Journal of The Korea Society of Computer and Information
Vol. 21 No. 5, pp. 79-89, May 2016
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.5.079

Object Classification Method Using Dynamic Random Forests and Genetic Optimization

Jae Hyup Kim*, Hun Ki Kim**, Kyung Hyun Jang***, Jong Min Lee****, Young Shik Moon*****

Abstract

In this paper, we proposed the object classification method using genetic and dynamic random forest consisting of optimal combination of unit tree. The random forest can ensure good generalization performance in combination of large amount of trees by assigning the randomization to the training samples and feature selection, etc. allocated to the decision tree as an ensemble classification model which combines with the unit decision tree based on the bagging. However, the random forest is composed of unit trees randomly, so it can show the excellent classification performance only when the sufficient amounts of trees are combined. There is no quantitative measurement method for the number of trees, and there is no choice but to repeat random tree structure continuously. The proposed algorithm is composed of random forest with a combination of optimal tree while maintaining the generalization performance of random forest. To achieve this, the problem of improving the classification performance was assigned to the optimization problem which found the optimal tree combination. For this end, the genetic algorithm methodology was applied. As a result of experiment, we had found out that the proposed algorithm could improve about 3~5% of classification performance in specific cases like common database and self infrared database compare with the existing random forest. In addition, we had shown that the optimal tree combination was decided at 55~60% level from the maximum trees.

▸Keyword: Object Classification, Random Forest, Genetic Algorithm, Classifier Ensemble

• First Author: Jae Hyup Kim, Corresponding Author: Young Shik Moon
*Jae Hyup Kim(jaehyup.kim@hanwha.com), Dept. of Image Sensor Team, Hanwha Thales Co.
**Hun Ki Kim(hunki.kim@hanwha.com), Dept. of Image Sensor Team, Hanwha Thales Co.
***Kyung Hyun Jang(kyunghyun.jang@hanwha.com), Dept. of Image Sensor Team, Hanwha Thales Co.
****Jong Min Lee(jmlee@visionlab.or.kr), Dept. of CSE, Hanyang University
*****Young Shik Moon(ysmoon@hanyang.ac.kr), Dept. of CSE, Hanyang University
• Received: 2015. 11. 30, Revised: 2015. 12. 22, Accepted: 2016. 01. 20.

I. Introduction

영상으로부터 추출된 객체의 특징 정보를 토대로 객체의 종류를 구분하는 분류기 모델링은 머신 러닝(machine learning) 분야의 오랜 연구 분야중 하나다. 국내외를 통틀어 많은 연구가 진행되어 왔으며 큰 틀에서 대표적인 연구를 언급한다면, 80년대의 신경망(neural networks)[1], 90년대의 SVM(support vector machine)[2]을 들 수 있다. 현재에도 대표적인 머신 러닝 학습 알고리즘으로 간주되고 있는 이 기법들은 경사하강 (gradient descent)기반의 최적화, 마진(margin) 최적화 등의 수학적 기반을 바탕으로 많은 분야에서 주목받아왔고, 다양한 분류 시스템에 적용되었다[3-6]. 그러나 시스템과 환경의 급속한 발전에 따라 대용량 학습 샘플의 처리, 빠른 분류, 높은 신뢰성 등의 새로운 요구에 직면하면서, 기존의 단일 분류기의 성능과 최적화에 대한 문제가 대두되었다. 이러한 문제에 대한 해결방안의 하나가 바로 분류기 앙상블 개념이다.

분류기 앙상블은 다수의 분류기를 합하여 군집화된 분류 모델을 설계하는 개념이다. 단일 분류기로 해결할 수 없는 복잡도가 매우 높은 분류 공간 문제에서 상대적으로 약한 해결능력을 가지는 분류기들을 모아서 최종의 복잡한 문제를 해결해 내는, 일종의 분할정복(divide-and-conquer)의 패러다임이다. 80년대 앙상블 분류기에 대한 실험적 연구가 진행되었으나, 90년대 후반 재샘플링(resampling) 방법론인 부스팅(boosting)과 배깅(bagging, bootstrap aggregating)을 토대로 두 가지 앙상블 분류기가 제안됨으로써 높은 관심과 연구가 진행되어 왔다[7]. 여기서 언급한 부스팅 기반의 앙상블 분류기는 에이다부스트(adaboost, adaptive boosting)[8] 알고리즘이며, 배깅 기반의 앙상블 분류기가 랜덤포레스트(random forest)[9] 이다.

에이다부스트는 주어진 분류 문제를 낮은 수준으로 해결할 수 있는 약한 분류기(weak classifier)를 설정하고, 단계적으로 반복 생성된 약한 분류기들을 결합하여 점차 강한 분류기(strong classifier)로 개선해 나가는 알고리즘이다. 이때, 반복마다 이전 단계에서 오류로 판단된 학습 샘플에 대해 가중치를 높게 둠으로써 다음 단계에서는 해당 학습 샘플을 우선적으로 분류하여 전체적인 성능을 점진적으로 부스팅 되도록 한다. 에이다부스트는 학습과 분류 단계에서 설계자가 고려해야할 파라미터가 매우 적고, 복잡한 결정 경계도 충분히 모델링할 수 있음이 증명되어 있다. 이러한 강력한 성능과 편의성으로 인해 에이다부스트는 국내외에서 여러 분류 문제에 적용되고 있으며, 우수한 성능을 입증하고 있다[10][11]. 그러나 주어진 학습 샘플 간의 분포에 대한 고민이 없고 약한 분류기 개수의 적정선이 불분명하기 때문에, 오버피팅(overfitting)의 위험에 빠질 수 있다. 또, 앙상블 분류기의 주요 관점인 단위 분류기의 다양성(diversity)의 측면에서도 약점을 보인다.

랜덤포레스트는 이러한 오버피팅의 문제를 극복하고 전체 분류기의 일반화 성능(generalization performance)을 보장하는 알고리즘이다. 랜덤포레스트는 트리(decision tree)를 모아 숲을 이룬다는 의미로써, 여기서의 트리는 조건과 분기를 통해 분류를 수행하는 결정트리(decision tree)[10]를 의미한다. 배깅으로 샘플링된 학습 샘플에 대해 랜덤하게 선택된 특징에 대해 분리를 반복하면서 트리를 구성하며, 무작위로 행해진 많은 수의 트리의 결과를 종합하여 랜덤포레스트의 결과가 결정된다. 이러한 생성 과정의 랜덤 요소들을 통해 단위 분류기의 다양성을 보장하고 있으며, 이를 기반으로 뛰어난 일반화 성능을 보여주고 있다.

제안하는 기법에서는 랜덤포레스트가 가지는 뛰어난 일반화 성능에 주목하였다. 더불어 일정 개수 이상의 트리가 확보될 경우, 에이다부스트 등 최고수준(state-of-the-art)의 분류 성능을 보일 수 있다[12][13]. 이런 장점은 객체(또는 표적)에 대한 충분한 샘플의 확보가 어렵고, 최고 수준의 정확도가 요구되는 군사시스템 분야의 분류 문제에 매우 적합하다고 볼 수 있다. 그러나 Breiman 등[9]의 랜덤포레스트의 가장 큰 문제는 적정한 수준의 트리의 개수에 대한 기준이 없는 것이다. 즉, 설계자에 따라 임의로 트리의 개수가 정해지기 때문에, 현재 학습된 랜덤포레스트가 최적의 분류 성능 상태인지에 대해 확인할 수 없다. 또, 성능 보장을 위해 트리의 개수를 늘릴 경우 각각의 트리가 독립적이고 임의적으로 추가되기 때문에, 랜덤포레스트의 분류 성능을 저하시키는 트리가 포함될 가능성이 있다[14][15].

많은 연구에서, 랜덤포레스트의 단점을 보완하고 성능을 향상시키기 위한 변형 알고리즘이 제안되었으며, 대표적으로 동적 랜덤포레스트(dynamic random forest)를 들 수 있다. Tripoliti 등[16]은 일정 개수의 트리를 생성한 후, 추가적으로 성능 최적화를 위한 트리를 한 개씩 추가해 나가는 동적 랜덤포레스트를 제안하였다. 트리가 추가될 때마다 성능변화 그래프를 커브피팅(curve fitting)하여 최적의 트리 조합에서 멈추는 방법이다. 이 방법은 초기 해를 설정하고 순차적으로 성능 향상에 도움이 되는 해를 추가해 나가는 순차 탐색(sequential search)의 개념과 유사하다. 그러나 최적해 집합을 찾는 문제에서 이러한 탐욕적(greedy) 방법론은 지역 최적점(local minima)에 빠질 가능성이 있다. 즉, 임의성을 가지는 트리들의 집합에서 순차적으로 합해지는 트리들이 성능을 단조적(monotonically)으로 향상시킨다는 보장이 없으므로, 트리 생성이 멈춘 상황에서 최대의 성능을 가지는 트리 조합이라는 보장이 없다. Bernard 등[17]은 차례로 생성되는 트리들이 이전 단계에서 오분류된 학습 샘플에 적합하도록 가중치를 주어 점진적으로 성능 최적화가 되도록 하는 동적 랜덤포레스트를 제안하였다. 이 방법은 랜덤 트리와 부스팅 개념을 결합한 것으로 Breiman 등의 랜덤포레스트에 비해 우수한 분류 성능을 보이지만, 생성되는 트리들이 독립적이지 않아 랜덤포레스트 본래의 일반화 성능을 저해할 가능성이 크며, 노이즈(noise) 샘플, 아웃라이어(outlier) 데이터에 민감할 수 있다. 또, Tripoliti 등과 Bernard 등의 동적 랜덤포레스트 모두 분류 성능 최대화의 관점에서만 트리를 평가하여 결합하고 있으며, 랜덤포레스트의 주요 장점중 하나인 다양성을 통한 일반화 성능에 대해서는 고려되어 있지 않다.

본 논문에서는 과생성(overproduce)된 랜덤 트리들을 대상으

로, 분류 성능과 다양성의 측면에서 최적의 트리 조합을 결정하는 유전적 동적 랜덤포레스트 알고리즘을 제안한다. 제안하는 알고리즘은 랜덤포레스트 알고리즘에 과생성과 선택의 개념(overproduce and choose paradigm)[18]을 도입하여 생성된 랜덤 트리들 중에서 최적의 트리 조합을 찾는 최적화 문제로 확장하였다. 최적화 문제에 대해서는 탐욕적 개념의 방법들보다 전역 최적점(global minima)을 찾을 가능성이 높은 유전 알고리즘(genetic algorithm)[19][20]을 적용하였으며, 유전적 진화의 방향을 분류 성능과 다양성을 동시에 평가하도록 하였다.

본 논문에서는 2장에서 랜덤포레스트의 기본 개념과 동적 랜덤포레스트에 대해서 설명하고, 3장에서는 유전 알고리즘 기반의 최적화에 대해 설명한다. 4장에서는 제안하는 알고리즘을 설명하고, 5장에서는 다양한 데이터베이스에 대한 분류성능과 다양성에 대한 실험 결과를 설명하고, 6장에서는 결론을 설명한다.

II. Random Forests

랜덤포레스트는 부트스트랩으로 만들어진 학습 샘플을 이용하여 결정트리를 만드는 과정의 반복이다. 표 1은 Breiman 등의 랜덤포레스트 알고리즘의 개요를 나타낸다. T는 포함될 트리의 개수, K는 특징의 개수, H는 특징에 대한 임계치 개수, D_{max}는 트리의 최대 깊이를 나타낸다. 알고리즘에 따라 주어진 학습 샘플로부터 부트스트랩으로 T개의 샘플 부집합(subset)을 생성하여 각각에 대한 트리를 구성한다. 트리에서는 랜덤하게 선택된 특징과 임계치를 이용하여 주어진 샘플을 나누었을 때, 가장 큰 게인(gain)을 가지는 특징과 임계치를 저장하며, 나누어진 데이터는 다시 다음 노드 생성과정에 사용된다. 이러한 반복과정은 게인이 0이 되어 더 이상 분리할 필요가 없거나, 노드의 깊이가 최대치에 도달했을 때 중지된다. T개의 트리를 합하여 랜덤포레스트가 되며, 분류 단계에서는 주어진 테스트 샘플에 대해 T개의 트리의 결과를 합산하여 최종 분류 결과를 결정한다.

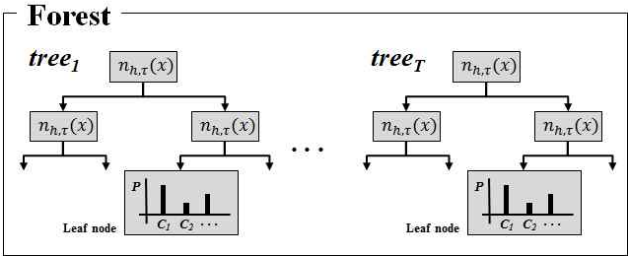

Fig. 1. The diagram of random forest

Table 1. The random forest algorithm

Algorithm 1

In : training samples $\varsigma = \{x_1, \ldots, x_N\}$
Define : T, K, H, D_{max}
Function RF(ς)
01: for t = 1 : T
02: make bootstrap subset ς_t from ς
03: $tree_t$ = TREE(ς_t)
04: end for
05: return forest = $\{tree_1, \ldots, tree_T\}$
Function TREE(ς)
06: $\Delta G_{max} \leftarrow \infty$, $\tau_{max} \leftarrow 0$, $f_{max} \leftarrow 0$
07: random selection $f = \{f_1, \ldots, f_k\}$, $\tau = \{\tau_1, \ldots, \tau_h\}$
08: for k = 1 : K
09: for h = 1 : H
10: split $\varsigma \rightarrow \{\varsigma_l, \varsigma_r\}$ using f_k, τ_h
11: Compute gain ΔG
12: if $\Delta G > \Delta G_{max}$
13: $\Delta G_{max} \leftarrow \Delta G$, $f_{max} \leftarrow f_k$, $\tau_{max} \leftarrow \tau_h$
14: end if
15: end for
16: end for
17: store τ_{max} , f_{max} to this node
18: if $\Delta G_{max} = 0$ or tree depth $\geq D_{max}$
19: store the probability distribition P (leaf node)
20: return this node
21: end if
22: TREE(ς_l)
23: TREE(ς_r)
24: return this node

그림 1은 랜덤포레스트의 구성 개념을 나타낸다. 각각의 단위 트리에서 잎노드(leaf node)는 도달한 학습 데이터들의 클래스 분포를 저장하며, 그 외의 노드는 샘플을 분기시키는 파라미터를 가진다. (그림 1에서는 $n_{h,\tau}(x)$로 표현되었다.)

표 1은 랜덤포레스트 알고리즘을 나타낸다. 일종의 메타휴리스틱(meta-heuristic) 알고리즘 이므로 각 단계별로 세부적인 알고리즘이 생략되었는데, 특징과 임계치의 랜덤 선택 범위, 게인의 계산, 분류 단계에서 독립적 트리의 결과에 대한 투표(voting) 등이다. 특징과 임계치의 선택은 필요에 따라 단일 또는 다수의 특징에 대해 평가하여 노드를 분기시킬 수 있다. 다수의 특징에 대해 평가하여 분리할수록 그 정확도는 높을 가능성이 크다. 그러나 앙상블 분류 모델의 관점에서 약한 분류기의 성능은 전체 분류 성능에 크게 영향을 미치지 않는다. 결과의 투표에서는 Breiman 등은 다수결 투표(majority voting)을 적용하였다. 즉, 각각의 트리가 잎노드에 저장된 P에 대해 가장 큰 값을 가지는 분류 결과를 출력하며, 각 투표 결과를 취합하여 가장 많은 표를 획득한 결과를 선택한다. 이 외에도, P의 값을 종합적으로 고려하는 가중치 투표(weighted voting) 방법이 일반적으로 사용된다. 게인의 계산 단계는 주어진 학습 샘플을

하위 노드로 분기하는 측정기준이다. 현재의 기준에 대한 분류 정도를 측정하는 단계로써, 엔트로피(entropy), 지니 지수(gini index), 분류율(classification ratio) 등 다양한 기준의 적용이 가능하다.

식 (1)은 지니 지수를 이용하여 분기 조건을 선택하는 가장 일반적인 방법을 나타낸다.

$$G_n = 1 - \sum_i P(c_i)^2 \qquad (1)$$
$$\Delta G = G_n - \frac{N_l}{N} G_l - \frac{N_r}{N} G_r$$

식 (1)의 n은 현재의 노드를 의미하며, l과 r은 각각 분기후의 왼쪽과 오른쪽 노드를 의미한다. c는 주어진 샘플데이터의 클래스 레이블을 의미하며, $P(c_i)$는 해당 클래스 샘플의 비율을 의미한다. N은 샘플의 개수를 나타낸다. 지니 지수는 현재 보유한 샘플의 클래스가 여러 종류일수록 큰 값을 가지며, 반대로 클래스가 동일한 샘플들로 구성될수록 작은 값을 가진다. 즉, 선택된 특징과 임계치에 따라 샘플을 분기 했을 때, 현재 노드의 지니 지수와 분기 후의 각 노드의 지니 지수를 비교하여 가장 변화량이 큰 경우를 선택하는 방법이다.

표 2는 Bernard 등의 동적 랜덤포레스트 알고리즘을 나타낸다. 알고리즘 2에서는 트리가 추가될 때마다 전체 샘플에 대한 가중치를 업데이트 하며, 단위 트리에서는 게인의 계산에서 샘플의 가중치를 반영함으로써 트리가 가중치가 높은 샘플을 우선적으로 분류하도록 한다.

5행의 o_t는 현재 트리에 대한 OOB(out-of-bag) 샘플을 의미한다. 11행의 $tree_{oob}$에 대한 정의는 식 (2)와 같다.

$$tree_{oob} = \{ tree_j \mid x_i \in o_j \ , 1 \le \ j < t \} \qquad (2)$$

즉, 샘플 x_i가 OOB 샘플로 포함된 트리의 집합을 의미한다. $tree_{oob}$가 없을 경우 이전의 가중치를 유지하며, 반대의 경우 해당 트리들에 대한 x_i의 분류 결과값을 이용하여 가중치를 업데이트 한다. 가중치 업데이트를 위한 함수는 식 (3)과 같다.

$$c(x) = \frac{1}{|tree_{oob}|} \sum_{tree_j \in tree_{oob}} I(tree_i(x) = y) \qquad (3)$$

I 는 분류 결과에 대한 지시함수(indicator function)이다. 함수 $c(x)$는 샘플을 OOB로 선택하고 있는 트리에서의 분류 결과의 평균을 의미한다. 즉, 이전의 트리들에서 잘 분류되고 있는 샘플은 높은 값이 부여된다. 따라서 알고리즘 2의 12행은 샘플의 분류가 잘 안될수록 1에 가까운 값으로 가중치를 업데이트 시킨다. 다음 반복에서는 역시 부트스트랩으로 대상 샘플들이 선택되고, 선택된 샘플들 중 가중치가 높은 샘플을 잘 분류하는 방향으로 트리가 구성된다.

Table 2. The dynamic random forest algorithm

Algorithm 2

Function DRF(ς)

01 : for i = 1 : N
02 : $D_1(x_i) = \frac{1}{N}$
03 : end for
04 : for t = 1 : T
05 : make bootstrap subset ς_t, o_t from ς
06 : $\varsigma_t \leftarrow \varsigma_t$ weighted with D_t
07 : $tree_t = TREE(\varsigma_t)$
08 : Z = 0
09 : for i = 1 : N
10 : $D_{t+1}(x_i) = D_t(x_i)$
11 : if $tree_{oob} \neq \varphi$
12 : $D_{t+1}(x_i) = 1 - c(x_i)$
13 : end if
14 : $Z = Z + D_{t+1}$
15 : end for
16 : for i = 1 : N
17 : $D_{t+1}(x_i) = \frac{D_{t+1}(x_i)}{Z}$
18 : end for
19 : end for
20 : return forest = $\{tree_1, ..., tree_T\}$

알고리즘 2는 단계별로 생성되는 트리가 이전에 생성된 트리들에서 잘 분류되지 않은 샘플에 대해 최적화되는 부스팅 개념을 도입하였다. 따라서 최종적으로 생성된 포레스트는 트리의 개수에 비례하여 학습 샘플을 최대한 잘 분류할 수 있는 가능성이 높아진다. 그러나 학습 샘플 중 존재할 수 있는 아웃라이어에 최적화된 트리들이 추가될 가능성이 높으며, 이는 전체 분류기의 일반화 성능을 저하시킬 수 있다. 이 단점은 학습 샘플의 개수가 적을수록 그 영향이 커지기 때문에 분류 문제의 상황에 따라 그 성능이 크게 달라질 수 있다. 반면에 충분한 학습 샘플이 확보된 상황이나 양질의 학습 샘플이 확보되는 상황에서는 높은 성능을 보이게 된다. 아울러, 알고리즘 2에서도 포레스트의 크기, 즉 트리의 개수가 동적이지 못하므로 최적의 트리 조합을 위해서는 충분한 크기를 설정해야만 한다.

III. Genetic Algorithm

유전 알고리즘은 확률적 최적화(stochastic optimization) 분야의 대표적인 방법이다.

Table 3. The genetic algorithm

Algorithm 3

In : solution space, $\theta \in S$

Define : G_{max} , K , N

Function GA

01 : random select $P = \{\theta_1, \ldots, \theta_N\}$

02 : for n = 1 : G_{max}

03 : calculate $Q(\theta_i), F(\theta_i)$, $i = 1, \ldots, N$

04 : for i = 1 : K

05 : random select θ_{p1}, θ_{p2}

06 : $\theta_i^{\circ} = \text{mutation}(\text{crossover}(\theta_{p1}, \theta_{p2}))$

07 : end for

08 : $p \leftarrow \theta_1^{\circ}, \ldots, \theta_k^{\circ}$ (replacement)

09 : if stop criteria

10 : break

11 : end if

12 : end for

13 : $\hat{\theta} = \arg\max_{\theta} Q(\theta_i)$, $i = 1, \ldots, N$

14 : return $\hat{\theta}$

라그랑제(lagrange) 등을 이용한 수학적 방법론이나 경사(gradient) 등을 이용한 탐욕적 방법론들과는 달리, 초기에 랜덤한 여러 개의 해를 집단(population)으로 설정한 후, 확률적으로 좋은 방향으로 교차(crossover)와 변이(mutation) 연산을 반복하여 해 집단을 최적의 해를 향해 진화시켜 나가는 방법론이다. 랜덤한 해 집단으로부터 출발하여 진화해 나가기 때문에 확률적으로 지역 최적점에 빠질 가능성이 매우 낮다. 여러 연구에서 복잡도가 높은 해 공간에서 탐욕적 방법론들 보다 우수한 성능을 보임이 실험적으로 입증되었다[19]. 또, 확률적 반복을 통한 최적화 기법이기 때문에 수학적 모델링이 어려운 복잡한 최적화 문제에도 적용할 수 있다. 표 3은 유전 알고리즘을 나타낸다. 유전 알고리즘은 기본적으로 찾고자 하는 최적해가 존재하는 전체 공간을 대상으로 하여 랜덤하게 N개의 해를 선택하여 초기해 집단을 만든다. 해집단을 모집단으로 하여, 두 개의 해를 임의 선택한 후, 교차와 변이연산으로 자식해(offspring solution)를 생성한다. K개의 자식해를 모집단의 K개의 해에 대해 대치하며, 이 과정을 반복한다. 반복이 종료된 후, 가장 좋은 해를 최적해로 선정한다.

G_{max}와 N은 사용자 정의 파라미터로써 유전진화 단계에 대한 반복수와 집단의 크기를 나타낸다. K는 유전진화마다 생성하고 대치할 해의 개수를 나타내는 사용자 정의 파라미터이며, K의 크기가 클수록 최적해를 찾을 확률은 높아지고 수행시간은 지수적으로 증가한다. 3행과 13행의 Q함수는 해에 대한 질(quality)를 계산하는 함수이다. 즉, 해의 최적도를 판단하는 함수로써 주어진 최적화 문제 자체를 의미한다. 3행의 F함수는 Q함수의 값을 이용하여 해의 적합도를 산술적으로 계산하는 함수를 의미한다. 유전 알고리즘은 자식해 생성을 위한 모집단 선택, 교차와 변이 방법, 대치 방법, F와 Q함수의 정의 등에서 많은 알고리즘이 정의될 수 있으며, 주어진 문제에 대해 변형이 필요하다. 제안하는 알고리즘에서도 최적의 트리 조합을 찾기 위해 단계별로 적합한 기법을 정의하였다.

IV. The Proposed Algorithm

본 논문에서는 과생성과 선택의 패러다임 하에서, 배깅으로 생성된 트리들에 대해 유전 알고리즘을 이용하여 분류 성능과 다양성의 측면에서 가장 우수한 조합을 찾아내는 유전적 동적 랜덤포레스트 알고리즘을 제안한다. 기본적인 알고리즘 구조는 표 4와 같다. 알고리즘 4.1의 1~5행은 과생성된 단위 트리로 구성된 랜덤포레스트의 구성 과정이다. 세부 수행과정은 알고리즘 1과 같다.

6행에서는 유전알고리즘으로의 적용을 위하여, N개의 염색체를 임의 선택한다. 유전알고리즘의 기본구조에 따라 염색체(chromosome)가 정의되며, 6행의 ch로 표기하였다. 염색체는 유전자(gene, 6행의 g)로 구성되며, 각 유전자는 0에서 1사이의 실수값을 가진다. 제안하는 알고리즘에서 염색체는 생성된 트리의 개수만큼의 유전자를 가지게 되며, 각 유전자의 값은 해당 트리의 가중치를 의미한다. 각각의 유전자의 초기값은 랜덤하게 결정하였다. 6행의 결과, T개의 유전자를 가지는 염색체 N개($ch_1, \ldots, ch_N$)가 정의된다.

알고리즘 4.1의 7행~18행에서는 유전 알고리즘의 수행단계를 나타낸다. 유전알고리즘의 주요 함수인 Q함수는 염색체의 최적도를 평가하는 함수이며 표 5와 같이 정의된다.

제안하는 알고리즘의 Q함수는 검증 샘플을 이용한 분류율과 다양성 지수를 이용한다. 다양성 지수는 Roli 등[18]의 패러다임을 응용하였다. 분류율은 테스트 샘플에 대한 각 단위 트리의 결과를 결합하여 성능을 평가한다. 이때 단위 트리의 결과는 해당 유전자의 값에 의한 가중치 투표(weighted voting)으로 결정된다. 다양성 지수는 랜덤하게 선택된 두 개의 샘플에 대해, 두 샘플 모두 오분류 하는 트리의 개수와 한 개의 샘플에 대해 오분류 하는 트리의 개수의 비율로 계산된다. 총 S회 반복한 후, 그 평균을 다양성 지수로 산출하며, S는 실험적으로 결정하였다.

Table 4. The proposed algorithm

Algorithm 4.1

Function GDRF(ς)

01 : for t = 1 : T
02 : make bootstrap subset ς_t from ς
03 : $tree_t$ = TREE(ς_t)
04 : end for
05 : forest = $\{tree_1, ..., tree_T\}$
06 : random select $ch_i = [g_1, ..., g_T]^T$, $0 \le g \le 1$, $i = 1, ..., N$
07 : for n = 1 : G_{max}
08 : calculate $Q(ch_i), F(ch_i)$
09 : for i = 1 : K
10 : random select ch_{p1}, ch_{p2}
11 : $ch_i^\circ = mutation(crossover(ch_{p1}, ch_{p2}))$
12 : end for
13 : $p \leftarrow ch_1^\circ, ..., ch_k^\circ$ (replacement)
14 : if stop criteria
15 : break
16 : end if
17 : end for
18 : $ch_{max} = \arg\max_{ch} Q(ch_i)$, $i = 1, ..., N$
19 : $g_i = 0$, if $g_i \in ch_{max}$ and $g_i \le 1/(5T)$
20 : return ch_{max}

그림 2는 집합 구조를 이용한 다양성 지수의 개념을 나타낸다.

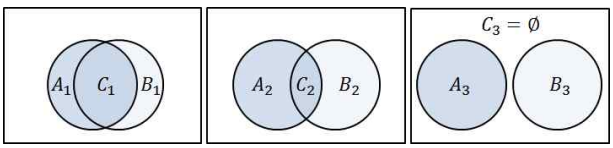

Fig. 2. The example of diversity concept

그림 2는 하나의 샘플을 오분류하는 트리의 집합을 A, 다른 하나의 샘플을 오분류하는 트리의 집합을 B로 설정한 예이다. 그림 2에서의 세 가지 경우 각각에 대한 다양성 지수는 식 (4)의 관계를 가진다. 두 샘플에 대해 동일하게 오분류하는 트리의 집합이 작을수록 다양성 지수는 1에 가깝게 된다. 따라서 Q함수는 주어진 샘플에 대한 분류율이 높으면서도 최대한 서로 다른 결과를 내는 트리들이 조합되어 있을수록 높은 결과를 출력한다.

$$d_1 = 1 - \frac{C_1}{A_1 + B_1 + 2C_1}$$
$$d_2 = 1 - \frac{C_2}{A_2 + B_2 + 2C_2} \quad (4)$$
$$d_3 = 1 - \frac{C_3}{A_3 + B_3 + 2C_3} = 1$$
$$d_1 \le d_2 \le d_3 = 1$$

Table 5. Q function

Algorithm 4.2

Function Q(ch)

00 : c $\leftarrow$ classification rate for validation samples
01 : d = 0
02 : for i = 1 : S
03 : random select v_1, v_2 in validation samples
04 : $p_1 \leftarrow$ count misclassified trees for both
05 : $p_2 \leftarrow$ count misclassified trees for v_1
06 : $p_3 \leftarrow$ count misclassified trees for v_2
07 : $d = d + \left(1 - \frac{p_1}{p_2 + p_3}\right)$
08 : end for
09 : d = d / S
10 : return (c + d) / 2

Table 6. F function

Algorithm 4.3

Function F(ch)

00 : $q_{min} = 1$, $q_{max} = 0$
01 : for i = 1 : N
02 : $q_i = Q(ch_i)$
03 : if $q_i > q_{max}$, $q_{max} = q_i$, end if
04 : if $q_i < q_{min}$, $q_{min} = q_i$, end if
05 : end for
06 : for i = 1 : N
07 : $q_i = (q_i - q_{max}) + (q_{max} - q_{min})/r$
08 : end for
09 : return q

표 6은 염색체의 적합도를 평가하는 F함수를 나타낸다. F함수는 각각의 염색체의 질의 값에 따라 상대적 비율값을 부여하여 염색체 선택의 기준이 된다. 제안하는 알고리즘에서는 Q함수값에 비례하여, 최대값이 최소값에 r배가 되도록 하였다. 제안하는 알고리즘에서는 선택압력(selection pressure)를 낮추기 위하여 r값을 1.5로 선택하였다.

알고리즘 4.1의 10행에서는 F함수를 통해 부여된 적합도 수치에 따라 두 개의 염색체를 랜덤 선택한다. 선택은 룰렛(roulette)방식을 적용하여, 높은 적합도의 염색체가 확률적으

로 더 잘 선택되도록 하였다. 알고리즘 4.1의 11행에서의 교차 연산은 표 7과 같이 균일 교차(uniform crossover)를 적용하였다. 두 염색체에 대해 랜덤 생성된 숫자에 따라 어느 한쪽의 유전자를 그대로 물려받는 기법이다. 두 염색체 ch_1과 ch_2가 입력으로 주어지며, 랜덤하게 생성된 rv변수의 값에 따라, rv가 0.5 이상이면 ch_1의 유전자를 물려받고, 반대의 경우 ch_2의 유전자를 물려받는다. 각각의 T개의 유전자에 대해 차례로 수행되어, 결과로 새로운 교차 염색체 ch를 생성한다.

Table 7. The crossover

Algorithm 4.4

Function CROSSOVER(ch_1, ch_2)

00 : for i = 1 : T
01 : rv ← random variable , $0 \le rv \le 1$
02 : if rv ≥ 0.5
03 : $g_i \leftarrow i_{th}$ gene in ch_1
04 : else
05 : $g_i \leftarrow i_{th}$ gene in ch_2
06 : end if
07 : end for
08 : ch = $\{g_1, \ldots, g_T\}$
09 : return ch

알고리즘 4.1의 11행에서의 변이 연산은 표 8과 같다. 생성된 교차 염색체에 임의적인 변형을 가하는 과정이다.

Table 8. The mutation

Algorithm 4.5

Function MUTATION(ch)

00 : pm = 0.25
01 : for i = 1 : T
02 : $rv_1 \leftarrow$ random variable , $0 \le rv_1 \le 1$
03 : if rv < pm
04 : $rv_2 \leftarrow$ random variable , $0 \le rv_2 \le 0.1$
05 : $g_i = g_i + rv_2$
06 : end if
07 : end for
08 : ch = $\{g_1, \ldots, g_T\}$
09 : return ch

두 개의 랜덤 실수가 생성되고 하나의 랜덤 실수가 미리 정해진 pm의 값보다 낮으면, 해당 유전자의 값에 두 번째 랜덤 실수를 더해주는 변이를 수행한다. 제안하는 알고리즘에서는 pm을 0.25로 설정하였으며, 유전자에 더해지는 두 번째 랜덤 실수는 0~0.1의 범위에서 결정되도록 하였다. 알고리즘 4.1의 9행~12행에서 이와 같은 선택, 교차, 변이를 통해 총 K개의 자식 염색체를 생성한다. 알고리즘 4.1의 13행에서는, 생성된 K개의 자식 염색체를 기존의 염색체 집단에 대치시킨다. 우선 자식 염색체를 생성하는데 사용된 부모 염색체들을 대상으로 적합도가 낮은 K개를 선택하여 제거한 후, 자식 염색체로 대치하는 기법을 적용하였다.

알고리즘 4.1의 14~16행은 유전알고리즘의 정지 조건으로써 제안하는 알고리즘에서는 별도의 조건을 부여하지 않았다. 18행에서는 최대 반복수까지의 진화가 완료된 염색체 중 최대의 성능을 보이는 염색체를 최종 트리 조합으로 결정한다(제안하는 알고리즘에서는 Q함수의 값, 즉 분류율과 다양성 지수의 평균을 성능으로 적용하였다). 19행은 유전자 소거 과정으로써 선택된 염색체의 각 유전자 값이 $1/(5T)$ 이하의 유전자는 0으로 바꾸며, 이는 현재의 트리 조합에 대한 성능의 결정에 영향이 매우 작은 트리를 제거하기 위한 단계이다. 예를 들어, 선택된 염색체의 유전자 중에서 0이 아닌 값을 가지는 유전자의 개수가 300개라고 가정한다면, 유전자의 값이 1/1500 이하의 값을 갖게 되므로 해당 트리의 가중치가 너무 낮아 전체 분류율의 결정에 영향을 미치기 어렵다고 판단하여 제거하였다.

V. Experimental Results

본 논문에서는 제안하는 기법의 성능을 측정하기 위하여, 공용 데이터베이스인 UCI 데이터베이스[21]와 자체적으로 확보한 적외선 영상 데이터베이스를 이용하였으며, 앞서 언급한 BRF(Beiman's random forest, 알고리즘 1) 및 DRF(Bernad's dynamic random forest, 알고리즘 2)와의 분류 성능을 비교하였다.

1. UCI 데이터베이스 실험

Table 9. The dataset for experiment

Datasets	Instances	Attributes	Classes
MGT	19020	11	2
ISO	7797	617	26
LETTER	20000	16	26
MUSK	6598	168	2
OPT	5620	64	10
PAGE	5473	10	5
PEN	10992	16	10
SEG	2310	19	7
SPAM	4601	57	2
STAT	946	18	4
WV	5000	40	3

UCI 데이터베이스는 머신 러닝 분야에서 가장 많이 사용되는 공용 데이터베이스이다. 총 325가지의 데이터 종류가 공유되어 있으며, 지속적으로 업데이트가 되고 있다. 성능 검증을 위하여 본 논문에서 사용한 샘플은 아래 표 9와 같다.

데이터베이스의 명칭이 길기 때문에 편의상 약칭을 사용하

였다. 약칭과 정식 명칭은 MGT(magic gamma Telescope), ISO(isolet), LETTER(letter recognition), MUSK(musk version 2), OPT(optical recognition of handwritten digits), PAGE(page blocks classification), PEN(pen-based recognition of handwritten digits), SEG(image segmentation), SPAM(spam base), STAT(statlog vehicle silhouettes), WV(waveform database generator version 2)이다.

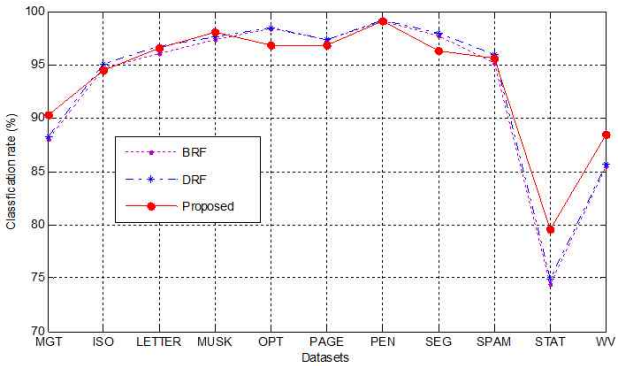

Fig. 3. The example of diversity concept

Table 10. The experimental results on UCI dataset

Datasets	BRF	DRF	Proposed	STD
MGT	88.01	88.26	**90.25**	1.23
ISO	94.65	**95.04**	94.44	0.30
LETTER	96.06	**96.69**	96.52	0.33
MUSK	97.38	97.59	**98.03**	0.33
OPT	98.35	**98.49**	96.82	0.93
PAGE	97.3	**97.36**	96.8	0.31
PEN	99.09	**99.13**	99.05	0.04
SEG	97.67	**97.98**	96.32	0.88
SPAM	95.12	**95.96**	95.56	0.42
STAT	74.4	74.93	**79.57**	2.84
WV	85.52	85.65	**88.41**	1.63

분류 성능 분석을 위하여, 각각의 데이터베이스에 대해 2/3를 학습 샘플로 임의 선정하고 나머지 1/3을 테스트 샘플로 선정하였다. 트리의 개수는 최대 500개로 설정하였으며, 각각의 랜덤포레스트 과정을 10회 반복하여 분류 성능을 평균하였다. 제안하는 알고리즘에서는, 초기 염색체를 1000개로 하였으며, 반복 단계에서 100개의 자식 염색체를 생성하여 대치하였다.

그림 3과 표 10은 UCI 데이터베이스에 대한 분류 실험 결과를 나타낸다. 단위는 분류율(%)이다. MGT를 비롯한 상위 9개의 데이터베이스에서는 DRF와 거의 유사한 분류 성능을 확인할 수 있다. BRF, DRF, 제안하는 알고리즘 모두 표준편차가 1% 이하의 매우 작은 분류 성능 분포를 보이므로 샘플 자체가 매우 분류가 잘 되는 양질의 샘플임을 알 수 있다. 그러나 STAT과 WV 데이터베이스에서는 상대적으로 분류성능의 차이가 발생한다.

표 11은 STAT 데이터베이스의 분류 성능 기준을 판단하기 위하여 기존의 다른 분류 연구에서의 분류 실험 결과와 비교하여 나타낸다. 기존의 연구에서도 74~81% 사이의 낮은 분류 결과를 도출하는 데이터베이스임을 확인할 수 있다. 여러 연구에서 STAT 데이터베이스의 분류 결과가 최대 80% 수준에서 평가되고 있으며, 제안하는 알고리즘은 이에 근접한 분류 결과를 나타냄을 확인하였다.

Table 11. The comparison of experimental results on STAT database

Proposed	Region Boost[23]	IE+IP[24]	Adaboost C4.5[25]
79.57	81	74	77.8

표 12는 UCI 데이터베이스에 대한 제안하는 알고리즘의 상세 결과를 나타낸다. 평균적으로 약 55% 수준의 트리 개수에서 최적화가 완료되었으며, MGT, PEN, SEG, STAT, WV 데이터베이스에 대해 최대 트리개수의 절반이하의 조합이 결정되었으며 높은 다양성 지수를 나타내고 있다. 특히, STAT와 WV 데이터베이스의 경우 모두 200개 내외의 최적화 트리 조합이 결정되었으며, 다양성 지수도 상대적으로 매우 높게 측정되었다. 따라서 BRF 및 DRF에서는, 해당 데이터베이스에 대해 과도한 개수의 트리를 생성하여 특정 샘플에 치우친 트리(즉, 전체 분류 성능에 악영향을 줄 수 있는 트리)들이 다수 포함되어 낮은 분류 성능을 나타내고 있다고 판단된다.

Table 12. The experimental result of proposed algorithm

Datasets	Classification rate	Trees (Mean)	Diversity (Mean)
MGT	90.25	215.1	**0.94**
ISO	94.44	343.2	0.68
LETTER	96.52	477.5	0.72
MUSK	98.03	254.3	0.85
OPT	96.82	322.4	0.65
PAGE	96.8	283.6	0.88
PEN	99.05	185.0	**0.91**
SEG	96.32	228.5	0.82
SPAM	95.56	291.3	0.84
STAT	88.57	193.3	**0.92**
WV	93.41	218.4	0.83

2. 적외선 영상 데이터베이스 실험

본 논문에서는 표적 분류 실험을 위하여 지상표적에 대한 적외선(infrared) 영상을 사용하였다. 전차, 장갑차, 군용트럭, 상용차량의 4종의 표적에 대하여 5차수에 걸쳐 획득한 영상이며, 표적 종류별 영상의 전체 개수는 표 13과 같다. 그림 4는 표적 종류에 따른 적외선 영상의 예를 나타낸다.

Table 13. The infrared target image database

획득차수	전차	장갑차	상용차량	군용트럭
1	418,170	384,940	230,180	377,128
2	107,945	131,915	126,975	131,330
3	237,070	154,815	80,965	197,785
4	230,490	242,620	169,790	97,710
5	280,350	259,305	204,160	255,545

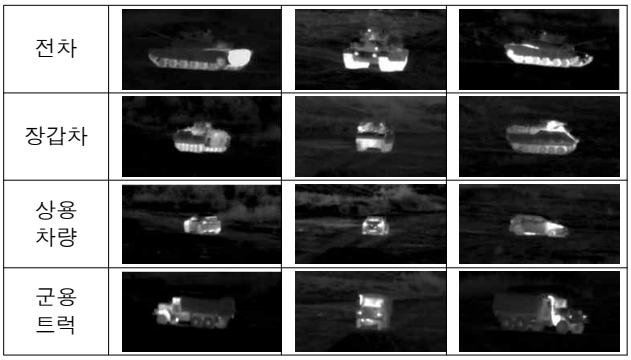

Fig. 4. The example of infrared target images

특징값으로 구성된 UCI 데이터베이스와는 달리 적외선 영상만을 획득했기 때문에, 본 논문에서는 위 영상들로부터 특징값을 추출하여 사용하였다. 특징값은 영상의 텍스처 정보를 기반으로 추출되는 HOG(histogram of gradient)[22]를 사용하였다. 실험적으로는 다수의 특징에 대해서도 수행했으나 본 논문에서는 특징에 대한 상세 내용은 별도로 기술하지 않는다. 표 13에서의 차수는 각각 획득 장소와 날짜 다름을 의미하며, 1, 2 차수가 동일 지역 다른 날짜의 영상이며, 3~5 차수가 동일 지역 다른 날짜의 영상이다. 또, 각 차수에 따라 영상 획득 당시의 고도와 날씨가 모두 다르다. 본 논문에서는 제안하는 알고리즘의 분류 성능을 평가하기 위하여 네 개의 차수의 영상을 학습 샘플로, 나머지 하나의 차수의 영상을 테스트 샘플로 하여 분류 성능을 평가하였다. 학습 샘플은 선택된 각각의 차수에 대해서 표적별로 1,000장씩을 랜덤선택 하였으며, 4개 차수 4개 표적에 대해 총 16,000장의 샘플 영상을 이용하였다. 이때 표적의 서로 다른 각도의 영상이 포함되도록 일정 구간을 균등하게 나눈 후, 구간 내 영상 중 한 개를 랜덤 선택 하였다. 트리의 개수는 최대 500개로 설정하였으며, 각각의 랜덤포레스트 과정을 10회 반복하여 분류 성능을 평균하였다. 제안하는 알고리즘에서는 초기 염색체를 1000개로 하였으며, 반복 단계에서 100개의 자식 염색체를 생성하여 대치하였다.

그림 5와 표 14는 적외선 표적 영상에 대한 분류 실험 결과를 나타낸다. 표 15는 제안하는 알고리즘의 상세 결과를 나타낸다. 전체적인 분류 성능은 기존의 랜덤포레스트 수준의 결과를 나타내며, 다소 분류 성능이 떨어지는 2차수 데이터베이스에서도 높은 다양성 지수와 함께 우수한 분류성능을 확인할 수 있다. 또, 기존의 랜덤포레스트들에 비해 약 60% 수준의 트리 개수에서 최적 조합으로 결정됨을 확인하였다. 이는 양질의 대용량 샘플에 대해서 많은 개수의 트리를 생성하더라도 상당부분 분류 성능에 영향이 미미하거나 악영향을 미치는 트리가 있음을 의미한다. 또한, 상대적으로 적은 수의 트리조합에서 높은 다양성을 가짐을 확인할 수 있으며, 앙상블 분류모델의 관점에서 트리 개수의 증가가 분류 성능과 일반화 성능에 역효과를 초래할 가능성이 있음을 의미한다. 따라서 분류 성능의 극대화와 다양성의 확보를 위한 고려가 함께 이루어져야 한다.

Table 14. The experimental results on infrared target images

Datasets	BRF	DRF	Proposed	STD
1	94.25	96.18	**96.43**	1.19
2	90.1	93.2	**96.35**	3.13
3	94.85	**96.8**	95.03	1.08
4	**93.22**	91.58	91.17	1.08
5	96.45	**97.46**	96.82	0.51

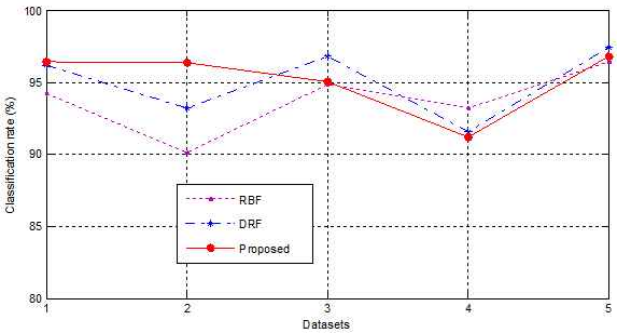

Fig. 5. The experimental results on infrared target images

Table 15. The experimental result of proposed algorithm

Datasets	Classification rate	Trees (Mean)	Diversity (Mean)
1	96.43	288.3	0.82
2	96.35	225.8	**0.89**
3	95.03	308.5	0.81
4	91.17	259.2	0.78
5	96.82	388.2	0.71

VI. Conclusions

본 논문에서는 과생성된 트리들로부터 분류 성능과 다양성을 평가하여 유전 알고리즘을 통해 최적의 트리 조합을 생성하는 유전적 동적 랜덤포레스트 알고리즘을 제안하였다. 공용 데이터베이스와 적외선 표적 영상 데이터베이스를 이용한 실험 결과, 설정된 최대 트리 개수 대비 55~60% 수준의 트리 개수에서 최적 조합이 결정되었으며, 특정 데이터베이스에 대해 약 3~5% 가량의 성능 향상을 확인할 수 있었다. 그러나 제안하는 알고리즘은 충분한 개수의 과생성된 트리가 필요하다. 이는 샘플의 양과 질에 따라 기하급수적으로 큰 수가 필요할 수 있다. 일정한 개수의 트리가 생성될 때 마다 최적화를 단계적으로 수

행하는 순차 탐색 개념과의 결합도 고려해볼 수 있다.

제안하는 알고리즘은 전통적인 오프라인 학습과 온라인 분류의 환경에 적합한 기법이다. 최근의 분류 시스템 분야에서 대두되고 있는 웹 기반 환경은, 대용량 데이터가 지속적인 입력되는 환경으로써, 온라인 학습과 온라인 분류를 지향하고 있다. 이와 같은 환경에 대해서는 제안하는 알고리즘은 적용하기 어려운 단점이 있다.

마지막으로, 제안하는 알고리즘의 구성이 되는 배깅과 유전 알고리즘은 확률적인 관점에서의 개념들이다. 따라서 다양한 대용량 데이터에 대한 실험과 분석을 통해 그 타당성을 입증해야만 한다.

REFERENCES

[1] D. E. Rumelhart and J. L. McClelland, "Parallel distributed processing: explorations in the microstructure of cognition," MIT Press, 1986.

[2] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Journal of Data Mining and Knowledge Discovery, Vol.2, pp.121-167, 1998.

[3] H. Y. Yeom, J. H. Kim, and Y. S. Moon, “Gene Classification Method using Neural Networks and Membership Function,” Journal of IEEK, Vol. 42CI, No. 4, pp.33-42, 2005.

[4] S. K. Kang, Y. U. Kim, I. M. So, and S. T. Jung, "Enhancement of the Correctness of Marker Detection and Marker Recognition based on Artificial Neural Networks," Journal of KSCI, Vol. 13, No. 1, pp. 89-97, Jan. 2008.

[5] Kwang Seong Kim and Doosung Hwang, "Support Vector Machine Algorithm for Imbalanced Data Learning," Journal of KSCI, Vol. 15, No. 7, pp. 11-17, July 2010.

[6] J. H. Kim, T. W. Cho, S. W. Chun, J. M. Lee, and Y. S. Moon, “Gunnery Classification Method Using Profile Feature Extraction in Infrared Images,” Journal of KSCI, Vol. 19, No. 10, October 2014

[7] Robi Polikar, "Ensemble based systems in decision making," IEEE Circuit and Systems, Vol.6, No.3, pp.21-45, 2006.

[8] P. Viola and M. Jones, "Robust real-time face detection," Int. Journal of Computer Vision, Vol.57, No.2, pp.137-154, 2004.

[9] L. Breiman, "Random forest," Machine Learning, Vol.45, pp.5-32, 2001.

[10] J. H. Kim, K. H. Jang, J. H. Lee, and Y. S. Moon, "Multi-target Classification Method Based on Adaboost and Radial Basis Function," Journal of IEEK, Vol. 47 CI, No. 3, pp. 22-28, May 2010.

[11] K. Jung, J. Choi, and K. Jang, "Facial express recognition using registration and Adaboost," Journal of IEEK, Vol. 51, No. 11, pp.193-201, 2014.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone," Chapman and Hall, 1993.

[13] R. Banfield, L. Hall, K. Bowyer, D. Bhadoria, W. P. Kegelmeyer, and S. Eschrich, "A comparison of ensemble creation technique," Proc. of Multiple Classifier Systems, Vol.1, pp.223-232, 2004.

[14] S. Bernard, L. Heutte, and S. Adam, "Influence of hyperparameters on random forest accuracy," Proc. of Workshop on Multiple Classifier Systems, Vol.1, pp.171-180, 2009.

[15] S. Bernard, L. Heutte, and S. Adam, "On the selection of decision trees in random forest," Proc. of Joint Conf. on Neural Networks, pp.302-307, 2009.

[16] E. Tripoli, D. Fotiadis, and G. Manis, "Dynamic construction of random forests: Evaluation using biomedical engineering problems," Proc. of IEEE Int. Conference on Information Technology and Application in Biomedicine, Vol.1, pp.3-5, 2010.

[17] S. Bernard, S. Adam, and L. Heutte, "Dynamic random forests," Journal of Pattern Recognition Letters, Vol.33, No.12, pp.1580-1586, 2012.

[18] F. Roli, G. Giacinto, and G. Vernazza, " Methods for designing multiple classifier systems," Proc. of 2nd International Workshop MCS2001, pp.78-87, 2001.

[19] I. S. Oh, J. S. Lee, and B. R. Moon, "Hybrid genetic algorithms for feature selection," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.26, No.11, pp.1424-1437, 2004.

[20] K. S. Hu and I. S. Oh, “Genetic Algorithm for Node Pruning of Neural Networks,” Journal of IEEK, Vol.46CI, No.2, pp.65-74, 2009.

[21] http://archive.ics.uci.edu/ml/

[22] C. M. Kim, Y. M. Baek, and H. Y. Kim, “An Efficient Pedestrian Recognition Method based on PCA Reconstruction and HOG Feature Descriptor,” Journal of IEIE, Vol.50, No.10, pp.162-170, 2013.

[23] R. Maclin, “Boosting Classifiers Regionally,” In Proc. of the 15th National Conference on Artificial Intelligence, Vol.1, pp.700-705, 1998.

[24] Venkatadri M. and Srinivasa R. K., “A multiobjective genetic algorithm for feature selection in data mining,” Journal of Computer Science and Information

Technology, Vol.1, No.5, pp.443-448, 2010.

[25] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," Journal of Machine Learning, Vol. 40, No.2, pp.139-157, 2010.

Authors

Jae Hyup Kim received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 2001, 2003 and 2008, respectively

From 2008 to 2009, Dr. Kim had been a post Doc. researcher at the Ambient Intelligence SW Research Institute in Hanyang University. In 2009, he joined the Hanwha-Thales Co., Republic of Korea, as senior researcher, and is currently a researcher. He is interested in machine learning, computer vision.

Hun Ki Kim received the B.S. and M.S. degrees in Electrical, Information & Control Engineering from Hongik University, Korea, in 2004 and 2007, respectively

In 2008, he joined Hanwha-Thales Co., Republic of Korea, and he is currently a researcher. He is interested in Surveillance, Tracking and computer vision.

Kyung Hyun Jang received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 2005, 2007 and 2014, respectively

In 2014, he joined the Hanwha-Thales Co., Republic of Korea, and is currently a senior researcher. He is interested in computer vision and pattern recognition.

Jong Min Lee received his B.S. and M.S. degrees in computer science and engineering from Hanyang University, Republic of Korea, in 2007 and 2009, respectively.

Mr. Lee is currently working toward his Ph.D. degree at the computer vision and pattern recognition laboratory, Department of Computer Science and Engineering, Hanyang University, Republic of Korea. His major interests include computer vision, image enhancement, and object detection.

Young Shik Moon Hong received the BS and MS in electronics engineering from Seoul National University and Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 1980 and 1982, respectively, and Ph.D. degree in electrical and computer engineering from the University of California at Irvine, CA, in 1990.

From 1982 to 1985, Dr. Moon had been a researcher at the Electronics and Telecommunications Research Institute, Daejon, Republic of Korea. In 1992, he joined the department of Computer Science and Engineering at Hanyang University, Republic of Korea, as an Assistant Professor, and is currently a Professor. His research interests include computer vision, image processing, pattern recognition, and computational photography.