

Adaboost 학습을 이용한 얼굴 인식

정 종 료, *최 병 옥

한양대학교 전자통신전파공학과, *한양대학교 정보통신학부

E-Mail : bellaw@ihanyang.ac.kr

Face Recognition Using Adaboost Learning

Abstract

In this paper, we take some features for face recognition out of face image, using a simple type of templates. We use the extracted features to do Adaboost learning for face recognition. Using a carefully-chosen feature among these features, we can make a weak face classifier for face recognition. And doing Adaboost learning on and on with those chosen several weak classifiers, we can get a strong face classifier. By using Adaboost learning, we can choose particular features which is not easily subject to changes in illumination and facial expression about several images of one person, and construct face recognition system. Therefore, the face classifier built like the above way has robustness in both facial expression and illumination variation, and it finally gives capability of recognizing face fast due to the simple feature.

I. 서론

생체 인식 기술 중에 하나인 얼굴 인식은 입·출입 시스템, 감시 시스템, 보안 시스템 등의 응용 분야에

널리 사용되고 있는 기술 중의 하나이다. 사람을 대신 하여 얼굴을 인식하는 얼굴 인식 시스템은 사용자 편의성 면에서 생체 인식 시스템 중 가장 탁월한 시스템이라 할 수 있다. 따라서 얼굴 인식에 대하여 많은 사람들이 관심을 가지고 지속적으로 연구를 진행하고 있다. 이러한 얼굴 인식 연구의 주요한 부분은 얼굴에서 얼굴 인식을 위하여 적합한 특징들을 추출하는 방법과 추출된 특징 정보를 사용하여 얼굴 인식을 하는 효과적인 방법을 찾아내는 것이다. 또한, 얼굴 인식 시스템은 조명 변화, 얼굴 표정 변화, 얼굴의 각도 변화, 액세서리 착용 등의 다양한 환경 변화에 따라 인식률에 영향을 받기 때문에 이에 대한 연구도 중요시되고 있다.

본 논문에서는 표정 변화와 조명 변화를 가지는 여러 장의 학습 이미지에서 쉽게 추출할 수 있는 간단한 얼굴 특징들을 사용하여 Adaboost 학습을 함으로써 조명이나 얼굴의 표정 변화에 강인하며 인식 속도가 빠른 얼굴 인식을 구현하고자 한다.

II. 특징 정보 추출

본 논문에서는 학습 이미지에서 얼굴 인식을 위한 특징을 얻기 위하여 간단한 형태의 특징 정보들을 사용하였다. 이 특징들은 그림 2.1과 같이 두 종류의 템플릿을 통하여 얻어진다. 특징 정보를 얻어내기 위한 템플릿은 2개와 3개의 부분 영역들을 내부에 가지고 있다. 그리고 개별 부분 영역 안에 있는 픽셀들의 값을 더하여 영역 합을 구하고 그 값들에 가중치를 곱하여 합을 계산함으로써 하나의 스칼라 값을 얻는다.

각 부분 영역의 크기와 위치는 전체 영역 안에서 자유롭게 변하기 때문에 하나의 이미지 영역에서 많은

특징 정보 값을 얻을 수 있다.

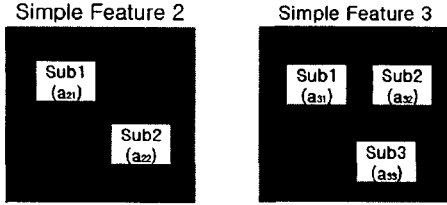


그림 2.1 특징 정보 추출의 위한 템플릿

$$SF_{2,i} = a_{21} \times SF_{2,i,sub1} + a_{22} \times SF_{2,i,sub2}$$

$$SF_{3,i} = a_{31} \times SF_{3,i,sub1} + a_{32} \times SF_{3,i,sub2} + a_{33} \times SF_{3,i,sub3}$$

$SF_{i,j}$: Simple Feature i에서 j번째 형태의 특징 값

$SF_{i,j,sub(k)}$: Simple Feature i에서 j번째 형태의 k번째

부분 영역안의 픽셀들의 합의 값

a_{ij} : 각 부분 영역의 가중치

III. 간단한 얼굴 분류기

위에서 두 종류의 템플릿을 사용하여 얻어진 특징 값들은 얼굴인식을 위한 얼굴 분류기를 만들기 위하여 Adaboost 학습을 하는데 사용하게 된다. Adaboost 학습에서는 매 학습 루프마다 위에서 추출된 간단한 특징들 중에서 주어진 학습 데이터에 가장 적합한 하나의 특징을 선택한 후 이를 사용하여 간단한 얼굴 분류기를 구성한다. 그 후 간단한 얼굴 분류기들을 조합하여 성능이 좋은 얼굴 분류기를 구성하게 된다.

학습에 사용되는 이미지는 개인에 대하여 표정 변화와 조명 변화를 가지는 여러 가지 이미지들이 사용된다. 이 이미지에 대하여 선택된 템플릿의 형태를 사용하여 특징 값을 추출해 낼 수 있다. 그리고 추출된 특징 값들을 사용하여 개인에 대한 이미지의 모임(class)에서 하나의 대표 값을 얻어내게 된다.

$$\arg \min_{Cv_i} \sum_{\# \text{ of Class } i} \sqrt{(Cv_i - Cm_{ij})^2}$$

Cv_i : Class i의 대표값

Cm_{ij} : Class i의 j번째 이미지의 특징 값

이렇게 얻어진 클래스 대표 값은 클래스 내에서 클래스 내부의 이미지들의 특징 값과 가장 작은 차이를

가지는 값이 된다. 이 클래스 대표 값은 다음 그림 3.1과 같이 모든 학습 이미지의 특징 값들과 거리를 계산하여 Adaboost 학습 중에 사용될 간단한 얼굴 분류기를 결정하는데 사용된다.

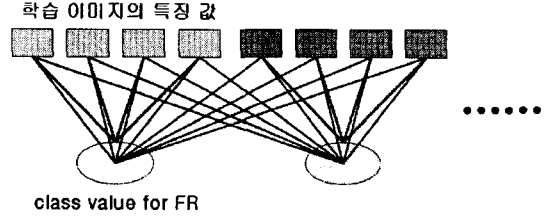


그림 3.1 간단한 얼굴 분류기의 결정

$$dis. = \arg \min_{i,j} \sqrt{(Cv_i - Cm_{jk})^2}$$

if $i = j$ then $h(Cm_{jk}) = +1$

만약 가장 작은 거리 값에서 $i=j$ 가 된다면 올바르게 결정된 간단한 얼굴 분류기라고 할 수 있다. 그러나 두 번째나 세 번째 작은 값에서 $i=j$ 가 된다면 완벽하지 않은 간단한 얼굴 분류기가 될 것이다. 따라서 이 경우에 학습 되는 데이터에 대하여 간단한 얼굴 분류기를 통하여 인식이 되는 정도에 따라 얼굴 분류기의 인식률 에러를 계산할 때 사용할 수 있도록 가중치를 부여하게 된다.

$$w_n = \frac{d_n - d_1}{d_{n-1} - d_1} \times (w_{n-1} - w_1) + w_1 \quad (3 \leq n \leq T_w)$$

$$w_1 = -1 \quad (T_w < n)$$

w_i : i번째 순위의 데이터 가중치 값 ($0 < w_i \leq 1$)

d_i : class 대표 값과 데이터 값 사이의 i순위 거리 값

T_w : 가중치가 부여되는 마지막 순위

이 가중치를 사용하면 간단한 얼굴 분류기를 사용하여 학습을 할 경우 데이터가 인식된 정도에 따라 가중치를 적용함으로써 효율적으로 Adaboost 학습을 할 수 있다.

IV. Adaboost 학습

얼굴 인식을 위한 학습을 위하여 Adaboost 학습을 사용하였다. 그러나 본 논문에서 제한한 간단한 얼굴 분류기가 데이터와 학습에 미치는 영향을 표현하기 위

한 데이터 가중치(w_d)를 일반적인 Adaboost 학습 알고리즘에서는 사용할 수 없기 때문에 본래의 알고리즘에 제안한 데이터 가중치가 영향을 미치는 부분에 대하여 수정하여 학습에 적용하였다.

V. 실험 결과

실험을 위하여 40명에 대하여 정규화된 40x40 크기의 이미지가 6장씩 사용되었다. 그 중 학습 이미지로는 그림 5.1에서 보는 것과 같이 개인에 대하여 표정 변화와 조명 변화를 가지는 4장의 이미지가 사용되었다. 인식을 위한 이미지로는 표정 변화와 조명 변화가 있을 때 인식률을 알아 볼 수 있도록 그림 5.2와 같이 표정과 조명변화에 관한 얼굴 이미지를 개인에 대하여 2장씩 총 80장을 사용하였다. 실험에는 펜티엄3 1GHz PC와 Visual C++ 6.0가 사용되었다.

Adaboost 학습

1. 주어진 데이터 $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in \{1\}$ 에 대하여 분포를 $D_1(i)=1/m$ 초기화
2. $t=1, \dots, T$ 에서 학습을 진행한다.
 - 2.1 우선 D_t 와 데이터를 사용하여 간단한 얼굴 분류기를 학습하고 가장 작은 인식 에러(ϵ_t)를 가지는 간단한 얼굴 분류기(h_t)를 선택한다.

$$\epsilon_t = \sum_{i: h_t(x_i)=-1} D_t(i) + \sum_{i: h_t(x_i)=+1} (1-w_d) \times D_t(i)$$
 - 2.2. $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ 를 계산한다.
 - 2.3. D_t 를 업데이트 한다.

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times e^{-\alpha_t \times w_d}$$
3. 결과적으로 성능이 좋은 얼굴 분류기를 얻을 수 있다.

$$R_v(x) = \sum_{adaboost_loop} [\alpha_t \times w_d \times h_t(x)]$$

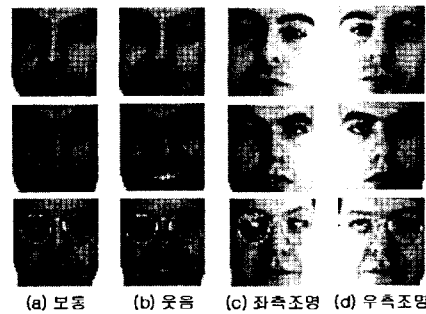


그림 5.1 학습에 사용된 이미지

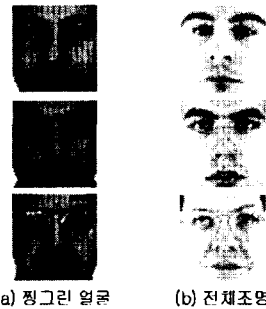


그림 5.2 인식에 사용된 이미지

위에서 언급된 Adaboost 학습 알고리즘을 사용하여 학습을 진행하면 최종적으로 얼굴 인식에 사용되는 성능이 좋은 얼굴 분류기를 다음과 같이 얻을 수 있다.

$$R_v(x) = \sum_{adaboost_loop} [\alpha_t \times w_d \times h_t(x)]$$

R_v : 얼굴 인식시 같은 얼굴이라고 판별되는 정도를 나타내는 값 (Recognition Value)

이 분류기를 사용하여 입력 얼굴 이미지를 인식 한다. 이 때 인식하고자 하는 얼굴 이미지와 DB상의 얼굴 이미지 사이에서 R_v 를 구하여 R_v 값이 최대가 되는 DB상의 얼굴을 인식된 얼굴로 선택하게 된다.

인식률은 표정 변화를 가지는 이미지와 조명 변화를 가지는 이미지에 대하여 인식률을 계산하고 다시 총 이미지에 대하여 인식률을 계산하였다. 그림 5.3과 같이 인식이 된 가장 가까운 순서 순으로 5번째 순위까지의 인식률을 그래프 형식으로 표시하였다.

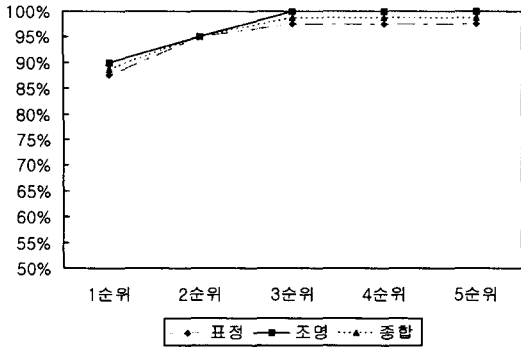


그림 5.3 얼굴 인식률

그림 5.4에서는 얼굴 인식에 사용된 특징의 개수에 따른 인식률의 변화에 대하여 그래프로 표시하였다. 그림에서 보는 바와 같이 얼굴 인식에 사용되는 특징의 개수가 증가하면 인식률도 같이 증가하는 추세를 보인다. 그러나 20개 이상의 특징에서는 인식률의 명확한 증가가 나타나지 않는다.

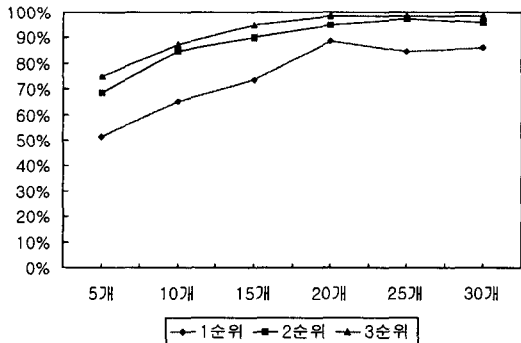


그림 5.4 사용된 특징 개수에 따른 인식률

실험 결과 표정 변화나 조명 변화에 대하여 특별한 전처리 과정을 거치지 않았음에도 불구하고 비교적 좋은 인식 성능을 보였다. 동시에 간단한 특징 정보들을 얼굴 인식에 사용함으로써 얼굴 이미지를 읽고 특징을 추출한 후 DB에서 인식하는데 평균 2.42 ms 정도의 시간이 걸려 빠른 얼굴 인식이 가능하였다.

VI. 결론

위와 같은 얼굴 인식 방법은 Adaboost 학습을 사용함으로써 특별한 특성이 없는 간단한 특징 정보들만을 사용하여 원하는 특성이 있는 얼굴 인식 시스템을 구현할 수 있다. 동시에 간단한 특징만을 사용함으로써

비교적 빠른 처리 속도를 가진다. 다만 Adaboost 학습 시 많은 특징 정보를 사용하기 때문에 학습 시간이 오래 걸리고 DB추가가 용이하지 않은 문제점이 있다.

본 논문에서는 Adaboost 학습을 사용하여 개인의 여러 장의 이미지에서 얻은 특징들 중에서 얼굴의 표정 변화나 조명 변화에 민감하지 않은 특징을 선택하여 얼굴 인식 시스템을 구성하여 보았다. 결과 간단한 영역 계산만을 통하여 빠른 처리가 가능하면서도 얼굴 표정 변화와 조명 변화에 강인한 얼굴 인식 시스템을 구현할 수 있었다.

참고 문헌

- [1] ZhenQiu Zhang, MingJing Li, Stan Z. Li, HongJiang Zhang, "Multi-View Face Detection with FloatBoost" Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on , 2002
- [2] Stan Z. Li, Long Zhu, ZhenQiu Zhang, HongJiang Zhang, "Learning to detect multi-view faces in real-time", Development and Learning, 2002. Proceedings. The 2nd International Conference on , IEEE 2002
- [3] Vladimir Pavlovic, "Boosted Detection of Objects and Attributes", Computer Vision and Pattern Recognition, IEEE 2001
- [4] P. Viola and M.J. Jones, "Robust real-time object detection", Technical Report Series, Compaq Cambridge Research Laboratory, CRL 2001/01, Feb.2001