안드로이드 기반의 휴대용 스마트폰을 이용한 실시간 얼굴 검출

구모세*, 김상훈*
*한경대학교 전기전자제어공학과
e-mail: kimsh@hknu.ac.kr

Android-based Face detection using OpenCV

Mose Koo*, Sang-Hoon Kim*
*Dept of Electrical, Electronic and Control, Hankyong National University

본 논문에서는 현재 활발히 연구 중에 있는 얼굴 인식의 전 과정인 얼굴 검출단계를 OpenCV를 이용한 안드로이드 기반의 휴대용 스마트폰으로 실시간 얼굴 및 눈 영역을 검출하는 어플리케이션의 개발을 수행하였다. 얼굴 검출 및 눈 검출 기술은 OpenCV에서 제공하는 실시간 얼굴 인식을 위해 이미지에서 얼굴의 특징을 찾는 기법 중 하나인 Haar-like Feature을 이용한 검출 방식을 사용하였다. 얼굴 검출 및 눈 검출에 대해 스마트폰에서 촬영한 이미지를 사용하여 구현 결과를 평가하였다.

1. 서론

처음 Apple이 스마트폰을 세상에 내놓은 이후로, 스마트폰, 갤럭시탭/아이패드 등과 같은 휴대 단말 관련 기술이 진화하고, 보급이 급속하게 증가하고 있다. 스마트폰의 등장은 Wifi, Bluetooth, 고성능의 카메라 및 다양한 각종 센서 등을 탑재하여 단순한 통화, 카메라 기능이 중심이었던 핸드폰 시장을 특별한 기능을 수행하는 어플리케이션(앱) 프로그램을 이용할 수 있는 환경으로 변화시켜 스마트폰의 활용의 다양성을 비약적으로 증가시켰다. 안드로이드는 리눅스 기반의 모바일 환경을 위한 오픈소스 플랫폼(Linux Platform)으로 스마트폰 컴퓨팅 환경 중의 한 가지이다. Google이 주도하는 모바일 플랫폼으로써 급격한 정보 사회화의 변화에 기인해 지속적인 발전을 하고 있다. 스마트폰의 활용과 편의성이 증대함에 따라서 사용자의 생활에 밀접하게 관여하고 있고 자연스럽게 보안에 대한 요구도 증가하게 되면서 보안을 위한 생체인식시스템이 중요성이 강조되고 있다.[1] 생체인식 분야 중 얼굴 검출 및 인식 기술은 가장 활발한 연구가 진행되는 분야이다. 얼굴 인식 시스템(face recognition system)은 얼굴의 특징을 추출해 별다른 과정 없이 카메라로부터 받은 영상의 얼굴을 검출해 검출한 영역의 얼굴의 대상이 누구인지 빠르게 인식할 수 있어 연구가 활발히 이뤄지고 있다.[2] 이러한 얼굴 인식 시스템에서 가장 먼저 수행되어야 하는 단계는 얼굴 검출(face detection)이다. 얼굴 검출이란 주어진 영상에 대하여 얼굴의 유무를 판단하고 얼굴이 존재할 경우 얼굴의 위치를 찾아내는 과정을 말한다.[3]

본 논문에서의 주요 내용은 안드로이드 기반 스마트폰에서 Open CV를 이용한 얼굴 영역과 눈 영역 검출시스템을 적용하여 어플리케이션으로 구현하였으며, 그에 대한 검출 성능평가를 바탕으로 본 연구를 응용해 적용할 수 있는 다양한 가능성을 제시한다.

2.본론

2.1. 개발 툴, 안드로이드 스튜디오

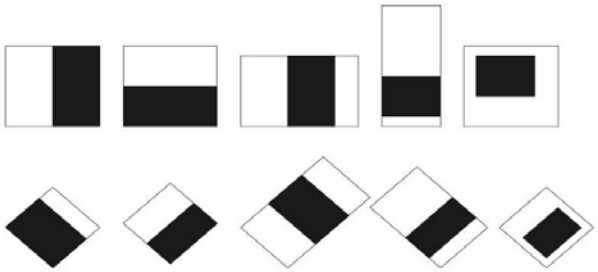
2013년 5월 16일, 구글 I/O 컨퍼런스에서 발표한 안드로이드 스튜디오는 안드로이드 및 안드로이드 전용 어플(앱) 제작을 위한 공식 통합 개발환경 (IDE)이다. 소프트웨어용 Java의 통합 개발 환경인 IntelliJ를 기반으로 코드 편집 및 개발 도구를 통합한다. 안드로이드 운영 체제 내에서 앱 개발을 지원하기 위해 Gradle 기반 빌드 시스템, 에뮬레이터, 코드 템플릿 및 Github등을 통합하여 사용한다. 안드로이드 스튜디오가 개발되기 전에는 Eclipse를 통한 개발을 하였지만, 더 이상 Google에서 ADT(Android Developement Tool)을 지원하지 않기 때문에 안드로이드 스튜디오를 통한 어플리케이션 제작을 하였다.

2.2. OpenCV를 이용한 얼굴 검출

Open CV는 인텔에서 개발한 실시간 영상처리, 컴퓨터 비전과 기계학습을 위한 소프트웨어 라이브러리로, 안드로이드로도 구현 가능하다. 컴퓨터가 인간처럼 시각을 통해 받아들이는 정보로 다양한 기능들을 구현할 수 있도록 한다. 이 라이브러리는 수많은 영상처리와 이를 통한 기계학습 알고리즘들을 가지고 있으며 이러한 알고리즘들은 사람의

얼굴을 인식 및 감지, 객체 및 움직임 추적, 전체 장

면을 위한 이미지들의 연결 등을 가능하게 하는 오픈소스 라이브러리이다.[4] 얼굴 검출을 위해 OpenCV에서 제공하는 실시간 얼굴 인식을 위해 이미지에서 얼굴의 특징을 찾는 기법중의 하나인 Haar-like Feature을 이용한 검출 방식을 사용할 것이다. 그림 1과 같은 사각형 흑백 영역에 대한 픽셀값의 평균차에 의한 임계치 구분에 의해 특징을 판단, 파악하는 기법으로 연산과정이 간단하여 빠른 얼굴 검출에 적합하다. Haar-like Feature를 이용한 얼굴 검출의 경우에 눈을 예로 들면 눈의 영역은 주변보다 어두운 특징이 있기 때문에 이러한 차이점을 이용해 얼굴뿐만이 아닌 눈 영역 검출에도 이용할 수 있는 방법이 된다.[5]



<그림 1> Haar-like feature

2.3 얼굴 검출, 전처리

이 어플리케이션의 주요 기능인 얼굴 검출과정에서는 휴대용 스마트폰의 카메라에서 영상을 받아 사람의 얼굴 영역과 눈 영역을 검출한다. 얼굴 검출의 알고리즘은 위에서 언급한 Haar-Like Feature을 이용한 검출 방식을 이용할 것이다. 또한, 얼굴뿐만이 아닌 정확한 얼굴 판별을 위해 눈 영역도 사용하였다. OpenCV에서 제공하는 검출기들은 눈, 코, 입, 눈썹 등의 검출기를 제공해 주지만 눈영역 검출이 정면 얼굴 검출시 항상 수평을 유지하고 환경에서의 제약도 다른 특징점들 보다 덜 받기에 검출이 유용해 얼굴 영역과 눈 영역을 검출해볼 것이다.

Face Detector	haarcascade_frontalface_alt.xml
Eye Detector (with glasses)	haarcascade_eye_tree_eyeglasses .xml

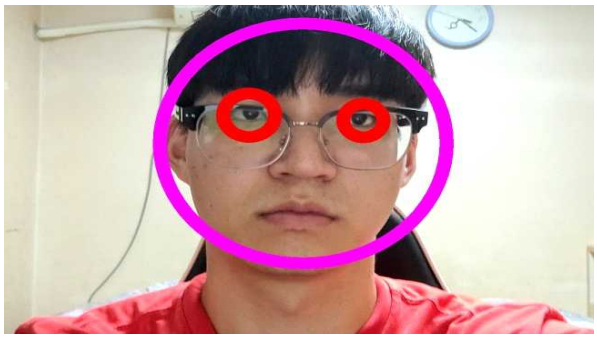
<표 1> Haar 분류기 XML 파일

OpenCV에서 제공하는 표 1의 두 가지 Haar 분류기 XML 파일을 CascadeClassifier 함수를 통해 불러온다. 이를 통해 검출된 얼굴은 전처리 과정을 거쳐 얼굴 검출을 수행하기 위한 형태의 이미지 파일로 변환되게 된다. 또한 얼굴 검출은 조명, 얼굴의 방향, 얼굴의 표정 등의 변화에 대해 매우 취약하다. 때문에 더 좋은 얼굴 검출 결과를 얻기 위해서는 얼굴 전처리 과정이 필수라고 말할 수 있다. 사용한 전처리 과정은 히스토그램 균등화이다. 히스토그램이란 영상 안에서 각 레벨이 가지는 화소의 개수(빈도)를 그래프 형식으로 나타낸 것이다. 서로 다른 조건에서 생성된 두 개의 영상이 동일한 히스토그램을 갖도록 만든 후, 각 영상의 히스토그램을 grayscale 전 구간에서 골고루 나타나도록 변경하여 준다. 히스토그램 균등화 과정은 첫 번째로 원 영상의 히스토그램을 생성하여 준다. 이후에 히스토그램의 값들을 정규화하여 누적 합을 계산해 그 값들을 이전 영상의 위치에 매핑하여 줌으로써 균등화 분포를 가진 히스토그램을 얻을 수 있다. 이에 따라 어두운 영상은 밝게, 밝은 영상은 조금 어둡게 하는 등 적당한 명도 값을 유지시켜 주게 되는 것이다. OpenCV에서 제공하는 히스토그램을 균등화시키는 equalizeHist() 함수를 사용해서 전처리 과정을 진행하였다.

3. 구현 결과

본 실험에서는 안드로이드 스마트폰 어플리케이션 개발 환경인 안드로이드 스튜디오에서 NDK를 이용해 스마트폰 카메라를 이용해 영상을 받아 영상인식 및 처리를 수행하여 얼굴 영역 검출 및 눈 영역 검출을 하고, 검출된 영상을 캡쳐하는 것을 수행하였다. 그림 2에서는 기본적인 순서적인 제어 흐름을 나타내었다. 1) 카메라를 초기화 및 영상 정보 영상 정보를 입력 후 2) 얼굴 영상 입력 및 전처리 과정을 거친다. 3) 얼굴 및 눈 영역을 검출하고 실시간으로 사용자에게 검출되고 있음을 보여주기 위해 얼굴 영역은 분홍색 원형으로, 눈 영역은 빨간색 원형을 검출된 얼굴과 눈동자에 각각 표현하여 준다. 얼굴 검출은 기본적으로 OpenCV에서 제공하는 Haar 검출에 대한 detectMultiScale() 함수를 이용하였다. 4) 실시간으로 얼굴 및 눈 영역이 검출되는 것을 캡쳐버튼을 추가해서 캡쳐를 누르게 되면 스마트폰의 갤러리로 캡쳐된 화면이 넘어가게 된다. 그림 3에서는 어플리케이션을 실행해 캡쳐한 결과화면으로, 얼굴 영역 및 눈 영역이 검출되는 모습을 볼 수 있다. 한 명의 얼굴 영상뿐만 아니라 다수의 얼굴 영상에서도 검출되고, 컴퓨터 화면에서의 사람의 얼굴이 나온 이미지에서도 얼굴 영역과 눈 영역 검출이 가능하다. 이미지 프로세싱 혹은 컴퓨터 비전 기술 특성상 장소에 따라 제대로 검출이 안 되거나 오류가 있을 수 있다. 그 원인으로는 장소마다 조명 상태가 다르기 때문에 적절한 조명 상태가 갖춰진 환경이 아니라면 검출에 제약이 따를 수도 있다. 실험을 수 차례 실행해본 결과, 평균적으로 300lux 아래로 떨어지는 조명 상태가 아닌 경우에는 실시간 얼굴 영역과 눈 영역이 검출이 정상적으로 작동하는 것을 확인하였다.

<그림 2> 얼굴 및 눈 영역 검출과정



<그림 3> 얼굴 영역 및 눈 영역 검출 결과

4. 결론

본 논문에서는 안드로이드 스튜디오를 이용한 얼굴 검출 과정과 그 실행 결과를 제안하였다. 현재 얼굴 영역 검출 뿐만 아닌 눈 영역 검출까지 구현되어 있다. 결과적으로 얼굴 영역과 눈 영역은 높은 검출률을 가진다. 이를 이용해 눈 영역이 일정 시간 검출되지 않았을 경우 또는 얼굴 영역이 앞으로 기울어짐을 인지하는 방법 등을 적용하여 운전 중 졸음 방지 시스템으로 응용 가능성에 대해 구상할 것이다. 하지만 얼굴을 검출해 그것이 어떤 대상인지 인식하는 얼굴 인식이 아닌 검출로만 한정되어 있기 때문에 개선이 필요한 것은 사실이다. 데이터 베이스를 구축해 받은 영상의 얼굴이 누구의 얼굴인지 판단할 수 있는 방법으로 얼굴을 인식하거나 딥러닝 기반의 학습을 통한 고성능 얼굴 인식 기술 등 활발히 연구되고 있다. 그러한 얼굴 인식의 전초 단계인 얼굴 검출을 구현해보고 높은 검출률을 가지는 결과를 얻은데 그 의미가 크다.

참고문헌

[1] 황지휘, 정강훈, 이민형, 문현준, 안드로이드 환경의 MCT 기반 얼굴검출 알고리즘 성능 향상 연구, 2014
[2] 이호훈, 주혜진, 윤보미, 전명근, 안드로이드 기반의 얼굴 인식 시스템, 2014
[3] 최성필, 주일양, 문현준, 안드로이드 환경을 위한 얼굴 검출 시스템 최적화 연구, 2011
[4] https://opencv.org/
[5] 김동현, 임재현, 김대희, 김태경, 백준기, 실시간 영상에서 피부색상 정보와 Haar-Like Feature를 이용한 얼굴 검출 및 추적, 2009