



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위 논문

지도교수 임 태 호

고속 해상 객체 검출을 위한 양자화
적용 기반 CNN 딥러닝 모델 시스템에
관한 연구

A Study on CNN Deep Learning Model System Based on
Quantization Application for High-Speed Marine Object
Detection

2020

호 서 대 학 교 대 학 원

해양융합기술학과

이 성 주

고속 해상 객체 검출을 위한 양자화
적용 기반 CNN 딥러닝 모델 시스템에
관한 연구

A Study on CNN Deep Learning Model System Based on
Quantization Application for High-Speed Marine Object
Detection

이 논문을 석사학위 논문으로 제출함.

2021년 2월

호 서 대 학 교 대 학 원

해양융합기술학과

이 성 주

이성주의 석사학위
논문을 합격으로 판정함.

심사위원장 김용관 인

심사위원 진현준 인

심사위원 임태호 인

2020년 12월

호 서 대 학 교 대 학 원

연구윤리 서약서

본인은 호서대학교 대학원생으로서 이 학위논문 작성과정에서 다음과 같은 연구윤리 기본원칙의 준수를 서약하고 이를 지켰음을 확인합니다.

첫째, 지도교수의 지도에 따라 정직하고 엄정하게 연구를 수행하여 학위논문을 작성한다.

둘째, 논문작성 시 타인에의 대필의뢰, 타 논문의 표절, 데이터의 위조와 변조 등 학문적 진실성을 훼손하는 어떠한 연구부정행위도 하지 않는다.

2020 . 12. 4.

학 위 명 : 석 사

학 과 : 해양융합기술학과

성 명 : 이 성 주

주 인
이 성 주

호서대학교 대학원장 귀하

목 차

I. 서	론	1
II. 관련 연구		3
1. 영상 처리		3
2. CNN(Convolutional Neural Network) 알고리즘		5
가. 딥러닝 알고리즘 경량화 기술 동향		6
나. 경량 딥러닝 Network 기술 동향		7
III. 훈련 데이터 전처리 구성		8
1. 데이터 셋 영상 처리		8
가. Grayscale		8
나. Adaptive Histogram Equalization		10
다. ZCA(Zero Component Analysis) Whitening		14
라. RGB 8bit Color Image		17
2. 해양 환경 데이터 영상 처리		19
IV. CNN 모델 구성		20
1. CNN		20
2. CNN 모델 구성		23
3. Object Detection 모델 구성		28
가. SSD		28
나. Mobilenet v2		31

4. 경량화 알고리즘: 양자화	33
V. Neural Network 실험 결과	36
1. 실험 환경	36
2. 실험 결과	37
가. 전처리 별 모델 정확도 결과	37
나. 객체 인식 네트워크 모델 성능 분석	41
다. TPU 기반 객체 인식 모델 성능 분석	44
VI. 결 론	48
참고문헌	49
영문초록	50

그 립 목 차

그림 1 Grayscale 처리 전과 후	2
그림 2 Histogram 처리 및 누적 분포도	3
그림 3 Grayscale 이미지	3
그림 4 Histogram Equalization	3
그림 5 Adaptive Histogram Equalization	3
그림 6 120x90 원본 이미지	3
그림 7 ZCA 미백화 처리	16
그림 8 ZCA 미백화 이미지	16
그림 9 8bit Color Image 처리 전과 후	18
그림 10 객체 검출 학습용 데이터 셋	19
그림 11 인공신경망 뉴런의 흐름도	20
그림 12 CNN 흐름도	23
그림 13 Max Pooling 방식	24
그림 14 SSD와 YOLO의 네트워크 구조	29
그림 15 uint8의 양자화 흐름도	34
그림 16 히스토그램의 Feature map (1)	39
그림 17 히스토그램의 Feature map (2)	40
그림 18 SSD MobileNet V2 추론 결과 (1)	41
그림 19 SSD MobileNet V2 추론 결과 (2)	42

그림 20 SSD MobileNet V2 추론 결과 (3)	43
그림 21 SSD MobileNet V2 uint8 양자화 추론 결과 (1)	44
그림 22 SSD MobileNet V2 uint8 양자화 추론 결과 (2)	45
그림 23 SSD MobileNet V2 uint8 양자화 추론 결과 (3)	46

표 목 차

표 1 CNN 기반 딥러닝 모델 구성	26
표 2 임베디드 보드 사양	36
표 3 각 전처리별 모델 성능	37
표 4 Raspberry 환경 성능 분석	47
표 5 Coral Dev Board 환경 성능 분석	47

I. 서론

최근 하드웨어의 기술이 지속해서 발전하면서 암흑과도 같았던 DNN의 새로운 길을 열어주었다. 이러한 기술들은 인공지능 기술을 산업 전반에 적용하여 연구할 수 있게 되었으며 컴퓨터 비전 분야에서 영상에 특화된 CNN 기반의 모델을 적용하는 연구들이 중점적으로 연구되고 있다. 특히 자율운행에서는 안전과 자율운행에 필요한 실시간성을 확보하기 위한 연구가 중점적으로 진행되었다.

해양 환경에서 선박을 운항하는 경우 기존에는 선박 주변의 상황을 인식하기 위해 레이더와 같은 센서를 사용하거나 사람이 눈으로 직접 보고 판단하여 위험한 상황에 대해 대처하였다. 하지만 최근에는 카메라를 이용하여 인간의 시각적 정보를 대신해 컴퓨터가 물체를 인식하여 주변 상황을 파악하여 더 안전하고 정확한 판단을 내려주는 인공지능 기술이 연구되고 있다. 이러한 기술이 국내외에서도 선박 물동량이 증가하며 해양 환경에서도 관심이 증가하면서 자율운항선박에 대해서도 관심이 증가하였다. 자율운항선박(MASS)은 무인으로 선박을 운항하기 위해 주변 환경 및 위치 정보 등을 활용하여 스스로 판단하고 시스템을 제어하는 통합적인 시스템을 의미한다[1].

해양수산부에서는 스마트 해양교통정책 추진전략을 통해 기존 해양사고의 80%가 넘는 인적 과실로 인해 발생하는 사고를 인공지능 기술을 활용하는 해상 내비게이션 서비스를 시행하여 선제적 예방을 하겠다고 발표하였다. 이러한 인공지능 서비스를 도입하기 위해서는 암초나 선박과 같은 충돌 위험이 존재할 수 있는 해상의 부유물들을 감지하거나 선박 내부의

화재 및 기타 다양한 안전 문제를 야기할 수 있는 객체에 대해 인식하여 다양한 기술이 접목되어 스스로 판단할 수 있어야 하므로 데이터가 실시간으로 처리되지 않는다면 사람의 지속적인 감시가 없이는 선박의 안전을 보장할 수 없다. 이러한 문제를 해결하기 위해서 프로세서의 장비를 고성능으로 대체하여 데이터 처리 속도를 높일 수 있으나, 제품의 가격이 급격하게 상승할 수 있는 단점이 존재해 소형 선박을 운항하는 선주들에게는 비용적인 문제로 인하여 시스템을 설치하는 이점을 취할 수 없다. 이를 해결하기 위해서 고성능의 프로세서가 존재하지 않아도 시스템의 알고리즘에서 연산량이 적게 설계하여 처리 속도 및 성능을 유지하는 것이 매우 중요하다.

본 논문에서는 시스템의 알고리즘에서 사용하는 딥러닝 모델의 정확도를 유지하면서 처리 속도를 향상시킬 수 있는 방법을 제안한다. 자율운행에 있어서 중요한 요소 중 하나인 주변 환경이 육상과 달리 해상에서는 주변 영상의 변화가 적고 대부분의 배경이 하늘이나 바다와 같은 부분의 영역을 차지하기 때문에 안전의 위협을 가할 수 있는 객체를 비교적 적은 연산량으로도 탐지가 가능할 수 있다. 이를 고려하여 해상에서의 암초나 선박처럼 운항의 안전을 위해 반드시 검출되어야 하는 부유물을 전처리를 진행하여 딥러닝 모델에서 효율적인 방법을 제안한다. 또한, 입력되는 데이터들을 고속으로 처리하기 위해서 알고리즘의 경량화 기법의 하나인 모델 양자화 기법을 시스템에 적용하여 처리 속도를 증가함과 동시에 메모리 사용률을 감소시키는 연구를 진행하였다.

II. 관련 연구

1. 영상 처리

디지털 화상 처리, 디지털 영상 처리로 일컬어지는 영상 처리는 컴퓨터 알고리즘을 사용하여 디지털 이미지를 변화시키는 처리 방식의 전반적인 것을 의미한다.

디지털 영상을 처리하기 위해서는 저장, 디스플레이, 전송과 같은 다양한 기술이 접목되기 때문에 컴퓨터 기술의 발전이 매우 중요하였다. 이러한 기술의 대부분은 하드웨어의 성능이 향상되면서 1960년대부터 점진적으로 발전하였는데, 위성 사진 및 유선을 통한 사진 변환부터 시작하여 의료, 영상 통화, 생물 분야, 산업 현장 등 다양한 분야에서 활용되고 있다. 이러한 영상 처리 기술은 화질이 떨어지는 이미지를 복원하거나 특정 목적에 맞게 변화를 주어 보다 더 질적인 데이터로 만드는 것에 의미가 있다.

하드웨어의 성능이 향상됨과 동시에 딥러닝 분야에서도 혁신적인 연구 결과가 나타나기 시작하면서, CNN 기반의 모델에 대해서도 지속적인 발전이 이루어졌다. CNN 기반 모델의 신경망을 구성하고 학습을 진행할 때 기존의 연구방식에서는 학습되는 데이터의 양을 증가시키는 방식이 적용되거나 질적 향상을 위해 회전 및 색상을 보정하는 방식과 같은 기존 데이터를 가공하는 다양한 처리 기법들을 사용하였다. 하지만 딥러닝 모델은 학습되는 데이터가 충분하지 않을 때에는 정확도 향상이 좋지 못하다는 단점이 존재하며, 이에 따라 데이터의 명도나 색상 정보의 변화, 입력 데이터에서의 객체 간 겹침 등의 현상들이 딥러닝 모델에서는 잡음으로

인식될 수 있는 주변 환경의 변화를 잘 처리해내지 못하므로 CNN 모델의 필터에서 고수준의 특징점을 추출할 수 있는 질적인 데이터를 생산하기 위해서 보다 다양한 처리 기법을 통한 연구가 필요하다.

본 논문에서 설계하는 모델의 경우에는 해상 환경에서 위협이 될 수 있는 다양한 부유물들을 분류하고 검출해야 하므로 카메라로 데이터를 입력받을 때 RGB 3채널의 값으로 들어온다면 검출해야 하는 객체를 제외하고는 비교적 불필요한 바다와 같은 색상 정보는 중요하지 않기 때문에 제거하고 객체의 형태를 살려내어 학습시키는 데이터로 사용하는 것이 오히려 모델의 정확도에 대해 좋은 영향을 줄 수 있다. 따라서 색상을 제거하고 전처리 기법을 적용한 데이터 셋과 기존 RGB 3채널을 비교하여 어떠한 전처리 방법이 좋은지를 파악한다. 또한, 바다 영역의 색상을 제거할 때 데이터의 크기가 기존 RGB 24bit의 크기에서 1채널의 8bit로도 표현이 가능하므로 메모리의 비중을 줄일 수 있다. 그리고 질적으로 향상된 데이터를 생성하기 위해 동일한 데이터 셋에 8bit color, Gray scale, Histogram equalization, ZCA whitening과 같은 전처리 방법이 적용되었다.

2. CNN(Convolutional Neural Network) 알고리즘

최근 급속도로 성장하고 있는 인공지능 기술들은 대부분의 실생활에 적용하기 위해서는 저전력 임베디드 보드에서도 처리할 수 있는 수준의 실시간적 속도가 필요하므로 처리 속도 개선을 위한 연구가 활발히 진행되고 있다.

이미지, 소리, 텍스트 등 시계열 데이터부터 다양한 종류로 이루어지고 무수하게 많은 양의 데이터들을 판단하기 위해서 GPU(Graphics Processing Unit) 기반으로 학습 및 추론을 진행하여 각각의 목적에 부합하는 딥러닝 모델들을 만들어내지만, 저전력 임베디드 보드, 모바일이나 IoT 기술에서의 직접적인 학습 및 추론에서는 성능이 효율적이지 못하다. 따라서 학습된 모델들을 연산량을 감소시키는 방식을 적용하고 이에 따라 메모리 용량도 절약하면서 정확도의 하락은 최소화하는 딥러닝 모델의 경량화에 관한 연구가 활발히 진행되는 실정이다. 경량화에 관한 연구는 가장 보편화된 CNN 기반의 모델을 통해 연구가 활발하게 진행되고 있으며 이러한 모델에서의 경량화 연구는 크게 경량 알고리즘 연구와 알고리즘 경량화 연구로 나뉜다[2].

가. 딥러닝 알고리즘 경량화 기술 동향

먼저 경량 알고리즘은 알고리즘 자체를 적은 연산으로 활성화하기 위한 효율적인 구조를 설계하여 기존의 딥러닝 모델보다 처리 속도를 향상하는 연구를 칭한다. 경량 알고리즘 연구는 모델 구조, 합성곱 필터 변경 및 자동 모델 탐색까지도 경량 알고리즘 연구에 속한다.

일반적으로 신경망이 깊어질수록 계층이 늘어나면서 정확도가 떨어지는 문제가 존재하는데, 경량 알고리즘 연구는 다양한 계층 구조를 이용하여 파라미터를 축소하는 방식으로 잔여 블록(Residual block)을 사용하여 깊은 신경망에서도 정확도를 향상할 수 있으며 필터 크기를 변경하여 연산 구조를 변경한 ResNet과 같은 네트워크가 이에 속한다. 다음으로 합성곱 필터 변경 방식은 CNN의 가장 큰 연산량이 존재하는 합성곱 필터의 연산을 축소하는 방식으로, 채널을 분리하여 학습시키면서 연산량을 감소시키는 MobileNet과 같은 네트워크가 이에 속한다. 마지막으로 자동 모델 탐색은 모델에 대해 지연 시간이나 전력 소모 등과 같은 어떠한 특정 요소가 주어질 때, 강화 학습을 적용하여 최적의 모델을 찾아내는 방식을 의미하며 NetAdapt와 같은 방식이 이에 속한다.

나. 경량 딥러닝 Network 기술 동향

알고리즘 경량화는 앞서 설명한 경량 알고리즘과는 달리 모델이 특징점을 저장하고 판단할 수 있는 요소인 파라미터들에서 불필요한 가중치들을 삭제하여 처리 속도를 향상하는 연구를 칭한다. 이러한 알고리즘 경량화는 모델 압축, 지식 증류, 하드웨어 가속화 등이 존재한다. 지식 증류는 이미 이전에 학습되었던 모델을 활용하여 새로운 모델을 구성할 때 파라미터값을 사용하여 학습 시간을 줄이는 방식으로, 전이 학습(Transfer Learning) 등이 이에 해당한다. 모델 압축의 경우에는 파라미터 값에서 불필요한 가중치를 제거하여 모델의 표현력을 제어하는 방식으로, 가지치기(Pruning) 및 이진화(binarization)/양자화(Quantization) 등이 이에 해당한다. 하드웨어 가속화는 모바일 기기 등을 중점적으로 뉴럴 프로세싱 유닛(NPU)이나 FPGA 등의 하드웨어를 통해 처리 속도를 개선하는 방식이 이에 해당한다[3].

본 논문에서는 모델의 크기를 압축시키면서 모바일 기기, 저전력 임베디드 보드에서도 성능이 충분히 발휘되기 위하여 알고리즘 경량화 방식을 진행하였으며, 그 중 양자화 방식을 통해 모델의 구성을 개선하였다. 양자화 기술은 파라미터 값에서 일정 부분을 제거하는 방식이기 때문에 파라미터 수가 과도하게 많이 존재하거나 신경망의 깊이가 깊은 모델에 대해서 추론 속도를 상당히 개선할 수 있으므로[4] 선박과 같은 객체더라도 모습이 각기 다른 해상 환경에서의 부유물을 검출해내기에 비교적 깊은 신경망을 써서 정확도를 높여도 처리 속도에서 이점을 가져올 수 있으므로 해당 기법을 적용해 진행하였다.

Ⅲ. 훈련 데이터 전처리 구성

1. 데이터 셋 영상 처리

CNN 기반의 모델에 대해 학습용으로 구성된 데이터는 비교적 질적으로 향상된 데이터로 가공하기 위해 전처리를 진행한다. 기존에 RGB 3채널로 입력되는 데이터를 원본 데이터라고 가정한다. 전처리 기법으로는 Grayscale과 이를 적용한 데이터에서 Adaptive Histogram Equalization, ZCA(Zero Component Analysis) Whitening 기법을 포함하여 RGB 8bit Color Image의 4종류의 방법으로 구성하였다.

가. Grayscale

Grayscale 기법은 카메라로부터 입력받는 데이터가 RGB 3채널의 데이터일 경우에 광도의 정보만을 저장하게끔 처리하여 색상 정보를 제거하는 방식이다. 영상 처리 분야에서 Grayscale 기법은 영상 안에서 경계선과 같은 특징을 추출하는 경우에도 사용되며 이러한 방식을 적용한 이미지는 흑백 사진이라고도 알려진다. 일반적인 1bit로 광도 정보를 표시할 때는 검은색과 흰색 두 가지로만 데이터를 표현할 수 있으나 Grayscale의 경우에는 8bit로도 표현이 가능하다. 입력되는 데이터들의 RGB 값은 각 채널당 0~255의 값으로 표현되어 입력되며, 3채널의 값을 모두 더해 각 채널에 3등분하여 재구성해주는 방식부터 각 채널의 비중을 조금씩 다르게 하여 모두 더한 뒤에 3등분하는 방식까지 다양하다. 즉, Grayscale 기법은 1bit로 표현되는 검은색과 흰색 사이에 이보다 훨씬 많은 회색으로 음영

을 구현하는 방식이다. 각 채널의 비중에 변화를 주어 Grayscale로 전환하는 것은 사람의 시각에 대해 민감도 차이가 발생하기 때문인데, 이를 해양 환경을 목적으로 하는 모델에 적용한다면 바다와 같은 비교적 불필요한 색상 정보의 비중을 감소시켜 객체의 형태를 더 유리하게 보존할 수 있다.

아래 식 1은 본 논문에서 사용한 Grayscale 관계식이며 그림 1은 모델 학습에 사용될 데이터 중 선박 이미지 한 장을 추출하여 Grayscale을 적용하기 전과 후를 비교하였다.

$$L = R*0.299 + G*0.587 + B*0.114 \quad (1)$$



그림 1. Grayscale 처리 전과 후
Figure 1. Before & After Grayscale Processing

나. Adaptive Histogram Equalization

Histogram이란 데이터의 분포를 그래프로 표현한 것이다. 예를 들어 이미지는 0~255의 수로 표현된 각 픽셀들의 값들을 분포하는 수를 바탕으로 그래프로 표현하면 해당 그래프가 이미지의 Histogram이다.

Histogram의 경우 분포가 왼쪽으로 치우쳐져 있을수록 어두운 이미지를 의미하며 반대인 오른쪽으로 치우쳐져 있을 경우 이미지가 밝게 보이는 현상을 갖는데, Histogram으로 나타낸 이미지의 경우 선박의 색상은 대체로 흰색으로 구성되어 있으며, 그 외에 부분은 비교적 어두운 색상으로 구성되어 있으므로 아래 그림 2의 Histogram에서도 흰색으로 표현되는 픽셀값 255에 가까운 부분과 그 외의 부분이 나누어져서 나타난 것을 확인할 수 있다. 일반적으로 밝기 조절이 잘되어 있는 이미지들은 0~255의 평균값을 중심으로 좌우로 잘 분배되어 있다.

Histogram은 일반적으로 막대그래프로 표현하며, Histogram을 구하기 위해서는 픽셀의 강도 범위, 표현할 막대의 개수, 막대의 범위를 고려하여야 한다. 픽셀의 강도 범위는 그래프에 표현할 강도 범위를 의미한다. 전체 그래프를 확인하기 위해서는 0~255의 범위로 지정하여 표현한다. 표현할 막대의 개수는 강도 범위를 표현할 막대의 개수를 의미한다. 막대의 범위는 막대 하나당 표현할 픽셀 강도 범위를 의미한다.

그림 2에서는 그림 1에서 Grayscale로 변환된 이미지의 전체 범위에서의 Histogram과 누적 분포도를 나타내었다.

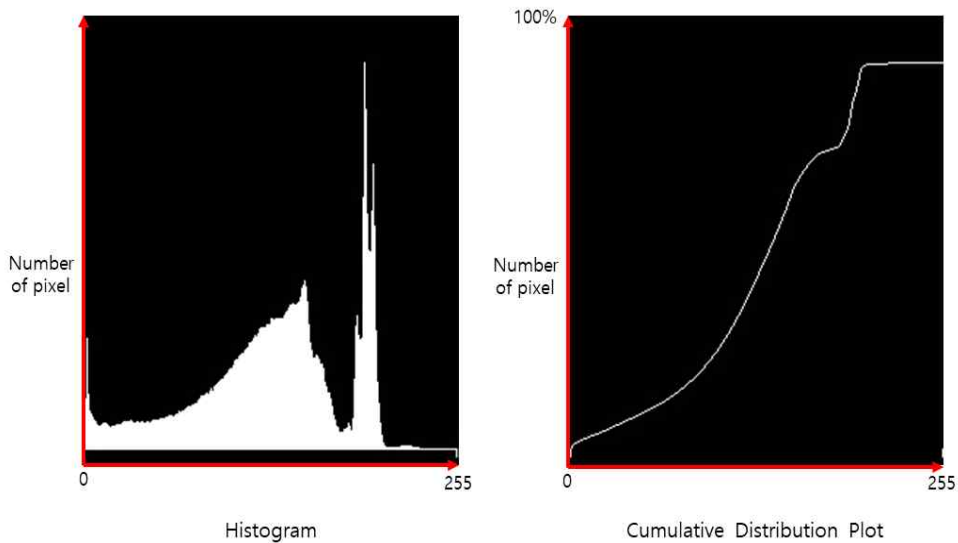


그림 2. Histogram 처리 및 누적 분포도
Figure 2. Histogram Processing & Cumulative Distribution Plot

Histogram Equalization 기법은 입력되는 데이터가 Histogram으로 나타내어 전체적인 범위에서 균등하게 분포시키는 방식으로 밝기 차이를 의미하는 Contrast를 향상하여 광도의 정보 값을 균등하게 분포시킴으로써 전체 데이터의 밝기가 평균화된다는 특징이 있다. 일반적으로 이미지의 선명도나 세부적인 표현들에 영향을 줄 수 있다. Histogram이 특정 영역에만 분포되면 좋은 결과를 만들어 낼 수 있으나 그렇지 못한 경우인 픽셀 강도가 높은 경우에는 전체적인 밝기가 평균화되는 특성상 좋은 결과를 도출해내기는 어렵다.

본 논문에서는 이를 해결하여 기존보다 질적인 데이터로 학습을 진행하기 위해 Adaptive Histogram Equalization를 사용한다. 즉 전체 이미지를 각각의 작은 블록 단위로 영역을 분할하여 Histogram Equalization을 적용하고, Contrast 값을 제한하여 각 영역에서의 noise 값의 증가를 제한하

는 방식으로 사용한다. 8x8 사이즈의 크기별로 Histogram Equalization을 진행하였으며 이렇게 작은 블록 단위로 평활화를 진행하고 대비값을 제한하여 Histogram Equalization을 진행하는 방식은 전체적인 이미지에서 평균을 취하는 것이 아니므로 사람의 시각으로 처리된 이미지를 확인하면 각각의 영역에서의 특징점들이 더 확연히 나타나는 효과를 가진다[5]. Adaptive Histogram Equalization은 이처럼 Contrast 값을 제한하기 때문에 CLAHE(Contrast Limited Adaptive Histogram Equalization)이라고 정의하기도 한다.

아래 그림 3은 Grayscale의 이미지를 Histogram으로 표현한 것이며, 그림 4, 5는 기존 Grayscale의 이미지를 바탕으로 Histogram Equalization 및 Adaptive Histogram Equalization을 적용해 나타나는 이미지와 함께 표현한 것이다. Grayscale 이미지와 일반적인 Histogram Equalization을 적용한 이미지는 객체를 검출하기에는 모든 부분의 밝기가 동일한 것으로 보이지만 Adaptive Histogram Equalization을 적용한 이미지에서는 작은 블록 단위로 진행하였기 때문에 선박의 세부적인 부분도 명확하게 보이는 것으로 확인된다.



그림 3. Grayscale 이미지
Figure 3. Grayscale Image

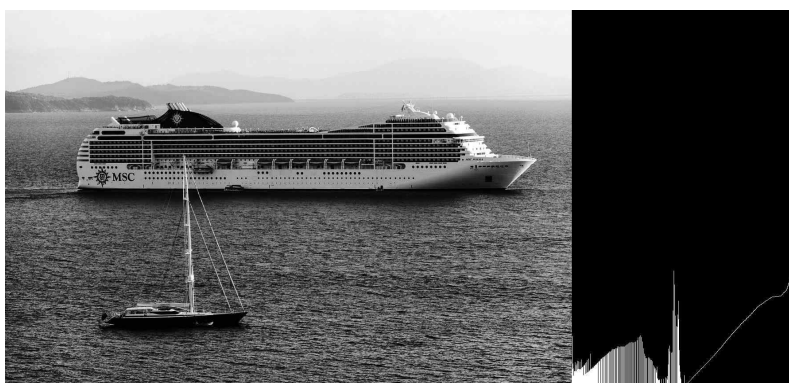


그림 4. Histogram Equalization
Figure 4. Histogram Equalization



그림 5. Adaptive Histogram Equalization
Figure 5. Adaptive Histogram Equalization

다. ZCA(Zero Component Analysis) Whitening

ZCA(Zero-Phase Component Analysis) 기법은 입력되는 전체 데이터들의 모든 픽셀에서 평균을 구하여 각각의 이미지에 평균값을 제거한 필터를 생성한다. 이렇게 생성되는 필터는 0 값을 중심으로 값이 재구성되며, 필터에서 PCA를 연산하면 공분산 행렬을 구할 수 있다[6]. 아래 식 2는 ZCA에 필요한 PCA와 ZCA의 관계식을 나타내었으며 U 는 PCA의 고유벡터 값으로 구성된 직교 회전 행렬을 의미하며 U^T 는 데이터의 상관관계를 제거하는데 적용된다. 즉, 원래 형상을 주성분에 매핑해주는 형식이다.

$$X_{ZCAwhite} = UX_{PCAwhite} \quad (2)$$

이 후 ZCA를 위해서는 분산값을 1로 해주기 위해 공분산 행렬을 제공근으로 나누어주고 작은 파라미터 값으로 미백 계수인 ϵ 를 추가한다. 아래 식 3은 ZCA whitening의 관계식을 의미한다. $\text{diag}()$ 는 대각 행렬을 의미하며 S 는 고유 벡터를 이용한 공분산 행렬이다.

$$X_{ZCAwhite} = U(1\sqrt{\text{diag}(S)+\epsilon})U^TX' \quad (3)$$

이렇게 ZCA를 이미지에 적용하면 인접한 픽셀에 대해 위상 스펙트럼과 이미지의 배열은 보존하는 상태로 미백화를 진행해 주파수 스펙트럼도 평탄화된다. 이러한 방식의 미백화를 적용하면 원본 이미지보다는 세부적인

부분의 데이터는 없어지듯이 주변 픽셀과 비슷한 값을 가지지만, 선박과 같은 전체적인 부분의 경계선과 같은 고주파 성분이 존재하는 구간에서는 값이 살아남아 객체의 형태를 보존하게 된다.

아래 그림 6은 원본 이미지를 가로, 세로 픽셀의 크기로 120x90으로 재조정된 이미지이며, 그림 7은 그림 6처럼 재조정된 이미지 20장을 가지고 ZCA 미백화를 적용한 결과다. 따라서 전체 전처리 과정을 거쳐 실제로 사용되는 데이터는 그림 8과 같으며 이는 1채널 데이터로만 변환한 것이다. 그림을 확인해보면 20장만 사용하여 진행하였기 때문에 데이터가 흐려진 것처럼 나타나지만 실제로 학습에 사용되는 데이터를 바탕으로 전체 데이터에서 ZCA 미백화를 활용하면 큰 경계선만 남게 되는 것을 확인할 수 있었다.

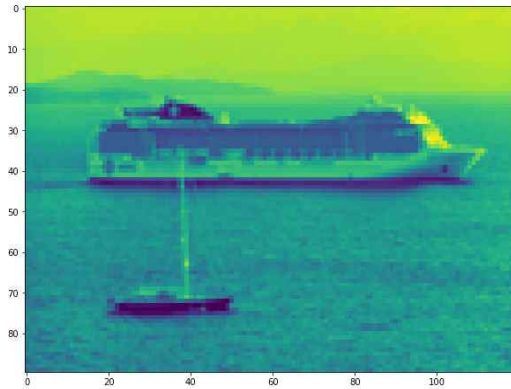


그림 6. 120x90 원본 이미지
Figure 6. 120x90 Original Image

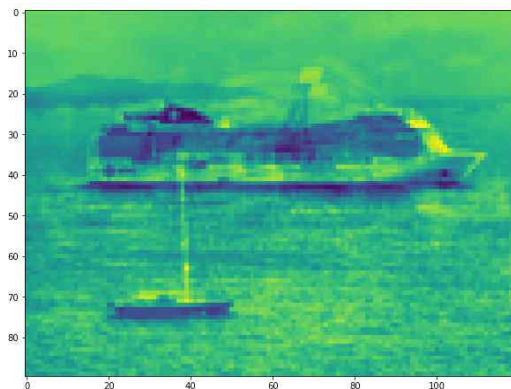


그림 7. ZCA 미백화 처리
Figure 7. ZCA Whitening Processing



그림 8. ZCA 미백화 이미지
Figure 8. ZCA Whitening Image

라. RGB 8bit Color Image

일반적으로 이미지의 한 픽셀당 24bit의 값을 가지는 RGB 채널의 경우에는 각각이 8bit를 보유하고 있으며 이는 결론적으로 16,777,216가지의 색상을 표현할 수 있다. 따라서 색감이 뛰어난 이미지를 표현할 때는 좋으나 이는 모든 정보가 필요한 데이터가 아니라면 오히려 객체를 인식하는 데이터로는 좋지 않을 수 있다. 따라서 비교적 불필요한 데이터를 감소시킨다면 객체를 검출하기에 적합한 질적인 데이터로 재생성하거나 연산량 부분에서 이점을 얻을 수 있다.

본 논문에서 사용하는 Color Image는 8bit로 재구성하여 256가지의 색을 표현하는 이미지로 변환하게 된다. 따라서 0~255의 수로 표현한다고 가정한다면 0은 RGB 채널의 (0,0,0)의 값으로 저장하는 팔레트를 저장하고 있는 방식으로 진행하였다.

앞에서 설명한 나머지 전처리 방식들은 8bit Grayscale에서 전처리를 진행하지만 8bit color image의 방식의 경우 실제로는 8bit의 데이터로 학습이 진행되나 256가지의 색상 정보를 추출할 수 있어 바다 색상을 비교적 적게 저장하고 있을 수 있다는 장점이 있어 데이터 셋으로 구성하였다. 이러한 특성을 이용하면 256가지의 색상이라도 사람의 눈에는 큰 차이가 없는 것처럼 보여 원본 이미지와 유사하게 인식되므로 특징점은 살려내 모델의 연산에서 좋은 결과를 얻을 수 있다. 아래 그림 9는 원본 이미지와 8bit로 구성된 이미지를 보여준 그림이다.



그림 9. 8bit Color Image 처리 전과 후
Figure 9. Before & After 8bit Color Image Processing

2. 해양 환경 데이터 영상 처리

그림 10에서는 해양 환경에서 선박, 부이, 암초 등과 같은 객체와 충돌하면 위험하다 판단될 수 있는 이미지 중 선박 이미지 각 5장씩 선택하여 각각의 영상 처리 결과를 나타낸 것이며, 각 모델의 비교를 위하여 원본 데이터를 포함하여 본 연구에서 활용된다.



그림 10. 객체 검출 학습용 데이터 셋

Figure 10. Dataset for Object Recognition Training

IV. CNN 모델 구성

1. CNN

1950년대 후반 고양이의 뇌에 전극을 심어 화면에 다양한 모양의 객체를 보여주었을 때 뉴런의 활성화 정도를 측정하여 시각 피질 내에 있는 뉴런들은 전체 이미지를 한 번에 받아들여 판단하는 것이 아니라, 소수의 뉴런(Neuron)이 모여 이룬 국부적으로 이미지의 일부분들을 각각 받아들여 판단한다는 것을 밝혀내었다. 따라서 인공신경망의 가장 기본적인 단위로 생물학적인 뉴런을 모방하여 제안되었다. 아래 그림 11에서는 모방하여 제안된 뉴런의 구조를 간략히 나타낸 것이다. X는 입력 데이터이며, Y는 출력 데이터를 의미한다.

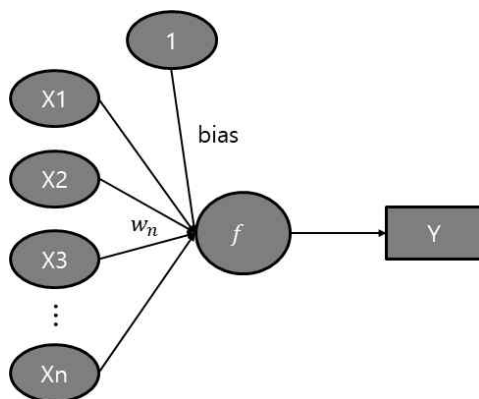


그림 11. 인공신경망 뉴런의 흐름도
Figure 11. Flow Diagram of Artificial Neural Network neurons

뉴런에서의 입력값을 통합하는데 필요한 값을 bias라고 정의하며, 고유값을 출력하는데 있어서 필요한 함수인 f 를 활성화 함수(activation

function)라고 정의한다. 이는 인공신경망에서 객체를 판단하고 다양한 기능을 할 수 있도록 하는 중요한 요소로서 활성화 함수의 경우는 대표적으로 sigmoid, tanh, ReLU 등이 범용적으로 많이 사용된다.

1980년대에 들어서면서 이러한 오류역전파(Back-propagation) 알고리즘이 제안되어 신경망 연구의 활성화가 다시 시작되었다. 오류역전파 알고리즘은 신경망의 학습을 위해 제안된 알고리즘으로, 모델의 학습 과정 중 입력이 X 이고 출력이 Y 인 경우 그 전달 방향이 입력에서 출력으로 가는 것을 순전파(feedforward)라고 한다면 오류 역전파는 역방향으로 오차를 전파하면서 각 계층으로 전달함으로써 최적의 학습 결과를 찾아갈 수 있게 해주는 알고리즘을 의미한다. 즉, 인공 신경망에서 출력값과 실제 정확한 값의 오차를 최소화하게끔 하는 방법 중의 하나이다. 이러한 오류역전파 알고리즘은 DNN(Deep Neural Network)에서도 동일한 방식으로 적용된다. DNN이란 깊은 인공신경망을 의미하며 모델 중간에 계층(layer)을 여러 개를 두어 사용하는 신경망을 의미한다. 따라서 첫 번째 층은 입력층으로, 마지막 층은 출력층으로 사용하며 나머지 중간에 존재하는 계층들은 은닉(Hidden)층으로 정의된다. 이러한 DNN의 학습에서는 그 학습하는 환경과 조건에 따라서 결과값의 변화가 다양하다. 이는 같은 학습 데이터라 할지라도 입력해주는 순서나, 역방향으로 오차를 전파시킬 때에 기존 값을 초기화하는 방식, 또는 은닉층의 개수와 같은 다양한 조건에 따라 쉽게 변할 수 있다. 따라서 가장 좋은 모델을 선택하기 위해서 다양한 조건으로 학습해보는 것이 일반적이다[7].

그러나 DNN에서 계층의 개수를 증가하면 함수적인 표현은 좋아지지만 오류역전파 학습의 성능이 떨어진다. 이는 sigmoid 함수의 미분값의 재귀적 특성 때문이며 계층이 많을수록 오차값이 빠르게 0에 하강하여 남는

계층에서 값을 갱신할 때의 효과가 떨어지기 때문이다. 또한 학습을 위해서 무수한 학습용 데이터가 필요하고 고성능의 하드웨어가 필요하다는 문제점이 제기되었다.

CNN(Convolutional Neural Network)은 1989년에 Backpropagation applied to handwritten zip code recognition 논문으로 처음 발표되었다. 이 논문은 손글씨인 zip code 인식을 위한 프로젝트이며 필기체 인식에서는 의미 있는 결과였으나 범용화에는 미흡한 부분이 존재하였다. 이후 1998년에 오류 역전파 알고리즘을 적용하고 발전시켜 CNN을 발표하였다. 그러나 CNN에서도 신경망의 깊이가 증가할수록 학습 속도가 느려지는 단점이 존재하고 오류 역전파 알고리즘이 적용되었어도 오류값이 손실되는 그래디언트(gradient) 손실의 문제가 존재하였다. 따라서 신경망에서의 학습 방법을 개선하기 위해 sigmoid를 사용하지 않고 입력이 음수면 0, 양수면 본연의 값 그대로 출력할 수 있는 ReLU와 같은 다른 활성화 함수를 사용하면서 해결하였고, 최근 들어 하드웨어의 기술도 발전하면서 개선된 CNN 기반의 인공지능망이 다양한 분야에서 사용되었다[8].

2. CNN 모델 구성

CNN은 영상과 이미지처럼 화면을 구성하는 것에 가장 기본이 되는 단위인 픽셀에 대한 데이터를 처리하여 인식하고 판단하는 것이 강점을 보이는 인공지능망으로써 연산의 방식 자체가 사람과 같지는 않으나 사람의 시각으로 들어오는 정보를 처리하는 방식과 유사한 딥러닝 기반의 네트워크이다. 기존의 FC(Fully Connected layer)에서는 1차원으로 평면화시켜서 데이터를 입력해 주는데, 이러한 방식은 입력되는 데이터의 공간적인 정보의 손실을 발생시킬 수 있다. CNN에서는 각 계층에서의 입출력 데이터의 공간 정보를 유지하면서 인접 픽셀과의 특징을 효과적으로 인식하고, 다양한 필터로 이미지의 특징을 추출하고 학습하는 특성을 지닌다. 이에 따라 입력 데이터에서 특징을 추출하는 부분과 객체의 클래스를 분류하는 부분으로 나누어 설명할 수 있다. 아래 그림 12는 CNN의 전반적인 흐름도를 그림으로 표현한 것이다.

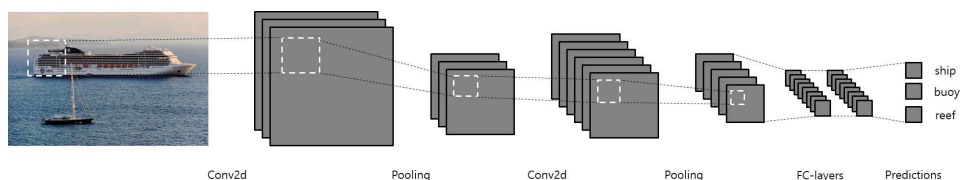


그림 12. CNN 흐름도
Figure 12. Flow Diagram of CNN

CNN은 Convolution 계층과 Pooling 계층의 반복적인 구조로 연산이 진행되며 이러한 연산을 통해 필터를 생성한다. 필터는 CNN에서 학습할 파라미터 값이 되며 이를 학습 결과에 따라 값이 생성된다. 따라서

Convolution, Pooling 계층이 특징을 추출하는 부분이며 FC 계층이 객체의 클래스는 분류하는 부분으로 정의한다. 가장 먼저 Convolution 계층에서는 입력되는 데이터가 RGB 3채널의 데이터라면 이를 합성곱 연산을 진행하고 활성화 함수를 통해 특징을 추출하며 이러한 결과는 Feature map에 저장된다. 그 후 Pooling 계층으로 전달되는데, 합성곱 연산을 통해 특징점이 추출된 결과값인 Feature map은 Pooling 구간에서 더욱 강한 특징 부분만 살아남게 된다. Pooling 방식은 Min Pooling, Average Pooling, Max Pooling 등과 같은 방식이 존재한다. 일반적으로 CNN에서의 Pooling 방식은 Max Pooling 방식이 적용되는데, 각각의 구역에서 가장 큰 값만 살아남게 되는 방식이다. 그뿐만 아니라 Pooling 방식은 학습하는 모델에서 파라미터가 존재하지 않아도 연산이 가능하며 채널의 수가 변화하지 않기 때문에 공간 정보도 유지가 가능하다. 아래 그림 13은 Stride 값이 2일 때의 Max Pooling 방식을 그림으로 간략히 나타낸 것이다.

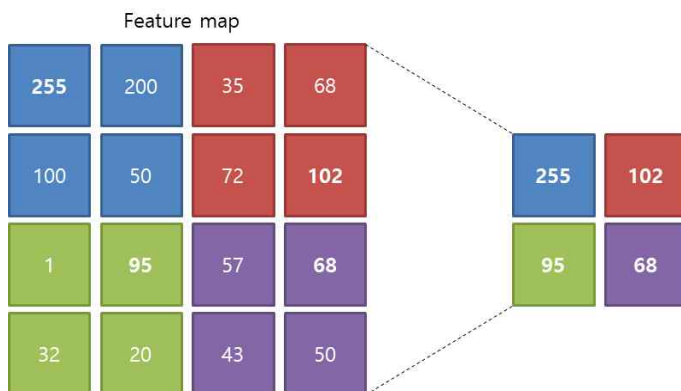


그림 13. Max Pooling 방식
Figure 13. Max Pooling Method

CNN에서는 객체의 특징을 추출하기 위한 정보들을 필터에 저장하고 Feature map은 FC 계층에 입력될 때 1차원으로 입력되어 마지막 계층에서 입력값을 0~1로 정규화해 출력값의 총합을 1로 반환하여 가장 큰 값만 True로 받을 수 있는 Softmax와 같은 함수를 통해 객체의 클래스를 분류하게 된다.

본 논문에서 적용하는 모델 구성으로는 앞에서 설명한 합성곱 신경망 모델을 사용한다. 구성하려는 모델은 해상 환경에서 선박과 같은 부유물을 검출하기에 적합한 모델이어야 한다. CNN의 경우 모델의 깊이가 깊어 질수록 정확도가 높아지는 것이 연구 결과로 이미 상당히 검증되었다. 하지만 학습에 사용되는 전처리 과정이 진행된 데이터 셋이 부유물을 검출하기 적합한 전처리 과정인지 검증하기 위해서 비교적 CNN 모델의 의존성이 낮을 수 있도록 LeNet과 같은 네트워크처럼 깊이가 깊지 않은 신경망을 설계해 실험에 적용하였다.

CNN 기반 모델의 정확도를 높게 가져가기 위해서는 학습을 진행할 모델 구성도 상당히 중요한 요소 중 하나가 되는데, 본 논문에서 구현한 신경망은 선박과 같은 해상 환경에서 존재하는 부유물 분류에 사용될 계층들에 대해 각 계층의 파라미터 수를 포함하여 모델의 흐름대로 다음 표 1에 세부적으로 기술하였다.

Layer	Output shape (Col, Row, Number of Channels)	Number of filter	Parameter
Input shape	(60, 90, 3)	·	·
Conv2D	(58, 88)	32	896
Conv2D	(56, 86)	32	9,248
Max-Pooling	(28, 43)	32	·
Conv2D	(26, 41)	64	18,496
Conv2D	(24, 39)	64	36,928
Max-Pooling	(12, 19)	64	·
flatten	14,592	·	·
FC-layer	·	32	466,976
FC-layer	·	3	99
Total Parameter : 532,643			

표 1. CNN 기반 딥러닝 모델 구성
Table 1. Deep-learning model configuration

해당 모델에서는 Convolution 계층이 4개, Max-pooling layer가 2개, FC 계층의 개수를 2개로 구성하였다. 모델의 깊이가 깊지는 않으나 각 Convolution 계층에서 합성곱 연산을 위해 학습으로 생성되는 필터의 개수를 늘려 입력되는 데이터를 통해 특징점을 더 많이 저장할 수 있도록 조정하였다. 합성곱 필터 개수를 증가시킬 경우, Convolution 계층에서는 특징점을 저장할 공간도 증가하지만, 모델의 파라미터 수가 증가하기 때문에 총 연산량이 증가하게 된다.

특징점을 추출하는데 사용되는 필터 크기는 3x3이며 다음 계층에 전달

되는 입력 데이터의 크기는 줄어들고 깊이가 깊어질수록 필터의 개수는 증가하게끔 진행하였다. 또한, 이에 따라 해당 모델에서의 필터를 통해 나오는 데이터를 1차원으로 나열한 후에 출력으로 연결해주는 FC-layer의 수를 2개 계층으로 지정하였다. 따라서 구성된 모델에서 파라미터가 존재하는 계층은 총 6개 계층이며, 입력 데이터부터 0~1로 정규화(Normalization)를 진행하여 하나의 파라미터마다 float32 값으로 진행되었다.

3. Object Detection 모델 구성

가. SSD

Faster R-CNN처럼 기존 Object Detection에서 속도와 성능이 향상된 알고리즘이 등장하였으나 실질적으로 실생활에서 사용하기에는 속도적인 측면에서 상당히 느린 문제가 존재했다. 이러한 문제점을 YOLO(You Only Look Once)가 등장하면서 Faster R-CNN의 속도보다 6~7배 향상해 비약적으로 발전하였으나 정확도를 향상하기에는 feature map이 하나만 생성되므로 비교적 작은 물체를 감지해내지 못하는 문제가 존재했다. 이러한 문제를 해결하기 위해 SSD(Single Shot multibox Detector)가 나타났다.

SSD는 출력값을 생성하는 공간을 나누어 각 feature map에서 다른 비율과 크기로 default box를 생성한다. 이후 모델을 사용하여 계산된 좌표와 클래스값에 대해 이 default box를 활용하여 최종 그려지는 bounding box를 생성한다. 아래 그림 14는 SSD 논문에서 설명한 기존의 YOLO 및 SSD의 네트워크 구조이다[9]. 기존의 YOLO와 비교하면 각 Convolution 계층마다 feature map을 그저 다음 계층으로만 넘겨주는 것이 아니라, 데이터를 넘겨주는 동시에 각 계층의 feature map을 객체를 검출하는데 사용한다.

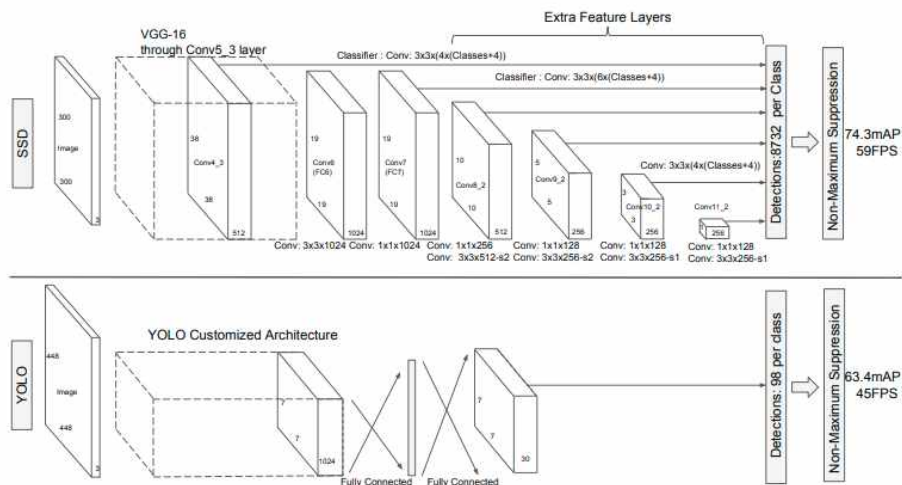


그림 14. SSD와 YOLO의 네트워크 구조

Figure 14. A comparison between two single shot detection models: SSD and YOLO

예를 들어 SSD의 구조를 사용하기 위해 입력 데이터를 300x300으로 가정하면 base network로 사용되는 VGG-16 네트워크를 통과시킨다. 전체를 통과시키는 것이 아닌 중간의 10번째 Convolution 계층(conv4_3)까지 통과되어 생성된 feature map을 SSD의 구조에서 사용한다. 이는 300x300의 3채널 이미지의 경우 conv4_3을 통과하면 38x38 크기의 512개 feature map이 생성된다. 따라서 이렇게 생성된 데이터부터 1x1의 크기까지 Convolution을 진행하며 각 단계에서 생성된 feature map을 바탕으로 객체 인식을 수행한다. 따라서 YOLO의 경우 7x7 크기의 grid 하나만 존재하지만, SSD의 경우 38x38부터 19x19, 10x10, 5x5, 3x3, 1x1까지 여러 개의 grid로 분할하여 객체를 검출하는 데 사용한다. 즉, 앞에서 설명한 바와 같이 FC 계층을 통과하면서 세부적인 데이터가 손실되는 문제를 SSD는 앞에서 생성된 feature map을 객체 검출에 끌어와 함께 적용함으

로써 해결하였다. 따라서 기존 YOLO의 입력 크기인 448x448보다 작은 300x300으로도 더 높은 정확도와 속도를 가질 수 있다[9].

본 논문에서는 이러한 SSD를 사용하여 객체 검출을 진행한다. 이는 전체 모델에 대해 양자화를 진행할 때 기존 Object Detector보다 정확도와 속도 측면에서 준수한 성능을 가졌으며 각 계층에서 feature map을 생성하고 검출하는 네트워크의 계층 구조가 가장 단순한 Convolution 계층으로 쉽게 연결되어 있으므로 각 단계의 파라미터를 양자화하기에도 유리하다.

나. MobileNet v2

딥러닝의 연구가 활발히 진행되면서 정확도를 향상하는 부분에서는 비약적으로 발전이 되었으나 실생활에서 사용하기 위해서는 실시간적인 처리 속도가 발목을 잡았다. 이를 해결하기 위해서는 고성능의 하드웨어를 사용하면 해결이 가능하였으나, 연산량이 증가함에 따라 전력 소모가 극심해졌고 이는 결국 상용화 단계에서 문제가 되었다. 따라서 기존 네트워크의 연산량이나 복잡도를 해결하고, 이에 따라 처리 속도를 증가할 수 있는 네트워크를 연구하는 경량화 네트워크에 대해 연구가 진행되기 시작했다.

MobileNet은 네트워크 구조를 변경하여 이러한 문제점을 개선하기 위해 나타난 네트워크로써 Convolution 연산 방식을 기존과 차별화하여 나타난 네트워크이다. 이보다 먼저 개발되었던 VGGnet에서는 필터에 대해 개선이 되었는데, 이는 5x5나 7x7의 필터 크기를 3x3으로 줄여서 연산하였을 때 연산량도 적고 데이터의 비선형성도 다양해져 데이터의 유실도

적다는 결과를 내었다. 기존에 진행하던 Convolution 연산 방식은 3x3 필터를 사용하여 각 채널의 모두 연산을 진행하여 합치지만 MobileNet은 각 채널 별로 연산하고 해당 위치에 그대로 출력해준다. 이러한 방식을 Depthwise Convolution이라고 정의하며 각각에 위치에 출력된 3개의 값을 1x1 Convolution 연산을 진행하여 묶어주는 방식을 Pointwise Convolution이라 정의한다. MobileNet에서는 이 둘을 합쳐 Depthwise Separable Convolution을 사용하여 연산량을 감소시켰다.

MobileNet v2의 경우는 기존 MobileNet의 구조를 사용하지만 Linear Bottlenecks 구조를 사용하여 정확도를 향상시키고, Inverted Residuals 구조를 적용하여 이전보다도 더 연산량을 감소시켰다[10].

본 논문에서 사용하는 Mobilenet v2는 다른 네트워크에 비해서 규모가 작아 메모리 사용량이 적으며 네트워크 구조 자체가 연산량을 상당히 감소시켰기 때문에 이미지 한 장을 처리할 때 처리 속도를 빠르게 가져올 수 있다는 장점이 존재하므로 고성능이 아닌 저전력의 임베디드 보드에서도 적용하기에 적합하다.

4. 경량화 알고리즘 : 양자화

양자화는 샘플링에서 연속적인 데이터를 이산적인 데이터로 표현하는 것처럼 파라미터값을 설계하고자 하는 비트로 압축하는 것으로, 일반적인 CNN 기반의 모델은 가중치와 같은 파라미터가 부동 소수점으로 구성되어 있다. 양자화 기법은 모델의 이러한 부동 소수점의 비트를 줄여서 연산량을 감소시키는 방식이다. 일반적인 합성곱 신경망의 경우에는 Convolution layer, FC layer처럼 가중치와 편향 값을 연산해야 하는 계층에서 파라미터값을 가지게 되는데 이러한 파라미터의 값들은 일반적인 모델에서 float32의 자료형에 저장되게 된다. 부동 소수점으로 구성되는 파라미터들은 일반적으로 모델이 가지고 있는 가중치에서 그 값이 작을 경우에는 모델에 정확도에 대해서 큰 영향을 주지는 않는다. 따라서 이러한 양자화 기법을 신경망 모델에 적용할 때는 정확도에 큰 영향을 주지 않는 선에서 모델의 경량화를 진행하여야 저전력 임베디드에 적용하는 데 의미가 있다.

모델에서 생성되는 이러한 파라미터 값은 0에 근사하게 정규화 과정을 거쳐 실질적인 객체를 분류하기 위한 값으로 데이터가 저장되게 되는데, 모델의 계층이 깊어지고 각 계층에서 파라미터의 비트가 많아질수록 연산량이 급격하게 늘어나게 된다. 즉, float32라는 자료형에 weights와 bias 값이 저장되는데 이러한 값들의 bit를 조절함으로써 메모리 용량을 절약하고 연산속도는 증가시킬 수 있다는 것이 양자화의 장점이다. 특히 영상과 같은 픽셀 데이터 처리 기반의 CNN과 같은 신경망 모델은 입력 데이터의 크기가 크거나 필터의 수가 많을수록 합성곱 계층에서의 데이터 연산량이 상당히 높아 임베디드 실험 보드에서 실시간적인 처리가 어려운데

양자화를 적용한다면 한 파라미터의 값마다 데이터 용량이 감소함으로써 저전력 임베디드와 같은 하드웨어에서 속도와 메모리 용량에서 이점을 볼 수 있게 된다[11].

본 연구에서는 학습을 진행하는 것은 float32에서 진행하고 결괏값으로 추출되는 파라미터 값들에 대해 양자화를 적용하는 학습 후 양자화 기법으로 진행하였으며, 이러한 방법은 파라미터의 수가 많을수록 비교적 정확도 하락 폭이 작아지는 특성이 있다. 양자화는 좀 더 세밀하게 표현할 수 있는 float32의 값들을 비교적 일정한 크기의 단위로 재배치하는 것이므로, 8bit로 양자화를 진행할 때 overflow 현상이 발생하면 객체를 정확하게 인식하지 못할 수 있으므로 이를 고려해 범위를 조절하여 압축을 진행하여야 한다.

float32에서 uint8의 형태로 양자화를 적용하는 경우에서 Convolution layer 하나의 계층에서 보면 다음 계층으로 데이터를 전달하기까지의 전체적인 흐름은 그림 15와 같다.

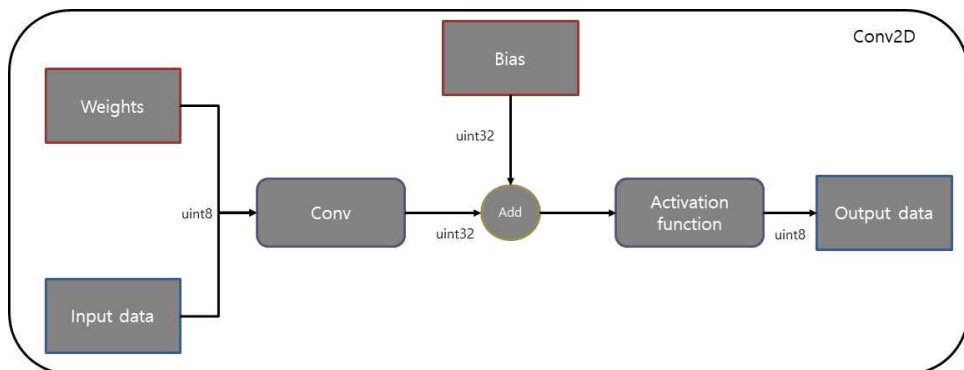


그림 15. uint8의 양자화 흐름도
Figure 15. Quantization Flow Diagram

실질적인 합성곱 연산에 사용되는 입력 데이터와 weights 값들이 0~255의 수로 표현되는 uint8로 데이터가 변환되고, 결과 값은 uint32에 저장되게 된다. 그 후 bias 값이 더해지게 된다. 이 bias 값의 경우 너무 작은 비트의 값으로 압축을 시킬 경우 정확도 하락에 큰 영향을 미칠 수 있다. 그 이유는 다음 계층에 출력값을 전달해줘야 하는데 객체의 클래스를 보다 정확하게 맞추기 위해서 활성화 함수를 사용하여 다음 계층에게 전달해줘야 한다. 이 과정에서 bias 값에 대한 정보는 weights의 데이터보다는 비교적 세밀하게 표현이 유지되어야만 활성화 함수의 그래프 이동이 가능해 객체를 분류하는 데 중요한 역할을 하므로 bias 값은 32bit로 유지하되 weights와 입력 데이터의 곱값과 연산을 위해 uint32로 저장한다.

이렇게 더해진 weights와 bias 값은 다시 uint8로 변환하여 Convolution layer 계층의 마지막 단계인 활성화 함수를 통해 출력되는 데이터가 다음 계층에게 입력 데이터로 전달되게 된다[11].

이렇게 출력되는 데이터는 각 계층에서의 피쳐 맵(feature map)으로 정의되며 필터 연산을 통해 추출된 특징점들을 저장할 수 있는 공간이 된다.

V. Neural Network 실험 결과

1. 실험 환경

본 논문에서 제안하는 전처리 데이터 셋의 정확도를 확인하기 위해서 Tensorflow를 백엔드로 사용하는 Keras를 사용하였으며, 네트워크를 학습하고 양자화를 적용하기 위해서는 Tensorflow를 사용하였다. 또한 성능 분석을 위해 필요한 저전력 임베디드 보드로는 Coral Dev board 및 Raspberry 4를 사용하였다. 이들을 사용할 때 고정 소수점 연산에 특화되어 있는 TPU를 사용함으로써 성능을 확인한다.

아래 표 2는 본 논문에서 실험에 사용된 임베디드 보드의 사양을 간략하게 나타낸 것이다. 모두 Linux 기반의 운영 체제를 사용하며 Coral Dev Board는 TPU가 내장되어 있으나 Raspberry의 경우 TPU Accelerator를 추가로 USB로 연결하여 사용한다.

종 류	CPU	GPU/TPU	RAM
Raspberry 4 b	1.5GHz ARM Cortex-A72	Broadcom VideoCore IV	4GB LPDDR4
Coral Dev board	NXP i.MX 8M SoC (quad Cortex-A53, Cortex-M4F)	Integrated GC7000 Lite Graphics	1GB LPDDR4

표 2. 임베디드 보드 사양

Table 2. Embedded Board Specifications

2. 실험 결과

가. 전처리 별 모델 정확도 결과

본 논문에서는 해양 환경의 객체를 검출해낼 때 적절한 전처리 방법을 확인하기 위하여 4가지 전처리 방법을 데이터 셋에 적용하였고, 이후 CNN 모델에 필터의 개수를 증가시켜 feature map을 추출하는 개수를 증가시켰다.

아래 표 3은 각 전처리를 진행한 데이터 셋을 바탕으로 같은 CNN 기반의 모델 구조에서 학습시킨 결과를 나타낸 것이며 모델의 총 파라미터 개수는 532,643개로 동일하다. 모델의 결과를 확인하기 위해서 사용된 데이터는 해상 부유물 이미지 데이터로 총 8,000장의 데이터가 사용되었다. 그 중 10%를 검증 데이터로 활용하였으며 아래 표는 이러한 검증 데이터를 사용하여 나타낸 결과이다. 여기서 정의되는 정밀도는 해당 모델이 선박이라 분류하였을 경우에, 실제 입력된 데이터가 선박일 경우의 확률을 의미한다.

모 델	정확도(accuracy)	정밀도(precision)
Grayscale	85.66%	96.84%
Histogram	90.39%	96.26%
ZCA	87.50%	97.23%
8bit Color	88.2%	96.12%

표 3. 각 전처리별 모델 성능

Table 3. Model performance for each Preprocessing

그림 16 ~ 17은 정확도 및 정밀도에서 준수한 성능을 보이는 히스토그램 데이터 셋으로 학습한 모델을 파라미터가 존재하는 처음 계층부터 FC feature map을 추출하여 0~255의 값으로 재구성하여 시각화하였다. 그림 16의 경우 32개의 필터부터 Pooling을 한번 진행한 부분까지의 흐름도이며 그림 17의 경우 그림 16에서 출력되는 데이터를 입력으로 받고 필터가 64개로 증가한 것을 나타낸 그림이다. 이러한 모델의 전체적인 흐름 및 필터 생성을 확인하면 학습의 결과를 확인할 수 있다.

그림의 x, y축은 픽셀 크기를 나타내며 padding 기법을 적용하지 않았기 때문에 3x3 필터를 통해 연산이 진행되면 가장자리 픽셀 크기가 1씩 감소하게 된다. 그림에서 활성화가 되지 않은 feature map들은 해당하는 입력 이미지에 활성화가 진행되지 않았기 때문에 단색으로 표현이 되었다. 전처리 결과를 판단하기 위하여 사용한 데이터의 클래스는 선박, 암초, 부이로 나뉘어진 총 3개이므로, 그에 대해 판단이 적절히 이루어지기 위해서는 feature map의 활성화가 적절히 이루어져야 한다. 따라서 그림 16 ~ 17을 확인해보면 선박의 이미지가 입력 데이터로 활용되었을 경우 feature map에서 선박이 존재하는 부분의 픽셀값과 선박이 존재하는 부분을 제외한 나머지 부분과의 픽셀값 차이가 많이 나타나는 것으로 확인되었고, 이는 모델이 해당 입력 데이터에 대해서 객체를 정확하게 판단하는 것으로 확인할 수 있다.

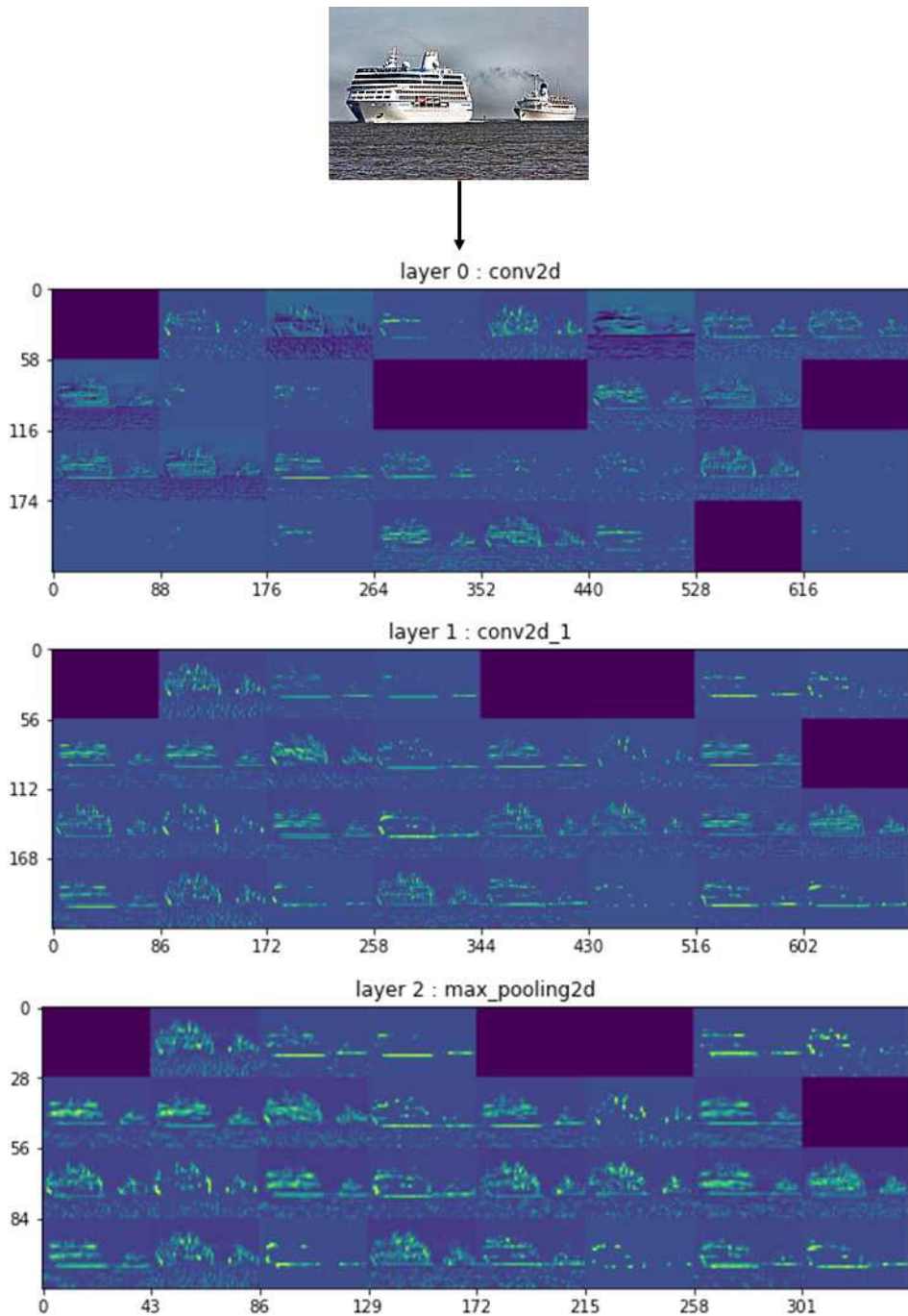


그림 16. 히스토그램의 Feature map (1)
Figure 16. Feature map of Histogram (1)

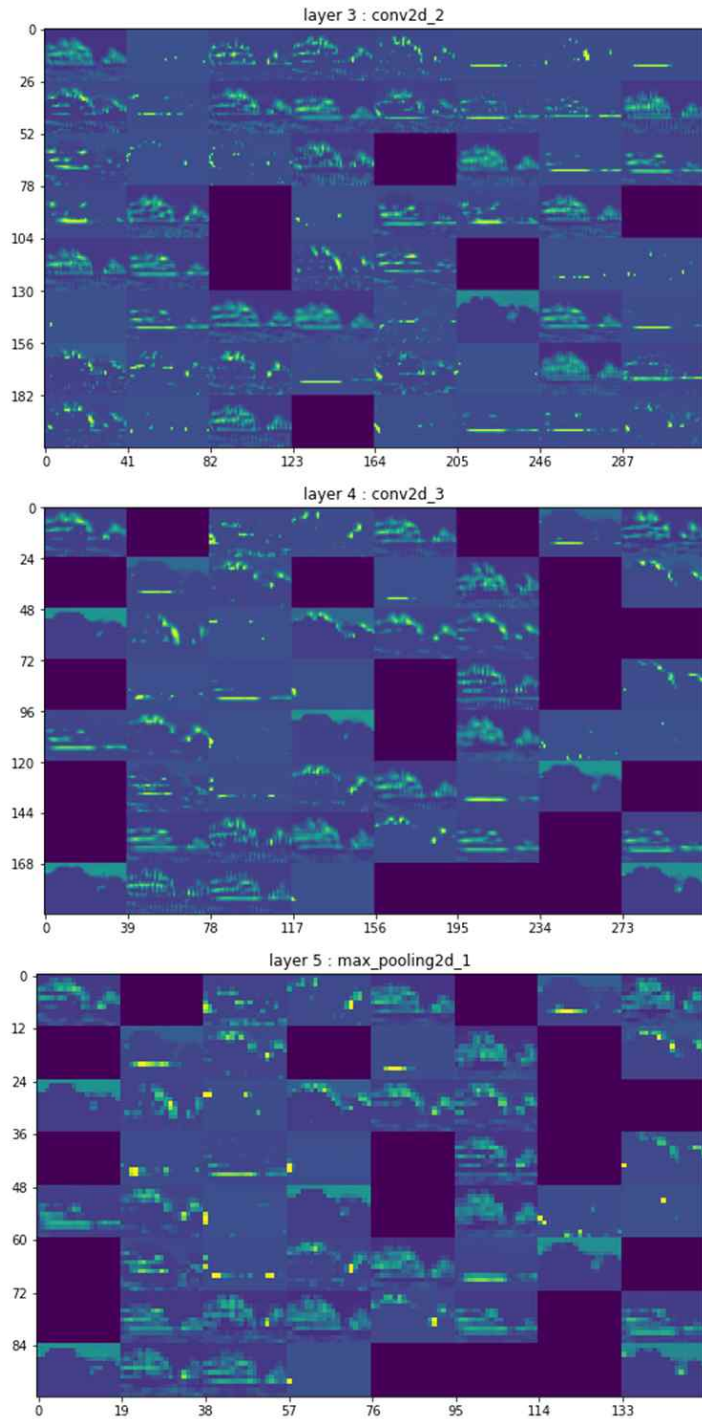


그림 17. 히스토그램의 Feature map (2)
Figure 17. Feature map of Histogram (2)

나. 객체 인식 네트워크 모델 성능 분석

본 논문에서 실험한 전처리 결과를 바탕으로 SSD 기반의 MobileNet v2에 학습을 진행한다. 모델 학습에는 선박, 부이 및 암초와 같은 전체적인 부유물 3개의 클래스로 앞서 진행하였던 CNN 모델의 데이터 셋과 같이 총 8,000장이 사용되었으며 이 중 10%는 검증 데이터로 사용되었다. 아래 그림 18~20은 양자화를 적용하기 전에 학습이 완료된 모델을 활용하여 임의의 데이터 중 10장의 이미지를 입력으로 전달하여 추론한 결과를 나타낸 사진이다.

초록색 박스는 선박을 의미하며 하늘색 박스는 부이를 나타내는 라벨이다. 그림으로 확인하였을 경우 선박과 부이와 같은 부유물에 대해서 비교적 잘 판단하는 것을 확인할 수 있다.



그림 18. SSD MobileNet V2 추론 결과 (1)
Figure 18. SSD MobileNet V2 Inference Result (1)

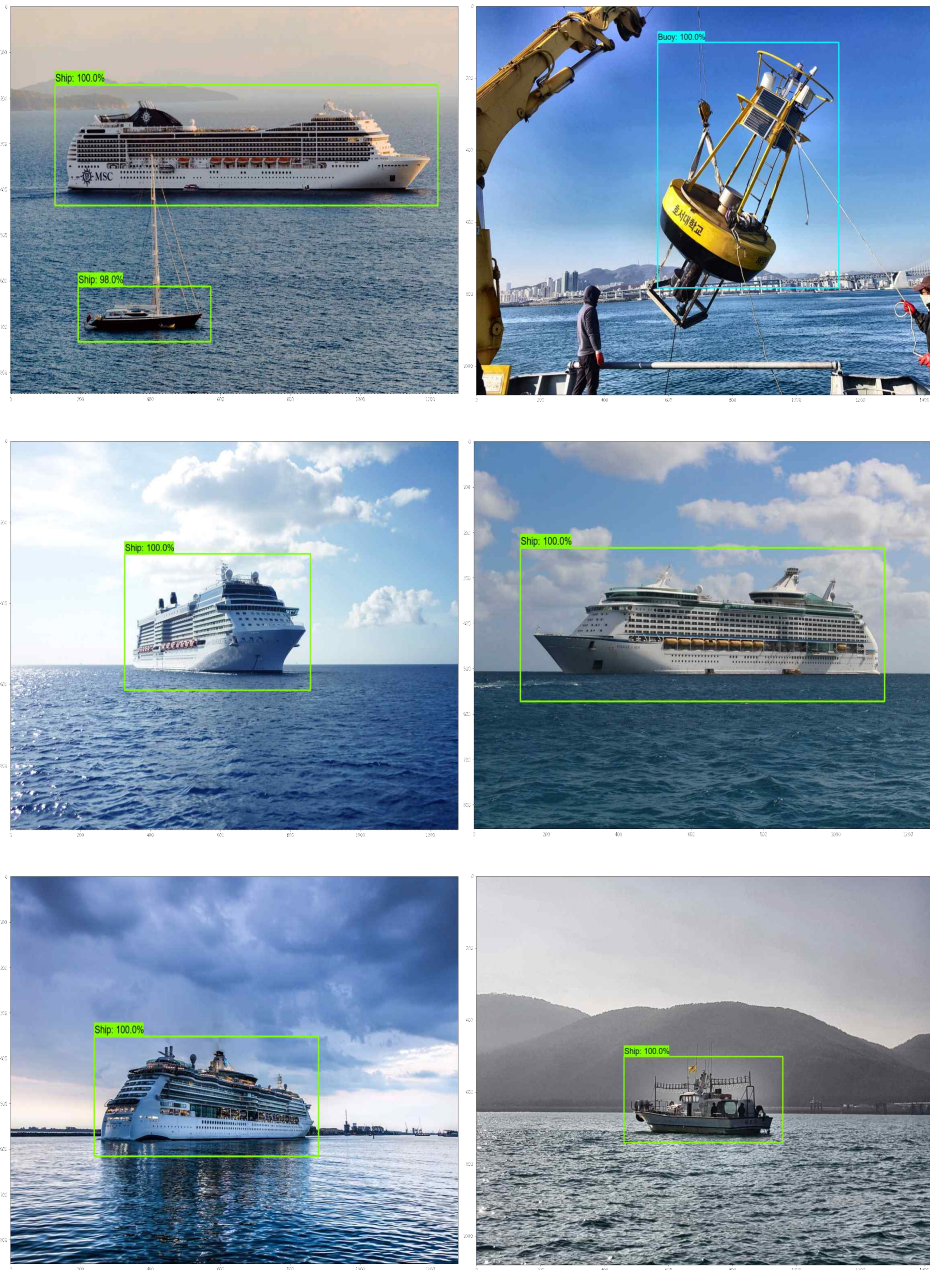


그림 19. SSD MobileNet V2 추론 결과 (2)
Figure 19. SSD MobileNet V2 Inference Result (2)

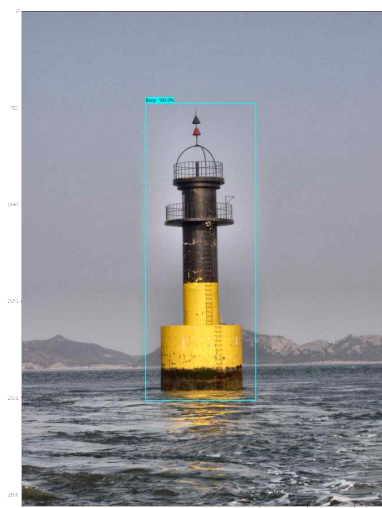
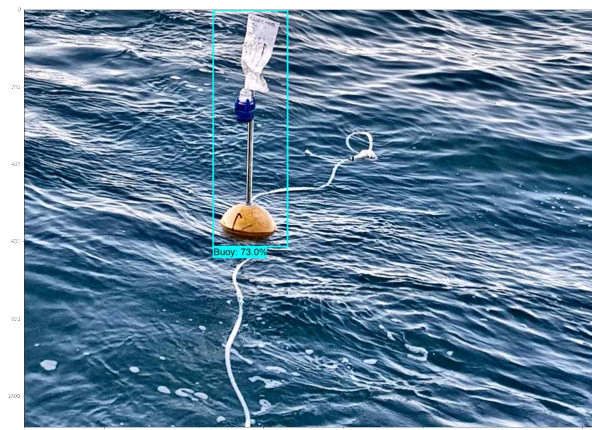
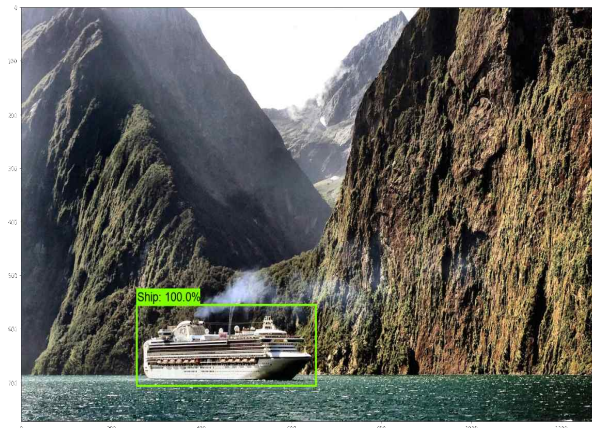


그림 20. SSD MobileNet V2 추론 결과 (3)
Figure 20. SSD MobileNet V2 Inference Result (3)

다. TPU 기반 양자화 적용 객체 인식 모델 성능 분석

학습시킨 SSD MobileNet v2 모델에 대해서 고정 소수점 연산에 특화된 가속기를 적용하기 위해 양자화를 적용한다. 따라서 같은 모델을 tflite 파일로 변환하여 비트 수를 조절하고 그에 대한 모델을 확인한다. 아래 그림 22 ~ 25는 uint8로 양자화를 진행한 모델로 추론을 진행하여 나타난 결과이다. 선박의 경우는 기존 모델과 거의 동일하게 판단하는 것처럼 보이나 그림 21과 같이 암초로 분류한 객체가 다른 클래스로 인식되는 경우가 존재하였으며, 부이의 경우 같은 사진이더라도 그림 20에서는 73%, 그림 23에서는 85%의 확률로 판단하는 것을 확인할 수 있었다.



그림 21. SSD MobileNet V2 uint8 양자화 추론 결과 (1)

Figure 21. SSD MobileNet V2 uint8 Quantization Inference Results (1)

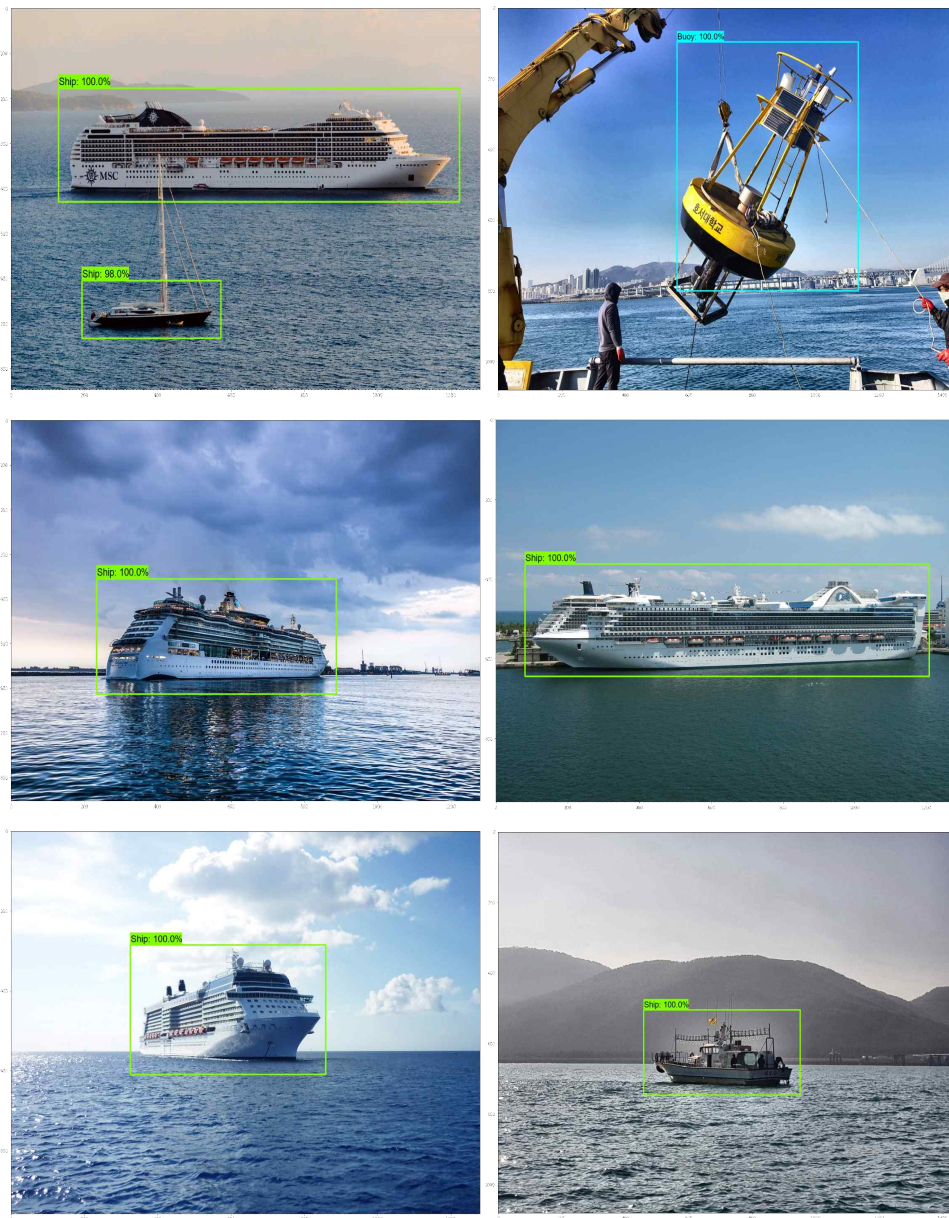


그림 22. SSD MobileNet V2 uint8 양자화 추론 결과 (2)

Figure 22. SSD MobileNet V2 uint8 Quantization Inference Results (2)

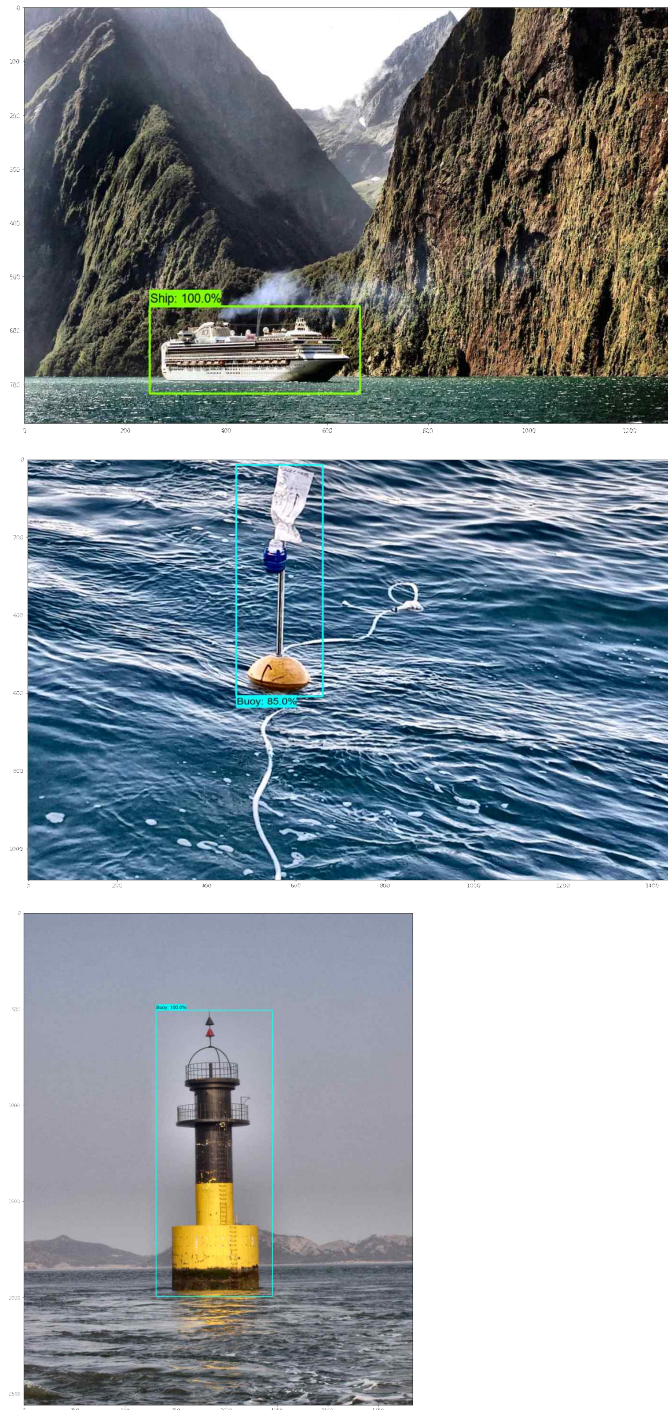


그림 23. SSD MobileNet V2 uint8 양자화 추론 결과 (3)
Figure 23. SSD MobileNet V2 uint8 Quantization
Inference Results (3)

아래 표 4는 라즈베리파이에서 진행하고 표 5는 Coral Dev Bo 양자화를 적용하기 전의 모델을 포함하여 양자화를 적용한 각 비트별로 정확도 및 처리 속도를 표현하였다. 모델의 평가를 위해 검증 데이터를 활용하였으며, 처리 속도는 모델이 입력된 이미지 데이터 중 하나를 추론하는데 걸리는 시간을 의미한다. FPS는 Frames Per Sec를 의미한다. 두 임베딩 보드에서는 정확도가 동일하게 나타났으며, 처리 속도의 경우 8비트로 양자화를 적용하였을 경우에 정확도의 손실은 미세하게 저하되지만 약 5배 정도의 처리 속도 향상이 되는 것을 확인하였다.

모 델	정확도(accuracy)	처리 속도	FPS (Frames Per Sec)
float32 (CPU)	87.21%	0.145	6.89
float16 (CPU)	87.21%	0.145	6.89
uint8 (CPU)	82.54%	0.175	5.71
uint8 (TPU)	82.54%	0.028	35.71

표 4. Raspberry 환경 성능 분석

Table 4. Model performance for each Preprocessing

모 델	정확도(accuracy)	처리 속도	FPS (Frames Per Sec)
float32 (CPU)	87.21%	0.088	11.36
float16 (CPU)	87.21%	0.088	11.36
uint8 (CPU)	82.54%	0.096	10.41
uint8 (TPU)	82.54%	0.018	55.5

표 5. Coral Dev Board 환경 성능 분석

Table 5. Model performance for each Preprocessing

VI. 결 론

본 논문은 딥러닝 CNN 모델을 해상 환경에서 실시간으로 객체 인식을 하기 위해 전처리 및 양자화를 적용하였으며 전처리를 통해 질적 데이터로 가공하고 객체 인식 모델에 양자화를 적용하여 정확도 손실은 최소화하면서 처리 속도를 개선하는 방법을 제안하였다.

입력되는 전처리 방법으로는 Gray Scale, Histogram equalization, ZCA whitening, 8bit color image 등을 적용하였고 그 중 Histogram equalization 방식이 정확도가 가장 좋았던 것을 확인하였다. 따라서 학습 데이터의 질적 향상이 필요할 경우 Histogram equalization 방식의 특징을 적용하면 해상 배경과 부유물의 경계선 처리가 확연해지고 이는 객체의 분별력이 향상된 것을 실험으로 확인할 수 있었다. 따라서 본 논문에서는 해당 전처리 방식을 활용해 양자화를 적용하여 정확도의 손실을 최소화하면서 기존 모델보다 속도는 최대 5배 정도 성능을 개선할 수 있었다.

현재 연구에서는 Tensorflow를 활용하여 양자화를 진행하였으며 지원되는 타입은 float32, 16으로 소프트웨어를 지원하고 정수 전용 가속기를 활용할 경우 uint8 형식의 고정소수점 형태로 가속을 진행하였다. 향후 FPGA에서 8비트 이하로 고정소수점을 지원할 수 있도록 소프트웨어를 설계하여 저전력 임베디드 보드에서도 고성능이 지원되는 해양 환경 딥러닝 모델을 개발할 예정이다.

참고문헌(Reference)

- [1] S. Li and K. S. Fung, “Maritime autonomous surface ships (MASS): implementation and legal issues,” *Maritime Business Review*, Vol. 4, No. 4, pp. 330–339, Nov. 2019.
- [2] J. S. Lee, S. K. Lee, D. W. Kim, S. J. Hong, and S. I. Yang, “Trends on Object Detection Techniques Based on Deep Learning”, *Electronics and Telecommunications Trends*, vol. 33, no. 4, pp. 23–32, Aug. 2018.
- [3] Y. J. Lee, Y. H. Moon, J. Y. Park, and O. G. Min, “Recent R&D Trends for Lightweight Deep Learning” *Electronics and Telecommunications Trends*, Vol. 34, No. 2, pp. 40–50, Apr. 2019.
- [4] S. W. Lee, G. D. Lee, J. G. Ko, S. J. Lee, and W. Y. Yoo, “Recent Trends of Object and Scene Recognition Technologies for Mobile/Embedded Devices,” *Electronics and Telecommunications Trends*, vol. 34, no. 6, pp. 133–144, Dec. 2019.
- [5] C. Cui, X. Wang, and H. Shen, “Improving the face recognition system by hybrid image preprocessing,” in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2016.
- [6] Kuntal Kumar Pal, Sudeep K. S, “Preprocessing for Image Classification by Convolutional Neural Networks”, in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2016.

- [7] 안성원, 추형석, 김수형, “Deep Learning Tutorial - Understanding Neural Network & Deep Learning”, 소프트웨어정책연구소, 2017
- [8] <http://times.kaist.ac.kr/news/articleView.html?idxno=4675>
- [9] W. Liu et al., “SSD: Single Shot MultiBox Detector,” in Computer Vision - ECCV 2016, Springer International Publishing, pp. 21 - 37, 2016.
- [10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
- [11] B. Jacob et al., “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
- [12] S. Y. Kahu and K. M. Bhurchandi, “JPEG-based Variable Block-Size Image Compression using CIE $L^*a^*b^*$ Color Space”, KSII Transactions on Internet and Information Systems, Vol. 12, No. 10, pp. 5056-5078, 2018

ABSTRACT

A Study on CNN Deep Learning Model System Based on Quantization Application for High-Speed Marine Object Detection

Seong - Ju Lee

Department of Ocean Convergence technology, The Graduate School

Hoseo University

Asan, Korea

(Supervised by professor Tae-Ho Im)

With the recent development of hardware technology, artificial intelligence technology is gaining attention and starting to be applied to autonomous driving, and studies that apply CNN-based deep learning models that are specialized in images are being studied. In particular, in the case of autonomous navigation vessels, real-time processing is important because systems that can be operated safely without having to directly care about safety such as collision prevention and fire are incorporated. However, the more features are added, the more high-speed processing of the entire processor is possible only when it is high performance, so a lightweight method is needed to apply the highly complex deep learning

model. Therefore, in this paper, we propose a method for implementing a deep learning model to enable real-time processing while maintaining accuracy. First, we conduct preprocessing for the detection of suspended solids at sea and utilize it in the learning and inference of deep learning models. Furthermore, after learning, the model's parameters are quantified to increase the processing speed while reducing the model's memory. We then proceed with inference on low-power embedded boards to compare accuracy and processing speed for each model. We utilize the best-performing models among the proposed methods to improve processing speed by approximately four times by applying them to the board.

