Automatic Construction of Image Transformation Algorithms Using Feature Based Genetic Image Network

Yuta Nakano, Shinichi Shirakawa, *Member, IEEE*, Noriko Yata, and Tomoharu Nagao, *Member, IEEE*

***Abstract*— Image processing and recognition technologies are becoming increasingly important. Automatic construction methods for image transformation algorithms proposed to date approximate adequate image transformation from original images to their target images using a combination of several known image processing filters by evolutionary computation techniques. In this paper, we introduce the adaptive image processing filters that process according to the features of an input image. The processing of the adaptive filters is decided based on the local features of an input image. We implement them to feed-forward genetic image network (FFGIN) that is one of the automatic construction methods for image transformations. Then we apply our method to the problems of segmentation of organs and tissues in medical images. Experimental results show that our method constructs the effective segmentation algorithms that extract multiple regions respectively.**

I. Introduction

Image processing and recognition technologies are becoming increasingly important, e.g., medical, industrial, and security images. Numerous image processing algorithms have been proposed and shown their effectiveness in various fields. It is, however, difficult to design effective image processing algorithms for given target problems because each image processing algorithm considerably depends on the target problems. Therefore, automatic construction methods for image processing algorithms are effective to given target problems are required.

Evolutionary computation (EC) techniques have been applied to automatic construction of image processing algorithms. Genetic programming (GP) [1], [2] is often applied to image classification tasks [3], [4], [5]. GP is also used to develop useful texture-feature extraction algorithms [6], [7]. In these studies, the evolved features are either at par or outperform human-designed features. Naturally, EC techniques are used for medical image processing fields [8], [9]. The Automatic Construction of Tree structural Image Transformation (ACTIT) system [10], [11], an automatic construction method for image transformation, has been proposed previously. ACTIT approximates adequate image transformation from original images to their target images by a combination of several known image processing filters. ACTIT constructs tree structured image processing filters using GP. It has constructed various types of image transformation algorithms automatically. The genetic image network (GIN) is a recent automatic construction method for image transformation [12], [13]. The representation of GIN is a network structure.

Yuta Nakano, Shinichi Shirakawa, Noriko Yata, and Tomoharu Nagao are with the Graduate School of Environment and Information Sciences, Yokohama National University, 79-7, Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa 240-8501, Japan (phone: +81 45 3394136; email: nakanock@gmail.com, shinichi.shirakawa@gmail.com, yata@nlab.sogo1.ynu.ac.jp, nagao@ynu.ac.jp).

In this paper, we introduce the adaptive image processing filters that match for automatic construction method. The processing of the adaptive filters is decided based on the local features of an input image. We implement them to feed-forward GIN (FFGIN) [13] and modify it to be parameter tunable. Then we apply it to the problems of segmentation of organs and tissues in medical images. It is not possible to segment target regions only by the intensity of the pixels in these problems. We solve these problems using the parameter tunable FFGIN.

The next section of this paper is an overview of several related works. In section III, we describe our proposed method. Next, in section IV, we apply the proposed method to the problems of automatic construction of image transformation for medical images and show several experimental results. Finally, in section V, we describe conclusions and future work.

II. Related Works

A. Automatic Construction of Image Transformation

In image processing, it is difficult to select filters satisfying the transformation from original images to their target images. The ACTIT system [10], [11] has been proposed previously. ACTIT approximates adequate image transformation from original images to their target images by a combination of several known image processing filters. ACTIT constructs tree-structured image processing filters using GP. The individual in ACTIT is a tree-structured image transformation. The terminal nodes of a tree are the original images, whereas the non-terminal nodes are several kinds of image processing filters. A root node represents an output image. Users provide training images, and the ACTIT system automatically constructs appropriate image processing procedures. 3D-ACTIT [11] is an extended method that automatically constructs various 3D image processing procedures, and is applied to medical image processing and video image processing [14], [15].

Recently, two other extensions of ACTIT, genetic image network (GIN) [12] and feed-forward genetic image network (FFGIN) [13], have been proposed. Instead of a tree representation, they are represented by a network structure. Their biggest difference from ACTIT is the structure of connections between image processing filters. In general, a network structure theoretically includes a tree structure (i.e., a network structure also represents a tree structure). Therefore, the descriptive ability of a network representation is higher than that of a tree structure. In GIN and FFGIN, a genotype is a string of integers that indicate image processing filter types and connections. Other studies show that GIN and FFGIN automatically construct a simple structure for complex image transformation using their network representation [12], [13]. The GIN for Image Classification (GIN-IC) [16] is the extended method of GIN, which constructs image classification algorithms containing image transformation (preprocessing) parts.

Other approaches using linear genetic programming (LGP) succeed in automatic construction of image transformations and recognition [17], [18], [19]. Bhanu and Lin evolve composite operators for object detection from combinations of primitive image processing operators [20]. The image segmentation programs for a biomedical image are also evolved by combining ordinary image processing operators in [21]. Colton and Torres evolve an image filter to approximate an artistic image filter such as a filter in Photoshop using tree-based GP [22].

B. Prospects of Previous Works

The automatic construction methods for image transformation such as ACTIT, GIN, and other techniques show the effectiveness in various tasks. The simple well known image processing filters are prepared in these techniques, e.g., mean filter, maximum filter, minimum filter, Sobel filter, Laplacian filter, gamma correction filter, binarization, linear transformation, difference, logical sum, logical prod, and so on. We, however, consider that they have following problems:

- It is necessary to prepare the image processing filters beforehand (The processing of prepared filters is fixed).
- It tends to construct image transformations that overfit to the training images.

These problems are often observed when applying them to the medical image processing in which texture pattern is important. The first problem means that we have to prepare sufficient filters to solve the given target problems. However, it is difficult to know the effective image processing filters for given target problems beforehand. The second problem suggests that the evolutionary optimization only performs to approximate to the target images without considering the features of the images. To overcome these problems, we believe that the adaptive image processing filters that consider the features of the input images are effective. It is possible to construct the image transformation procedures that consider the features of the original images and the transformed image.

III. Methodology

A. Adaptive Image Processing Filters Based on Local Features of Image

Here, we describe our proposed adaptive image processing filters based on local features of image.

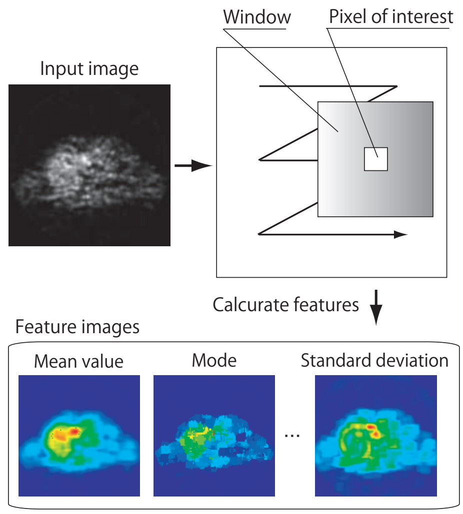

Fig. 1. Calculation of local features using sliding window.

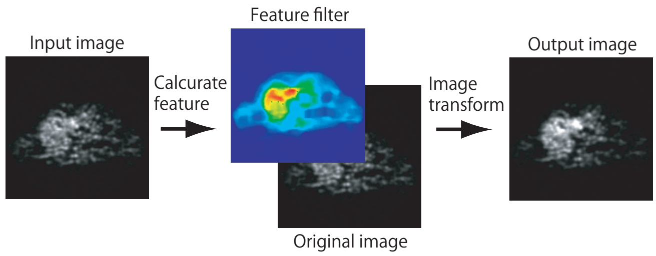

Fig. 2. Filtering using calculated local features.

1) Calculation of Local Features: We do not calculate one feature from the entire image but calculate the local features of each pixel to grasp the feature of each tissue. We prepare the sliding window considering the region of a pixel and around it. The sliding window is scanned in the image, and features in the window (local features) are calculated. In our method, we prepare the local texture features to grasp the features of each organ and tissue in medical images. These local features are useful to the segmentation of each organ and tissue that is difficult to extract only by pixel intensities. Fig. 1 shows the example of the images calculated local features in PET (positron emission tomography) images. The red color indicates high feature values and the blue color indicates low feature values in feature images. We can observe the different feature between the feature images and the original image.

2) Filtering Using Calculated Local Features: Image filtering is composed by the local features using sliding window described in III-A.1. Image transformation method is decided based on the values of the calculated local features. Concretely, the calculated values of the local feature are overlapped with the input image, as shown in Fig. 2. Then the pixel intensities of the input image are emphasized (or weakened) according to the feature values of the same position. The emphasizing image transformation (emphasizing

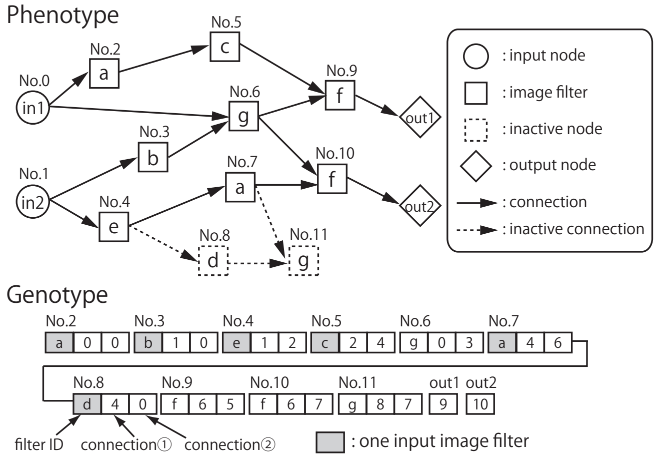

Fig. 3. Structure of FFGIN (phenotype) and the genotype.

filter) is performed by the following equation.

$$O_{ij} = \sum_{i=1}^{W} \sum_{j=1}^{H} I_{ij} V_{\text{weight}} (F_{ij} + V_{\text{offset}}) \tag{1}$$

The weakening image transformation (weakening filter) is performed by the following equation.

$$O_{ij} = \sum_{i=1}^{W} \sum_{j=1}^{H} I_{ij} V_{\text{weight}} ((1.0 - F_{ij}) + V_{\text{offset}}) \tag{2}$$

where O_{ij} are the pixel values of output image, I_{ij} are the pixel values of input image, and F_{ij} are the normalized values of local feature among $[0.0, 1.0]$. The numbers of pixels in the direction i and j are W, H respectively. V_{weight} and V_{offset} are weight value and offset value, respectively. We use $V_{\text{offset}} = 0.5$ in this paper. The parameter V_{weight} and the window size for calculation of the local features are optimized in evolutionary process.

This method calculates the local feature of the input image, and transforms it. Although this algorithm is simple, the adaptive image processing filters in which the processing changes according to the input image can be generated only with the preparation of the local features.

In this paper, the local features are calculated by simple statistical values, e.g., mean value, standard deviation, maximum value, minimum value, mode, 1 sigma in rate, 3 sigma out rate, skewness, kurtosis, and so on. The details of the local features are described in IV-A). We prepare emphasizing and weakening filters that are defined Equation (1) and (2) respectively for each local feature.

B. Parameter Tunable Feed Forward Genetic Image Network

1) Structure of FFGIN: FFGIN [13] constructs an acyclic network-structured image transformation procedure automatically. Fig. 3 shows an example of the phenotype (feed-forward network structure) and genotype (string representing the phenotype) of FFGIN. One of the benefits of this type of representation is that it allows the implicit reuse of nodes in its network. The nodes of FFGIN are categorized into three types: input nodes, image transformation nodes, and output nodes. Input nodes correspond to original images. Image transformation nodes execute image transformation using the corresponding image processing filters. The output images are obtained from output nodes. In FFGIN, the feed-forward network structure of nodes is evolved, as shown in Fig. 3. Numbers are allocated to each node beforehand. Increasingly large numbers are allocated, in order, to input nodes, image transformation nodes, and output nodes. The feedback structure is restricted at the genotype level. The nodes take their input from the output of previous nodes in a feed-forward manner. Therefore, it is possible to straightforward execution of network structured image transformation.

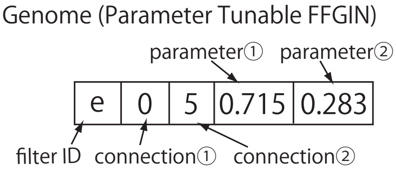

Fig. 4. Example of each genome that denotes a list of filter ID, connections, and parameters in parameter tunable FFGIN.

To adopt an evolutionary method, FFGIN uses genotype-phenotype mapping. This genotype-phenotype mapping method is similar to Cartesian genetic programming (CGP) [23], [24]. The feed-forward network structure is encoded in the form of a linear string. The genotype in FFGIN is a fixed length representation. To evolve the parameters of nodes simultaneously with the structure, we modify the genotype of FFGIN. We add the genome that denotes the parameters of nodes to the genotype. The genotype consists of a string that encodes the node function ID, connections, and parameters of each node in the network. The parameter values lie in $[0.0, 1.0]$. Fig. 4 illustrates the example of a genome that denotes a list of filter ID, connections, and parameters in FFGIN. The number of nodes in the phenotype can vary in a restricted manner, as not all of the nodes encoded in the genotype have to be connected. This allows the existence of inactive nodes. In Fig. 3, node No. 8 and 11 are inactive nodes.

The length of the genotype is fixed and equals to $N_{\text{trans}}(1 + n_{\text{in}} + n_{\text{para}}) + N_{\text{out}}$, where N_{trans} is the number of image transformation nodes, n_{in} is the maximum number of inputs of predefined filters, n_{para} is the maximum number of parameters of predefined filters, and N_{out} is the number of output nodes. Because FFGIN constructs a feed-forward network structured image transformation, it can represent multiple outputs. Therefore, FFGIN enables to construct multiple-output image transformation using only a single network structure.

2) Genetic Operators and Generation Alternation Model: To obtain the optimum structure, an evolutionary method is adopted. The genotype of FFGIN is a linear string. Therefore, it is able to use the standard genetic operators.

In this paper, we use uniform crossover and mutation as the genetic operators. The uniform crossover operator affects two individuals, as follows:

- Select several genes randomly according to the uniform crossover rate P_c for each gene.
- The selected genes that denote filter ID or connection are swapped between two parents. The selected genes that denote the parameter are updated to the value of $rp_1 + (1 - r)p_2$, where p_1 and p_2 are the selected parameters of parents 1 and 2, respectively, r is a uniformly distributed random number among $[0.0, 1.0]$.

The mutation operator affects one individual, as follows:

- Select several genes randomly according to the mutation rate P_m for each gene.
- Change the selected genes randomly under the constraints of the structure.

It uses the minimal generation gap (MGG) model [25] as a generation alternation model. The MGG model is a steady-state model proposed by Satoh et al., having a desirable convergence property of being able to maintain the diversity of the population, and shows higher performance than other conventional models in a wide range of applications (especially real parameter optimization [26], [27]). The MGG model in this paper is summarized as follows:

1) Set the generation counter $t = 0$. Generate N individuals randomly as the initial population $P(t)$.
2) Select a set of two parents M by random sampling from the population $P(t)$.
3) Generate a set of m children C by applying the crossover and the mutation operation to M.
4) Select two individuals from set $M + C$. One is the elitist individual and the other is the individual from the tournament selection. Then replace M with the two individuals in population $P(t)$ to get population $P(t+1)$.
5) Stop if a certain specified condition is satisfied; otherwise, set $t = t + 1$ and go to step 2.

The MGG model localizes its selection pressure, not to the whole population as simple GA or steady state does, but only to the family (children and parents). In the experiments we use children size m of 10 and tournament size 5.

IV. Experiments and Results

FFGIN enables to construct multiple-output image transformation procedures using only a single network structure. In this section, the image transformation algorithms that extract multiple regions from the PET (positron emission tomography) images are constructed by FFGIN. In conventional method, ACTIT, it is necessary to construct the image transformation algorithms of each region. Our proposed method is able to represent multiple image transformation procedures using a single network. We expect to construct compact structure by reusing of nodes.

TABLE I

Prepared adaptive image processing filters based on local features used in the experiments.

Mark		*Name of local feature*
Emphasizing	*Weakening*	
W-	w-	Mean value
WM	wM	Maximum value
Wm	wm	Minimum value
WS	wS	Standard deviation
WT	wT	Maximum value - minimum value
WI	wI	1 sigma in rate
WO	wO	3 sigma out rate
WF	wF	First quartile
WD	wD	Median
WQ	wQ	Third quartile
Wd	wd	Mode
Wk	wk	Skewness
WL	wL	Kurtosis

TABLE II

Prepared ordinary image processing filters for FFGIN used in the experiments.

Mark	*Name of image processing filter*
One-input one-output filters	
G2	Gamma correction ($\gamma = 0.2 \sim 2.2$)
I2	Inversion: $f \to V_{\max} - f$
Two-input one-output filters (f_1: input image 1, f_2: input image 2)	
2L	Logical sum: $max(f_1, f_2)$
2l	Logical product: $min(f_1, f_2)$
2A	Algebraic sum: $f_1 + f_2 - f_1 \cdot f_2/V_{\max}$
2a	Algebraic product: $f_1 \cdot f_2/V_{\max}$
2B	Bounded sum: $f_1 + f_2$
2b	Bounded product: $f_1 + f_2 - V_{\max}$
2u	Drastic sum: $f_1 = 0 \to f_2, \mathrm{or} f_2 = 0 \to f_1,$ $\mathrm{or} f_1, f_2 \neq 0 \to V_{\max}$
2U	Drastic product: $f_1 = V_{\max} \to f_2, \mathrm{or} f_2 = V_{\max} \to f_1,$ $\mathrm{or} f_1, f_2 \neq V_{\max} \to 0$
2D	Difference: $\lvert f_1 - f_2 \rvert$

A. Experimental Settings

The prepared adaptive image processing filters based on local features used in the experiments are shown in Table I. There are two filters to each local feature, which are emphasizing and weakening filters. Capital letter "W" of the mark in Table I indicates emphasizing filter, and small letter "w" of the mark in Table I indicates weakening filter. The parameters of each filter, window size and V_{weight}, are optimized simultaneously with the network structure. The window size takes any of the size of 7×7, 9×9, 11×11, and 13×13 according to the value of the first parameter gene. V_{weight} is a value within the range of $[0.3, 1.8]$ according to the value of the second parameter gene. Because offset value V_{offset} in Equation (1) and (2) is assumed to be 0.5, the pixel intensities of the input image are multiplied by the value among $[0.15, 2.7]$ by filtering once.

A part of ordinary image processing filters is also used for FFGIN. The ordinary image processing filters used in the experiments are shown in Table II. The parameter of γ in gamma correction is also optimized in evolutionary process. γ is a value within the range of $[0.2, 2.2]$ according to the first value of the parameter gene.

Although we do not prepare binarization filter for FFGIN,

TABLE III

PARAMETERS OF FFGIN USED IN THE EXPERIMENTS.

Parameter	*Value*
Number of generations	10000
Population size	200
Crossover rate	0.7
Uniform crossover rate (P_c)	0.1
Mutation rate (P_m)	0.05
Maximum number of image transformation nodes	50

the binarization is executed in the output nodes to obtain the binary output images. The threshold of the binarization is decided by discriminant analysis.

The parameters of FFGIN are shown in Table III. We use the mean error on the training images as a fitness function for FFGIN. The fitness function F used in the experiments is described as follows:

$$F = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}\left\{1 - \frac{\sum_{i=1}^{W}\sum_{j=1}^{H} w_{ij}^{mn}|o_{ij}^{mn} - t_{ij}^{mn}|}{V_{\max}\sum_{i=1}^{W}\sum_{j=1}^{H} w_{ij}^{mn}}\right\} \quad (3)$$

where o_{ij}^{mn} are the pixel values of transformed images (output images), t_{ij}^{mn} are their target ones, and w_{ij}^{mn} are their weight ones that indicate the important degree of pixel. The numbers of pixels in the direction i and j are W, H, respectively, the number of output images is M, the number of training images is N, and the maximum value of each pixel is $V_{\max}$ $(= 255)$. The target images are the images that users require after the image processing (ideal results). The range of this fitness function is $[0.0, 1.0]$. A higher numerical value indicates better performance.

B. Segmentation of Lungs, Heart, and Body Regions

The training images we used in this experiment appear in Fig. 5. The training images consist of an original image, target images, and weight images. This original image is PET (positron emission tomography) image. PET is a kind of medical image and indicates the metabolizing of human body. This technology can determine the presence and severity of cancers, thus PET is recently becoming a major diagnostic imaging method. All images used in the experiments are gray scale images and the size of 128 × 128 pixels. Because of low resolution and unclear indication, it is hard to distinguish borderline of internal organs. We prepared the target and the weight images manually. The purpose of this image transformation is to extract lung regions with weak accumulation, a heart region placed between lungs, and whole body region respectively. The target and the weight images 1, 2, and 3 correspond to the regions of lungs, heart, and body, respectively.

The obtained network structured image transformation procedure, the output images, and the images under transformation are displayed in Fig. 6. From this structure, a common structure is used for multiple outputs at the early stage of the processing. Our proposed adaptive filters that use standard deviation and mean value as the local feature are used at the early stage of the processing for smoothing. Then it branches after inversion filter (I2) is executed. In the first route, it processes in order of "W-" with the window size of 11×11, "WM" with the window size of 11×11 (emphasizing the region of lungs and the background), and "wM" with the window size of 13×13 (weakening the region of the background), and the output image 1 to which only the region of lungs is extracted is obtained finally. In the second route, it processes in order of "w-" with the window size of 7×7 (smoothing the heart accumulation part mainly), and "WM" with the window size of 9×9 (emphasizing the region of heart accumulation), and the output image 2 that is only extracted the region of heart is obtained finally. In the third route, it processes of "w-" with the window size of 7×7 for smoothing the heart accumulation part. The region of whole body transforms into the same pixel intensities because of this processing, and the output image 3 to which the region of whole body is extracted is obtained finally. All output images are similar to the target images, and the fitness value of this image transformation is 0.983. In addition, the constructed structure is compact and reuses the transformed images in its network.

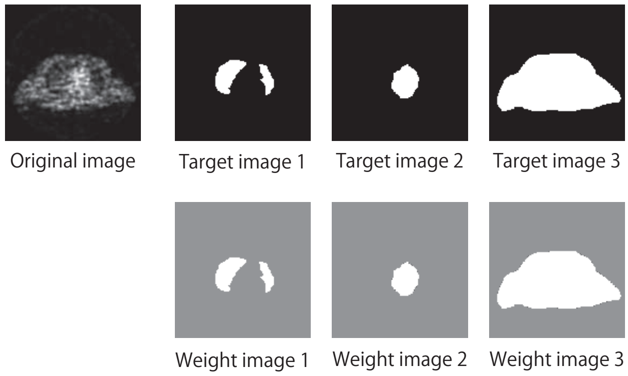

Fig. 5. Training images in the experiment of segmentation of lungs, heart, and body regions.

Next, we apply the constructed network structured image transformation procedure to the test images. The test images used in the experiments are shown in Fig. 7, which are not used in evolutionary process (the non-training images that are similar to training images). The number of the test images is three. The results (the output images) also appear in Fig. 7. From these output images, the obtained image transformation extracts each region correctly and outputs the ideal images. It shows that our method constructs general image transformation for segmentation from PET images through learning.

C. Segmentation of Lungs, Cancer, and Body Regions

The training images we used in this experiment appear in Fig. 8. This PET image is an image of the patient with the lung cancer. Strong accumulation exists in the lungs region where originally weak accumulation should be shown. This

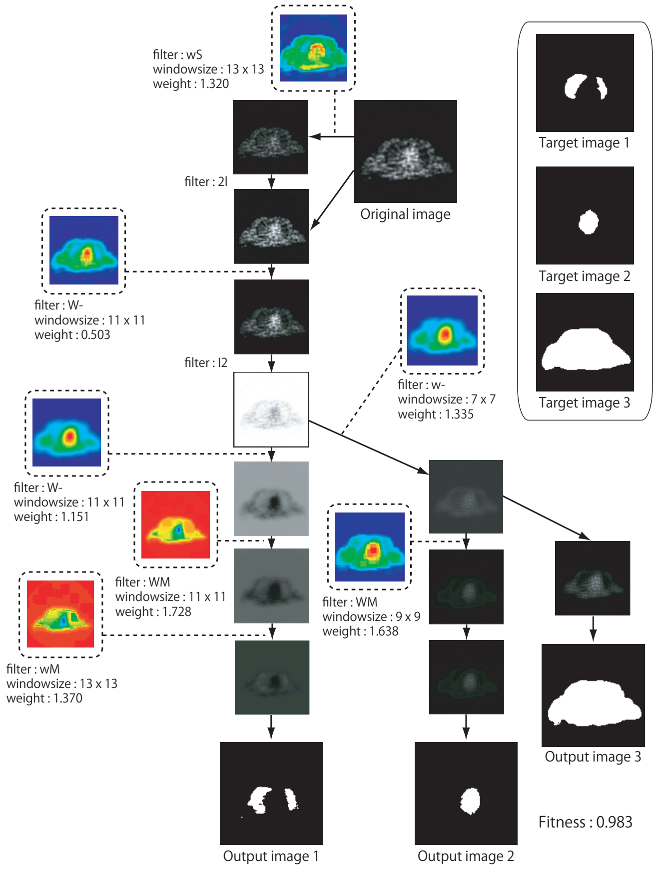

Fig. 6. Obtained network structured image transformation procedure for segmentation of lungs, heart, and body regions.

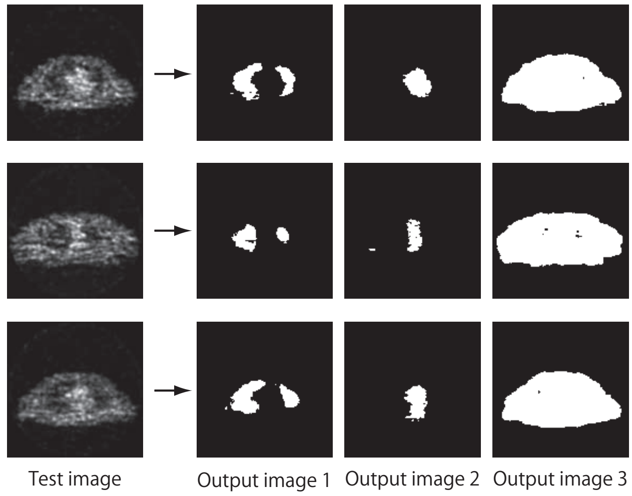

Fig. 7. Test images that are not used in evolutionary process, and their output images transformed by the obtained image transformation procedure for segmentation of lungs, heart, and body regions.

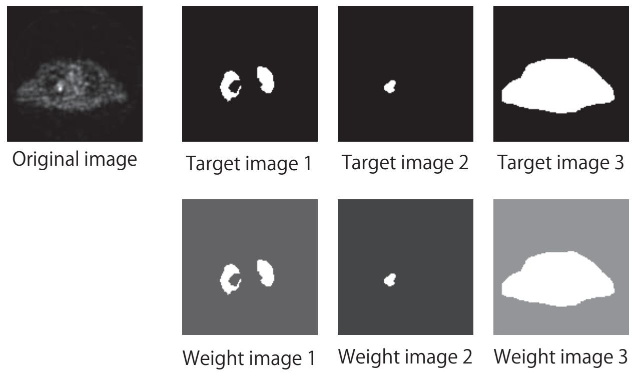

Fig. 8. Training images in the experiment of segmentation of lungs, cancer, and body regions.

strong accumulation appears from the carcinoma cell. The purpose of this image transformation is to extract normal lung regions, regions with abnormal accumulation, and whole body region, respectively. The target and the weight images 1, 2, and 3 correspond to the regions of lungs, abnormal accumulation, and body, respectively.

The obtained network structured image transformation procedure, the output images, and the images under transformation are displayed in Fig. 9. According to this structure, a reusing structure is used at the early stage of the processing as well as the previous experiment. It processes in order of "wT" with the window size of 15×15 (weakening the region of abnormal accumulations that have high intensity), and calculate the drastic product (2U) to obtain the output image 2 that is extracted only abnormal accumulation. To obtain output image 1 that is only extracted the region of lungs, the region of lungs in the inverted image is emphasized with "W-" with the window size of 9×9, then the background is weakened with "wM" with the window size of 15×15. All output images are similar to the target images as well as the previous experiment, and the fitness value of this image transformation is 0.984.

Next, we apply the constructed network structured image transformation procedure to the test images. The test images used in the experiment are shown in Fig. 10. The results (the output images) also appear in Fig. 10. From these output images, it succeeds in extraction of the lung cancer accumulations to which we cannot predict in the lung. Moreover, it extracts normal lung regions and whole body region simultaneously.

D. Comparison with Conventional ACTIT

We compare the proposed method to conventional ACTIT to verify the effectiveness of it. ACTIT constructs tree structured image processing filters to satisfy the given conversion from original to target images using predefined well-known filters. Thus, ACTIT only represents one output image transformation algorithms. In this experiment, we use only ordinary image processing filters without using our proposed adaptive image processing filters. We prepare simple and well-known image processing filters for ACTIT in the experiment (21 one-input one-output filters and 9 two-input one-output filters), e.g., mean filter, maximum filter, minimum filter, Sobel filter, Laplacian filter, gamma correction filter, binarization, linear transformation, difference, logical sum, logical prod, and so on.

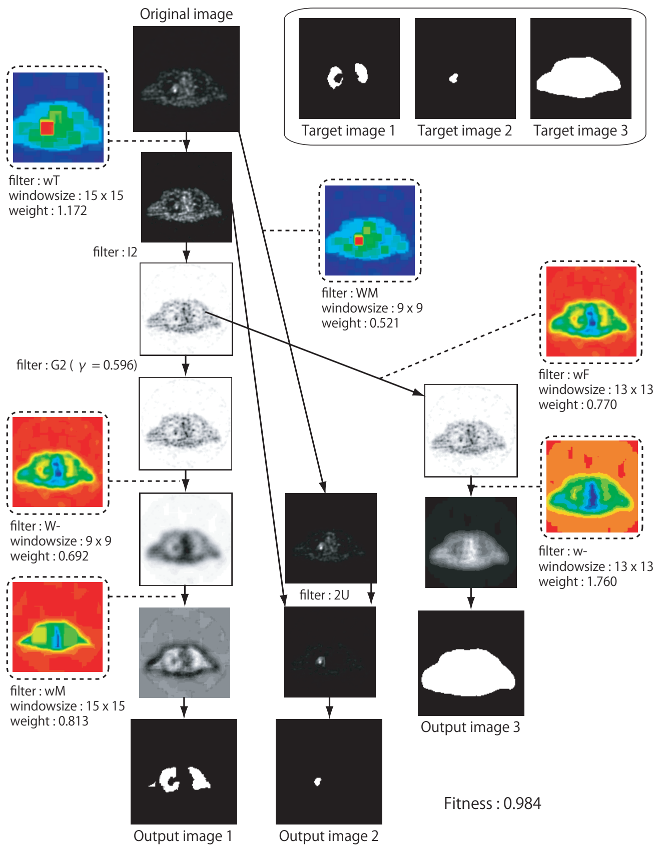

Fig. 9. Obtained network structured image transformation procedure for segmentation of lungs, cancer, and body regions.

We apply ACTIT to construction of a segmentation algorithm that extracts the region of lungs from the PET images of the normal patient. The training images and their output images used in this experiment appear in Fig. 11. We prepare three training images for ACTIT. The output images are relatively approximated to the target images although the shape of the output image is distorted. Fig. 12 shows the test images and their output images transformed by the obtained image transformation procedure by conventional ACTIT. From these output images, the over-extraction and extraction shortage of the lungs are observed. The performances of the extraction are worse than our proposed method. This cause is that the image processing filters prepared in conventional method cannot consider the local features. Therefore, it tends to construct the image transformation procedures that specialize in the training images. On the other hand, the general effective algorithms for the test image are constructed by our proposed method. In addition, our method constructs the multiple output image transformation using a single network.

V. Conclusions and Future Work

In this paper, we introduce the adaptive image processing filters that process according to the features of an input image to FFGIN. The processing of the adaptive filters is decided based on the local features of an input image. We implement them in FFGIN and modify it to be parameter tunable. Then we apply it to the problems of automatic construction of segmentation algorithms of organs and tissues in medical images. It is not possible to simply judge the segmented regions by the intensity of the pixels in these problems. Experimental results show that the image transformation procedures with multiple outputs are constructed, and we confirmed that our method is effective and general than the conventional method for medical images. In future work, we have to perform a quantitative analysis about this method using more images. Then, we will apply the proposed method to other medical images, such as CT, MRI, and DWI (diffusion weighted image). Moreover, we will introduce the other features to our method and construct more effective image transformation algorithms.

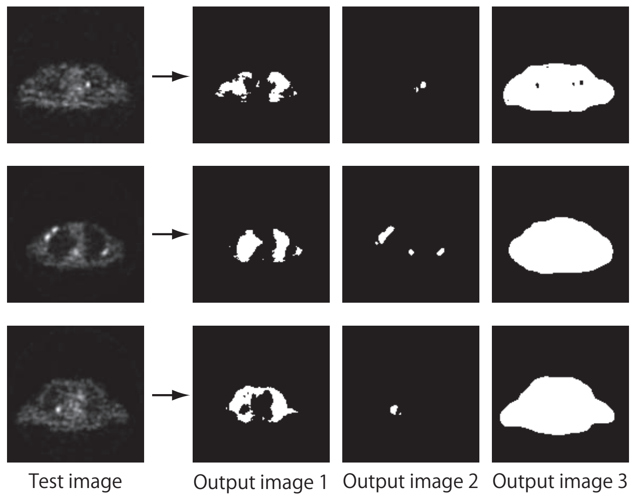

Fig. 10. Test images that are not used in evolutionary process, and their output images transformed by the obtained image transformation procedure for segmentation of lungs, cancer, and body regions.

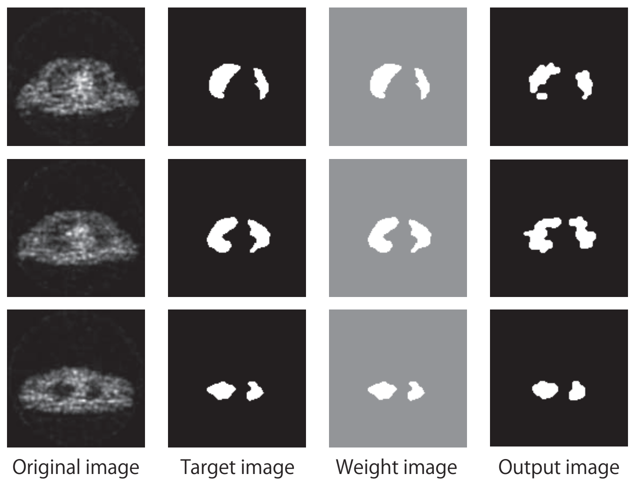

Fig. 11. Training images and their output images transformed by the obtained image transformation procedure by conventional ACTIT for segmentation of lungs.

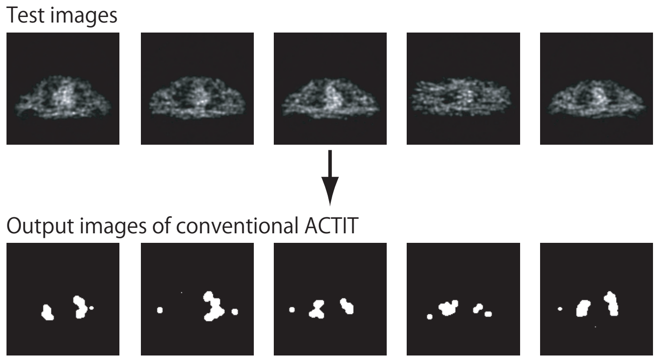

Fig. 12. Test images that are not used in evolutionary process, and their output images transformed by the obtained image transformation procedure by conventional ACTIT for segmentation of lungs.

References

[1] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[2] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza).

[3] W. A. Tackett, "Genetic programming for feature discovery and image discrimination," in *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA '93)*. Morgan Kaufmann Publishers, 1993, pp. 303–309.

[4] A. Teller and M. Veloso, "PADO: A new learning architecture for object recognition," in *Symbolic Visual Learning*, K. Ikeuchi and M. Veloso, Eds. Oxford University Press, 1996, pp. 81–116.

[5] M. Zhang and C. G. Fogelberg, "Genetic programming for image recognition: An LGP approach," in *Applications of Evolutionary Computing: EvoWorkshops 2007*, ser. LNCS, vol. 4448. Springer-Verlag, 2007, pp. 340–350.

[6] B. Lam and V. Ciesielski, "Discovery of human-competitive image texture feature extraction programs using genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference 2004 (GECCO '04), Part II*, ser. LNCS, vol. 3103. Springer-Verlag, 2004, pp. 1114–1125.

[7] M. Aurnhammer, "Evolving texture features by genetic programming," in *Applications of Evolutionary Computing: EvoWorkshops 2007*, ser. LNCS, vol. 4448. Springer-Verlag, 2007, pp. 351–358.

[8] P. Ghosh and M. Mitchell, "Segmentation of medical images using a genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference 2006 (GECCO '06)*. ACM, 2006, pp. 1171–1178.

[9] T. Kowaliw, W. Banzhaf, N. Kharma, and S. Harding, "Evolving novel image features using genetic programming-based image transforms," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC '09)*, 2009, pp. 2502–2507.

[10] S. Aoki and T. Nagao, "Automatic construction of tree-structural image transformation using genetic programming," in *Proceedings of the 1999 International Conference on Image Processing (ICIP '99)*, vol. 1. IEEE, 1999, pp. 529–533.

[11] 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 2004 (IWAIT '04)*, 2004, pp. 529–533.

[12] S. Shirakawa and T. Nagao, "Genetic image network (GIN): Automatically construction of image processing algorithm," in *Proceedings of the International Workshop on Advanced Image Technology 2007 (IWAIT '07)*, 2007, pp. 643–648.

[13] S. Shirakawa and T. Nagao, "Feed forward genetic image network: Toward efficient automatic construction of image processing algorithm," in *Advances in Visual Computing, Third International Symposium, ISVC 2007, Part II*, ser. LNCS, vol. 4842. Springer-Verlag, 2007, pp. 287–297.

[14] Y. Nakano and T. Nagao, "Automatic extraction of internal organs region from 3D PET image data using 3D-ACTIT," in *Proceedings of the International Workshop on Advanced Image Technology 2006 (IWAIT '06)*, 2006.

[15] Y. Nakano and T. Nagao, "Automatic construction of moving object segmentation from video images using 3D-ACTIT," in *Proceedings of the 2007 IEEE International Conference on Systems, Man, and Cybernetics (SMC '07)*, 2007, pp. 1153–1158.

[16] S. Shirakawa, S. Nakayama, and T. Nagao, "Genetic image network for image classification," in *Applications of Evolutionary Computing: EvoWorkshops 2009*, ser. LNCS, vol. 5484. Springer-Verlag, 2009, pp. 395–404.

[17] K. Krawiec and B. Bhanu, "Visual learning by coevolutionary feature synthesis," *IEEE Transactions on System, Man, and Cybernetics –Part B*, vol. 35, no. 3, pp. 409–425, 2005.

[18] K. Krawiec and B. Bhanu, "Visual learning by evolutionary and coevolutionary feature synthesis," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 635–650, 2007.

[19] U. Watchareeruetai, Y. Takeuchi, T. Matsumoto, and N. Ohnishi, "Transformation of redundant representations of linear genetic programming into canonical forms for efficient extraction of image features," in *Proceedings of the 2008 IEEE World Congress on Computational Intelligence (WCCI '08)*, 2008, pp. 1996–2003.

[20] B. Bhanu and Y. Lin, "Learning composite operators for object detection," in *Proceedings of the Genetic and Evolutionary Computation Conference 2002 (GECCO '02)*. Morgan Kaufmann Publishers, 2002, pp. 1003–1010.

[21] T. Singh, N. N. Kharma, M. Daoud, and R. Ward, "Genetic programming based image segmentation with applications to biomedical object detection," in *Proceedings of the Genetic and Evolutionary Computation Conference 2009 (GECCO '09)*. ACM, 2009, pp. 1123–1130.

[22] S. Colton and P. Torres, "Evolving approximate image filters," in *Applications of Evolutionary Computing: EvoWorkshops 2009*, ser. LNCS, vol. 5484. Springer-Verlag, 2009, pp. 467–477.

[23] J. F. Miller and P. Thomson, "Cartesian genetic programming," in *Genetic Programming: 3rd European Conference, EuroGP 2000*, ser. LNCS, vol. 1802. Springer-Verlag, 2000, pp. 121–132.

[24] J. F. Miller and S. L. Smith, "Redundancy and computational efficiency in cartesian genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 167–174, 2006.

[25] H. Satoh, M. Yamamura, and S. Kobayashi, "Minimal generation gap model for considering both exploration and exploitations," in *Proceedings of the IIZUKA'96*, 1996, pp. 494–497.

[26] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference 1999 (GECCO '99)*. Morgan Kaufmann Publishers, 1999, pp. 657–664.

[27] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evolutionary Computation*, vol. 10, no. 4, pp. 371–395, 2002.