

차량 검출용 CNN 분류기의 실시간 처리를 위한 하드웨어 설계

A Real-Time Hardware Design of CNN for Vehicle Detection

방 지원*, 정 용 진*

Ji-Won Bang*, Yong-Jin Jeong*

Abstract

Recently, machine learning algorithms, especially deep learning-based algorithms, have been receiving attention due to its high classification performance. Among the algorithms, Convolutional Neural Network(CNN) is known to be efficient for image processing tasks used for Advanced Driver Assistance Systems(ADAS). However, it is difficult to achieve real-time processing for CNN in vehicle embedded software environment due to the repeated operations contained in each layer of CNN. In this paper, we propose a hardware accelerator which enhances the execution time of CNN by parallelizing the repeated operations such as convolution. Xilinx ZC706 evaluation board is used to verify the performance of the proposed accelerator. For 36x36 input images, the hardware execution time of CNN is 2.812ms in 100MHz clock frequency and shows that our hardware can be executed in real-time.

요 약

최근 딥 러닝을 중심으로 빠르게 발전하고 있는 기계학습 분류 알고리즘은 기존의 방법들보다 뛰어난 성능으로 인하여 주목받고 있다. 딥 러닝 중에서도 Convolutional Neural Network(CNN)는 영상처리에 뛰어나 첨단 운전자 보조 시스템(Advanced Driver Assistance System : ADAS)에서 많이 사용되고 있는 추세이다. 하지만 차량용 임베디드 환경에서 CNN을 소프트웨어로 동작시켰을 때는 각 Layer마다 연산이 반복되는 알고리즘의 특성으로 인해 수행 시간이 길어져 실시간 처리가 어렵다. 본 논문에서는 임베디드 환경에서 CNN의 실시간 처리를 위하여 Convolution 연산 및 기타 연산들을 병렬로 처리하여 CNN의 속도를 향상시키는 하드웨어 구조를 제안한다. 제안하는 하드웨어의 성능을 검증하기 위하여 Xilinx ZC706 FPGA 보드를 이용하였다. 입력 영상은 36x36 크기이며, 동작주파수 100MHz에서 하드웨어 수행시간은 약 2.812ms로 실시간 처리가 가능함을 확인했다.

Key words : Convolutional Neural Network, ADAS, FPGA, Vehicle Detection, Machine Learning

* Dept. of Electronics & Communications Engineering, KwangWoon University

★ Corresponding author

vijeong@kw.ac.kr / 02-940-5551

※ Acknowledgment

This work was supported in part by the Research Grant of Kwangwoon University in 2015 and the IT R & D program of Ministry of Trade, Industry and Energy (10049192, Development of a Smart Automotive ADAS SW-Soc for a Self-Driving Car)

Manuscript received Nov. 30, 2016; revised Dec. 26, 2016 ; accepted Dec. 28, 2016

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

최근 딥 러닝(Deep learning)을 중심으로 빠르게 발전하고 있는 기계학습 분류 알고리즘은 기존의 방법들보다 뛰어난 성능으로 인하여 주목받고 있다. 딥 러닝이란 인간의 뇌가 사물을 구분할 때의 정보처리 방식을 기계에 적용한 형태인 인공신경망을 기반으로 대량의 데이터를 통하여 컴퓨터가 스스로 학습(training)하는 기계학습 알고리즘의 한 분야이고 음성 신호 처리, 영상 신호 처리, 데이터 마이닝 등 여러 분야에서 활용되고 있다.[1] 특히, 딥 러닝 알고리즘 중 하나인 Convolutional Neural Network(CNN)는 영상 신호 처리 분야에서 최근 각광을 받고 있는 분류기 중 하나이다. CNN은 사람의 시각 처리과정을 모방해 데이터 처리에 적합한 구조로 개발된 인공신경망으로 다른 딥 러닝 구조들과 비교해도 영상, 음성 신호 처리 분야 모두에서 좋은 성능을 보이며, Support Vector Machine(SVM)[2], Decision Tree[3], AdaBoost[4] 등 다른 기계학습 알고리즘과 비교하여도 우수한 성능을 나타내며 차세대 핵심 분류 알고리즘으로 떠오르고 있다.[5]

CNN, SVM, AdaBoost와 같은 기계학습을 이용한 분류 알고리즘들은 영상 처리 분야 중에서 물체 인식이나 장면 해석과 같은 컴퓨터 비전 기술로 자주 활용된다. 이러한 알고리즘들을 이용한 대표적인 예시로 첨단 운전자 보조 시스템(Advanced Driver Assistance System : ADAS)을 들 수 있다. ADAS에서는 보행자 검출(Pedestrian Detection : PD)[6], 차량 검출(Vehicle Detection : VD), 교통 표지판 검출(Traffic Sign Recognition : TSR)[7], 전방 차량 충돌 시스템(Forward Collision Warning System : FCWS)[8]과 같이 물체 인식을 활용하는 영상 처리 알고리즘이 많기 때문에, 현재 차량용 영상 처리 분야에서 분류기 알고리즘에 대한 연구가 활발히 진행되고 있다. 특히, 학습용 데이터의 양이 풍부해짐에 따라 Haar-like feature나 Histogram of Oriented Gradients(HOG) 등과 같은 특징점과 다른 분류 알고리즘을 사용하는 것보다 CNN을 이용하는 것이 물체 인식에 더 뛰어난다는 것이 최근의 연구들에서 증명되고 있다.

하지만 차량용 전장 시스템과 같은 임베디드

환경에서 CNN을 수행했을 때, 각 Layer마다 반복되는 Convolution 연산으로 인하여 수행시간이 오래 걸려 실시간 처리가 어렵다. 본 논문에서는 임베디드 환경에서 CNN의 실시간 처리를 위하여 CNN 알고리즘에서 연산량의 86% 이상을 차지하는 Convolution 연산을 포함한 반복 연산들을 병렬로 처리하여 CNN의 속도를 향상시키는 하드웨어 구조를 제안한다. 제안하는 하드웨어의 성능을 검증하기 위하여 Xilinx ZC706 보드를 이용하여 차량 검출 알고리즘에 대해 실험하였다. 실험에 사용된 입력 영상은 36x36 크기이며, 소프트웨어와 하드웨어 환경 모두 CalTech99 데이터셋에 대하여 99.69%의 검출률을 보였다. 동작주파수 100MHz에서 하드웨어 수행 시간은 약 2.812ms로 실시간 처리가 가능함을 확인하였으며, 이는 동일 보드의 소프트웨어 환경에서 측정된 506.7ms보다 180배 빠르다.

각 장의 내용은 다음과 같다. 2장에서는 다른 차량 검출 알고리즘과 CNN에 대한 관련 연구에 대하여 서술한다. 3장에서는 CNN 알고리즘 및 본 논문에서 제안하는 CNN 구조에 대해 간단히 설명하고, 4장에서는 구현한 하드웨어의 구조 및 성능 분석, 5장에서는 실험환경 및 결과, 끝으로 본 논문의 결론을 서술한다.

II. 관련 연구

CNN을 사용하는 분야는 앞서 서술한 것처럼 광범위하다. 이 장에서는 차량 검출을 위한 선행 연구들에 대하여 알아보고, 이와 비교하여 왜 CNN이 우수한 지 기술한 논문들에 대한 소개를 진행한다. 또한 CNN의 역사에 대해 알아보고 현재 이를 토대로 진행되고 있는 영상인식 분류기용 CNN 관련 연구 그리고 하드웨어 구현 관련 연구에 대하여 알아본다.

1. 차량 검출 알고리즘

CNN은 네트워크 안에서 자체적으로 학습을 통해 특징을 만들어 분류(classification)을 함께 하는 구조를 가지고 있지만, 다른 기계학습 알고리즘들은 특징을 이용하여 분류를 수행한다. 따라서 차량을 검출하기 위해서 차량의 특징을 잘 찾아낼 수 있는 특징점 추출기를 이용하고 추출한 특

징점을 이용하여 물체가 차량이 맞는지 분류기를 통해 확인한다. 차량을 검출하기 위한 대표적인 방법에는 PCA+SVM[9], Gabor Filter+SVM[10], Wavelet+SVM[11], Wavelet+Gabor+SVM[12], Haar-like + Cascaded Adaboost[13][14], Haar-like + Adaboost[15], Haar-like + AdaBoost + SVM[16]을 이용한 방법들이 있다. 하지만 해당 알고리즘들은 고정적인 특징을 이용한다는 공통적인 요소가 있고, 이는 유기적으로 특징을 학습하는 CNN보다 검출 성능이 낮다는 것이 ImageNet Large Scale Visual Recognition Challenge(ILSVRC)[17]를 통해 증명되고 있다.

2. Convolutional Neural Network

CNN은 1989년 필기체 우편 번호 인식을 위한 프로젝트로 Yann LeCun[18]에 의해 처음 소개가 되었다. LeNet-5[19]는 최초의 성공적인 CNN으로 주로 우편 번호나 숫자 등을 인식하기 위해서 개발되었다. CNN의 구조는 Convolution과 Pooling, Fully Connected Layer를 어떻게 구성하느냐에 따라 성능이 많이 달라진다. 최근 2010년부터 매년 열리는 컴퓨터 비전 분야의 성능 우열을 가리기 위한 대회인 ILSVRC에서는 CNN이 압도적으로 높은 성능을 보이며 영상처리 분야에서의 강세를 보여주고 있다. AlexNet[20]은 Alex Krizhevsky에 의해 개발된 CNN모델로 ILSVRC 2012에서 2등과의 압도적인 에러율 차이로 우승을 하여 컴퓨터 비전분야에서 CNN을 유명하게 만들었다. LeNet의 구조와 기본적으로는 유사하지만 기존의 구조와는 조금 다른 여러 개의 Convolution Layer가 연속적으로 이루어져있는 구조로 되어있다. GoogLeNet은 ILSVRC 2014년 대회에서 우승을 한 CNN으로 파라미터 개수를 대폭 줄이는 인셉션 모듈을 제안했다.

3. CNN 관련 연구 / CNN 하드웨어 연구

CNN은 영상처리 분야에서 다른 기계학습과 비교하여 우수한 성능으로 인하여 주목을 받고 있으며, 최근 차량 및 표지판 검출이나 보행자 인식 등 많은 ADAS 관련 연구가 이루어지고 있다. CNN을 이용한 차량 검출에 관한 연구들로 [21]에서는 CNN을 이용하여 차량 앞쪽의 번호판을 인식하며, [22]에서는 Caffe framework를 이용하여

차량 검출에 대한 학습과 분류 하였다. [23]는 저 품질의 교통 단속카메라에서 자동차 검출을 하였다. [24]는 차선과 차량을 찾아내었으며, [25]와 [26]에서는 교통 표지판과 보행자에 대해 검출하였다.

CNN은 컴퓨터 비전 기술이나 다른 영상처리 혹은 음성신호 처리 등 우수한 성능으로 다양한 분야에서 사용하지만, CNN은 엄청난 양의 단순 계산을 하기 때문에 계산량이 많다는 단점을 가지고 있기 때문에 하드웨어 가속기를 만들어 병렬처리, 파이프라이닝을 통해 계산량을 줄이는 연구들이 진행된다.[27][28]

III Convolutional Neural Network

CNN은 Convolution Layer와 Pooling Layer를 반복 수행하여 영상에 대한 특징을 추출하고, 최종적으로 Fully Connected Layer를 통해 분류를 수행하는 모델이다. 기본적으로 네트워크를 구성할 때 Convolution 마스크의 크기, 활성화 함수, Pooling 방법, feature map의 수, Layer의 개수 등을 이용하여 CNN의 구조를 정해주게 된다.

Convolution Layer는 입력 영상에 대해 Convolution 연산을 하여 feature map을 생성하는 단계이다. Convolution 연산에 사용되는 weight는 학습과정에서 얻어지며, 본 논문에서는 5x5 크기의 Convolution 마스크를 이용하여 연산을 하였다. Convolution 연산 후에는 bias를 더한 값에 활성화 함수를 사용한다. 활성화 함수의 종류로는 Linear function, Step function, Sigmoid function, Hyperbolic tangent function 등이 있으나 인공지능망을 학습시킬 때 많이 이용하는 Back-Propagation 알고리즘에서 낮은 Layer로 갈수록 전파되는 에러의 양이 적어지기 때문에 gradient 변화가 거의 없어져 학습이 일어나지 않는 gradient vanishing 현상이 있다. 하지만 Rectified Linear Unit(ReLU)의 도입으로 이 문제가 해결되어 낮은 층까지 학습이 가능해졌으며, ReLU의 성능 또한 다른 활성화 함수보다 뛰어나기 때문에 본 논문에서는 활성화 함수로 ReLU를 사용했다. Convolution 연산을 수행할 때 각각의 feature map은 서로 다른 Convolution 마스크를 가지며, 같은 feature map내에서는 같은

Convolution 마스크가 사용된다. Max Pooling Layer에서는 2x2 필터를 이용하여 필터 내의 4개의 특징 값 중 가장 큰 특징 값만을 선택하는 방법으로 계산과 파라미터의 양을 줄이기 위해 사용된다. 마지막 단계인 Fully Connected Layer에서는 이전 Layer에서 추출된 특징을 이용하여 분류를 수행한다.

본 논문에서 사용한 CNN은 차량 검출용 CNN으로 LeNet-5를 기반으로 네트워크 구조를 변경하여 구성하였다. 네트워크 구조를 결정하기 위해서 다수의 차량 데이터를 이용한 반복 실험을 통해 최적의 네트워크 구조를 선택하였다.

제안하는 네트워크 구조는 총 8개의 Layer로 이루어져 있으며, 반복되는 3개의 Convolution Layer와 Max Pooling Layer 그리고 2개의 Fully Connected Layer로 이루어졌다. 그림 1은 LeNet-5의 구조이며, 그림 2는 본 논문에서 사용한 CNN 구조이다.

IV 하드웨어

이번 장에서는 설계한 하드웨어 블록도 및 동작 과정을 설명한다. CNN 구조에서 Convolution Layer는 수행시간의 대부분을 차지하고 있기 때문에 다른 Layer의 수행 시간을 줄이는 것보다 Convolution Layer의 수행시간을 줄이는 것이 더 효율적이다. 따라서, 제안하는 하드웨어 구조는 Convolution Layer의 수행 시간을 최대한 줄이기 위해서 곱셈기 25개를 사용하여 Convolution 연산이 1 clk에 수행되게 하였으며, 더블 버퍼를 이용하여 연속적으로 계산이 이루어지도록 하였다. 그림 3은 설계한 CNN 하드웨어 가속기의 전체 블록도이며, Convolution 모듈, Max Pooling 모듈, Fully Connected 모듈로 구성되어있다. 동작 과정은 Bus Interface를 통해서 CNN 연산에 필요한 weight와 bias, 입력 영상을 Block Memory(weight memory, bias memory, Input image)에 저장한 후 Convolution1 모듈부터 순차

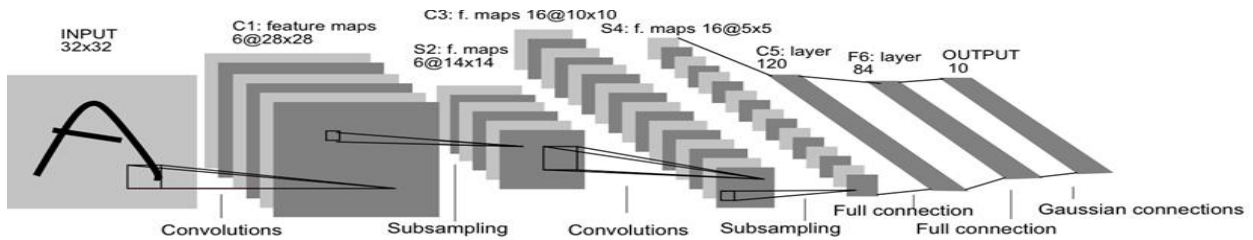


Fig. 1. Architecture of LeNet-5

그림 1. LeNet-5의 구조

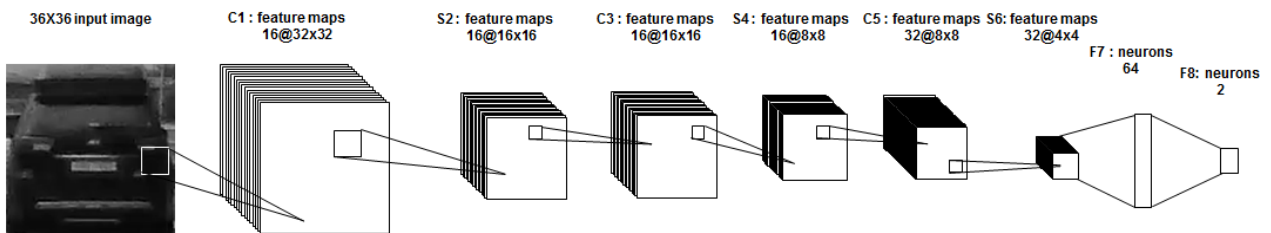


Fig. 2. Proposed CNN structure

그림 2. 본 논문에서 사용한 CNN 구조

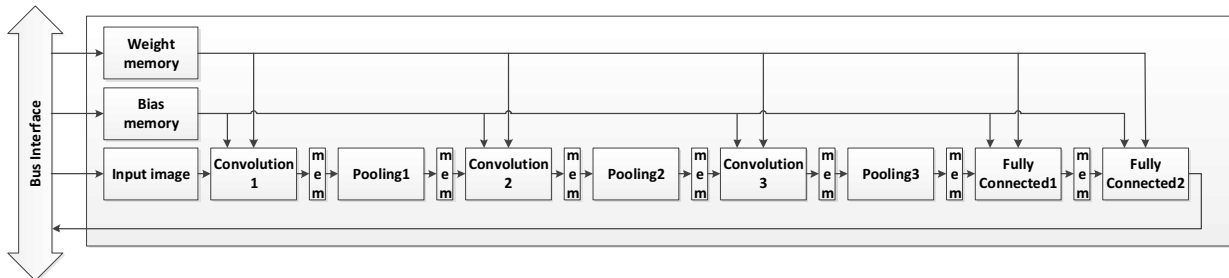


Fig. 3. Block Diagram of CNN Hardware Accelerator

그림 3. CNN 하드웨어 가속기 블록도

적으로 동작하게 된다. Max Pooling은 모든 feature map에 대해 동시에 수행되며 각 모듈의 연산 결과는 Block Memory에 저장된다.

1. Convolution Layer Module

그림 4는 본 논문에서 사용한 Convolution Layer 모듈의 블록도를 나타낸다. Convolution을 수행하는 방법은 입력 버퍼의 크기만 다를 뿐 각 Convolution Layer에서의 계산 과정은 같기 때문에 첫 번째 Convolution Layer에 대해 간략히 설명한다. 첫 번째 Convolution Layer에서 입력 버퍼의 크기는 36x5이며, Convolution 계산을 한 번에 하기 위해서 weight 버퍼의 크기는 5x5로 사용하였다. 처음 입력 버퍼와 weight 버퍼에 해당 버퍼 크기만큼 데이터를 저장한다. 그 후에 입력 버퍼와 weight 버퍼를 Convolution하여 32x1 크기의 feature map 한 줄을 생성하게 되고, 다른 weight 버퍼를 사용하여 다른 feature map 한 줄을 생성한다. 입력 버퍼 하나에 대해서 16개의 5x5 Convolution 마스크를 적용하여 16개의 feature map 한 줄을 생성하게 되고 그 후에는 입력 버퍼를 바꾸어 다시 16개의 Convolution 마스크를 이용하여 feature map의 다음 줄을 생성한다. 연산이 끝난 후 연속적으로 계산이 이루어져 계산 속도를 빠르게 하기 위해서 입력 버퍼와 weight 버퍼를 2개씩 사용하였다. 두 번째 Convolution Layer와 세 번째 Convolution Layer에서의 weight 버퍼의 크기는 5x5로 동일하며 입력 버퍼의 크기는 두 번째 Layer에서는 20x6, 세 번째 Layer는 12x8 크기의 입력 버퍼를 사용하여 각각의 Layer에서 하나의 입력 버퍼에 대해서 16x2, 8x4 크기의 feature map을 생성한다.

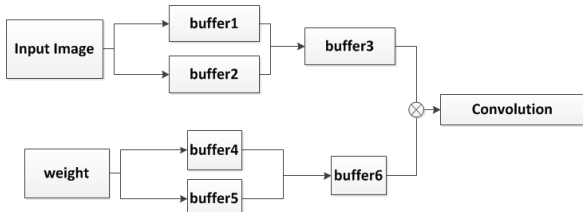


Fig. 4. Proposed Convolution Layer Module
그림 4. 제안하는 Convolution Layer 모듈

그림 5는 Max Pooling Layer에 대한 하드웨어 모듈의 블록도로 Max Pooling을 수행하기 위해 feature map으로부터 4개의 값을 가져온 뒤 두

개씩 비교하여 큰 값을 추출한 후, 다시 두 값을 비교해 최종적으로 4개의 값 중 가장 큰 값을 결과 값으로 사용한다. Feature map은 Block Memory에 저장되어 있기 때문에 1 clk에 한 개의 값밖에 가져오지 못하여 초기 결과 값은 6 clk 후에 나오게 되며 이후에는 4 clk 마다 Max Pooling의 결과 값이 나오게 된다. Max Pooling은 feature map의 개수만큼 동시에 수행되며 첫 번째와 두 번째 Max Pooling 모듈에서는 feature map 16개에 대해서 동시에 수행되며 세 번째 Max Pooling 모듈에서는 feature map 32개에 대해 동시에 수행된다. 그림 6은 Fully Connected Layer에 대한 하드웨어 모듈의 블록도로 Fully Connected Layer에서 사용되는 weight와 feature map의 값들은 Block Memory에 저장되어 있으며 하나씩 메모리로부터 불러와서 계산하게 된다.

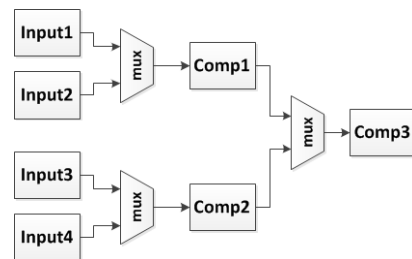


Fig. 5. Proposed Max Pooling Layer Module
그림 5. 제안하는 Max Pooling Layer 모듈

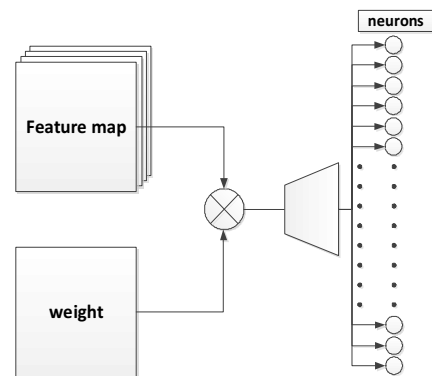


Fig. 6. Proposed Fully Connected Layer Module
그림 6. 제안하는 Fully Connected Layer 모듈

V 실험 환경 및 결과

1. Convolution Layer Module

본 논문에서 사용한 CNN 구조는 앞서 설명한

그림 2와 같은 구조이며, 차량 검출을 위한 학습 데이터셋은 2,128개의 positive 영상[29]과 주행 중 얻은 차량의 영상, 5,130개의 negative 영상[30]과 주행 중 얻은 영상에서의 도로, 가드레일 및 차선 등을 이용하였다. 학습과정에서 데이터를 학습시키는 걸리는 시간은 총 1,080초가 소요되었으며, 500개의 검증(validation)셋에 대해서 98.9%의 검출률을 보였다. 그림 7은 학습 시 사용한 positive 영상과 negative 영상의 일부를 나타낸 것이다.

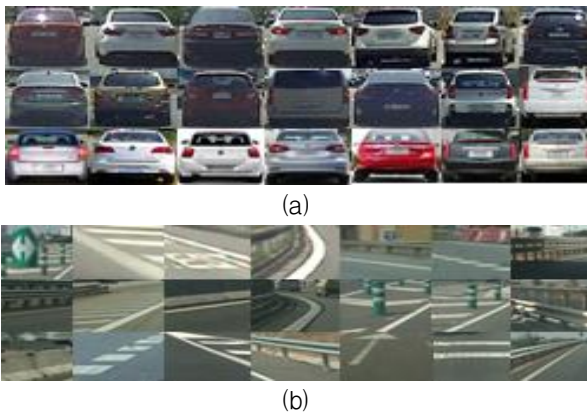


Fig. 7. Training Data Images
 (a) positive training image (b) negative training image
 그림 7. 학습 데이터 영상
 (a) positive 학습 영상 (b) negative 학습 영상

2. 차량 검출 결과

Table 1. CalTech dataset detection rate comparison
 표 1. CalTech 데이터셋 검출률 비교

Methods	Detection rate(%)
PCA+SVM[9]	86.59%
Gabor+SVM[10]	90.24%
Wavelet+SVM[11]	90.24%
Wavelet+Gabor+SVM[12]	91.06%
Haar-like+Cascaded AdaBoost[13][14]	92.38%
Haar-like+AdaBoost[15]	94.20%
Haar-like+AdaBoost+SVM[16]	94.41%
Proposed method	99.69%

CNN과 다른 기계학습 알고리즘의 성능을 비교하기 위해서 California Institute of Technology(CalTech)의 차량 데이터셋을 이용하여 차량 검출률을 비교하였다. 사용한 데이터셋은 CalTech에서 제공하는 652장의 차량 뒷면으로 이루어진 이미지를 이용하였다. 표 1은 앞선 2장에서 설명한 다른 기계학습 방법들과 본 논문에서

사용한 CNN의 차량 검출률을 나타낸 것으로 CNN을 사용한 방법의 차량 검출률은 99.69%로 다른 기계학습 방법들보다 우수한 성능을 보이는 것을 확인할 수 있다.

3. FCWS 알고리즘

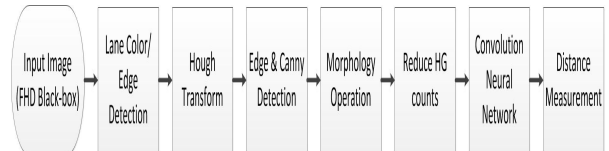


Fig. 8. FCWS Algorithm
 그림 8. FCWS 알고리즘 순서도

그림 8은 실험에 사용한 FCWS 알고리즘의 순서도이다. FCWS는 입력 영상에서 현재 주행 중인 차선을 찾아내고 차선 내에 차량이 있을만한 후보군을 추출한 뒤 해당 후보군이 차량인지 아닌지 확인 후에 차량검출 시 일정거리 이상 가까워져 충돌이 예상되면 운전자에게 경고를 주어 사고를 방지하는 시스템이다. FCWS 알고리즘 중 차량의 분류에 해당하는 CNN은 하드웨어로 구현하였고, 나머지 부분은 소프트웨어로 동작시켰으며 그림 9는 FCWS의 결과 영상으로 찾아낸 전방 차량은 차량 주변에 사각형을 그리는 것으로 표현한다.



Fig. 9. Vehicle detection result image
 그림 9. 차량 검출 결과 영상

4. 하드웨어 검증

그림 10은 Zynq를 이용한 임베디드 환경의 블록도로 CNN 하드웨어 구현을 검증하기 위해서 Zynq ZC706보드를 이용하였다. Processing

System(PS)의 CPU는 Dual Core ARM Cortex-A9, 1GB RAM, g++ 4.8.1 컴파일러, OpenCV 2.4.9, Linaro 14.04을 사용하였다. Programmable Logic(PL)에서 하드웨어 구현을 위해 사용한 언어는 Verilog이며, 사용한 합성기는 Vivado 2014.4.1.을 이용하였다. PS와 PL간의 데이터 전송은 AMBA AXI4 버스를 이용하였으며, 하드웨어의 동작 주파수는 100MHz이고, 실험에 사용한 입력 영상은 36x36 크기의 그레이 영상을 이용하였다.

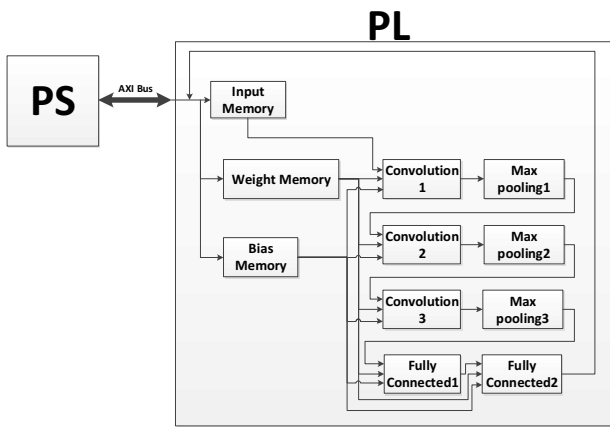


Fig. 10. Embedded environment using Zynq
 그림 10. Zynq를 이용한 임베디드 환경

Table 3. Execution time for each layer of CNN
 표 3. CNN Layer의 구간별 수행시간

(단위 : ms)

Layer	ZC-706(SW)	ZC-706(HW)
AXI(write)	-	0.67318
Conv1	71	0.16581
Pool1	1	0.01031
Conv2	287	0.65674
Pool2	0.479	0.00264
Conv3	144	0.32787
Pool3	0.221	0.00071
Fully Connected1	3	0.32787
Fully Connected2		0.00131
AXI(read)	-	0.00027
Total time	506.7	2.167

표 3은 Zynq를 이용한 임베디드 환경에서 CNN의 각 Layer에 대한 소프트웨어와 하드웨어의 수행시간을 각각 비교한 표이다. 하드웨어로 구현 후 Convolution Layer별로 428배, 437배, 439배의 속도 향상이 있으며, Pooling Layer는 96배, 181배, 311배의 속도가 증가하였고 Fully Connected Layer는 9배정도 증가하였다. 데이터 전송에 약 0.67ms정도가 소요되지만 데이터 전송

을 제외한 나머지 수행시간은 1.5ms로 구현한 CNN 하드웨어를 실시간으로 수행하는데 문제가 없다.

Table 4. Execution time for each HW section
 표 4. 하드웨어 구간 별 수행 시간

(단위 : ms)

	Execution time for one image
MMAP	0.645
Write Data	0.67318
CNN H/W	1.49355
Read Data	0.00027
Total	2.812

표 4에서는 하드웨어의 구간별 수행 시간을 나타낸 것이다. MMAP은 소프트웨어에서 하드웨어로 데이터를 전송하기 위해서 PS DDR 메모리에 데이터를 쌓는 과정으로 0.645ms가 걸렸다. Write Data는 하드웨어에서 사용할 데이터들을 PL의 BRAM(Block RAM)에 저장하는 시간이며, Read Data는 하드웨어의 결과 데이터들을 PS DDR로 전송하는 시간으로 각각 0.67318ms, 0.00027ms가 걸렸다. 따라서 하드웨어의 수행시간은 Write Data와 CNN 하드웨어 수행시간 그리고 Read Data의 수행시간을 합친 2.812ms로 한 장의 이미지를 처리하는데 약 355.62fps가 걸려 실시간 처리가 가능함을 확인했다. 표 5는 한 장의 이미지를 처리할 때 사용된 하드웨어의 리소스이다.

Table 5. CNN Hardware Resource
 표 5. CNN 하드웨어 사용량

Resource	Utilization	Available	Utilization(%)
Flip-flop	97,799	437,200	22.23
LUTs	144,393	218,600	66.05
BRAM(KB)	218	2,330	9.36
DSP	9	900	1

VI 결론

최근, 딥 러닝을 이용한 영상 처리 방법인 CNN은 영상 처리 분야에서 다른 기계학습 방법들보다 뛰어난 성능을 보이며 각광받고 있다. 특히, 운전자 보조 시스템에서는 차량 검출, 표지판 검출, 보행자 검출 등 CNN을 이용한 영상 처리 알고리즘에 대한 연구가 활발히 진행되고 있다. 하지만 차량용 전장시스템에서 CNN은 반복 연산이 많기 때문에 수행 속도가 오래 걸리는 단점이

있다. 따라서 CNN 알고리즘의 수행시간 중 86% 이상 차지하는 Convolution 연산 및 기타 연산들을 병렬로 처리하여 CNN의 수행 속도를 향상시키는 하드웨어 구조를 제안하였다. Convolution Layer에서는 곱셈기 25개를 사용하여 Convolution 계산이 한 번에 이루어지도록 하였으며, Convolution 계산 시 weight 버퍼와 입력 버퍼에 대해 더블 버퍼를 이용하여 결과 값이 끊이지 않고 1 clk마다 나오도록 설계하였다. Max Pooling Layer에서는 4개의 입력 값에 대해 동시에 두 개씩 계산하여 4 clk마다 결과가 나오도록 설계하였다. 제안하는 CNN 하드웨어의 성능을 검증하기 위해서 Xilinx사의 ZC706 보드를 이용하였고, 운전자 보조 시스템에서 사용하는 차량 검출 알고리즘에 적용하여 실험하였다. 임베디드 환경에서 소프트웨어로 수행시간이 506.7ms였었던 CNN을 하드웨어 구현 후 동작 주파수 100MHz에서 수행시간은 2.812ms로 실시간 처리가 가능해졌으며, 소프트웨어의 속도보다 180배 빨라졌다.

References

- [1] K. Grauman and T. Darrell, "The pyramid match kernel Discriminative classification with sets of image features," *Proceedings of IEEE International Conference on Computer Vision*, 2005.
- [2] C. Cortes and V. Vapnik, "Support-Vector Network," *Machine Learning*, Vol. 20, No. 3 273-297, 1995.
- [3] Quinlan, J. Ross, *CA. 5: programs for machine learning*, Elsevier, 2014.
- [4] Viola, Paul, and Michael J. Jones. "Robust real-time face detection," *International journal of computer vision* 57.2 pp. 137-154, 2004.
- [5] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview," *Neural Networks* 61 pp. 85-117, 2015.
- [6] Dollar, Piotr, et al. "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence* Vol. 34 No. 4 pp. 743-761, 2012.
- [7] Y. B. Jo, et al. "Traffic Sign Recognition using SVM and Decision Tree for Poor Driving Environment," *j.inst.Korean.electr.electron.eng* Vol. 18 No.4 pp. 485-494, 2014.
- [8] Salari, E., and D. Ouyang. "Camera-based Forward Collision and lane departure warning systems using SVM," *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, 2013.
- [9] J. Wu and X. Zhang, "A PCA classifier and its application in vehicle detection," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 1, pp. 600 - 604, 2001.
- [10] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," in *Proc. 14th Int. Conf. Digit. Signal Process*, pp. 1019 - 1022, 2002.
- [11] Z. Sun, G. Bebis, and R. Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection," in *Proc. 7th Int. Conf. Control, Autom., Robot. Vis*, pp. 1641 - 1646, 2002.
- [12] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining Gabor and wavelet features," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst*, pp. 130 - 135, 2002.
- [13] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. Int. Conf. Image Process*, pp. 900 - 903, Jan. 2002.
- [14] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," in *Proc. 25th German Pattern Recognit. Symp.* pp. 297 - 304, 2003.
- [15] Wen, Xuezhi, and Yuhui Zheng. "An improved algorithm based on AdaBoost for vehicle recognition," *The 2nd International Conference on Information Science and Engineering*. IEEE, 2010.

- [16] X. Wen et. al, "Efficient Feature Selection and Classification for Vehicle Detection," *IEEE Trans. On Circuits and Systems for Video Technology*, Vol.25, No.3, pp.508-517, Mar. 2015.
- [17] O. Russakovsky et. al, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Vol.115, No.3, pp.211-252, Dec.2015.
- [18] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition," *Neural computation*, Vol 1. No. 4: pp. 541-551, 1989.
- [19] LeCun, Yann, et al. "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* 86.11 pp. 2278-2324, 1998.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [21] Gao, Yongbin, and Hyo Jong Lee. "Vehicle Make Recognition Based on Convolutional Neural Network," *Information Science and Security (ICISS), 2015 2nd International Conference on. IEEE*, 2015.
- [22] LANGE, Stefan; ULBRICH, Fritz; GOEHRING, Daniel, "Online vehicle detection using deep neural networks and lidar based preselected image patches," *Intelligent Vehicles Symposium (IV)*, pp. 954-959, 2016 IEEE.
- [23] Bautista, Carlo Migel, et al. "Convolutional neural network for vehicle detection in low resolution traffic videos," *Region 10 Symposium (TENSymp)*, 2016 IEEE. IEEE, 2016.
- [24] Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F., Coates, A., and Ng, A. Y. "An Empirical Evaluation of Deep Learning on Highway Driving," *ArXiv e-prints*, April 2015.
- [25] Jung, Seokwoo, et al. "Real-time Traffic Sign Recognition system with deep convolutional neural network," *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on. IEEE*, 2016.
- [26] Fukui, Hiroshi, et al. "Pedestrian detection based on deep convolutional neural network with ensemble inference network," *2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015.
- [27] Li, Huimin, et al. "A high performance FPGA-based accelerator for large-scale convolutional neural networks," *Field Programmable Logic and Applications (FPL), 2016 26th International Conference on. EPFL*, 2016.
- [28] Li, Ning, et al, "A multistage dataflow implementation of a Deep Convolutional Neural Network based on FPGA for high-speed object recognition," *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, IEEE, 2016.
- [29] Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification," *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [30] J. Arróspide, L. Salgado, M. Nieto, "Video analysis based vehicle detection and tracking using an MCMC sampling framework," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, Article ID 2012:2, Jan. 2012.

BIOGRAPHY

Jiwon Bang ((Student Member)



2015 : BS degree in
Electronics and
Communications Engineering,
Kwangwoon University.
2015~ : Course of MS in
Electronics and

Communications Engineering, Kwangwoon
University.

Yongjin Jeong (Member)

1983 : BS degree in Control
and Instrumentation

Engineering, Seoul National
University.

1995 : MS, PhD degree in
Electronics and

ComputerEngineering, University of
Massachusetts, Amherst

1983~1989 : Research Engineer, ETRI

1995~1999 : Chief Researcher, Samsung
Electronics

1999~current : Professor, Dept. of Electronics
and Communications Engineering,
Kwangwoon Univ.