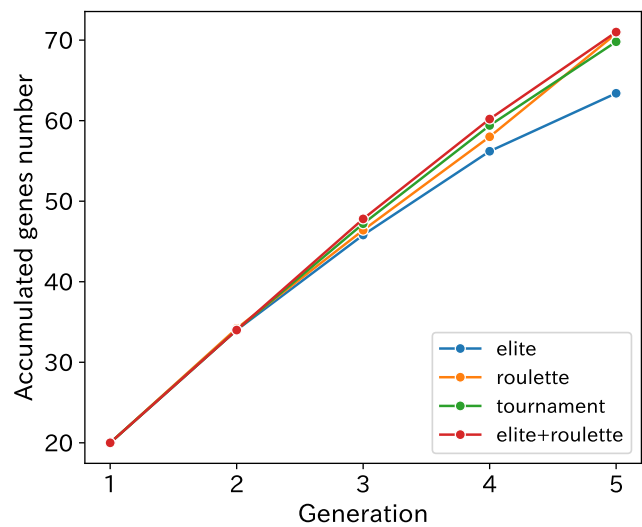


Fig. 7. Cumulative sum of the unique genotypes that appear in experiments

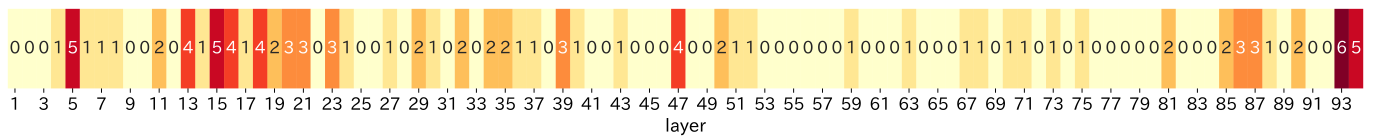


Fig. 8. Number of times that each layer of best gene in final generation was selected as an adjustable layer among 20 transfer learning based on genetic algorithm

adjustable is not uniform and extends to the entire network. This indicates that layers to be adjustable layers for improving the performance of transfer learning are not limited to the output layer side. The result contrasts the general intuition that layers on the output side should be adjustable because the CNNs capture macroscopic features on the input side and microscopic features on the output side.

VI. CONCLUSION

In this paper, we have proposed a method to automatically select layers whose weights should be updated in the transfer learning of CNNs using a genetic algorithm. Transfer learning experiments are performed using the InceptionV3 network that pre-trained ImageNet as source images to CIFAR-100 as target images using a genetic algorithm. The learning performance is evaluated by comparing transfer learning via the genetic algorithm using four selection methods and conventional learning methods (from-scratch and fine-tuning). The transfer learning results obtained by the genetic algorithm demonstrate that test accuracy is 10 – 15% greater than that of the conventional method for each selection method, and the genotype selection by the elite selection method yields the best result. We also compare the diversity of genotypes generated by the genetic algorithm with several selection methods. We find that the elite + roulette selection method passes superior genotypes to the next generation and maintains genotype diversity. In addition, the layers selected as adjustable by the genetic algorithm that contribute to transfer learning performance is distributed nonuniformly on both the input and output sides. This may be because layer selection by the genetic algorithm successfully captures the characteristics of CNNs that cannot be captured by manual layer selection.

In this study, we performed transfer learning on the CIFAR-100; however, in the future, experiments with other types of datasets are required to demonstrate the effectiveness of the genetic algorithm for transfer learning. In addition, in our experiment, we restrict the number of generations to five because of the computation cost. So it is also necessary to improve proposed methods to be computationally efficient.

REFERENCES

- [1] Pan. S. J., Yang. Q, "A survey on transfer learning," *IEEE Transactions on knowledge data engineering*, 2010, 22, pp. 1345–1359.
- [2] Sawada. Y, Kozuka. K, "Whole layers transfer learning of deep neural networks for a small scale dataset," *Int. J. Mach. Learn. Comput*, 2016, 6, pp. 27–31.
- [3] Zeiler. M. D, Fergus. R, "Visualizing and understanding convolutional networks," *In European conference on computer vision*, 2014, pp. 818–833.
- [4] Simonyan. K, Zisserman. A, "Very deep convolutional networks for large-scale image recognition," *CoPR*, 2014, abs/1409.1556.
- [5] Zunair. H, Mohammed. N, Momen. S, "Unconventional wisdom: a new transfer learning approach applied to bengali numeral classification," *In International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1–6.
- [6] Ghafoorian. M, Mehrtash. A, Kapur. T, Karssemeijer. N, Marchiori. E, Pesteie. M, Guttman. C, Leeuw. F, Tempany. C, Ginneken. B, Fedorov. A, Abolmaesumi. P, Platel. B, WellsIII. W, "Transfer learning for domain adaptation in mri: application in brain lesion segmentation," *In International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017, pp. 516–524.
- [7] Fernando. C, Banarse. D, Blundell. C, Zwols. Y, Ha. D, Rusu. A, Pritzel. A, Wierstra. D, "Pathnet: evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.
- [8] Imai. S, Kawai. S, Nobuhara.H, "Stepwise Pathnet: a layer-by-layer knowledge-selection-based transfer learning algorithm," *Nature Scientific Report* (to appear).
- [9] Harvey. I, "The microbial genetic algorithm," *In European Conference on Artificial Life*, 2019, pp. 126–133.
- [10] Szegedy. C, Vanhoucke. V, Ioffe. S, Shlens. J, Wojna. Z, "Rethinking the inception architecture for computer vision," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2818–2826.
- [11] Krizhevsky. A, Hinton. G, "Learning multiple layers of features from tiny images," *Technical Report, Citeseer* 2009.