

## Learning Algorithm for Multiple Distribution Data using Haar-like Feature and Decision Tree

Ju-Hyun Kwak<sup>†</sup> · Il-Young Woen<sup>\*\*</sup> · Chang-Hoon Lee<sup>\*\*\*</sup>

### ABSTRACT

Adaboost is widely used for Haar-like feature boosting algorithm in Face Detection. It shows very effective performance on single distribution model. But when detecting front and side face images at same time, Adaboost shows its limitation on multiple distribution data because it uses linear combination of basic classifier. This paper suggests the HDCT, modified decision tree algorithm for Haar-like features. We still tested the performance of HDCT compared with Adaboost on multiple distributed image recognition.

**Keywords :** Adaboost, Haar-like, Decision Tree, Pattern Recognition

## 다중 분포 학습 모델을 위한 Haar-like Feature와 Decision Tree를 이용한 학습 알고리즘

곽주현<sup>†</sup> · 원일웅<sup>\*\*</sup> · 이창훈<sup>\*\*\*</sup>

### 요 약

Adaboost 알고리즘은 얼굴인식을 위한 Haar-like feature들을 이용하기 위해 가장 널리 쓰이고 있는 알고리즘이다. 매우 빠르며 효율적인 성능을 보이고 있으며 하나의 모델이미지가 존재하는 단일분포 데이터에 대해 매우 효율적이다. 그러나 정면 얼굴과 측면 얼굴을 혼합한 인식 등 둘 이상의 모델이미지를 가진 다중 분포모델에 대해서는 그 성능이 저하된다. 이는 단일 학습 알고리즘의 선형결합에 의존하기 때문에 생기는 현상이며 그 응용범위의 한계를 지니게 된다. 본 연구에서는 이를 해결하기 위한 제안으로서 Decision Tree를 Haar-like Feature와 결합하는 기법을 제안한다. Decision Tree를 사용함으로써 보다 넓은 분야의 문제를 해결하기 위해 기존의 Decision Tree를 Haar-like Feature에 적합하도록 개선한 HDCT라고 하는 Haar-like Feature를 활용한 Decision Tree를 제안하였으며 이것의 성능을 Adaboost와 비교 평가하였다.

**키워드 :** 아다부스트, 하르-라이크, 결정트리, 패턴인식

### 1. 서 론

영상인식에서 얼굴 인식을 수행할 때 가장 널리 사용되는 방법은 Haar-like과 Adaboost를 결합한 방법이다. Haar-like 단순한 사각형 기반의 비교 이미지 다수를 결합하여 패턴을 만드는 기법이며 이런 각각의 단순한 비교이미지를 결합하기 위해 Adaboost 라는 기법을 사용한다[2,3,4]. 이 두 기법의 결합은 사진이나 영상 등에서 사람의 얼굴을 찾는 데 있어서 빠른 속도 및 정확도로 인해 널리 사용되고 있다.

이는 단순하지만 고속의 연산이 가능한 Haar-like를 단순

한 분류기(classifier)를 여러 개 동시에 사용함으로써 더욱 성능을 높일 수 있는 Adaboost와 결합함으로써 매우 효율적인 성능을 보인다. 또한 Adaboost는 결국 분류기들의 가중치 결합으로 연결한다는 점에서 SVM과 유사한 성격을 지닌다[1].

SVM은 상태공간에서의 패턴분류를 위해 각 입력 값의 가중치를 조정하여 특정 패턴에 수렴 학습시키는 방법으로써 군집적 패턴에 대해서는 매우 높은 효율을 지니지만 단층 신경망이 가지는 XOR 문제와 같은 다중 분포 등에 취약하다는 단점을 가지고 있다[1]. 이는 Adaboost 도 마찬가지로 사람의 얼굴처럼 특정 모델을 중심으로 군집된 정규분포형 데이터에 대해서는 높은 수렴률을 지니지만 둘 이상의 이산분포형태 이미지패턴에 대해서는 한계성을 보인다.

이는 초창기 인공지능에서의 단층신경망 모델이 가졌던 한계성과 유사하며 이를 해결하기 위한 방법으로 다층 신경망 모델이나 결정 트리 등의 다른 형태 학습방법이 제시되

<sup>†</sup> 준 회 원 : 건국대학교 컴퓨터공학과 공학석사

<sup>\*\*</sup> 정 회 원 : 호서서울전문대학교 사이버해킹보안과 교수

<sup>\*\*\*</sup> 중신회원 : 건국대학교 컴퓨터공학과 교수

논문접수 : 2012년 8월 1일

수정일 : 1차 2012년 9월 6일, 2차 2012년 9월 11일

심사완료 : 2012년 10월 9일

\* Corresponding Author : Il-Young Woen(decoz91@gmail.com)

어왔으며 이들은 복잡한 구조의 상태공간에 대해서는 단층 가중치 결합보다 더 높은 성능을 보인다.

따라서 본 논문에서는 Haar-like를 결합하기 위해 Adaboost가 적용되기 어려운 문제들에 대해 Adaboost기법의 한계점을 해결하기 위해 결정트리(Decision Tree)를 사용한 방법으로 문제를 해결할 수 있음을 제시한다.

## 2. 관련 연구

### 2.1 Haar-like feature 와 Adaboost

이미지의 인식, 학습, 분류(classification) 등에 사용되는 Haar-like feature는 사각형의 집합으로 구성된 특징(feature) 이미지의 집합으로써 이미지영역 누적테이블을 이용해 매우 빠른 시간에 feature의 적합도를 구할 수 있다는 점에서 널리 사용되고 있다[2].

그러나 기초적인 간단한 사각형들로만 구성된 feature 만으로는 복잡한 이미지를 인식하기 어렵기 때문에 단순한 Haar-like feature 의 집합을 각각의 가중치와 함께 합산 함으로써 복잡한 이미지에 대한 인식을 구성한다.

다양한 이미지에 대하여 적합한 Haar-like feature 집합과 가중치를 얻기 위해서는 학습알고리즘이 요구된다. 이때 가장 널리 사용되는 방법 중 하나가 Adaboost이다.

Adaboost는 실제로 여러 분류기나 학습기를 조합하여 하나의 메타학습기를 만드는 기법으로 하나의 학습방법에 의해 발생하는 약점을 커버할 수 있다는 장점이 있으나 여러 학습 및 분류기를 사용 함으로써 시간적인 오버헤드가 크다. 그러므로 Adaboost는 단순하지만 빠른 소위 약한 분류기(weak classifier)를 조합하여 하나의 분류기(final classifier)를 생성할 때 주로 사용된다.

Haar-like feature는 이런 Adaboost의 성격에 매우 적합하며 각각의 feature 를 일종의 classifier로서 간주하고 이들의 가중치를 조정해 하나의 최종 분류기(final classifier)를 생성한다.

### 2.2 Adaboost 의 단점

약 분류기에 의해 판단된 결과값을 입력으로 간주할 경우 최종 분류기는 각각의 입력에 가중치를 곱한 값들의 합산으로 생각할 수 있으며 이는 단층 신경망이나 SVM (Supported Vector Machine)과 같은 성격을 지니게 된다. 이러한 방법은 비슷한 데이터이거나 정규분포로 구성된 그룹들의 분류에는 효율적이거나 둘 이상의 종류로 나뉘어지는 그룹에 대해서는 그 구조상 표현이 어렵다.

Fig. 1과 같이 만일 비슷한 유형으로 구성된 데이터라면 비교적 유사한 feature들이 높은 가중치를 얻음으로써 자연스럽게 학습이 수행되지만 입력데이터가 둘 이상의 유형으로 분류될 경우 상반된 feature간의 충돌로 인하여 적합한 학습효과를 얻기 어렵다. 이는 Fig. 2와 같이 단층 신경망의 한계인 XOR문제와 유사하다.

우측으로 치우치는 요소와 좌측으로 치우치는 두가지 성격의 1자에 대해 Adaboost는 모호한 값으로 수렴된다.

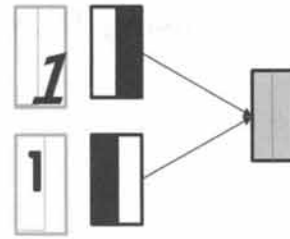
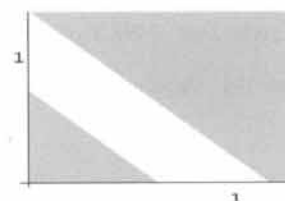
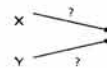


Fig. 1. Multi-case problem of Adaboost



0의 영역이 두부분으로 갈리는 구조



표준 군집이 둘이상으로 나뉘는 경우 가중치의 군집으로 표현이 어렵다.

Fig. 2. State space of XOR problem

신경망은 이 문제를 해결하기 위해서 다층 신경망 기법을 도입하였으며 그 외에도 이런 상태공간을 해결하기 위해 효율적인 알고리즘으로서 Decision Tree 나 IBL 등의 기계학습 기법 등이 널리 사용된다.

### 2.3 Decision Tree

결정 트리(Decision Tree)는 classification분야에서 널리 사용되는 방식으로 Fig. 3과 같이 각각의 feature들을 결정 트리 형태로 구성하여 입력 값에 대한 판단을 수행하는 방식이다. 학습 과정 중에 <수식 1>의 엔트로피 공식을 이용하여 [6] 더 효율적인 분류를 가능케 하는 feature 값을 상위 노드에 배치함으로써 작은 트리 구조로도 효율적으로 입력 값을 분류하는 분류기를 생성할 수 있다[5].

이는 위에서 언급한 XOR 문제에 대해서도 분류가 가능하며 복잡한 상태공간에 대한 적응성이 뛰어나다[10].

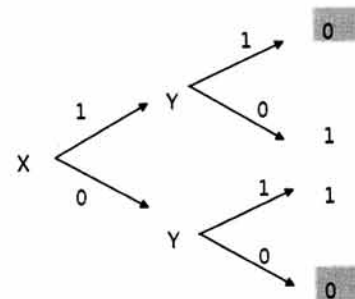


Fig. 3. Decision Tree for XOR problem

$$p_c = \frac{N_c}{N}$$

$$E_C = \sum_{c \in C} -p_c \log p_c \quad \text{<수식 1> 엔트로피 함수}$$

그러나 기존의 Decision Tree는 불연속 값을 갖는 feature 를 대상으로 개발되었으며 이를 보완하기 위해 C4.5 에선 linear value 를 대상으로 알고리즘이 확장되었다[7].

### 3. Haar-like feature Decision Tree (HDCT) 알고리즘

우리는 Haar-like feature에 Adaboost 대신 결정 트리 (Decision Tree)를 기반으로 한 최종 분류기 생성 방법을 제시함으로써 하나 이상의 유형을 가지는 이미지에 대한 학습이 어려운 Adaboost의 단점을 해결하기 위한 방법을 제안한다.

#### 3.1 제안 알고리즘

Haar Decision Tree 학습의 기본절차는 Decision Tree 와 동일하다. 다음과 같이 Fig. 4의 과정에서 볼 수 있다.

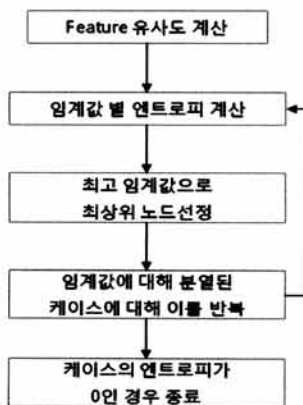


Fig. 4. Process diagram of HDCT

이 과정에서 일반적인 Decision Tree와 마찬가지로 엔트로피가 낮은 feature를 검색 후에 그것을 최상위 노드로 구분하는 것은 동일하다. 하지만 feature와 value로 구성되는 Tree의 하위노드와 달리 Haar-like feature는 연속적인 유사도값을 지닌다. 따라서 HDCT는 유사도를 구분하는 임계값을 사용한다. 또한 일반적인 Decision Tree와 달리 HDCT는 항상 두 개의 노드만 자식 노드로 지니는 Binary Tree로 구성한다.

이 경우 이미 특정 feature의 임계값에 의해 분할된 케이스라도 다시 feature의 다른 임계값에 의해 더 효율적으로 case 를 분할 할 경우도 생각해 볼 수 있기 때문이다. 그러므로 HDCT에서는 일반적인 Decision Tree와 달리 feature 노드의 중복이 발생한다.

```

Haars = loadFeatures()
function HDCT(images)
  for Haar in Haars by h
    for img in images by i
      fitness[][] = getFitness(Haar,img)
    end for
    sort(fitness[])
    bestHaar,bestFit = getMaxThresh(fitness,images)

    if bestFit > maxFit
      then maxHaar,maxFit = bestHaar,bestFit
    end if
    node r
    r.Haar = bestHaar
    r.thresh = bestFit
    if r.thresh > 0.1 then
      images1 = getUpperByHaar(images,bestHaar)
      images2 = getLowerByHaar(images,bestHaar)
      r.addChild( HDCT(images1) )
      r.addChild( HDCT(images2) )
    end if
  end for
end function
    
```

Fig. 5. HDCT Pseudo Code

#### 3.2 임계값 별 엔트로피 계산

N개의 feature f에 대해 M개의 이미지 각각 feature별로 N-1의 임계값을 생성한다. 이는 각 feature 별 이미지의 유사도의 크기로 나눈 경우 n번째 적합도를  $Sf_n$ 이라고 할 경우 1부터 N-1번째까지의 임계값  $B_n$ 은 다음과 같다[7].

$$B_n = \frac{Sf_n + Sf_{n+1}}{2} \quad \text{<수식 2> 임계값 계산}$$

<수식 2> 에 의해 연속적 값을 가진 feature라도 결국 측정된 값들의 중점을 임계 값으로 사용할 경우 유한한 임계 값을 가지게 된다[9].

임계 값의 적합도는 임계 값을 기준으로 나뉘어진 두 그룹의 이미지  $C_s$ 와  $S_l$ 에 대해서 각각의 엔트로피를 계산한다 [8,10].

$$E_n = E_{C_s} + E_{C_l} \quad \text{<수식 3> 임계값 별 적합도}$$

계산된 값의 적합도중에 가장 높은 적합도를 지닌 feature와 임계값이 상위 노드에 선택되며 해당 임계 값은 다음 후보 임계 값에서 제외된다.

### 4. 실험

실험은 현재 공개되어있는 opencv 2.1 상의 haar-feature 와 Adaboost 알고리즘을 이용하였으며 eclipse CDT 상에서 c++ 프로그래밍하였다.

4.1 문자인식

실험은 크게 두 가지 방향에서 수행했다. 첫 번째는 Adaboost 가 처리하기 힘들게 편향된 숫자 이미지들을 선정하였다. 이러한 이미지들은 숫자의 위치 및 폰트에 따라 하나 이상의 모델이미지를 생성하는게 더욱 유리하며 이를 통해 HDCT가 그것을 해결할 수 있음을 보여주하고자 했다. 두 번째는 Adaboost 가 특히 잘 적용되는 얼굴인식분야에 대해 성능을 비교 평가했다.

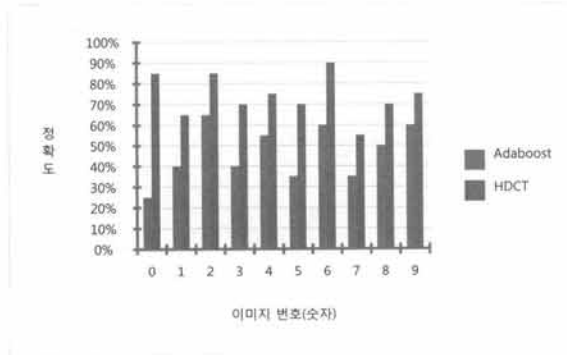


Fig. 6. Example of number images

이미지 사이즈는 평균 100x100 정방형 형태로 추출하였으며 0-9까지의 데이터에 대해서 각각 20개씩 총 200개의 이미지를 선정했다.

OpenCV의 Adaboost 라이브러리를 이용한 학습데이터를 이용하여 0-9까지 각각 학습기를 생성했다. 반면 HDCT는 그 트리의 노드가 각각 숫자에 대응되므로 하나의 트리로 생성됨으로써 단일 학습기로 구성하여 그 정확성을 측정했다.

Table 1. Comparison number image detection

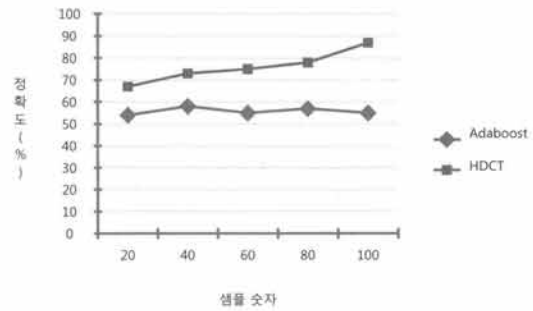


Adaboost와의 성능에 대한 비교 우위를 강조하기 위해 어느 정도 편향된 폰트나 이미지를 수집하여 실험하였기에 Adaboost의 성능은 그다지 높지 않은 결과를 보여준다. 1과 7의 정확도가 특히 낮은 이유는 많은 경우 이 둘이 서로 비슷하기 때문이다.

다음은 비교적 성능이 낮게 나온 학습샘플의 1과 7을 중심으로 학습 샘플의 양 늘려가면서 정확도를 측정하였다. 20개부터 100개까지 순차적으로 1의 이미지와 7의 이미지를 늘려가며 이중 랜덤하게 20개를 뽑아 Adaboost 와 HDCT 의 정확도를 비교했다.

Adaboost의 경우 학습 샘플이 매우 중요하며 위와 같이 특정 모델 이미지가 만들어지기 어려운 분산 샘플의 경우 샘플이 늘어나도 정확도의 향상이 거의 이루어지지 않았다. HDCT의 경우 이런 편향에 대해 트리 형태로 나뉘어

Table 2. Accuracy comparison by learning



학습함으로써 학습 샘플에 대한 정확도는 지속적으로 향상되었다.

특히 각각의 숫자별로 분류기를 구성, 학습시켜야 하는 Adaboost 에 비해 단일 학습기로 구성이 가능한 HDCT가 더 나은 성능을 보인다는 점은 이런 문제에 대해서 여러 면에서 유리하다고 볼 수 있다.

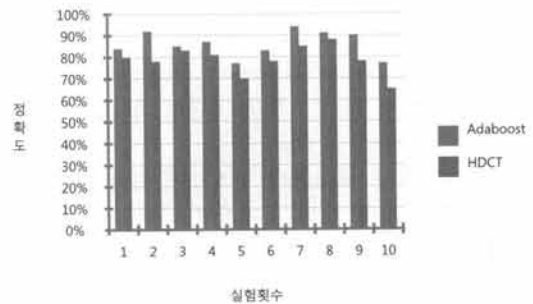
4.2 얼굴인식

얼굴 인식은 Adaboost가 가장 높은 성능을 나타내는 분야이다. 특히 OpenCV라는 오픈 라이브러리에 이 기능이 기본으로 제공되어 얼굴인식기능을 대중화시키는데 기여하였다. 이를 위해 널리 알려진 OpenCV 상의 Adaboost 에 의한 얼굴인식과 본 논문에서 제시하는 HDCT 상의 성능을 비교했다.

학습 얼굴 이미지는 <http://vis-www.cs.umass.edu/lfw/> 의 얼굴 샘플을 사용하였으며, testset은 <http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html> 의 이미지를 사용했다. 학습에 사용된 이미지는 500 개 정도로 제한하였으며 이때 50개의 테스트 이미지에서 정확히 얼굴의 위치를 인식하는지를 놓고 학습샘플을 랜덤하게 바뀌가며 10회를 테스트 후 정확도를 비교했다.

Adaboost 의 경우 평균 86% 이며 HDCT 는 평균 78% 의 정확도를 보여 Adaboost의 정확도에 비해 약 91% 가량의 정확도를 보여주었다. HDCT의 경우 트리를 구성하여

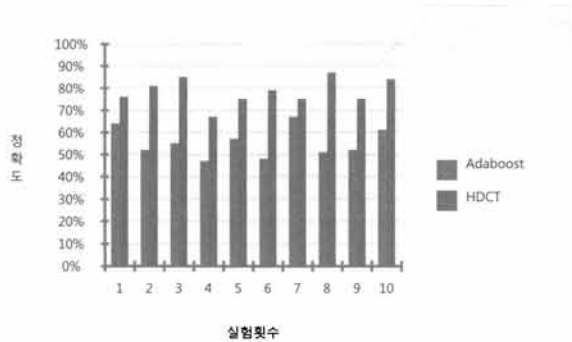
Table 3. Front face detection comparison



세세한 이미지까지 구분해 내기 때문에 노이즈에 대한 성능 저하가 나타났다.

다음은 50개의 사람의 옆면 얼굴까지를 포함한 이미지에 대해 분석을 수행한 결과이다.

Table 4. Mixed front and side image detection comparison



이 경우 Adaboost의 평균 정확도는 55%로 크게 떨어졌으며 HDCT는 78%로 정면 얼굴만의 결과에 비해 크게 떨어지지 않았다. 결과적으로 실험 결과 두 개 이상의 표준 모델을 가진 데이터에 대한 Adaboost의 성능 저하를 볼 수 있었다. 반면 HDCT는 정확도의 저하가 없었다.

결과적으로 단일 모델에 대해서는 Adaboost가 더 세밀하면서도 빠른 성능을 보이고 있지만 이 모델영역이 복잡해질수록 Adaboost는 feature 간의 선형결합에 의한 한계성을 지니고 있음을 볼 수 있었다. 그에 비해 HDCT는 그런 문제에 대해서는 Adaboost에 비해 성능 저하가 적게 나타났으며 이는 데이터 샘플이 늘어날수록 더욱 그 차이가 벌어졌다. 이를 통해 HDCT가 범용성에서 더 높은 가능성을 지니고 있음을 알 수 있었다.

### 5. 결론

Haar-like Feature는 그 단순함과 그에 따르는 속도적 이점에 의해 앞으로 더 많은 연구가 필요한 분야이다. Adaboost의 장점은 그 속도와 정규분포를 따르는 학습 데이터에 대해 대단히 빠른 수렴속도를 지니고 있다는 점이다. 그 결과 정면얼굴 인식이나 표준모델이 있는 이미지에 대해서는 높은 성능을 지니지만 현실세계의 다양한 물체인 식에는 그 한계를 보인다.

이와 달리 HDCT의 경우 Adaboost에 준하는 성능을 보여줌과 동시에 다중 분포 문제도 해결할 수 있음을 제시하였다. 이는 Adaboost보다 범용적인 데이터에 적용될 수 있음을 의미한다.

또한 본 연구에선 아직 수행되지 않았지만 이것이 트리형태의구조를 가진다는 특수성을 이용하여 얼굴영역검색과 같은 문제에 적용 시 Feature를 다 검색하지 않고도 해당 영

역에 대한 판단이 어느 정도 적정 영역에 대한 판단을 멈출 수 있기 때문에 속도적인 이점도 기대해 볼 수 있다.

### 참고 문헌

- [1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers, 1997.
- [2] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. CVPR, pp.511 - 518, 2001.
- [3] Carbonell, J.G., Michalski, R.S., & Mitchell, T.M. (1983). An overview of machine learning, In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), Machine learning: An artificial intelligence approach. Palo Alto: Tioga Publishing Company.
- [4] Pearl, J. (1978a). Entropy, information and rational decisions (Technical report). Cognitive Systems Laboratory, University of California, Los Angeles.
- [5] Quinlan, J.R. (1985b). Decision trees and multi-valued attributes. In J.E. Hayes & D. Michie (Eds.), Machine intelligence 11. Oxford University Press(in press).
- [6] T Kölsch, Diplomarbeit Im Fach Informatik, Lehrstuhl Fur Informatik Vi, Rheinisch-westfälische Technische Hochschule Aachen, Prof Dr. -ing, Prof Dr. -ing, H. Ney, H. Ney, Prof Dr, E. Vidal, Dipl Inform, D. Keysers, Tobias Gabriel, Tobias Gabriel, Benedikt Kolsch, Benedikt Kolsch, Local Features for Image Classification. 2003.
- [7] Quinlan, J.R. (1993). Program for machine learning, Morgan Kaufmann Publisher, Inc.
- [8] Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In D. Michie(Ed.), Expert systems in the micro electronic age. Edinburgh University Press.
- [9] J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.
- [10] Quinlan, J.R. (1983a). Learning efficient classification procedures and their application to chess endgames. In R.S. Michalski, J.G. Carbonell & T.M. Mitchell, (Eds.), Machine learning: An artificialintelligence approach. Palo Alto: Tioga Publishing Company.



### 곽주현

e-mail : decoz91@gmail.com  
 1995년 건국대학교 전자계산학과(학사)  
 1997년 건국대학교 컴퓨터공학과  
 공학석사  
 관심분야: 인공지능, 소프트웨어 공학,  
 이미지 처리, 정보보안



### 원 일 용

e-mail : decoz91@gmail.com

1998년 경원대학교 전자계산학과(석사)

2000년 건국대학교 컴퓨터공학과  
(공학석사)

2007년 건국대학교 컴퓨터공학과  
(공학박사)

2008년~현 재 호서서울전문학교 사이버해킹보안과 교수

관심분야: 인공지능, 정보보안, 이미지, 이미지 처리



### 이 창 훈

e-mail : chlee@konkuk.ac.kr

1977년 연세대학교 수학과(학사)

1980년 한국과학기술원 전산학과(석사)

1993년 한국과학기술원 전산학과(박사)

1996년~2002년 건국대학교 정보통신원장

2001년~2002년 건국대학교 정보통신대학과  
학장

1980년~현 재 건국대학교 컴퓨터공학과 교수

관심분야: 인공지능, 운영체제, 정보보안