

Haar-like 특징을 이용한 고성능 보행자 및 차량 인식 회로 설계

김수진[†] · 박상균^{**} · 이선영^{***} · 조경순^{****}

요약

본 논문은 Haar-like 특징을 이용한 고성능 보행자 및 차량 인식 회로를 제안한다. 제안하는 회로는 영상의 매 프레임 마다 슬라이딩 윈도우를 적용하여 Haar-like 특징을 추출하고 보행자 및 차량을 인식한다. Haar-like 특징 추출 회로는 슬라이딩 윈도우 당 200개의 Haar-like 특징을 추출하며, 추출된 특징들은 AdaBoost 인식 회로에서 사용된다. 제안하는 회로는 속도 향상을 위해 병렬 회로 구조를 적용하였으며 두 개의 슬라이딩 윈도우가 동시에 보행자 또는 차량을 인식한다. 제안하는 고성능 보행자 및 차량 인식 회로는 Verilog HDL로 설계하였으며 130nm 표준 셀 라이브러리를 이용하여 게이트 수준의 회로로 합성하였다. 합성된 회로는 1,388,260개의 게이트로 구성되며 최대 동작 주파수는 203MHz이다. 제안하는 회로는 640x480 영상을 초당 약 47.8장 처리할 수 있기 때문에 보행자와 차량을 실시간으로 인식하기 위해 사용될 수 있다.

키워드 : Haar-like 특징, 아다부스트, 보행자 및 차량 인식, 실시간 처리

Design of High-performance Pedestrian and Vehicle Detection Circuit using Haar-like Features

Soojin Kim[†] · Sangkyun Park^{**} · Seonyoung Lee^{***} · Kyeongsoon Cho^{****}

ABSTRACT

This paper describes the design of high-performance pedestrian and vehicle detection circuit using the Haar-like features. The proposed circuit uses a sliding window for every image frame in order to extract Haar-like features and to detect pedestrians and vehicles. A total of 200 Haar-like features per sliding window is extracted from Haar-like feature extraction circuit and the extracted features are provided to AdaBoost classifier circuit. In order to increase the processing speed, the proposed circuit adopts the parallel architecture and it can process two sliding windows at the same time. We described the proposed high-performance pedestrian and vehicle detection circuit using Verilog HDL and synthesized the gate-level circuit using the 130nm standard cell library. The synthesized circuit consists of 1,388,260 gates and its maximum operating frequency is 203MHz. Since the proposed circuit processes about 47.8 640x480 image frames per second, it can be used to provide the real-time detection of pedestrians and vehicles.

Keywords : Haar-like Features, AdaBoost, Pedestrian and Vehicle Detection, Real-time Processing

1. 서론

지능형 자동차를 위한 보행자 및 차량 인식 시스템을 구현하기 위해서는 실시간으로 인식이 이루어져야 하기 때문에 실시간 처리가 중요하다. 소프트웨어를 이용한 인식 속

도는 컴퓨터의 성능에 영향을 받기 때문에 실시간 처리를 위해서는 고성능 CPU를 탑재한 컴퓨터가 필요하다. 하지만 고성능 CPU를 탑재한 컴퓨터는 전력 소모가 높고 발열이 심하기 때문에 자동차에서 사용하기에 부적합하다. 따라서 본 논문에서는 실시간으로 보행자 및 차량을 인식하기 위한 차량용 고성능 하드웨어 회로 구조 및 설계를 제안한다.

제안하는 고성능 보행자 및 차량 인식 회로는 효율적인 메모리 관리를 통해 데이터를 저장하기 위한 메모리의 크기 및 지연 시간을 줄였다. 또한 병렬 회로 구조를 적용하여 슬라이딩 윈도우 두 개를 동시에 처리하며, 각 슬라이딩 윈도우마다 8개의 Haar-like 특징 값을 동시에 추출한다. 제안

※ 이 논문은 2012년도 한국외국어대학교 교내학술연구비의 지원에 의한 것임.

[†] 준회원: 한국외국어대학교 전자공학과 박사과정

^{**} 정회원: 한국외국어대학교 전자공학과 석사과정

^{***} 정회원: 전자부품연구원 SoC플랫폼센터 선임연구원

^{****} 정회원: 한국외국어대학교 전자공학과 정교수

논문접수: 2012년 3월 2일

수정일: 1차 2012년 4월 19일

심사완료: 2012년 4월 24일

* Corresponding Author: Kyeongsoon Cho(kscho@hufs.ac.kr)

하는 회로는 AXI (Advanced eXtensible Interface) 및 APB(Advanced Peripheral Bus) 버스를 이용한 데이터 전송이 가능하도록 설계되었기 때문에 AMBA(Advanced Micro controller Bus Architecture) 3.0 규격을 이용하는 다른 시스템에도 쉽게 적용되어 사용될 수 있다.

본 논문에서 제안하는 고성능 보행자 및 차량 인식 회로는 Verilog HDL(Hardware Description Language)을 사용하여 구현하였다. 또한 Synopsys사의 Design Compiler를 통해 130nm 표준 셀 라이브러리를 이용하여 게이트 수준의 회로로 합성하였다. 합성된 회로는 1,388,260개의 게이트로 구성되며 최대 동작 주파수는 203MHz이고, 초당 약 47.8장의 영상을 처리할 수 있다.

2. 보행자 및 차량을 인식하기 위한 방법

제안하는 회로가 보행자 및 차량 인식을 위해 사용한 방법은 Haar-like 특징 추출[1] 및 AdaBoost 인식 방법[2]이다. 본 절에서는 이 두 가지 방법에 대해 요약하여 설명하고자 한다.

2.1 Haar-like 특징 추출

보행자 및 차량을 효율적으로 인식하기 위한 연구들이 많이 발표되고 있다[3-6]. 보행자 및 차량을 인식하기 위해 대칭, 모서리, 텍스처, 음영, 색상 그리고 에지 등을 이용하여 특징 값을 추출하는 방법들이 많이 사용되고 있다. 대칭, 모서리, 텍스처를 이용한 방법은 복잡한 환경에서 인식을 실패할 확률이 높고, 음영이나 색상을 이용한 방법은 날씨와 빛의 영향을 많이 받기 때문에 인식률이 좋지 못하다. 반면 에지 특성을 이용한 방법은 파라미터 수에 따라 성능이 달라지는 단점이 있지만 다른 방법들 보다 인식률이 높은 장점이 있다[3][4]. Haar-like 특징 추출 방법은 에지 특성을 이용한 대표적인 방법이다.

2.1.1 Haar-like 특징

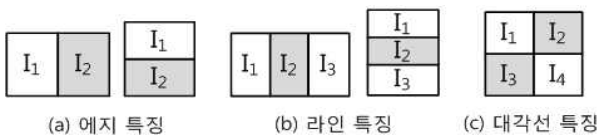


그림 1. Haar-like 특징의 종류
Fig. 1. Types of Haar-like features

Paul Viola가 제안한 Haar-like 특징 추출 방법에는 그림 1과 같이 에지 특징, 라인 특징, 대각선 특징 등이 있으며, 이러한 특징들은 영상에서 인식할 대상의 특징 값을 추출하기 위해 사용된다. 특징 값은 그림 1의 $I_1 \sim I_4$ 영역에 의한 변화량으로, 각 영역에 대한 적분 값의 차이를 계산함으로써 추출될 수 있다. 예를 들어 그림 1(a)의 Haar-like 에지 특징을 추출하기 위해서는 I_1 영역과 I_2 영역의 차이를 계산한다.

2.1.2 적분 이미지

Haar-like 특징 값을 추출하기 위해 각 영역 내의 모든 화소 값을 이용하여 적분 값을 계산하는 것보다 특정 위치마다 미리 누적된 적분 이미지를 이용하여 적분 값을 계산하는 것이 빠른 연산을 위해 더욱 효과적이다. 그림 2는 적분 이미지를 이용하여 특정 영역에 대한 적분 값을 계산하는 방법을 예로 나타낸다. 그림 2(a)의 원본 이미지 D영역에 대한 적분 값을 구하기 위해 각 화소 위치마다 미리 계산된 적분 이미지를 이용하면 계산을 쉽고 빠르게 할 수 있다. 그림 2(b)에 나타난 $P_1 \sim P_4$ 는 각각 그림 2(a)의 A영역, (A+B)영역, (A+C)영역, (A+B+C+D)영역에 대한 적분 값을 나타낸다. D영역에 대한 적분 값 I_D 는 미리 계산된 적분 값 $P_1 \sim P_4$ 를 이용하면 그림 2(b)에 나타난 연산식과 같이 쉽게 계산될 수 있다.

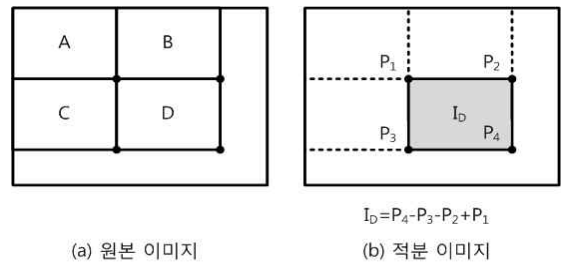


그림 2. 적분 이미지를 이용한 적분 값 계산
Fig. 2. Example of calculating integral image

2.2 AdaBoost를 이용한 인식

Y. Freund와 R. Schapire가 제안한 AdaBoost 알고리즘은 인식률을 높이기 위해 여러 개의 약 분류기를 결합하여 한 개의 강한 인식기를 만드는 것이다. AdaBoost 알고리즘은 간단하고 효율적이기 때문에 속도가 빨라 실시간 검출에 주로 사용된다[5][6].

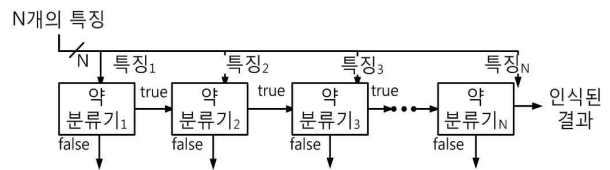


그림 3. AdaBoost 인식기
Fig. 3. AdaBoost classifier

그림 3은 N개의 약 분류기를 결합한 AdaBoost 인식기를 나타낸다. 특징 추출 과정에서 추출된 N개의 특징들은 각각의 약 분류기에 적용되며, AdaBoost를 이용한 인식은 첫 번째 약 분류기에서부터 시작된다. 첫 번째 약 분류기의 결과가 'false'이면 AdaBoost 인식기는 인식 과정을 멈춘다. 첫 번째 약 분류기의 결과가 'true'이면 두 번째 약 분류기가 인식을 수행하고, 이러한 과정은 N개의 약 분류기까지 반복된다. AdaBoost 인식기는 마지막 N번째까지의 분류기 결과가 'true'인 경우에 대해서만 인식이 성공했다고 판단한다.

3. 고성능 보행자 및 차량 인식 회로 설계

지능형 자동차를 위한 보행자 및 차량 인식 시스템에서 실시간 처리는 필수적인 조건이다. 인텔의 OpenCV 라이브러리는 PC를 기반으로 하는 시스템에서 빠른 물체 인식 처리가 가능하게 한다. 하지만 소프트웨어를 이용한 실시간 인식은 전력 소모가 높고 시스템의 크기가 커야 하기 때문에 지능형 자동차나 휴대용 단말 등과 같은 임베디드 시스템에 적용하기 어렵다. 따라서 지능형 자동차에 적용하여 보행자 및 차량을 실시간으로 인식하기 위해서는 전력 소모가 낮고 크기가 작은 하드웨어 시스템에 대한 연구 및 개발이 필요하다.

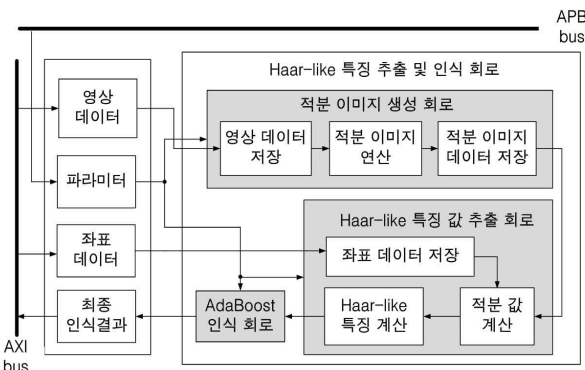


그림 4. 제안하는 고성능 보행자 및 차량 인식 회로
Fig. 4. Proposed pedestrian and vehicle detection circuit

그림 4는 본 논문에서 제안하는 고성능 보행자 및 차량 인식 회로의 전체적인 구조를 나타내며, 그림과 같이 적분 이미지 생성 회로, Haar-like 특징 값 추출 회로, AdaBoost 인식 회로로 구성된다. 제안하는 회로는 인식을 위해 필요한 데이터를 AXI 및 APB 버스를 통해 전송 받는다. 처리하는 영상 및 슬라이딩 윈도우의 크기 등은 APB 버스를 통해 입력되고, AdaBoost 인식기에서 사용되는 파라미터 및 입력 영상 데이터 등은 AXI 버스를 통해 입력된다. 보행자 및 차량에 대한 최종 인식 결과는 AXI 버스를 통해 출력된다.

본 논문에서는 보행자 및 차량을 인식하기 위해 640×480 크기의 입력 영상을 슬라이딩 윈도우 단위로 처리하며, 보행자를 인식하기 위한 슬라이딩 윈도우의 크기는 48×96이고 차량을 인식하기 위한 슬라이딩 윈도우의 크기는 64×64이다. 또한 각 슬라이딩 윈도우의 이동 크기로 4, 6, 8 화소 단위를 고려하였으며, 한 번에 두 개의 슬라이딩 윈도우를 처리할 수 있다. 보행자 및 차량을 인식하기 위해 사용하는 Haar-like 특징의 개수는 슬라이딩 윈도우 당 200개이며, 실시간 처리를 위해 8개의 특징 값을 한 번에 추출하여 사용한다. 제안하는 회로에서 사용된 Haar-like 특징의 종류는 OpenCV를 이용한 훈련 과정에서 결정되며, 훈련 과정에서는 인식하는 대상에 따라 서로 다른 종류의 Haar-like 특징이 결정되기 때문에 보행자와 차량을 인식하기 위해 사용된 Haar-like 특징의 종류는 다르다.

3.1 적분 이미지 생성 회로

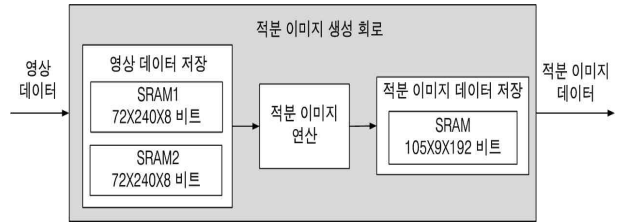


그림 5. 적분 이미지 생성 회로
Fig. 5. Integral image generator circuit

제안하는 적분 이미지 생성 회로는 그림 5와 같다. 본 논문에서 제안하는 보행자 및 차량 인식 회로는 한 번에 두 개의 슬라이딩 윈도우를 처리한다. 따라서 두 개의 슬라이딩 윈도우 크기를 고려한 양만큼의 영상 데이터 및 적분 이미지 데이터를 메모리에 저장한다.

3.1.1 영상 데이터 저장

640×480 크기의 영상 한 프레임을 모두 저장하기 위해서는 2,457,600비트(8비트×640화소×480화소) 크기의 메모리가 필요하므로 한 프레임 전체를 저장하는 것은 비효율적이다. 따라서 본 논문에서는 한 프레임 전체에 대한 영상 데이터를 저장하는 대신 필요한 만큼의 영상 데이터만을 저장하여 사용한다. 제안하는 회로는 한 번에 두 개의 슬라이딩 윈도우를 처리하기 때문에 슬라이딩 윈도우가 640×480 영상을 세로로 두 번 스캔할 수 있는 양 만큼의 영상 데이터를 메모리에 저장한다. 따라서 본 논문에서 영상 데이터를 저장하기 위해 사용한 메모리의 크기는 72×480×8(72화소×480화소×8비트)이다. 여기에서 72화소는 슬라이딩 윈도우의 최대 가로 크기인 64화소와 슬라이딩 윈도우의 최대 이동 단위인 8화소를 고려한 것이다. 본 논문에서는 실시간 처리를 위해 72×480×8 크기의 영상을 두 개의 그룹으로 나누어서 번갈아 저장한다. 따라서 각 메모리 그룹의 크기는 72×240×8 비트이며, 한 개의 그룹에 저장된 영상 데이터가 적분 이미지 연산을 위해 사용되는 동안 나머지 그룹에는 새로운 영상 데이터가 저장된다.

3.1.2 적분 이미지 연산

적분 이미지는 식 (1)~(3)을 이용하여 계산된다. 식에서 $O(x,y)$ 는 현재 위치의 화소 값, $I(x,y)$ 는 현재 위치에 대한 적분 값, 그리고 $C(x,y)$ 는 현재 위치까지 누적된 열의 합을 의미한다. 식 (1)은 현재 위치에 대한 적분 값이 현재 위치까지의 모든 화소들의 합과 같다는 것을 나타내고, 식 (2)는 현재 위치까지의 열의 합이 이전 열의 합에 현재 화소 값을 더한 것과 같다는 것을 나타낸다. 식 (3)은 현재 위치에 대한 적분 값이 바로 왼쪽 화소에 대한 적분 값에 현재 위치까지의 열의 합을 더한 것과 같다는 것을 나타낸다. 제안하는 회로는 식 (2), (3)을 이용하여 입력 영상의 각 화소 데이터에 대한 누적 연산을 통해 적분 이미지를 생성한다.

$$I(x,y) = \sum_{x' \leq x, y' \leq y} O(x',y') \quad (1)$$

$$C(x,y) = C(x,y-1) + O(x,y) \quad (2)$$

$$I(x,y) = I(x-1,y) + C(x,y) \quad (3)$$

3.1.3 적분 이미지 데이터 저장

화소 당 8비트의 크기를 갖는 640×480 영상 전체에 대한 적분 이미지는 최대 27비트로 표현되며, 이를 저장하기 위해서는 최대 641×481×27비트(641화소×481화소×27비트) 크기의 메모리가 필요하다. 하지만 본 논문에서는 한 번에 두 개의 슬라이딩 윈도우를 처리하기 때문에 두 개의 슬라이딩 윈도우 양 만큼의 데이터를 저장할 수 있는 크기의 메모리를 사용한다. 제안하는 적분 이미지 연산 회로는 슬라이딩 윈도우의 최대 가로 크기 64화소 및 최대 세로 크기 96화소와 슬라이딩 윈도우의 최대 이동 단위인 8화소를 고려하여 가로 72 화소, 세로 105 화소 크기에 대한 적분 이미지를 계산하여 메모리에 저장한다. 72×105 크기의 영상에 대한 적분 이미지의 각 데이터는 최대 24비트로 표현된다. 본 논문에서 계산된 적분 이미지는 105개의 9×192비트 메모리에 저장된다. 본 논문에서 제안하는 회로는 한 번에 두 개의 슬라이딩 윈도우를 처리하며, 슬라이딩 윈도우 당 한 번에 8 개의 Haar-like 특징 값을 계산한다. 따라서 메모리에 저장되는 각 데이터의 크기는 192비트(24비트×8개)이며 모두 9 개(72화소/8개)의 주소에 나누어 저장된다.

3.2 Haar-like 특징 값 추출 회로

제안하는 특징 값 추출 회로는 그림 6과 같다. 적분 이미지에 대한 좌표 데이터 및 Haar-like 특징의 종류는 AXI 버스를 통해 입력되며, 한 번에 8개의 Haar-like 특징 값을 추출하기 위해 8개의 25×111비트 크기의 메모리에 저장된다. 여기에서 111비트는 Haar-like 특징의 종류(3비트)와 Haar-like 특징을 계산하기 위해 필요한 좌표 데이터(9개×12비트)를 고려한 것이다. 본 논문에서 제안하는 회로는 보행자 및 차량을 인식하기 위해 슬라이딩 윈도우 당 200개의 Haar-like 특징 값을 추출하여 사용한다. 따라서 한 번에 8개의 Haar-like 특징 값을 추출하기 위해서 각 특징의 종류 및 좌표 데이터는 25개의 주소에 나누어 메모리에 저장된다. 제안하는 회로는 한 번에 8개의 특징 값을 추출하기

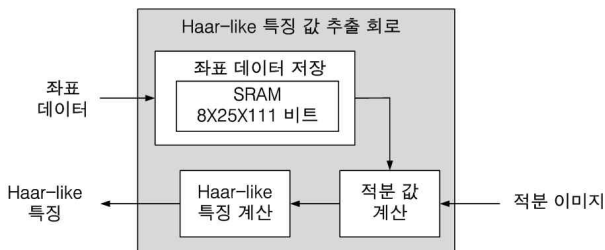


그림 6. Haar-like 특징 값 추출 회로
Fig. 6. Haar-like feature extractor circuit

위해 적분 값 계산 및 Haar-like 특징 계산 회로에 병렬 구조를 적용하였다.

3.2.1 적분 값 계산

그림 7은 특정 영역에 대한 적분 값을 구하기 위해 좌표 데이터에 따른 적분 이미지 데이터를 선택하는 과정을 나타낸다. 제안하는 회로는 먼저 Y좌표에 따라 105개의 SRAM 중에서 하나의 SRAM을 선택한다. X좌표는 Y좌표를 이용하여 선택된 SRAM에서 하나의 24비트 적분 이미지 데이터를 선택하기 위해 사용된다.

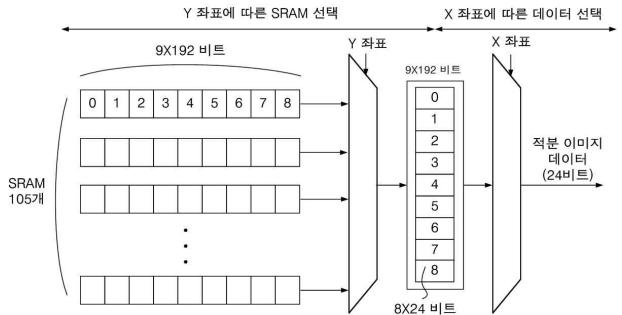


그림 7. 적분 이미지 데이터 선택
Fig. 7. Selection of an integral image data

그림 7과 같은 과정을 통해 선택된 적분 이미지 데이터는 그림 2(b)에 나타난 연산을 통해 그림 1의 I₁~I₄ 영역에 대한 적분 값을 계산하기 위해 사용된다. 그림 2(c)에 나타난 대각선 Haar-like 특징을 추출하기 위해서는 4개의 적분 값이 필요하기 때문에 제안하는 회로는 최대 9개의 (X, Y) 좌표 데이터를 이용하여 적분 값을 계산한다.

3.2.2 Haar-like 특징 계산

그림 1의 I₁~I₄ 영역에 대한 적분 값은 Haar-like 특징 값을 계산하기 위해 사용되며, 각 특징의 종류에 따라 식 (4)~(6)의 연산을 이용한다. 본 논문에서 제안하는 Haar-like 특징 값 추출 회로는 슬라이딩 윈도우 당 한 번에 8개의 Haar-like 특징 값을 추출한다. 또한 고성능으로 보행자 및 차량을 인식하기 위해 슬라이딩 윈도우 두 개를 동시에 처리하므로, 한 번에 16개의 Haar-like 특징 값을 추출한다.

$$\text{예지 특징 값: } I_1 - I_2 \quad (4)$$

$$\text{라인 특징 값: } I_1 - (I_2 \gg 1) + I_3 \quad (5)$$

$$\text{대각선 특징 값: } I_1 - I_2 - I_3 + I_4 \quad (6)$$

3.3 AdaBoost 인식 회로

그림 8은 본 논문에서 제안하는 AdaBoost 인식 회로를 나타낸다. 각 분류기의 판별 상수, 신뢰도 등은 AdaBoost 훈련 과정에서 결정된다. 제안하는 회로에서 사용된 약 분

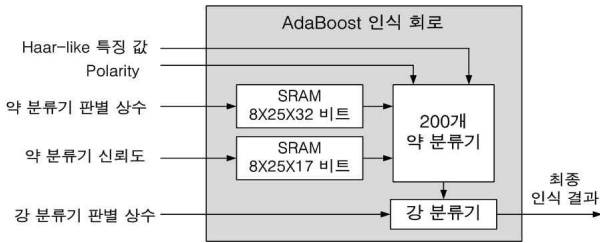


그림 8. AdaBoost 인식 회로
Fig. 8. AdaBoost classifier circuit

류기의 개수는 슬라이딩 윈도우 당 추출하는 Haar-like 특징의 개수와 같은 200개이다. 각 분류기를 위한 파라미터들은 버스를 통해 입력되어 내부 메모리에 저장된다. 제안하는 회로는 여러 개의 약 분류기를 하나의 스테이지로 처리한다. AdaBoost 인식 과정을 여러 스테이지로 나누어 처리하면 각 스테이지의 인식 결과에 따라 인식 과정을 중간에 멈출 수 있기 때문에 속도를 빠르게 하는 장점이 있다. 하지만 모든 특징을 이용하지 않기 때문에 인식률이 떨어질 수 있다는 단점이 있다. 따라서 제안하는 회로는 모든 특징을 이용하여 보행자 및 차량을 인식하기 위해 인식 과정을 한 개의 스테이지로 처리하며, 스테이지를 나누지 않아도 실시간 처리가 가능하다. 제안하는 회로에서는 각 약 분류기의 결과가 판별 상수보다 크면 약 분류기의 결과 값을 누적시킨다. 200개의 약 분류기에 대한 누적 값이 강 분류기 판별 상수보다 작으면 보행자 및 차량에 대한 인식이 성공했다고 판단하여 해당 슬라이딩 윈도우의 좌표를 인식 결과로 출력한다.

4. 실험 결과

본 논문에서 제안하는 고성능 보행자 및 차량 인식 회로는 Verilog HDL를 사용하여 RTL로 구현하였으며, Cadence사의 NC-Verilog 및 AMBA 3.0 버스 환경을 이용하여 회로의 동작을 검증하였다. 또한 Synopsys사의 Design Compiler를 통해 130nm 표준 셀 라이브러리를 이용하여 게이트 수준의 회로로 합성하였다. 제안하는 회로는 640×480 크기의 영상을 슬라이딩 윈도우 단위로 처리하여 보행자 및 차량을 인식한다. 보행자 및 차량을 인식하기 위한 슬라이딩 윈도우의 크기는 각각 48×96과 64×64이며 이동 단위는 4, 6, 8 화소이다.

표 1은 제안하는 회로의 성능 및 합성 결과를 나타낸다. 제안하는 회로는 슬라이딩 윈도우 두 개를 동시에 처리하는 병렬 회로 구조를 적용하였으며 각 슬라이딩 윈도우에서 한 번에 8개의 Haar-like 특징 값을 추출한다. 슬라이딩 윈도우의 이동 단위를 4화소 단위로 적용할 경우 한 프레임 당 처리해야 하는 슬라이딩 윈도우의 최대 개수는 35,996개이다. 본 논문에서 제안하는 회로는 640×480 영상 한 장과 640×480 영상을 80% 크기로 연속적으로 축소시킨 서로 다른 크기의 영상 10장(9단계의 스케일링 다운 적용)을 모두

표 1. 제안하는 회로의 성능 및 합성 결과
Table 1. Performance of the proposed circuit

| | | |
|------------------------|---------------|---------|
| 영상 크기 | 640x480 | |
| 약 분류기 개수 | 200 | |
| 슬라이딩 윈도우 이동 단위 | 4, 6, 8 화소 | |
| 한 번에 처리하는 슬라이딩 윈도우 개수 | 2 | |
| 프레임 당 처리하는 슬라이딩 윈도우 개수 | 35,996 | |
| 프레임 당 필요한 사이클 수 | 4,711,924 사이클 | |
| 슬라이딩 윈도우 크기 | 48x96 | 64x64 |
| 처리 속도 | 47.8fps | 43.2fps |
| 최대 동작 주파수 | 203MHz | |
| 지연 시간 | 4.91ns | |
| 게이트 수 | 1,388,260 | |

처리하는데 총 4,711,924 클럭 사이클이 소요된다. 제안하는 회로를 130nm 표준 셀 라이브러리로 합성하였을 때 최대 동작 주파수는 203MHz이고 회로의 크기는 1,388,260 게이트이다. 본 논문에서 제안하는 고성능 보행자 및 차량 인식 회로는 9단계의 스케일링 다운을 적용한 640×480 영상을 초당 약 47.8장 처리할 수 있다. 그림 9는 제안하는 회로의 보행자 인식 실험 결과를 나타내며, 영상 내에서 보행자가 인식되면 그림과 같이 해당 슬라이딩 윈도우의 좌표를 인식 결과로 출력한다.

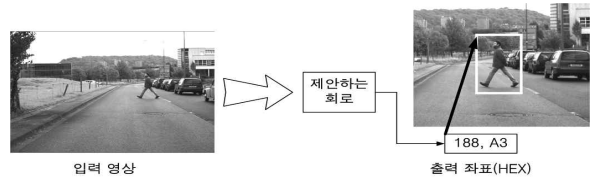


그림 9. 보행자 인식 결과
Fig. 9. Pedestrian detection result using proposed circuit

표 2. 다른 회로와의 성능 비교
Table 2. Comparison results with other circuits

| | [5] | [6] | 본 논문 |
|----------------|---------|---------|----------------|
| 입력 영상 크기 | 640×480 | 640×480 | 640×480 |
| 스케일링 정도 | - | 0.5 | 0.8 |
| 스테이지 수 | 22 | 20 | 1 |
| 특징 개수 | 2,135 | 1,715 | 200 |
| 슬라이딩 윈도우 크기 | 20×20 | 80×60 | 64×64 48×96 |
| 슬라이딩 윈도우 수 | 1 | 1 | 2 |
| 슬라이딩 윈도우 이동 단위 | 22 | 5 | 4, 6, 8 |
| 동작주파수 | - | 100MHz | 203MHz |
| 처리 속도 | 6.84fps | 64fps | 47.8fps |

표 2는 본 논문에서 제안하는 회로를 다른 논문의 회로들과 비교한 것이다. 얼굴을 인식하기 위한 논문 [5]의 회로는 2,135개의 Haar-like 특징을 이용하여 640x480 크기의 영상을 초당 6.84장 처리하며, 차량 및 보행자를 인식하기 위한 논문 [6]의 회로는 1,715개의 Haar-like 특징을 이용하여 320x240 크기의 영상을 초당 64장 처리한다. 본 논문에서 제안하는 회로는 64x64 또는 48x96 크기의 슬라이딩 윈도우를 이용하여 640x480 크기의 영상을 처리하며, 두 개의 슬라이딩 윈도우를 동시에 처리하기 때문에 한 번에 400개의 특징을 이용한다. 논문 [5], [6]의 회로들과 본 논문에서 제안하는 회로의 성능 비교를 위해 입력 영상과 슬라이딩 윈도우의 크기, 스케일링 정도, 한 번에 이용하는 Haar-like 특징 개수 등을 동일하게 적용하면, 본 논문에서 제안하는 회로는 640x480 크기의 영상을 초당 약 48장 처리할 수 있는 반면, 같은 조건에서 논문 [5]와 논문[6]의 회로는 각각 초당 13장과 14장을 처리할 수 있다. 따라서 본 논문에서 제안하는 회로는 Haar-like 특징의 수, 슬라이딩 윈도우의 크기 및 이동 단위 등을 종합적으로 고려했을 때 다른 회로들보다 성능이 더 좋다.

5. 결 론

본 논문은 Haar-like 특징과 AdaBoost 인식을 이용한 고성능 보행자 및 차량 인식 회로를 제안한다. 제안하는 회로는 효율적인 메모리 관리를 통해 데이터를 저장하기 위한 메모리의 크기 및 지연 시간을 줄였다. 또한 병렬 회로 구조를 적용하여 슬라이딩 윈도우 두 개를 동시에 처리한다. 보행자 및 차량을 추출하기 위해 사용하는 Haar-like 특징의 개수는 슬라이딩 윈도우 당 200개이며, 실시간 처리를 위해 8개의 Haar-like 특징 값을 동시에 추출한다. 제안하는 회로는 9단계의 스케일링 다운을 적용한 640x480 영상을 초당 약 48 프레임을 처리할 수 있기 때문에 실시간으로 보행자와 차량을 인식하기 위해 사용될 수 있다.

참 고 문 헌

[1] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Vol.1, pp.511-518, Dec., 2001.

[2] Y. Freund and R. E. Schapire, "A Decision-theoretic Generalization of Online Learning and an Applications to Boosting," Journal of Computer Vision and Pattern Recognition, pp.886-893, 2005.

[3] Z. Sun, *et al.*, "On-road Vehicle Detection: A Review." IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.28, No.5, pp.649-711, May, 2006.

[4] P. Dollar, *et al.*, "Pedestrian Detection a Benchmark," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp.304-311, Jun., 2009.

[5] Junguk Cho, *et al.*, "Parallelized Architecture of Multiple Classifiers for Face Detection," in Proceedings of IEEE International Conference on Application-specific Systems, Architecture and Processors, pp.75-82, Jul., 2009.

[6] C. Kyrkyou and T. Theocharides, "A Flexible Parallel Hardware Architecture for AdaBoost-based Real-time Object Detection," IEEE Trans. on VLSI Systems, pp.1034-1047, Jun., 2011.



김 수 진

e-mail : ksjsky9888@hufs.ac.kr
 2007년 한국외국어대학교 전자공학과(학사)
 2009년 한국외국어대학교 전자공학과
 (공학석사)
 2009년~현 재 한국외국어대학교
 전자공학과 박사과정
 관심분야: SoC 설계 등



박 상 군

e-mail : sirius@hufs.ac.kr
 2010년 한국외국어대학교 전자공학과(학사)
 2010년~현 재 한국외국어대학교
 전자공학과 석사과정
 관심분야: SoC 설계 등



이 선 영

e-mail : drleesy@keti.re.kr
 1998년 한국외국어대학교 전자공학과(학사)
 2000년 한국외국어대학교 전자공학과
 (공학석사)
 2009년 한국외국어대학교 전자공학과
 (공학박사)

2001년~2006년 (주)이시티 반도체설계연구소 선임연구원
 2009년~현 재 전자부품연구원 SoC플랫폼센터 선임연구원
 관심분야: SoC 설계 등



조 경 순

e-mail : kscho@hufs.ac.kr
 1982년 서울대학교 전자공학과(학사)
 1984년 서울대학교 전자공학과(공학석사)
 1988년 미국 Carnegie Mellon University
 대학원 전기 및 컴퓨터 공학과
 (공학박사)

1988년~1994년 삼성전자(주) 반도체 총괄 선임, 수석 연구원
 1994년~현 재 한국외국어대학교 전자공학과 조교수, 부교수,
 정교수
 관심분야: SoC 설계 등