Data Augmentation using Evolutionary Image Processing

Kosaku Fujita
Yokohama National University
Yokohama, Japan
fujita-kosaku-df@ynu.jp

Masayuki Kobayashi
Yokohama National University
Yokohama, Japan
kobayashi-masayuki-xc@ynu.jp

Tomoharu Nagao
Yokohama National University
Yokohama, Japan
nagao@ynu.ac.jp

***Abstract*—In the machine learning community, data augmentation techniques have been widely used to make deep neural networks invariant to object transition. However, less attention has been paid to data augmentation in traditional classification methods. In this paper, we take a closer look at traditional classification methods and introduce a new data augmentation technique based on the concept of image transformation. Starting with a few existing examples, we add noise and generate new data points to reduce sparseness in a given feature space. Then, we generate images corresponding to the new data points, although this is usually an ill-posed problem. Herein, the novelty is in constructing an image transformation tree and generating new data from a small number of instances. This allows us to reduce sparseness in the feature space and build more robust classifiers. We evaluate our method on the Caltech-101 dataset to verify its potential. In the context of the situation where the amount of training data is limited, we demonstrate that the support vector machine-based classifiers trained with an augmented dataset using our method outperform classifiers trained with the original dataset in most cases.**

***Index Terms*—machine learning, data augmentation, evolutionary computing, evolutionary image processing**

I. Introduction

To train classifiers for image recognition, we require a lot of data to maximize the generalization capability of classifiers. For instance, recent deep neural network models have millions of parameters, and they require a large amount of data to achieve good performance. Not having enough training data will result overfitting and poor generalization performance. Therefore, in the deep learning community, many techniques, including data augmentation, have been widely used to address this problem.

In traditional classification methods, less attention has been paid to data augmentation. In traditional classification methods, all data are mapped from data space and represented as points in a feature space. Therefore, in order to improve the classification performance, we need to reduce sparseness in the feature space. The simplest way to do this is to apply some transformations and generate new examples in the feature space. Thus, we can manipulate the feature space and easily augment the datasets. However, this is usually an ill-posed problem and it is difficult to see the image corresponding to a given data point in the feature space. This can be a barrier to assigning correct labels to the generated points. Here, we present a new data augmentation technique for generating images that reduce sparseness in a given feature space.

We start with a few existing examples, add noise to the original data points, and generate new data points in the feature space. Once a new data point has been generated, we employ ACTIT (Automatic construction of tree-structural image transformations) [1], [2], to generate images corresponding to each data point. The performance of each individual (i.e., *fitness*) is measured by the closeness of the generated data point and the target one. To determine whether the generated images are valid for augmented data or not, we evaluate them on the basis of several perspectives.

We evaluate our method on the Caltech-101 dataset to verify its potential. In the context of the situation where the amount of training data is limited, we demonstrate that the support vector machine (SVM) [3] classifiers trained with an augmented dataset using our method outperform classifiers trained using the original dataset. The innovations and contributions of this paper are summarized as follows:

- We propose constructing an image transformation tree and generating new data from a small number of instances.
- We focus on traditional classification methods, and employ an evolutionary computing approach to augment the dataset efficiently. Previously, data augmentation techniques for maximizing the generalization capability of deep neural networks have been proposed.
- We apply our data augmentation technique to two-class classification, achieving superior performance over the baseline in most cases, and demonstrate the potential of our method.

The rest of this paper is organized as follows. Section II provides a review of the related work. In Section III, we describe our data augmentation method. We apply our method and show the advantage over the baseline in Section IV. Finally, we discuss the results and an outlook for future work in Section V.

978-1-5386-6602-9/18/$31.00 ©2018 IEEE

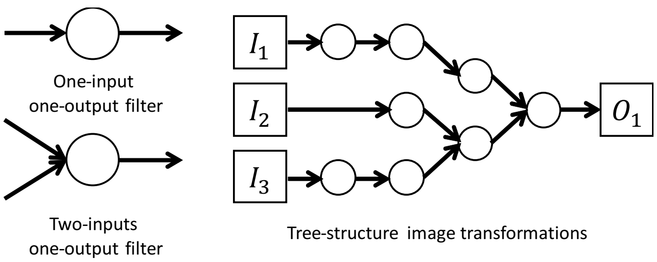


Fig. 1: An example of filters and phenotype used in ACTIT.

II. Related Work

A. Data Augmentation

In a deep learning community, data augmentation has been widely used to maximize the generalization capability of deep neural networks. Traditional data augmentation consists of a series of affine transformations, such as random cropping, image mirroring and color shifting.

Although these data augmentation techniques have been designed manually, recent studies tend to rely on the deep neural network models to generate new training samples automatically [4]–[6]. Generative adversarial networks [7] have also been used to generate the augmented data directly [8]–[11]. Another interesting approach using reinforcement learning was recently attempted; this demonstrated outstanding performance [12].

Although these are very powerful data augmentation techniques, they typically fail to learn expressive models from a small number of instances. Thus, it is difficult to apply them to traditional classifiers (e.g., support vector machine).

B. The ACTIT Method

ACTIT (Automatic construction of tree-structural image transformations) [1], [2] is a [13], [14] GP-based method that aims at automatically constructing image transformation, which is built upon one-input one-output and two-inputs one-output filters. Every tree node has an image processing filter, and every leaf node has an input image. This tree structure is called *phenotype* and is encoded by *genotype* [15]. For given input and target images, the phenotype is modified by applying a number of genetic operators (i.e., crossover and mutation) to obtain better image-to-image transformations. An example of phenotype is given in Fig. 1. We have extended ACTIT and applied this method to several image processing tasks, such as medical image processing [16], [17] and style transfer [18], and demonstrated its efficiency.

Our approach was inspired by the recent success of ACTIT in a variety of image processing tasks, and we exploited the potential of this method for data augmentation.

III. Proposed Method

Fig. 2 illustrates an overview of our method. In our method, we consider that each example is projected into a feature space. These examples in the feature space are referred to as *data point*s. In order to apply data augmentation, we must

Algorithm 1 Proposed data augmentation method. In this paper, we use $T = 1$, which is the least expensive option.

Input: D: Dataset, N: Number of samples to generate from each existing sample, $\sigma_{\max}$: Maximum distance.

for $i = 1$ to T **do**
  Chose n samples, $x_1, \ldots, x_n$ from dataset D.
  for $j = 1$ to n **do**
    repeat
      $\sigma \leftarrow Uniform(0, \sigma_{\max})$
      Apply a transformation to the data point $\boldsymbol{f}_{x_j}$ and generate a new data point $\boldsymbol{f}'_{x_j}$ which is σ distant from $\boldsymbol{f}_{x_j}$:
      $$\boldsymbol{f}'_{x_j} = \boldsymbol{f}_{x_j} + \alpha \boldsymbol{X}$$
      Employ ACTIT to generate image x'_j corresponding to the data point $\boldsymbol{f}'_{x_j}$:
      $$fitness = 1.0 - \frac{1}{n} \|\boldsymbol{f}'_{x_j} - \boldsymbol{f}_{x'_j}\|_2$$
      Evaluate generated image x'_j on the basis of two perspectives:
      - Appropriateness
      - Naturalness
    until a valid data point has been found.
    $N-1$ new data points are randomly generated within the distance σ from $\boldsymbol{f}_{x_j}$, and added to the dataset D
  end for
end for

interpolate among the data points and reduce sparseness in the feature space. To do this, we first choose n samples, $x_1, \ldots x_n$ from the original data points. For each data point, we then apply a transformation to generate a new data point that is σ distant from the original point, where σ is sampled from $Uniform(0, \sigma_{\max})$.

There are many transformations to be considered; however, in our method, we simply add noise to the original data points. Formally, let x represent some image in the dataset, and let $\boldsymbol{f}_x$ be the data point of an image x in the feature space. Given the data point $\boldsymbol{f}_x$, a new data point $\boldsymbol{f}'_x$ can be generated:

$$\boldsymbol{f}'_x = \boldsymbol{f}_x + \alpha \boldsymbol{X} \tag{1}$$

where α is a scale parameter and $\boldsymbol{X}$ is a random vector drawn from a uniform distribution.

The new data point, called a *target data point*, is first obtained and the images corresponding to this point generated, although this is usually an ill-posed problem. In order to address and overcome the ill-posed nature of this problem, we introduce an image transformation technique and perform the transformation on the input image to manipulate the feature space. In our method, we employ ACTIT to apply some transformation to the original images. During an evolution process, each individual is evaluated by the closeness of the generated data point and the target data point. We call this measurement *fitness*. For the fitness function, we use the

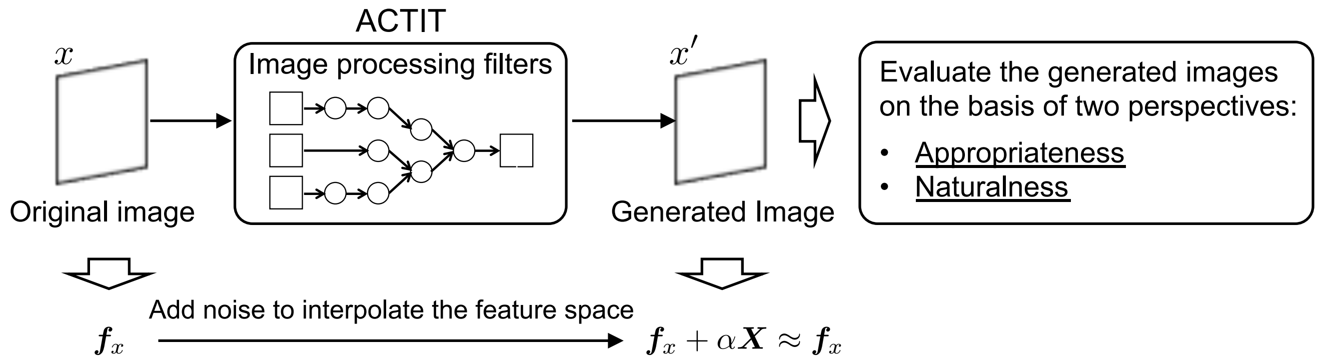


Fig. 2: An overview of our method.

Euclidean distance between the target and the generated data points in the feature space:

$$fitness = 1.0 - \frac{1}{n}\|\boldsymbol{f}'_{x_j} - \boldsymbol{f}_{x'}\|_2 \tag{2}$$

where we denote the j-th distant target data point as $\boldsymbol{f}'_{x_j}$, the generated image as x', the data point of image x' as $\boldsymbol{f}_{x'}$, and n is the number of dimensions. The range of this fitness value is $[0.0, 1.0]$, and a higher value indicates that the generated data point is closer is to the target data point.

This enables us to obtain the images corresponding to the target data point. However, there is a possibility that the generated examples are not appropriate for augmented data, or they are unrecognizable. To determine whether the generated examples are valid for augmented data or not, we designed the following task: Given the most distant generated data point from the original data point, we evaluate the generated image and identify the class it belongs to. The generated image is evaluated on the basis of two perspectives: appropriateness and naturalness. We define the appropriateness as the closeness of the generated images to any of the known classes, and the naturalness as how natural the generated images are for augmented data (i.e., less noisy).

We do this iteratively by determining valid generated data points. Once a valid data point has been found, all data points within its distance are treated as augmented data. Therefore, $N-1$ new data points are randomly generated within the distance σ from the original data point, and added to the dataset. This comes from the assumption that it is much easier to obtain a given image corresponding to the near target data points.

The procedure of our data augmentation is listed in Algorithm 1.

IV. Experiment and Results

A. Dataset

We evaluated our method on image classification task using the Caltech-101[1] [20] dataset. Although Caltech-101 is a large dataset, which 101 classes, for simplicity we selected several classes and evaluated our method on a two-class classification problem. Our experiment is designed to address the following classification problems: flamingo vs. leopard, lotus vs. sunflower, airplane vs. ferry, and crab vs. lobster. We choose these classes because the former two problems can be easily classified (i.e., as either red/yellow) and the latter two cannot. Example images used in our experiment are shown in Fig. 3.

It must be noted that we consider the situation where the amount of training data is limited. Therefore, in our experiment, 40 images are sampled from each class. Then, five images are used for training and the remaining 35 images are used for testing. In this paper, each image is resized from its original size to 128×128 pixels by linear interpolation.

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

TABLE I: 58 Low-level features used in our method.

Feature	Description
Colors	RGB, L*a*b*
Statistical Values	Mean, Maximum, Minimum, Standard deviation, First quartile, Median, Third quartile, Mode
ULBP [19]	$P = 10, R = 2$

TABLE II: Image processing filters used in our method.

Filter name	Description
One-input one-output filters	19 filters (Mean, Maximum, etc.)
Two-input one-output filters	9 filters (Logical sum, Average, etc.)

B. Experimental Settings

We extracted 58 low-level features, including 48 simple statistical values from the input image and the uniform local binary pattern (ULBP) [19] (see TABLE I). For each example in the dataset, we generated $N = 1, 3, 5$ examples from each original data point, and investigated the effect of the numbers of augmented samples. ACTIT constructs image transformations using the predefined image processing filters shown in TABLE I, each of which transforms one or two 2-D images to a single 2-D image of the same size. The parameter settings for ACTIT are shown in TABLE III.

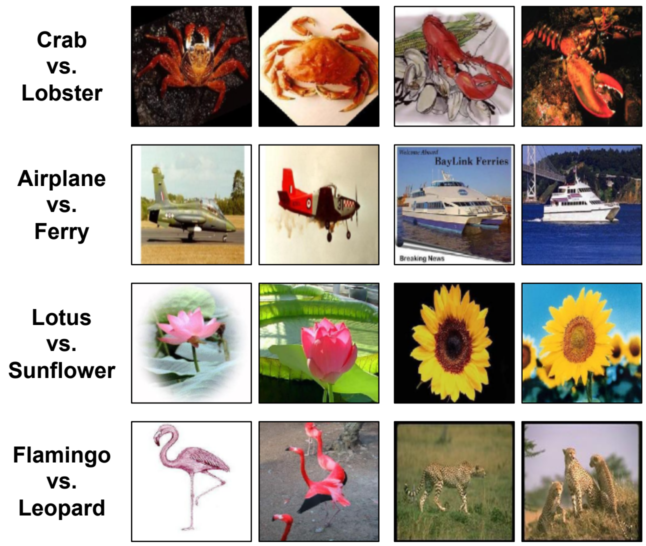


Fig. 3: Examples images used in our experiment.

TABLE III: Parameter settings for ACTIT.

Parameter	Value
Generation alternation model	MGG [21]
Number of generations	$1,000$
Population size	10
Children size	5
Mutation Rate	0.9
Tournament size	2
Minimum number of filters	1
Maximum number of filters	20

TABLE IV: Classification performance compared with baseline.

(a) Flamingo vs. Leopard

Model	Accuracy (%)
SVM (Baseline)	91.4
SVM + data augmentation ($N = 1$)	90.0
SVM + data augmentation ($N = 3$)	90.0
SVM + data augmentation ($N = 5$)	**92.9**

(b) Lotus vs. Sunflower

Model	Accuracy (%)
SVM (Baseline)	**61.4**
SVM + data augmentation ($N = 1$)	57.1
SVM + data augmentation ($N = 3$)	57.1
SVM + data augmentation ($N = 5$)	55.7

(c) Airplane vs. Ferry

Model	Accuracy (%)
SVM (Baseline)	**90.0**
SVM + data augmentation ($N = 1$)	85.7
SVM + data augmentation ($N = 3$)	88.6
SVM + data augmentation ($N = 5$)	88.6

(d) Crab vs. Lobster

Model	Accuracy (%)
SVM (Baseline)	57.1
SVM + data augmentation ($N = 1$)	**62.9**
SVM + data augmentation ($N = 3$)	60.0
SVM + data augmentation ($N = 5$)	60.0

We use a SVM [3] with radial basis function (RBF) kernel as the classifier. Due to the lack of training data, we do not extensively tune the hyper-parameters in SVM classification but simply use the RBF kernel with fixed cost and gamma values. The cost parameter c is set to 10 and the gamma parameter γ is set to 1.0, respectively. For the SVM solver, we use the implementation of LIBSVM [22]. The maximum distance $\sigma_{\max}$ is set to 0.2 in our experiment.

C. Results

1) Comparison with Baseline: The comparison of the classification performance is summarized in TABLE IV. As can be seen from this table, the SVM classifiers trained with an augmented dataset using our method outperform classifiers trained with the original dataset except two cases (Lotus vs. Sunflower and Airplane vs. Ferry). Especially, our model outperformed by 5.7% in the dog vs. cat classification, while it underperformed by 4.7% in the lotus vs. sunflower classification.

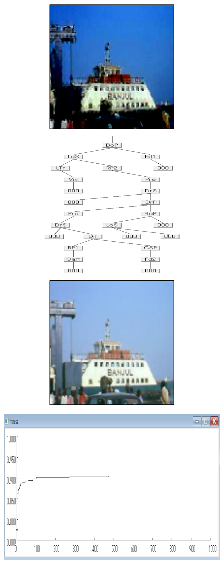
In our experiment, we can see no effect of the number of samples N. However, it may highly depend on the dataset and the maximum distance $\sigma_{\max}$. Therefore, we must consider the parameters when the number of samples N is determined.

Notice that we did not extensively tune the hyper-parameters in the SVM classification and simply used the RBF kernel with fixed cost and gamma values. This may suggest that using more augmented data with tuned SVM would achieve better performance.

2) Analysis of Obtained Images: Fig. 4 shows the generated images by our method, along with their image processing filters, original images and fitness progresses. We observe that ACTIT applied mostly color-based transformations. For instance, the "sky" region in Fig. 4a has become darker and the background in Fig. 4b has become green. This is likely to be because we use color-based features for the most part (i.e., 48 simple statistical values from input image) for classification in our experiment. On the other hand, we can also see that some blurred images are generated, which are not appropriate for data augmentation; see Fig. 4c. Furthermore, the generated images like Fig. 4d and Fig. 4e are also interpreted meaningfully in the sense of interpolation of a feature space, although only small changes can be seen in the input images.

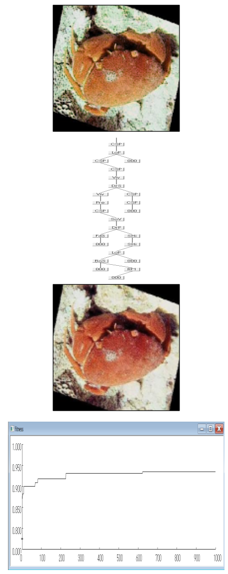
V. Conclusion

In this paper, we proposed a new data augmentation method, with a focus on the traditional image classifiers. For a given feature space, our method generates new data points and their corresponding images. This method enables users to reduce sparseness in the feature space and build more robust



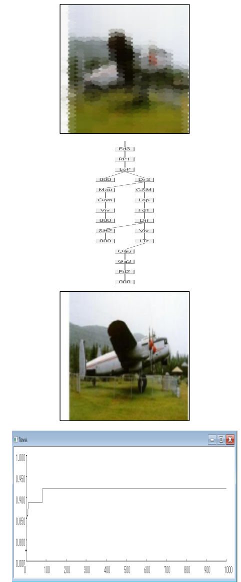
$\mathit{fitness} = 0.904$

(a) Ferry.



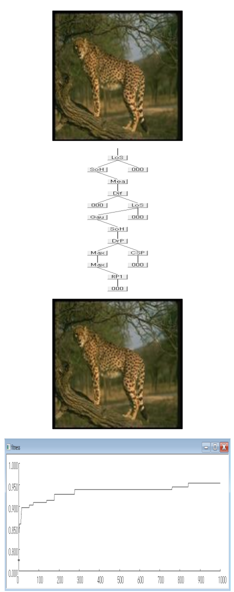
$\mathit{fitness} = 0.934$

(b) Crab.



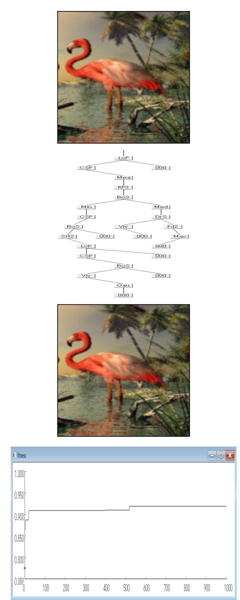
$\mathit{fitness} = 0.921$

(c) Airplane (blurry, unrecognizable and invalid for data augmentation).



$\mathit{fitness} = 0.954$

(d) Leopard.



$\mathit{fitness} = 0.918$

(e) Flamingo.

Fig. 4: Examples of images generated using our method along with their image processing filters, original images, and fitness progresses. Examples of (a), (b) good quality images, (c) bad quality image (blurry, unrecognizable and invalid for data augmentation), and (d), (e) good quality images; however, only small changes can be seen.

classifiers. We evaluated our method for image classification using the Caltech-101 dataset and showed that our method yields better performance in most cases.

For data augmentation, we simply added random noise to some original data points. However, there are a number of ways to generate new data points in a given feature space. In the future, we would like to consider the sparseness, and determine where to add new data points in a feature space. Thus, our method can be made more flexible and suitable for data augmentation. Furthermore, to make our method more effective in improving the performance of image classifiers, we could use two or more images and merge them to generate augmented data. Another direction of future research is to apply our method to practical image datasets, thereby extending the capabilities of our method.

REFERENCES

[1] S. Aoki and T. Nagao, "Actit: Automatic construction of tree-structural image transformation," pp. 888–894, 1999.

[2] ——, "Automatic construction of tree-structural image transformations using genetic programming," in *Proceedings of the 10th International Conference on Image Analysis and Processing (ICIP '99)*, Venice, Italy, 1999, pp. 136–141.

[3] V. N. Vapnik, *The nature of statistical learning theory*. Springer, 1995.

[4] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, "A bayesian data augmentation approach for learning deep models," in *Proceedings of Advances in Neural Information Processing Systems (NIPS' 17)*, Long Beach, USA, 2017, pp. 2794–2803.

[5] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation - Learning an optimal data augmentation strategy," pp. 5858–5869, 2017.

[6] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," 2017.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Advances in Neural Information Processing Systems (NIPS' 14)*, Montreal, Canada, 2014, pp. 2672–2680.

[8] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[9] S. Mun, S. Park, D. K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using svm hyperplane," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE'17)*, Munich, Germany, 2017.

[10] X. Zhu, Y. Liu, Z. Qin, and J. Li, "Data augmentation in emotion classification using generative adversarial networks," *arXiv preprint arXiv:1711.00648*, 2017.

[11] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[12] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," 2018.

[13] J. R. Koza, "Genetic programming: On the programming of computers by means of natural selection," *MIT Press*, 1992.

[14] ——, "Genetic programming ii: Automatic discovery of reusable programs," *MIT Press*, 1994.

[15] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. SpringerVerlag, 2003.

[16] Y. Nakano and T. Nagao, "3d medical image processing using 3d-actit; Automatic construction of tree-structural image transformation," in *Proceedings of the International Workshop on Advanced Image Technology (IWAIT '04)*, Singapore, Singapore, 2004, pp. 529–533.

[17] ——, "Automatic extraction of internal organs region from 3d pet image data using 3d-actit," in *Proceedings of the International Workshop on Advanced Image Technology (IWAIT '06)*, Okinawa, Japan, 2006.

[18] M. Saito and T. Nagao, "Evolutionary image processing application to painting," pp. 26–35, 2015 (in Japanese).

[19] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[20] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, Washington, DC, USA, 2004.

[21] H. Sato, I. Ono, and S. Kobayashi, "A new generation alternation model of genetic algorithms and its assessment," *Journal of Japanese Society for Artificial Intelligence*, vol. 12, no. 5, pp. 734–744, 1997 (in Japanese).

[22] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.