

퍼지 Adaboost를 이용한 객체 검출

Object Detection using Fuzzy Adaboost

김기상, 최형일
승실대학교 미디어학과

Kisang Kim(illusion1004@gmail.com), Hyung-Il Choi(hic@ssu.ac.kr)

요약

Adaboost 학습 알고리즘은 학습 단계마다 가장 좋은 특징을 선택하도록 하는 학습 알고리즘이다. 각 학습 단계에서는 최적의 특징을 선택하기 위해 특정 임계값과 그에 대한 최소 오차율을 가지는 특징을 선택하도록 되어 있다. 하지만, 임계값을 이용하는 방법은 최적의 오차율을 검출하는데 있어 효율적인 방법이 아니다. 본 논문에서는 최적의 오차율을 검출하기 위한 퍼지 Adaboost 기법을 제안한다. 퍼지를 통해 결정 경계를 유연하게 한 Adaboost는 학습 단계가 적어도 좋은 성능을 보이는 장점이 있다. 기존의 Adaboost는 학습 전에 학습데이터에 대한 가중치를 동일하게 할당한다. 하지만, 본 논문에서는 이에 대한 가중치를 확률을 이용하여 초기 가중치를 다르게 줌으로서, 적은 학습에도 좋은 결과를 보이는 방법을 제안한다. 실험 결과에서는 기존의 Adaboost와 제안하는 방법에 대한 성능 평가를 통해, 퍼지 Adaboost가 기존 방법에 비해 좋은 결과를 보였다.

■ **중심어** : | Adaboost | 퍼지 추론 | 데이터 분포 분석 | 객체 검출 |

Abstract

The Adaboost chooses a good set of features in rounds. On each round, it chooses the optimal feature and its threshold value by minimizing the weighted error of classification. The involved process of classification performs a hard decision. In this paper, we expand the process of classification to a soft fuzzy decision. We believe this expansion could allow some flexibility to the Adaboost algorithm as well as a good performance especially when the size of a training data set is not large enough. The typical Adaboost algorithm assigns a same weight to each training datum on the first round of a training process. We propose a new algorithm to assign different initial weights based on some statistical properties of involved features. In experimental results, we assess that the proposed method shows higher performance than the traditional one.

■ **keyword** : | Adaboost | Fuzzy Inference | Data Distribution | Object Detection |

1. 서론

입력된 영상에서 주변 환경에 상관없이 원하는 객체

를 검출해 내는 것은 매우 중요한 문제이다[1]. 특히, 실시간으로 시스템이 사용자를 검출 및 인식하여 다양한 서비스를 사용자에 맞춰 제공하는 인터랙션 기술은 현

* 본 연구는 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입니다.

접수일자 : 2016년 03월 18일

수정일자 : 2016년 04월 12일

심사완료일 : 2016년 04월 21일

교신저자 : 최형일, e-mail : hic@ssu.ac.kr

제 많은 연구가 진행되고 있을 뿐만 아니라 다양한 분야에 활용되고 있다[2]. 다양한 분야에 활용되기 위한 객체 검출 알고리즘은 실시간 시스템을 위한 검출시간과 신뢰성을 높이기 위한 정확성은 매우 중요한 요소이다. 객체 검출 및 분류 방법은 여러 가지가 있다. 특히, 기존의 알고리즘들 중 SVM(Support Vector Machine) [3] 과 Neural Network[4] 방법이 자주 사용될 뿐만 아니라 다양한 다른 분류기와 융합되어 연구가 진행되고 있다. 최근에는 연관정(soft decision) 및 분류기를 선택할 때 객체 데이터를 모두 포함하는 방법 등[7] 기존의 Viola와 Jones가 제안한 Adaboost[5]를 확장하여 보다 정확한 분류 및 검출하는 연구가 시도되고 있다.

Adaboost 알고리즘은 학습데이터를 효율적으로 분류할 수 있는 유용한 특징들을 가중치를 변화시키면서 순차적으로 찾는 방법이다. 초기 데이터의 경우 각 데이터에 대한 정보가 없기 때문에 가중치를 동일하게 할당한다. 그러나, 다음 결정 특징을 선택하는데 있어 잘못 분류된 데이터는 높은 가중치가 할당되며, 잘 분류된 데이터는 낮은 가중치가 할당하여 잘못 분류된 데이터를 잘 분류할 수 있게 한다. 가장 좋은 특징은 특징들 가운데 오차율 값이 가장 낮은 특징으로 선택하며, 각 특징에 대한 오차율을 계산하기 위해 각 특징마다 최소 오류값을 갖는 임계값을 설정한다. 하지만 임계값으로 설정하는 것은 임계값 주변에서 잘못된 결정을 내리는 오류가 빈번히 일어나는 문제가 있다. 이러한 문제를 해결하기 위해 본 논문에서는 퍼지를 이용한 최소 오차율을 계산하는 방법과 학습데이터 중 객체 데이터에 대해 데이터 분포를 이용하여 초기 가중치를 설정함으로써, 기존 방법에 비해 분류가 더 정확한 방법을 제안한다.

[그림 1]은 본 논문에서 제안하는 방법의 전체 흐름을 나타낸다. 학습 데이터들 중 객체 데이터의 초기 가중치를 설정하기 위해 각 특징들의 평균값과 이 평균값들을 이용하여 각 데이터의 분포를 이용한다. 최적의 특징과 분류기를 선택하기 오분류를 산출하며, 산출 방법으로는 연관정을 이용한다.

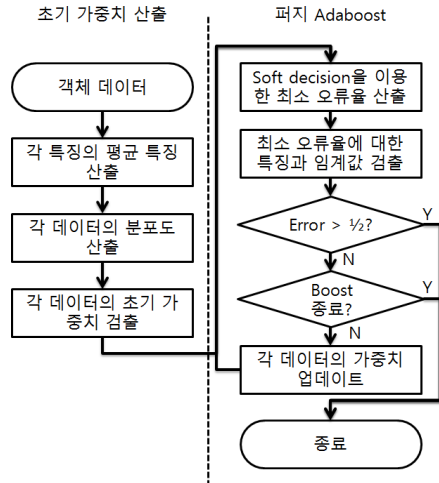


그림 1. 시스템 전체 흐름도

II. 본론

1. Adaboost와 Haar-like 특징

Haar-like 특징은 Adaboost 뿐만 아니라 다양한 객체 검출, 분류 및 인식 등 다분야에 많이 사용되는 특징이다. Haar-like 특징은 사용자가 설정한 특정크기의 인접한 여러 개의 윈도우를 생성한다. 각 윈도우 안에 있는 픽셀값들의 합으로 구성되며, 이 윈도우간의 차또는 합을 통해 특징을 추출하는 방법을 말한다. 이 특징은 특히 사람 얼굴을 검출하는데 많이 사용된다. 대부분의 사람 얼굴은 눈 영역이 있으며 이 영역은 대비가 뚜렷하여 구분쉬운 특징이 있기 때문이다. 그러므로 얼굴 검출을 위한 Haar-like 특징들 중 얼굴 검출에 유용한 특징으로는 한 영역은 눈 영역을 나타내며, 다른 한 영역은 볼 영역을 나타내는 좋은 특징을 구성할 수 있다. 검출이나 인식 단계에서는 검색 윈도우의 크기와 위치를 변화하여 입력 이미지로부터 찾게 되며, 검색윈도우의 값은 미리 학습된 임계값과 비교하여 객체 인지 배경인지 구분하게 된다.

[그림 2]에서는 눈 주변에서 Haar-like특징 검출에 대한 예제 그림이다. 같은 위치더라도 Haar-like 윈도우에 따라 특징 값이 다르게 검출되는 것을 [그림 2]에서 확인 할 수 있다. 이러한 특징을 이용하여 Adaboost 학

습 단계에서는 효율적인 특징, 윈도우의 형태, 그리고 임계값에 대해 추출하게 된다.

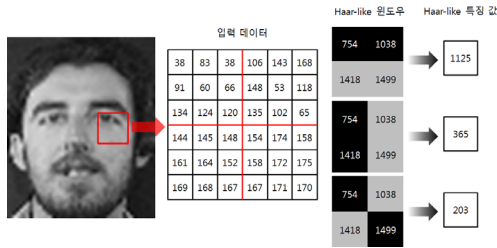


그림 2. Haar-like 특징에 대한 산출 예제. Haar-like 윈도우에서 같은 색은 합, 다른 색은 차로 계산

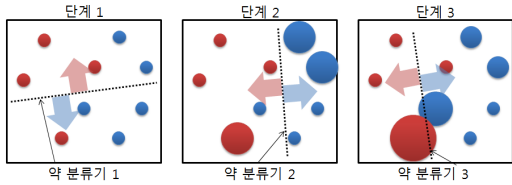


그림 3. 객체들의 특징 분포와 각 객체들의 가중치 그리고 분류기 선정 과정에 대한 예

[그림 3]은 Adaboost를 이용할 경우, 특징에 대한 가중치 변화와 그에 따른 분류기 생성 방법에 대해 나타내는 예제 그림이다. 입력된 데이터를 구분하기 위해 여러 개의 특징을 이용한다. 각 특징은 후보 특징들로부터 학습 데이터를 잘 분류할 수 있는 것들로 선택된다. Adaboost의 핵심은 특징이 선택된 후, 각 데이터의 가중치를 변경하여 다음 특징을 선택할 때, 잘못 분류된 데이터를 잘 분류할 수 있게 만드는 알고리즘이다. [그림 3]에서 원의 크기는 가중치의 크기를 나타내며, 점선은 분류기로 인한 경계선을 나타낸다.

표 1. 기존의 Adaboost 알고리즘

입력 : 학습데이터 - I_1, I_2, \dots, I_n 학습 알고리즘 : for (j = 1; j < n; j++) $W_1(j) = 1/n$ for (t = 1; t < T; t++) { 수식 (1)에서 ϵ_t 가 최소화 되는 F_t^* 와 T_t^* 를 찾음. $\epsilon_t = \sum_{\substack{j: \text{pos data} \\ T_t^* \leq F_t(j)}} W_t(j) + \sum_{\substack{j: \text{neg data} \\ T_t^* > F_t(j)}} W_t(j)$ (1) 만약 $\epsilon_t > 1/2$ 이면, 학습 종료.

$$\alpha_t = \frac{1}{2} \log\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) \text{와 같이 설정.}$$

업데이트

$$W_{t+1}(j) = \begin{cases} \frac{W_t(j) \exp(-\alpha_t h_t(I_j))}{Z_t}, & \text{if } j \text{ is pos data} \\ \frac{W_t(j) \exp(\alpha_t h_t(I_j))}{Z_t}, & \text{if } j \text{ is neg data} \end{cases} \quad (2)$$

Z_t 는 정규화 인자이며, h_t 는 다음과 같다.

$$h_t = \begin{cases} 1, & F_t^*(I_j) > T_t^* \\ -1, & \text{Otherwise} \end{cases}$$

분류 알고리즘 :

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (3)$$

[표 1]은 기존의 Adaboost 알고리즘을 나타낸다. 학습에 앞서 모든 데이터의 가중치는 동일하게 설정한다. 학습을 진행하면 Adaboost는 수식 (1)의 오차율이 최소가 되는 특징 F_t^* 와 그에 해당하는 임계값 T_t^* 을 추출한다. 수식 (1)의 $F(i, j)$ 는 j 번째 데이터에서 i 번째 특징 값을 나타낸다. 수식 (1)에서 오차율은 오 분류된 데이터에 가중치를 곱한 것들에 대한 전체 합으로 구성된다. 그러므로, 오차율은 다시 말하면 잘못 분류된 데이터에 대한 확률을 나타낸다. [표 1]에서 $h_t(I_j)$ 는 t 번째 학습 시, j 번째 데이터에 대해 객체 혹은 배경으로 분류하는 추정함수이다. 만약 특징이 임계값보다 클 경우 1로 설정되며, 그 반대의 경우 -1로 설정된다. 최소 오차율로 인해 최적의 특징과 그에 대한 임계값이 계산되었을 경우 그 다음으로 가중치를 업데이트 한다. 가중치를 변경하기 위한 수식으로는 수식 (2)가 사용된다. 수식 (2)에서 가중치 변경은 지수를 이용하여 변경되며, 잘못 분류되었을 경우 지수의 특징 상 급격하게 커지며, 잘 분류되었을 경우는 가중치가 작아지지만 0보다는 큰 값을 가지게 된다. 마지막으로 학습으로부터 추출된 특징과 임계값을 이용하여, 수식 (3)을 통해 입력 데이터가 들어 왔을 경우 학습된 데이터들로부터 추출된 추정함수를 통해 분류하게 된다.

2. 데이터 분포를 이용한 초기 가중치 설정

일반적으로 초기 가중치는 전반적인 학습 알고리즘에 많은 영향을 미친다[14]. 이에 지금까지 초기 가중치에 대해 많은 연구가 진행되고 있다[11]. 일반적으로 초

기 가중치가 다를 경우 항상 최적의 결과를 가져온다고 볼 수 없다. 그렇기 때문에 가치가 있는 변수를 구분하여 초기 가중치를 다르게 할당하는 알고리즘에 대한 연구가 있었다[15]. 기존의 Adaboost 알고리즘은 특징에 대해 학습 전에는 그 가치를 알 수 없다고 하여 학습 전 데이터의 가중치를 동일하게 할당하였다[5]. 하지만, 본 논문에서는 다음의 가설을 이용해 초기 가중치를 다르게 할당하여 더 좋은 결과를 볼 수 있다. 첫 번째로, 일반적으로 유용한 특징은 학습 데이터 중 객체 데이터에서 높은 값을 갖는다. 두 번째로, 배경 데이터는 특징이 여러 곳에 퍼져있는 반면, 객체 데이터는 한군데로 모이는 경향이 있다. [그림 4]는 객체 데이터와 배경 데이터의 분포도를 나타낸 결과 이다. 수평 방향과 수직 방향은 두 개의 다른 특징의 분포를 나타낸다. 각 데이터의 특징 값은 하나의 점으로 표시했다. [그림 4]에서 볼 수 있듯이 객체 데이터는 배경 데이터에 비해 한곳에 밀집해 있는 것을 알 수 있다.

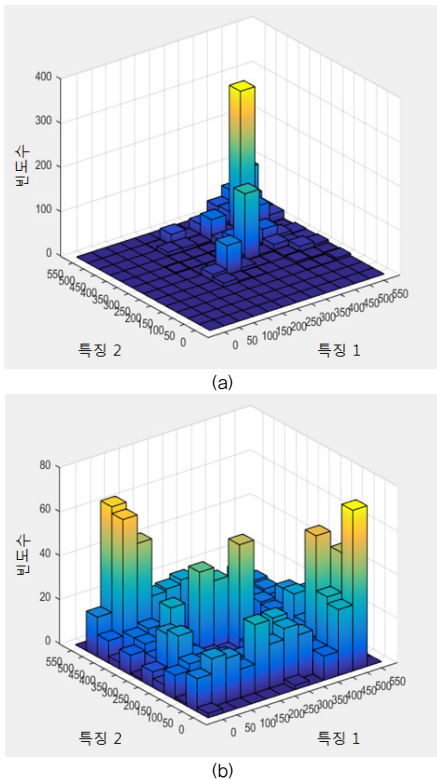


그림 4. 데이터 분산 (a) 긍정 데이터 (b) 부정 데이터

이 가설을 이용하여 각 객체 데이터와 배경 데이터에 초기 가중치를 다르게 할당하는 방법을 제안한다. 배경 데이터의 경우 기존과 동일하게 초기 가중치를 할당한다. 객체 데이터의 경우 [그림 4]에서와 같이 중심에 가까이 있는 데이터에는 경우 높은 가중치를 제공하며, 멀리 있는 데이터에는 낮은 가중치를 제공한다. 이를 위해 객체 데이터에서 각 특징에 대해 평균값을 생성하며, 이 평균값에 대해 분포를 계산한다. [그림 5]는 가중치 설정 알고리즘을 나타내는 예제 그림이다. 수식 (4)에서 $F(i, j)$ 는 j 번째 학습 데이터에서 i 번째 특징 값을 나타내며 $Avg(i)$ 는 N 개의 객체 데이터에서 i 번째 특징에 대한 평균값을 나타낸다.

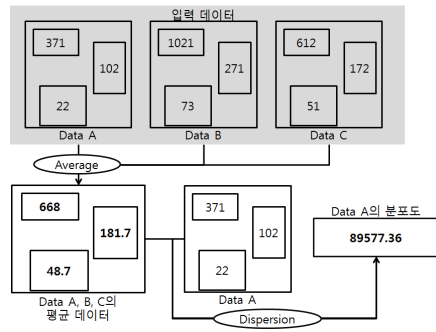


그림 5. 평균과 분포를 이용한 가중치 산출 예

$$Avg(i) = \frac{1}{N} \sum_{j=1}^N F(i, j) \tag{4}$$

수식 (5)에서는 수식 (4)에서 얻은 평균값을 이용하여 분포도를 계산한다. 수식(5)에서 $Disp$ 는 각 데이터에서 특징 값들의 분포도를 나타낸다. Avg_{min} 은 평균값들 중 최소값을 나타내며, Avg_{max} 은 평균값들 중 최대값을 나타낸다. M 은 특징들의 총 개수를 말하며 이를 이용해 분포를 계산한다.

$$Disp(j) = \frac{1}{M} \sum_{i=1}^M \frac{Avg(i) - Avg_{min}}{Avg_{max} - Avg_{min}} \cdot (F(i, j) - Avg(i))^2 \tag{5}$$

수식 (6)에서는 각 데이터의 분포도를 통해 가중치를

설정한다. 낮은 분포도면 낮은 가중치를 할당하며, 높은 분포도면 높은 가중치를 할당한다. 또한, 분모가 0이 되는 문제를 해결하기 위해 분모의 최소값이 1이 되도록 한다. $Disp_{min}$ 은 수식 (5)에서 계산한 값들 중 최소값을 나타내며 $Disp_{max}$ 는 최대값을 나타낸다.

$$D(j) = \frac{Disp_{max} - Disp_{min}}{1 + (Disp(j) - Disp_{min})} \quad (6)$$

마지막으로 초기 가중치의 정규화를 위해 수식 (7)과 같이 계산한다.

$$W(j) = \frac{D(j)}{\sum_{j=1}^M D(j)} \quad (7)$$

3. 특징을 선택하기 위한 퍼지 결정

Adaboost는 학습 할 때마다 좋은 특징과 이에 해당하는 최적 임계값을 찾아내는 알고리즘이다. 좋은 특징을 찾기 위해 학습 데이터를 분류할 때 최소 오차율을 이용하여 임계값과 특징을 추출한다[5]. 이러한 임계값을 이용하여 분류하는 방법을 경관정(hard decision)이라고 하며, 이는 분류하는데 참과 거짓으로만 구분이 된다. 이러한 방법은 경계 주변으로 구성된 입력에 대해서는 큰 오차를 발생하는 문제가 있다. 따라서, 본 논문에서는 퍼지를 이용한 연관정을 사용한다. 이 방법을 사용할 시, Adaboost 알고리즘에 유연성을 제공할 뿐만 아니라 학습 데이터의 수가 적어도 좋은 결과를 가져오는 장점이 있다.

[그림 6]은 퍼지 결정에 대한 전반적인 내용을 설명한다. [그림 6]에서 수평축은 특징 값을 나타내며 수직축은 참 그룹($pm(j)$)에 속하는 멤버십을 나타낸다. 수식 (8)과 수식 (9)는 각 데이터가 객체 그룹에 속하는 정도를 결정하는 멤버십에 관한 수식으로서, 학습 데이터의 특징이 Th_1 값보다 작을 경우 객체 그룹에 속하므로 1로 할당하게 된다. 또한, 특징 값이 Th_1 과 Th_2 값 사이에 속할 경우 수식 (8)과 수식(9)의 멤버십으로부터 나온 결과로 할당하게 된다. Th_2 보다 클 경우에는 배경

그룹에 속하므로 0으로 할당한다. 배경 그룹의 멤버십은 객체 그룹 멤버십 결과와 상반된 결과이므로 1에다가 배경 그룹의 멤버십 결과를 뺀 값이 된다.

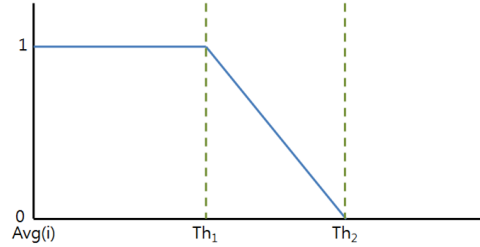


그림 6. 퍼지를 이용한 결정 경계 예

$$Th_k(i) = Avg(i) + \frac{k}{N} \sum_{j=1}^N \sqrt{(Avg(i) - F(i,j))^2} \quad (8)$$

$$pm(j) = \frac{1}{M} \sum_{i=1}^M \begin{cases} 1, & \text{if } F(i,j) < Th_1(i) \\ \frac{F(i,j) - Th_2(i)}{Th_1(i) - Th_2(i)}, & \text{if } Th_1(i) \leq F(i,j) \leq Th_2(i) \\ 0, & Th_2(i) < F(i,j) \end{cases} \quad (9)$$

퍼지 Adaboost는 매 학습 때마다 특징 F_i 와 이에 해당하는 임계값 T_i 를 수식 (10)을 이용한 최소 오차율을 계산함으로써 선택하게 된다. 수식 (10)은 최소 오차율에 대한 수식으로서 오류가 난 데이터의 가중치와 해당 멤버십의 곱셈으로 구성되어 있다. 수식 (10)의 앞 부분은 배경 그룹 멤버십과 객체 데이터로 구성되어 있으며, 뒷 부분은 객체 그룹 멤버십과 배경 데이터로 구성되어 있다. 이 두 개를 더함으로써 최소 오차율을 추출하는 수식이다.

$$\epsilon_i = \sum_{j: pos\ data} W_i(j) \cdot (1 - pm(j)) + \sum_{j: neg\ data} W_i(j) \cdot pm(j) \quad (10)$$

수식 (10)으로부터 최적의 특징과 그의 임계값을 찾고 이를 이용하여 다음 학습 때 사용할 가중치에 대하여 업데이트 한다. [표 2]는 제안하는 방법의 알고리즘을 나타낸다.

표 2. 제안하는 방법의 알고리즘

```

입력 : 학습데이터 -  $I_1, I_2, \dots, I_n$ 
학습 알고리즘 :
초기화 단계 -
for( i = 1; i < M; i++ )
수식 (4)를 통해 각  $Avg(i)$ 를 산출
for( j = 1; j < N; j++ )
수식 (5)를 통해 각  $Disp(j)$ 를 산출
for( j = 1; j < N; j++ )
수식 (6)을 통해 각  $D(j)$ 를 산출
for( j = 1; j < N; j++ )
수식 (7)을 통해 각 가중치  $w(j)$ 를 산출

학습 단계 -
for( t = 1; t < T; t++ ) {
수식 (10)에서  $\epsilon_t$ 가 최소화 되는  $F_t^*$ 와  $T_t^*$ 를 찾음.
만약  $\epsilon_t > 1/2$ 이면, 학습 종료.
 $\alpha_t = \frac{1}{2} \log(\frac{1-\epsilon_t}{\epsilon_t})$ 와 같이 설정.
수식(2)를 통해 다음 가중치  $w_{t+1}$ 를 산출
}

분류 알고리즘은 수식(3)과 동일
    
```



그림 7. 학습데이터 (a) 얼굴 데이터, (b) 보행자 데이터, (c) 배경 데이터

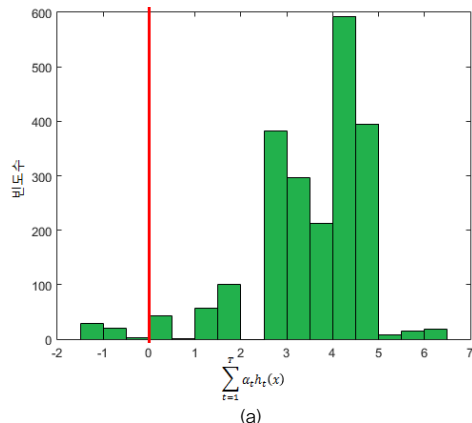
초기 가중치를 설정하는데 걸리는 시간은 대략 4.2초의 계산 시간이 필요하며 리소스는 특징의 수를 저장할 수 있는 만큼의 메모리를 필요로 한다. Adaboost는 최적의 특징을 뽑아내는데, 루프 한번에 약 8분 이상 소요되며, 메모리도 초기 가중치를 계산하는 메모리에 비해 훨씬 많은 리소스가 소요 된다. 따라서, 초기 가중치 설정은 최적의 특징을 추출하는데 있어 시간 복잡도나 시스템 메모리에 대해 영향이 거의 없는 것으로 판단된다.

[그림 8]은 수식 (3)의 여러 개의 가중치와 추정함수의 합($\sum_{t=1}^T \alpha_t h_t(x)$)들로 구성된 히스토그램의 결과를 나타낸다. 히스토그램의 빈은 적절히 양자화 된 결과로 표기하였다. 수평축은 빈을 나타내며 수직축은 빈에 속하는 학습 데이터로부터 도출된 결과의 누적 개수를 나타낸다. [그림 8]에서 확인 할 수 있듯이, 대부분의 객체 데이터의 결과는 양수 값을 갖으며, 배경 데이터의 결과는 음수 값을 갖는다. 극히 드물게 어떤 객체 데이터는 -2를 갖으며 배경 데이터는 50 근처에 있지만, 개수가 적으며, 또한 확인 특징이 늘어나면 해당 오류도 줄어드는 것을 확인 할 수 있었다.

III. 실험 결과 및 결론

1. 실험 결과

본 시스템을 개발 및 실험하기 위한 컴퓨터 환경으로는 Intel(R) Core™ i7-4790 3.60 GHz CPU와 8GByte 메모리를 사용하였다. 운영체제는 Windows 8 Enterprise K 64bit OS를 사용하였으며, 개발 툴은 Visual Studio 2013의 MFC환경과 OpenCV 2.4.9를 사용하였다. 두 가지 환경에서 실험하였으며, 학습 환경으로는 하나는 객체 데이터로 400장의 얼굴 이미지와 배경 학습 데이터로 800장의 랜덤 이미지를 사용하였으며 다른 하나는 객체 데이터를 300장의 보행자 이미지와 600장의 랜덤 이미지를 사용하였다. 객체 데이터는 뉴욕 대학에서 제공하는 얼굴 데이터베이스[12]와 MIT에서 제공하는 보행자 데이터베이스[13]를 사용하였으며, 배경 데이터는 인터넷에 있는 임의의 데이터를 사용하였다. [그림 7]은 학습하는데 사용한 데이터에 대한 일부이다.



(a)

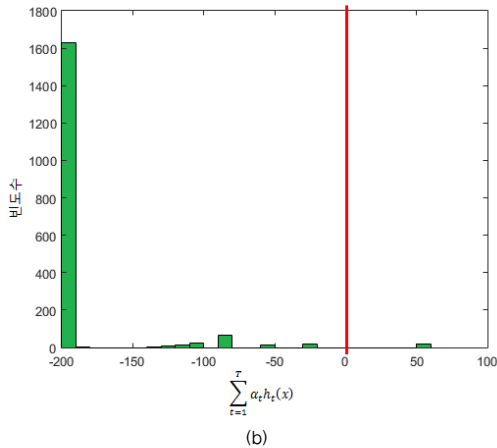


그림 8. 가중치와 추정함수들의 합으로 이루어진 가중치에 대한 (a) 객체 데이터, (b) 배경 데이터

실험을 위한 특징으로는 Haar-like 특징을 사용하였다. [그림 9]에서는 약 100,000개 이상의 특징들로부터 퍼지 Adaboost을 통한 학습 결과로 가장 좋은 특징 20개를 추출한 결과이다. [그림 9]에서 보듯이 결과들 중 많은 부분이 얼굴 주요 구성요소인 눈, 코, 입이 포함 된 것을 확인 할 수 있다.

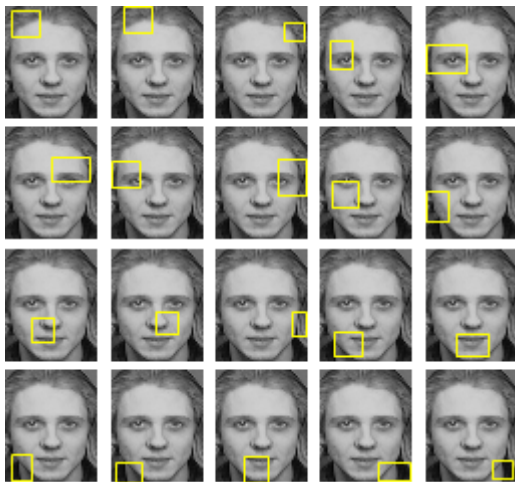


그림 9. 분류기로부터 검출된 객체 검증을 위한 특징들의 위치

[그림 10]은 기존의 Adaboost와 퍼지 Adaboost간의 ROC 곡선(Receiver Operating Characteristics)을 나타낸 결과이다. 두 분류기 모두 분류를 위해 30개의 특징

을 사용하였다. 이 곡선은 거짓 긍정률과 인식 성공률 간의 연관성에 대해 설명한다. 곡선의 분석 결과 전반적인 측면에 있어 제안하는 알고리즘이 기존의 Adaboost 알고리즘에 비해 모든 범위의 거짓 긍정률에 있어 인식 성공률이 더 좋은 결과를 나타낸다. 특히 거짓 긍정률이 낮을수록 더욱 좋은 결과를 보인다.

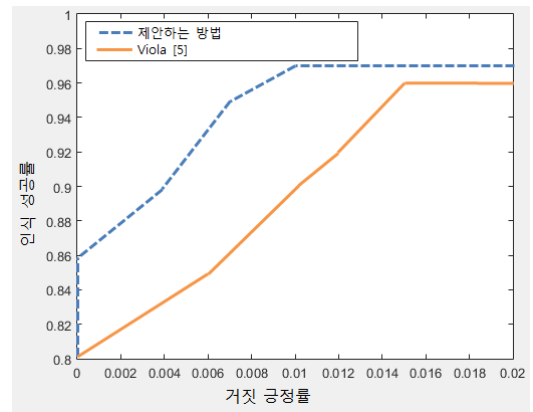


그림 10. 제안하는 방법과 기존 방법의 ROC 곡선 비교

[표 2]와 [표 3]은 기존의 Adaboost 알고리즘과 제안하는 퍼지 Adaboost 알고리즘 간의 실험 성능 비교를 나타낸다. [표 3]은 객체 데이터가 얼굴 데이터 때의 결과를 나타내며, [표 4]는 객체 데이터가 보행자 데이터에 대한 결과를 나타낸다. 성능 분석을 위해 전체 오차율, 거짓 긍정률, 거짓 부정률의 유형으로 비교 분석한다. [표 3]에서 확인할 수 있듯이, 거짓 긍정률이 거짓 부정률에 비해 오차율이 약간 높은 것을 확인할 수 있다. 이러한 결과는 [그림 9]의 히스토그램에서도 확인할 수 있는 부분이다. 또한, 제안하는 알고리즘들이 기존의 Adaboost 알고리즘들에 비해 모든 면에서 성능이 좋음을 확인할 수 있다. 데이터 분포를 이용한 초기 가중치 설정은 특징이 적을수록 더 잘 분류하는 것을 확인할 수 있다. 퍼지 결정을 이용한 방법은 특징이 적어도 인식률이 기존 방법에 비해 좋은 결과가 나타날 뿐만 아니라 특징이 많아도 인식률이 좋아지는 것을 확인할 수 있다.

표 3. 얼굴 검출 분류기 성능 비교 (단위 : %)

특징 개수	10	15	20	25	30
Viola [5]					
전체 오차율	18	15	13	10	7
거짓 긍정률	11	9	8	6	4
거짓 부정률	7	6	5	4	3
Joo [7]					
전체 오차율	17	13	10	8	6
거짓 긍정률	13	9	7	5	4
거짓 부정률	4	4	3	3	2
제안하는 방법					
전체 오차율	10	9	7	6	5
거짓 긍정률	6	5	4	3	3
거짓 부정률	4	4	3	3	2

표 4. 보행자 검출 분류기 성능 비교 (단위 : %)

특징 개수	10	15	20	25	30
Viola [5]					
전체 오차율	18	13	8	5	5
거짓 긍정률	3	3	2	1	1
거짓 부정률	15	10	6	4	4
Joo [7]					
전체 오차율	14	11	7	5	4
거짓 긍정률	7	6	3	3	3
거짓 부정률	7	5	4	2	1
초기 가중치 설정 + 퍼지 결정					
전체 오차율	9	5	4	3	2
거짓 긍정률	2	1	1	1	1
거짓 부정률	7	4	3	2	1

[그림 11]은 제안하는 알고리즘으로는 잘 분류되지만 기존의 Adaboost로는 오분류를 일으키는 데이터의 일부를 나타낸다. 두 분류기 모두 10개의 특징을 사용하였다. [그림 11]에서 볼 수 있듯이, 얼굴이 정면이 아닌 회전되었을 경우 분류에 실패하는 경우가 발생하는 것을 알 수 있다.



그림 11. 기존 방법의 False negative 예제

2. 결론

본 논문에서는 초기 가중치 설정 방법과 퍼지 Adaboost 알고리즘을 제안했다. 초기 가중치 설정은 학습을 적게 하여 특징을 적게 추출했을 때 효율적인 분류 결과를 보였다. 퍼지 결정은 더 좋은 특징을 추출하여 전반적으로 더 좋은 분류 결과를 보였다. 이러한 결과를 토대로 퍼지를 기반으로 한 Adaboost는 다양한 분야에서 기존에 방법보다 더 좋은 결과를 보일 것으로 예상된다.

참고 문헌

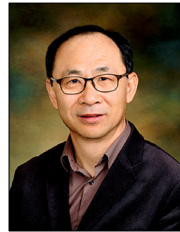
- [1] R. Rout, "A survey on object detection and tracking algorithms," PhD Thesis, National Institute of Technology Rourkela, 2013.
- [2] D. Prasad, "Survey of the Problem of Object Detection in real images," International Journal of Image Processing, Vol.6, Issue.6, pp.441-466, 2012.
- [3] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," The journal of machine learning research, pp.45-66, 2001.
- [4] M. Hagan, H. Demuth, M. Beale, and O. Jesus, *Neural Network Design*, Boston:Pws Pub., 1996.
- [5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Computer Vision and Pattern Recognition, 2001.
- [6] G. Ratsch, T. Onoda, and K. Muller, "Soft margins for AdaBoost," Machine learning, Vol.42, Issue.3, pp.287-320, 2001.
- [7] S. Joo, S. Weon, and H. Choi, "Real-time depth-based hand detection and tracking," The Scientific World Journal, 2014.
- [8] R. Lienhart and M. Jochen, "An extended set of haar-like features for rapid object detection,"

Image Processing, 2002

- [9] J. Zhu, S. Rosset, H. Zou and T. Hastie, "Multi-class adaboost," Statistics and its Inference, Vol.2, No.3, pp.349-360, 2009.
- [10] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," Cybernetics, IEEE transactions on, Vol.44, No.1, pp.66-82, 2014.
- [11] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural computation, Vol.18, No.7, pp.1527-1554, 2006.
- [12] <http://www.cs.nyu.edu/~roweis/data.html>
- [13] <http://cbcl.mit.edu/software-datasets/PedestrianData.html>
- [14] S. Lomax and S. Vadera, "A survey of cost-sensitive decision tree induction algorithms," ACM Computing Surveys (CSUR), Vol.45, Issue.2, 2013.
- [15] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," Acoustics, Speech and Signal Processing (ICASSP), 2013.

최 형 일(Hyung-II Choi)

정회원



- 1972년 2월 : 연세대학교 전자공학과(공학사)
- 1983년 8월 : 미시간대학교 전자공학(공학석사)
- 1987년 6월 : 미시간대학교 전자공학(공학박사)
- 1987년 9월 ~ 현재 : 숭실대학교 미디어학과 교수
<관심분야> : 컴퓨터비전, 인공지능, 패턴인식, 영상처리, 기계학습

저 자 소 개

김 기 상(Kisang Kim)

정회원



- 2007년 2월 : 숭실대학교 컴퓨터학과(공학사)
- 2009년 2월 : 숭실대학교 미디어학과(공학석사)
- 2014년 3월 ~ 현재 : 숭실대학교 미디어학과 박사과정

<관심분야> : 컴퓨터비전, 인공지능, 기계학습