

Edge Computing을 이용한 딥러닝 기반 마케팅 타깃 검출 및 분석 시스템

곽수지 · 김연우 · 신우정 · 이채원

세종대학교 지능기전공학부

Deep Learning Based Marketing Target Detection and Analysis System Using Edge Computing

Suzy Kwak · Yeon-Woo Kim · Woo-Jung Shin · Chae-Won Lee

School of Intelligent Mechatronics Engineering, Sejong University

Abstract

본 프로젝트는 길거리에서 흔히 접할 수 있는 디지털 광고판 상단에 임베디드 시스템 일체형 카메라를 설치하여 마케팅 타깃을 검출/추적 및 분석하는 시스템을 제안한다. 임베디드 시스템 일체형 카메라 한 대로 마케팅 타깃 검출/추적이 가능하며 CNN 기반 검출기를 통해 광고판 앞 남/여 보행자와 남/여 광고 응시자를 검출이 가능함을 확인했다. 카메라에 적용한 검출 모델을 실시간 시스템에 적용하기 위해 Channel Pruning과 양자화 인식 학습 기법(Quantization-aware Training)을 적용하여 경량화하였다. 검출 결과에 추적 알고리즘과 계수 알고리즘을 적용하여 광고판 앞 남/여 보행자와 남/여 광고 응시자를 카메라 내부 CPU에서 실시간으로 계수했고, 계수 결과를 LAN 통신을 통해 PC로 전송하여 현재 광고에 대한 마케팅 효과 분석과 앞으로의 마케팅 전략을 위한 웹 서비스를 제작하였다. 웹 서비스를 통해 하루 동안의 광고판 앞 통행량, 보행자 중 광고 응시자 비율 등의 정보를 다양한 그래프를 통해 확인할 수 있다.

1. 서론

여러 디지털 매체의 발달과 성장으로 디지털 광고 시장은 전년 대비 13 % 고성장을 기록하며 광고 시장을 이끌고 있다. 하지만, 모바일 기기나 PC를 통한 광고와 다르게 야외 디지털 광고는 광고 효과를 제대로 파악하기 어렵다는 단점이 있다. 따라서 야외 디지털 광고 또한 광고 효과 분석이 필요하다고 판단했고, 이를 위해 본 프로젝트에서는 광고판 중앙 상단에 임베디드 시스템 일체형 카메라를 설치하여 현재 송출되는 광고 효과 분석과 앞으로의 마케팅 전략 수립을 위한 자료를 웹을 통해 제공하고자 한다. 임베디드 시스템 일체형 카메라에서 검출할 대상은 남자 보행자, 여자 보행자, 남자 광고 응시자, 여자 광고 응시자로 총 네 가지이다. 보행자를 남/여로 구분하여 검출했고, 보행자의 광고 응시 여부를 판단을 위해 남/여의 정면 얼굴을 검출하였다. 검출 결과를 이용하여 객체를 추적하기 위해 SORT(Simple Online and Realtime Tracking) Algorithm을 사용했다. 객체의 검출, 추적, 계수 시스템을 모두 임베디드 카메라 안에서 구현하기

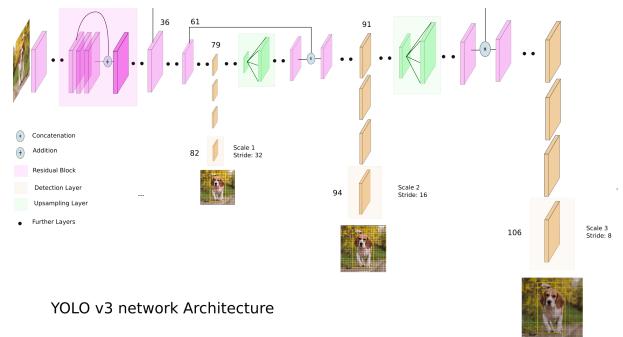
위해 Channel Pruning과 Quantization-aware Training(QAT)을 검출 모델에 적용한 후 임베딩하였다. 실험 진행 결과, 제안한 70 % Channel Pruning과 양자화가 적용된 YOLO v3 검출 모델은 시연 환경에서 79.37 %의 성능과 영상 당 약 70ms 처리 속도를 보이며 실제 환경에서 사용 가능함을 확인하였다. 검출 및 추적에 따른 계수 결과를 웹을 통해 보여줌으로써 통행량과 광고 응시자의 비율 등에 관한 통계를 확인할 수 있다. 웹 서비스를 이용하여 현재 송출되는 광고에 대한 마케팅 효과를 파악할 수 있고, 앞으로의 마케팅 전략 수립을 위한 기초 자료로 활용이 가능하다. 본 논문의 구성은 다음과 같다. 2장에서는 CNN 기반 검출기인 YOLO v3와 검출 모델 최적화 기법인 Channel Pruning과 Quantization-aware Training에 대해 기술한다. 3장에서는 본 프로젝트에 적용한 객체 추적 알고리즘인 SORT Algorithm에 관해 설명하고, 4장에서는 보행자 및 광고 응시자 계수 알고리즘에 대해 기술한다. 5장에서는 사용한 임베디드 시스템 일체형 카메라를 제안하며, 6장에서는 1~5장에서 언급한 내용을 바탕으로 본 프로젝트 실험 내용에 대해 자세히 기술한다.

2. CNN 기반 검출 시스템

2.1 검출기

객체 검출(Object Detection)이란 입력 영상이 주어질 때, 모든 검출 대상에 대해서 분류(classification)와 위치 정보(localization)를 찾는 문제이다. 최근 대부분의 객체 검출 기술은 CNN(Convolutional Neural Network) 기반의 다중 레이어 구조 모델로 연구 및 개발이 활발하게 진행되고 있다. 검출 방식은 크게 두 가지로 나뉘는데, 관심 후보 영역 추출과 관심 영역 추출 후 관심 영역에 대한 분류 및 위치를 추정하는 2-Stage 방식과 2-Stage 방식의 두 가지 task가 동시에 이루어지는 1-Stage 방식이 있다. 2-Stage 방식은 연산량이 많아 검출 속도가 느려 실시간 적용이 어렵다는 단점이 있다. 본 프로젝트는 실시간 시스템에 적용하기 위해 1-Stage 방식의

YOLO v3 검출기 [1]를 기반으로 모델을 제작하였다. YOLO v3 검출기는 물체 인식(Object Detection)을 수행하기 위해 고안된 심층 신경망으로서 영역 추출과 분류를 동시에 실행하는 통합 인식(Unified Detection) 검출기이다. SSD(Single Shot multibox Detector)와 비슷한 mAP 성능을 보이지만 속도는 3 배 더 빠르다고 알려져 있다. 현재 YOLO v5까지 발표되었으며 본 프로젝트에 적용한 검출기는 YOLO v3이다. YOLO v3는 53개의 layer를 가지는 Darknet-53을 Backbone Network로 사용한다. Darknet-53에서는 YOLO v2의 Backbone Network인 Darknet-19에 ResNet [7]에서 제안된 Skip Connection 개념을 적용하여 레이어를 훨씬 더 많이 쌓은 구조이다. <그림 1>은 YOLO v3의 네트워크 구조를 보여준다.



<그림 1> YOLO v3 네트워크 구조

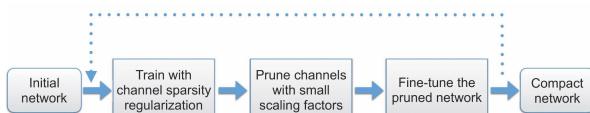
본 프로젝트에서 검출할 대상은 남자 보행자, 여자 보행자, 남자 광고 응시자, 여자 광고 응시자로 총 4 가지이다. <그림 2>는 검출기를 통해 검출할 대상을 보여준다.



<그림 2> 검출 대상

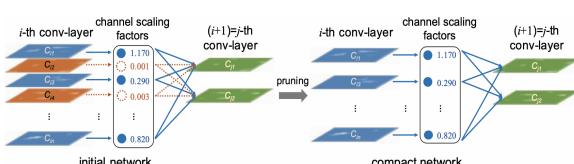
2.2 딥러닝 네트워크 최적화

제작한 검출기를 임베디드 시스템 일체형 카메라에서 실시간으로 처리하기 위해서 네트워크 최적화 과정이 필요하다. 네트워크를 최적화하면 모델 성능은 최대한 유지하면서도 네트워크 크기를 줄이며, 빠른 추론 속도를 얻을 수 있다는 장점이 있다. 본 프로젝트는 [3]을 기반으로 YOLO v3 검출기 최적화를 수행하였으며, Sparsity Training, Channel Pruning, Fine-tuning 순서로 최적화를 진행하였다. Fine-tuning 과정에서 Quantization-aware Training(QAT)를 도입하여 학습을 진행했다. <그림 3>은 YOLO v3 기반 검출기 최적화 과정을 보여준다.



<그림 3>검출기 최적화 과정

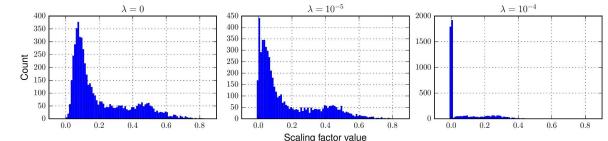
일반적으로 딥러닝 네트워크는 사람에 의해 수동으로 설계되는데, 이는 결과적으로 중복되거나 중요하지 않은 요소들을 포함할 수 있다. 따라서 네트워크 성능에 크게 관여하지 않는 불필요한 채널들을 제거하면 성능 하락을 최소화하면서 네트워크를 가볍게 만들 수 있다.



<그림 4> Channel Pruning 적용

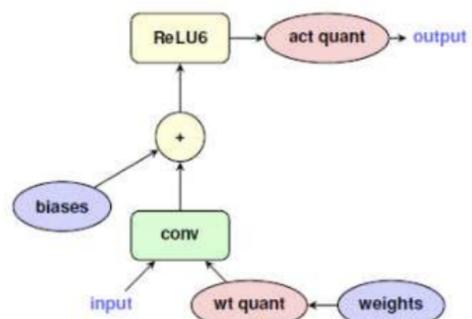
먼저, 불필요한 채널 판정을 위해 각 채널에 scaling factor를 부과하여 채널 별 희소성을 표현했다. <그림 4>는 Channel Pruning 적용 전후의 layer 구성을 보여준다. Batch Normalization의 scale factor Γ 를

scaling factor로 설정하고, 이 절대값을 채널 중요도의 척도로 설정했다. 성능 개선에 중요한 채널과 그렇지 않은 채널을 효과적으로 구별하기 위해 Γ 에 L1 정규화를 도입한 Channel-wise Sparsity Training을 진행했다.



<그림 5> 파라미터 Γ 별 Sparsity Training 결과

결과적으로 많은 scaling factor가 0에 가까운 모델을 얻게 되며, scaling factor가 거의 0에 가까운 채널들을 전역 임계 값을 도입하여 제거하는 Channel Pruning을 진행했다. <그림 5>에서 파라미터 Γ 에 따른 Sparsity Training 결과를 확인할 수 있다. Quantization-aware Training은 딥러닝 모델 학습을 진행할 때, 추론시 양자화 될 것을 미리 고려하여 학습하는 방식이다. 16비트 소수 형태의 가중치를 8비트의 정수형으로 줄이는 양자화 기법을 적용하면 연산량을 크게 감소할 수 있고 메모리 대역폭을 줄이며 전력 효율이 향상되는 효과를 얻을 수 있다.



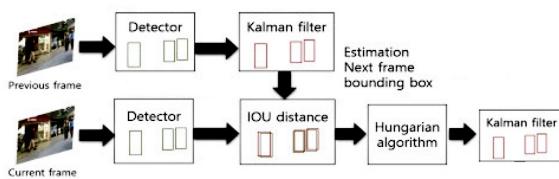
<그림 6> 학습 시 양자화 시뮬레이션 적용

Channel Pruning 과정에서 제거된 채널들로 인한 모델의 일시적 성능 하락은 QAT 적용과 동시에 Fine-tuning 학습이 진행되어 모델 성능을 개선할 수 있다. <그림 6>은 학습 시 양자화 시뮬레이션 과정을 보여준다.

3. 객체 추적 시스템

3.1 SORT Algorithm

SORT(Simple Online and Realtime Tracking) Algorithm은 Kalman Filter와 Hungarian Algorithm을 적용한 객체 추적 알고리즘이다. SORT Algorithm의 개요는 <그림 7>과 같다.



<그림 7> SORT Algorithm

Kalman Filter를 통해 전 프레임에서 검출된 객체의 경계 상자(bounding box)를 이용하여 다음 프레임의 객체 위치를 예측한다. 예측된 객체의 위치와 현재 프레임에서 얻은 객체 위치의 IoU(Intersection over Union)를 계산하여 같은 대상이라고 판단할 경우, 전 프레임을 통해 예측된 객체와 현재 프레임 객체를 매칭시켜주고 해당 객체에 고유 ID를 할당한다. 이 후 다시 Kalman Filter를 통해 결과 보정 단계를 거치게 된다. 모든 프레임마다 다음과 같은 과정을 반복하며 검출된 객체들을 추적한다.

4. 객체 계수 시스템

4.1 보행자 계수 알고리즘

보행자 계수를 연산하기 위해 추적 결과인 경계 상자(bounding box)의 위치 좌표(x, y, width, height), 객체의 ID, 객체 class 값을 사용하였다. 1920 x 1080 크기의 영상에서 x좌표 범위가 800이상 1120이하일 때 객체를 계수했고, 이미 계수된 ID는 다시 계수하는 경우가 없도록 구현하였다.

Algorithm 1: 보행자 계수 알고리즘

Input: [Box x,y,w,h,Id, Type]

Output: Increasing count value

Function Object(Input)

Begin

man_cnt ← 0

woman_cnt ← 0

Stack ← { }

ManStack ← { }

WomanStack ← { }

If iter->obj_id=0 **then** Stack← Id,x,y,w,h and ManStack ← Stack

Stack clear

Ptr ← find(ManStack.begin(), ManStack.end(), Id)

If ptr = ManStack.end() **and** (800<=x and x<=1120)

Then man_cnt+=1

Man ← Id

If iter->obj_id=1 **then** Stack← Id,x,y,w,h and WomanStack ← Stack

Stack clear

Ptr ← find(WomanStack.begin(), WomanStack.end(), Id)

If ptr = WomanStack.end() **and** (800<=x and x<=1120)

Then woman_cnt+=1

Woman ← Id

<그림 8> 보행자 계수 알고리즘 의사 코드

<그림 8>은 보행자 계수가 알고리즘에 관한 의사코드이다.

4.2 광고 응시자 계수 알고리즘

광고 응시자 얼굴의 ID(남자 정면 얼굴, 여자 정면 얼굴의 ID)와 광고 응시자의 ID(남자, 여자의 ID)를 같은 사람의 것이라고 판단하기 위해 광고 응시자의 ID를 기준으로 계수하였다. 이전에 계수되지 않은 객체에 한하여 객체의 경계상자(bounding box) 내부에 광고 응시자의 얼굴이 검출되는 경우에만 계수하는 방식으로 알고리즘을 작성했다.

Algorithm 2: 광고 응시자 계수 알고리즘

Input: [Box x,y,w,h,Id, Type]

Output: Increasing count value

Function Pedestrian(Input)

Begin

```
fman_cnt ← 0  
fwoman_cnt ← 0  
Stack ← {}  
FaceManStack ← {}  
FaceWomanStack ← {}  
If iter->obj_id=2  
then ptr_id ← man_id.back()[0] and x_min ← x-(w/2)  
and y_min ← y-(h/2) and x_max ← x+(w/2)  
and y_max ← y+(h/2)  
if x_min ≤ x_box and x_box ≤ x_max  
and y_min ≤ y_box and y_box ≤ y_max  
then ptr ← find(face_man_begin(), face_man_end(),ptr_id)  
if ptr = face_man_end then fman_cnt++  
  
If iter->obj_id=3  
then ptr_id ← woman_id.back()[0] and x_min ← x-(w/2)  
and y_min ← y-(h/2) and x_max ← x+(w/2)  
and y_max ← y+(h/2)  
if x_min ≤ x_box and x_box ≤ x_max  
and y_min ≤ y_box and y_box ≤ y_max  
then ptr ← find(face_woman_begin(), face_woman_end(),ptr_id)  
if ptr = face_woman_end then fwoman_cnt++
```

〈그림 9〉 광고 응시자 계수 알고리즘 의사 코드

〈그림 9〉는 광고 응시자를 계수하는 알고리즘에 대한 의사코드이다.

5. 임베디드 시스템

5.1 임베디드 시스템 구성

본 프로젝트에서 사용한 카메라는 Qualcomm 사의 QCS605 SoC가 내장된 임베디드 보드와 8020PL102 2M C-MOS 카메라 렌즈로 구성된 임베디드 시스템 일체형 카메라이다. QCS605는 IoT용 스마트 기기 개발을 위한 SoC 제품으로 기기에서 인공지능 모델을 활용하여 에지 컴퓨팅(Edge Computing)이 가능하다.

QCS505의 사양을 아래의 〈표 1〉과 같다.

QCS 605	
DSP	AI Engine/Hexagon 685 DSP with dual hexagon vector extensions
CPU	Kryo300:64-bit octa-cores, 2x Gold(2.5 GHz) + 6x Silver(1.7 GHz)
GPU	Adreno 615 @ up to 780 MHz

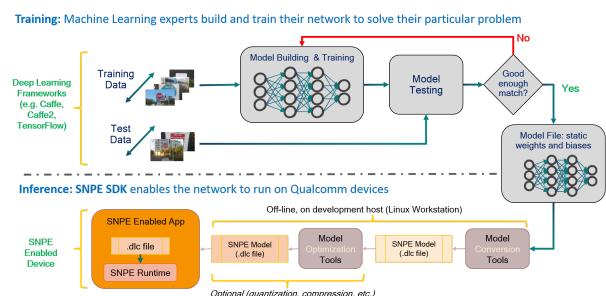
〈표 1〉 QCS605의 사양



〈그림 10〉 임베디드 시스템 일체형 카메라

〈그림 10〉은 본 프로젝트에서 사용한 임베디드 시스템 일체형 카메라를 보여준다. 임베디드 카메라의 운영체제는 Google 사의 Android 6.0이며, 학습 모델을 임베디드 시스템 일체형 카메라에서 작동시키기 위해 SNPE(Snapdragon Neural Processing Engine) SDK를 사용하였다. SNPE는 Qualcomm 사에서 개발한 모바일 기기용 AI 프레임워크이다.

5.2 모델 임베딩 과정



〈그림 11〉 SNPE SDK를 이용한 모델 임베딩 과정

검출 모델에 Quantization-aware Training(QAT)을 적용하여 얻은 학습 그래프는 시뮬레이션 노드 등 추론에 불필요한 요소가 많으므로 학습 그래프를 추론 그래프로 만들기 위한 freezing 과정을 거쳤다. 추론 그래프는 pb 파일로 저장되며, 이 파일을 카메라에 임베딩하기 위해 Snapdragon에 최적화된 dlc 파일로 변환하였다. <그림 11>은 SNPE SDK를 이용한 모델 임베딩 과정을 도식화한 것이다. 이후 ADB(Android Debug Bridge)를 사용하여 dlc 파일로 저장된 모델을 카메라에 임베딩하였다. 모델이 임베딩된 카메라는 LAN 통신을 이용하여 검출/추적 결과와 계수 연산 결과를 PC로 전송한다. PC에서 비디오 플레이어를 이용하면 실시간으로 임베디드 카메라에서 객체 검출 및 추적 영상을 확인할 수 있으며, PC의 터미널에서 검출/추적 결과와 계수 연산 결과를 확인할 수 있다.

6. 실험

6.1 환경

검출기 학습에 사용한 운영체제는 Ubuntu 18.04 LTS이며, CUDA 9.0, cuDNN 7.1.8 라이브러리와 NVIDIA TITAN RTX 24GB의 GPU를 사용하여 모델을 학습시켰다. 모델 학습 과정 중 Channel Pruning 과정은 Pytorch 환경에서 학습을 진행했고, Quantization-aware Training(QAT) 과정은 Tensorflow 환경에서 학습을 진행했다.

6.2 검출기 학습 데이터

검출 대상은 총 네 가지이며 남자 보행자, 여자 보행자, 남자 정면 얼굴, 여자 정면 얼굴로 이루어져 있다. 남자와 여자를 검출 대상으로 하여 광고판 앞 성별 별 보행자를 검출하였고, 성별 별 정면 얼굴을 검출하여 보행자의 광고 응시 여부를 판단하였다. 검출기 학습을 위해 <표 2>와 같이 데이터셋을 구성하였다. 공용 데이터셋과 수집 데이터셋을 합쳐서 사용하였으며, 공용 데이터셋은 Pascal VOC

2007과 Celeb A 데이터 중 본 프로젝트에 적합한 데이터 7335장을 선별하여 사용했다. 수집 데이터는 임베디드 시스템 일체형 카메라를 이용하여 1920 x 1080 크기의 영상 3879장을 취득하여 사용했다. 데이터 취득 환경은 총 세 가지로 (1) 세종대학교 정문 부근 버스정류장 광고판 앞, (2) 어린이대공원역 지하철 내 광고판 앞, (3) 어린이대공원역 1번 출구 부근 지상 광고판 앞으로 이루어져 있다. 모델 학습에 사용되는 Train 데이터와 모델 평가에 사용되는 Test 데이터는 8.5:1.5 비율로 무작위로 나누었다.

Data Type	Dataset	Split rate (8.5:1.5)	Total
Train	Pascal VOC 2007 + Celeb A	6234 개	9531 개
	수집 데이터	3297 개	
Test	Pascal VOC 2007 + Celeb A	1101 개	1683 개
	수집 데이터	582 개	

<표 2> 검출기 학습을 위한 데이터셋 구성



<그림 12> 공용 Dataset 예시

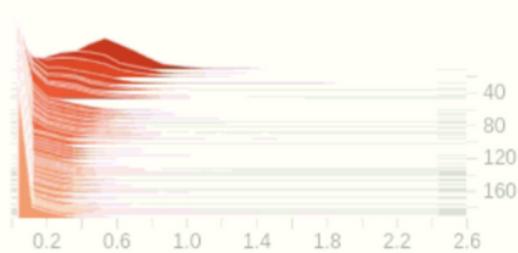


<그림 13> 수집 Dataset 예시

<그림 12>는 학습에 사용한 공용 데이터셋의 예시이며, <그림 13>은 임베디드 시스템 일체형 카메라를 이용하여 직접 수집한 Dataset의 예시이다.

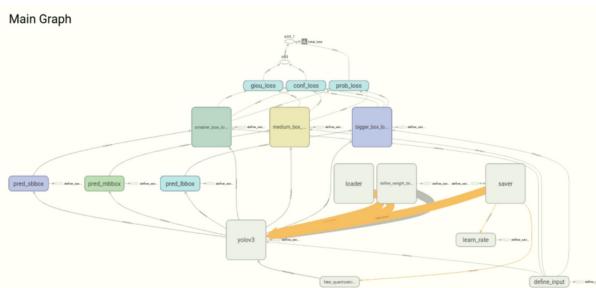
6.3 학습

Input 영상의 크기는 416×256 , Batch 크기는 8, 학습률(learning rate)은 0.0001로 학습을 진행했다. 먼저, Channel Pruning 적용을 위해 Channel-level 희소성 유도 정규화 학습을 진행하여 <그림 14>와 같이 많은 scaling factor가 0에 가까운 모델을 얻었다. scaling factor가 낮은 70 % Channel을 제거하는 Channel Pruning을 검출 모델에 적용하여 모델을 최적화하였다.



<그림 14> Sparsity Training 학습 과정 히스토그램

이후, Channel Pruning으로 최적화된 모델에 양자화 인식 학습을 적용했다. 양자화 인식 학습을 할 때, 동시에 fine-tuning이 진행되어 mAP 성능을 개선했다. 양자화 후 추론 그래프는 <그림 15>와 같다.



<그림 15> 양자화 인식 학습 적용후 추론 그래프

6.4 시스템 적용

6.4.1 추적 시스템 적용

70 % Channel Pruning과 양자화 인식 학습을 적용한 YOLO v3 모델을 임베디드 시스템 일체형 카메라에 임베딩하였고, 카메라를 통해 얻은 Input 영상에 대한 결과로 검출된 객체의 위치 좌표를 얻는다. 검출 결과에 SORT Algorithm을 적용하여 객체를 추적하며, 객체 추적 결과는 객체의 위치 좌표와 고유 ID, 객체 class이다. 객체 추적 알고리즘은 임베디드 시스템 일체형 카메라 내부 CPU에서 동작한다.

6.4.2 계수 시스템 적용

객체 추적 결과에 보행자 계수 알고리즘과 광고 응시자 계수 알고리즘을 적용하여 객체를 계수했다. 임베디드 시스템 일체형 카메라 내부 CPU에서 연산된 보행자 계수 결과는 LAN 통신을 통해 PC로 전달된다.

6.4.3 웹 서비스

PC로 전송된 보행자 계수 결과를 분석하여 다양한 그래프와 차트로 나타내는 마케팅 타깃 분석 웹 서비스를 제작했다. Highchart 툴을 통해 그래프 구현을 하였고, 계수 연산이 진행되는 시간과 LAN 통신을 통해 값이 전송되는 지연 시간을 고려하여 8000ms마다 서버에서 데이터를 읽어오도록 설정했다.



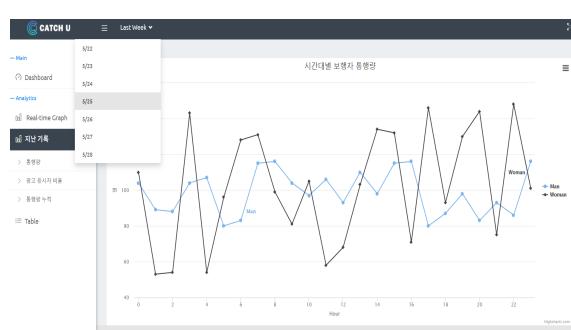
<그림 16-a> Catch U Web의 메인 페이지



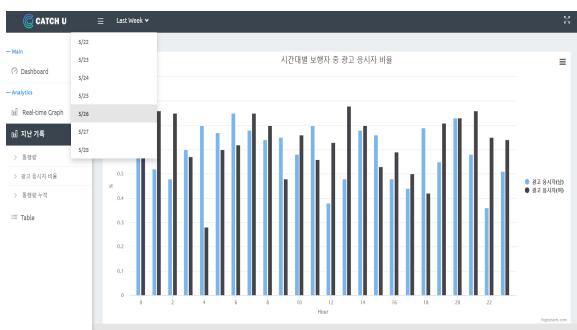
〈그림 16-b〉 실시간 보행자 통계 그래프

통행량 정보 Table of 2021-09-18				
Hour	Man	Woman	Face_man	Face_woman
0	107	141	62	119
1	102	105	52	97
2	89	70	67	47
3	100	99	80	82
4	113	95	71	70
5	96	114	39	104
6	100	136	75	119
7	91	102	43	97
8	106	130	85	104
9	93	110	46	95
10	101	117	79	96

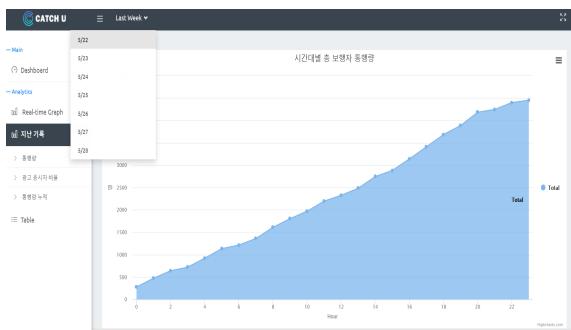
〈그림 16-f〉 보행자 계수 결과(Raw Data)



〈그림 16-c〉 시간대별 통행량 그래프



〈그림 16-d〉 시간대별 광고 응시자 비율 그래프



〈그림 16-e〉 시간대별 통행량 누적 그래프

〈그림 16-a〉는 웹의 메인 페이지로 마케팅 타겟 분석에 대한 전체적인 현황을 보여준다. 윗단에서는 오늘과 어제의 통행량과 총 광고응시자의 수를 나타냈고, 아랫단은 한 주 간의 일별 남여 광고 응시 비율과 광고판 앞 남여 통행 비율을 나타냈다. 〈그림 16-b〉는 실시간 광고판 앞 통행량과 보행자 중 남/여 광고 응시 비율을 확인할 수 있는 그래프이다. 〈그림 16-c〉는 일별 광고판 앞 통행량을 시간대 별로 확인할 수 있는 그래프이다. 지난 일주일 간의 기록도 확인 가능하다. 〈그림 16-d〉에서는 광고 응시자(남)은 남자 광고 응시자와 남자 보행자의 비를 백분율로 나타내었고, 광고 응시자(여)는 여자 광고 응시자와 여자 보행자의 비를 백분율로 나타냈다. 어느 성별이 해당 광고에 더 많은 관심을 갖는지 알아보기 위한 그래프이다. 지난 일주일 간의 기록 또한 확인할 수 있다. 〈그림 16-e〉는 하루 간의 광고판 앞 보행자 통행량을 누적한 그래프를 보여주는 페이지이다. 이 페이지 또한 지난 일주일 간의 기록을 확인할 수 있도록 구현했다. 〈그림 16-f〉는 임베디드 시스템 일체형 카메라로부터 실시간으로 받아오는 보행자 계수 결과를 확인할 수 있는 페이지이다. 해당 정보를 이용하여 추후 다양한 보행자 분석 자료로 사용하기 위함이다.

6.5 평가

6.5.1 검출기 정량 평가

본 논문에서는 검출기의 성능평가지표로 mAP(mean Average Precision)을 사용하였다. mAP는 각 물체

종류마다 검출 정확도의 평균인 AP(Average Precision)를 구한 후, 모든 클래스에 대한 AP의 평균을 구하여 얻은 수치이다.

$$\text{검출정확도(Precision)} = \frac{TP}{TP+FP}$$

〈식 1〉 Precision

Ground Truth와 검출된 결과의 IoU(Intersection over Union)가 0.5 이상인 경우를 True Positive로 설정하였다. 〈표 3〉은 딥러닝 모델 별 객체 검출 성능을 비교 분석한 결과이다. YOLO v3 모델에 양자화를 적용하였을 때, 양자화를 적용하지 않은 모델보다 약 2.39 % mAP가 하락했다. 본 프로젝트에서 사용한 모델인 70 % Channel Pruning된 양자화 모델의 전체 테스트 데이터셋에 대한 mAP는 77.81 %로, Channel Pruning을 하지 않고 양자화만 적용한 YOLO v3 모델보다 약 4.43 % 낮은 성능을 보였고, 양자화를 적용하지 않은 YOLO v3 모델보다 약 6.82 % 성능이 하락했다.

딥러닝 모델	공용 데이터셋 mAP	수집 데이터셋 mAP	전체 데이터셋 mAP
YOLO v3	74.70 %	87.99 %	80.20 %
YOLO v3 (QAT)	73.98 %	82.91 %	77.81 %
70 % pruned YOLO v3 (QAT)	69.09 %	79.37 %	73.38 %

〈표 3〉 딥러닝 모델 별 mAP 성능 평가

Channel Pruning과 Quantization-aware Training을 모델에 적용하면 mAP가 소폭 하락하는 단점이 있지만, 대신 보다 가벼운 네트워크로 추론시 Inference Time이 크게 감소하는 효과를 얻을 수 있다. 〈표 4〉에서 Prune Rate가 각각 다르게 적용된

모델들의 Parameter 개수와 GPU 환경에서의 Inference Time을 확인할 수 있다. Inference Time은 6.1에서 언급한 PC 환경에서 측정했다.

딥러닝 모델	Parameter 수 (개)	Inference Time (GPU)
YOLO v3	61539889	99 ms
50 % pruned YOLO v3	25759884	70 ms
60 % pruned YOLO v3	20681735	65 ms
70 % pruned YOLO v3	17105045	63 ms
80 % pruned YOLO v3	14836092	62 ms

〈표 4〉 딥러닝 모델 별 성능 비교

mAP와 Inference Time 간의 trade-off를 고려하여 본 프로젝트에서는 70 % Channel Pruning된 양자화 모델을 사용했다.



〈그림 17〉 Test Dataset에 대한 검출 결과

〈그림 17〉은 시연 환경 Test Dataset에 대한 검출 결과 예를 보여준다.

6.5.2 계수 시스템 평가

계수 시스템 평가는 수집한 데이터셋 환경과

마찬가지로 (1) 세종대학교 정문 부근 버스정류장 광고판 앞, (2) 어린이대공원역 지하철 내 광고판 앞, (3) 어린이대공원역 1 번 출구 부근 지상 광고판 앞의 총 3 가지 환경에서 진행했다. 계수 시스템 평가는 보행자 총 100명을 대상으로 진행했고, 세 가지 환경에 대한 계수 시스템 평가는 <표 5>에서 확인할 수 있다. 계수 오차율 계산은 아래의 식을 사용했다.

$$\text{계수오차율}(\%) = \frac{|\text{참값} - \text{실험결과 값}|}{\text{참값}} \times 100$$

<식 2> 계수 오차율 계산 식

대부분의 오차 발생 이유는 객체 여러 개가 동시에 겹친 상황 이후에 각 객체에 대한 추적이 끊기고, 객체에 새로운 ID가 할당되어 중복으로 계수된 경우이다.

	보행자 (남)	보행자 (여)	광고 응시자 (남)	광고 응시자 (여)
어린이대공원 지하철 내	5 %	6 %	11 %	13 %
어린이대공원 1번 출구 (지상)	7 %	9 %	10 %	8 %
세종대학교 정문 앞 버스정류장	9 %	7 %	12 %	10 %

<표 5> 세 가지 환경에 대한 계수 시스템 평가 표

6.5.3 향후 연구 방향

본 프로젝트에 적용한 SORT Algorithm은 이전 프레임의 객체의 위치를 이용하여 다음 프레임의 객체 위치를 예측하고, 예측된 객체와 현재 프레임의 객체의 IoU(Intersection over Union)를 계산하여 객체를 연계 및 추적하는 방식이다. 이 알고리즘은 객체의 위치 값만을 이용하여 객체를 추적하는 방식으로, 객체 여러 개가 동시에 겹치는 상황에서는 객체의 추적이 정확히 이루어지지 않을 수 있다는 단점이 있다. 따라서 계수 시스템 평가는 객체 여러

개가 겹치는 상황을 최대한 배제하고 진행하였다. 이러한 문제점은 향후 연구에서 다중 객체 추적을 위한 CNN 기반 추적 알고리즘을 도입하여 해결할 수 있을 것으로 사료된다.

7. 결론

본 프로젝트는 보다 효율적인 마케팅을 위하여 에지 컴퓨팅 환경을 위한 마케팅 타깃 검출 및 분석 시스템을 제안했다. 객체 검출 시스템을 임베디드 시스템 일체형 카메라에서 실시간으로 작동시키기 위해 Channel Pruning과 Quantization-aware Training(QAT)을 적용하여 모델을 최적화하였다. 검출된 결과에 SORT 알고리즘을 적용하여 객체를 추적하였으며, 추적 결과에 보행자 계수 알고리즘을 적용하여 임베디드 시스템 일체형 카메라 내부 CPU에서 보행자를 계수하였다. LAN 통신을 통해 PC로 최종 보행자 계수 결과를 전송했으며, 이를 이용하여 마케팅 타깃 분석을 위한 웹 서비스를 제작하여 광고판 앞 통행량, 보행자 중 광고 응시 비율 등의 내용을 다양한 차트와 그래프로 제시했다. Pascal VOC 2007, Celeb A 공용 데이터셋과 직접 수집한 데이터셋을 이용하여 CNN 기반 검출기 학습을 진행함으로써 남/여 보행자와 광고 응시자 검출 및 추적이 가능함을 보였다. 본 프로젝트에 적용한 70 % Channel Pruning된 YOLO v3 양자화 모델은 시연 환경에서 mAP 79.37 % 성능과 영상 당 70 ms 처리 속도를 보이며 인공지능 카메라가 실제 환경에서 사용 가능함을 확인하였다. 향후 연구에서는 여러 객체가 동시에 겹치는 상황에서의 객체 추적이 어려웠던 상황을 개선할 계획이다.

참고문헌

- [1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.
- [2] Luis Felipe de Araujo Zeni, Claudio Rosito Jung, "Real-Time Gender Detection in the Wild Using Deep

Neural Networks" 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018.

[3] Zhuang Liu, "Learning Efficient Convolutional Networks through Network Slimming", IEEE International Conference on Computer vision, 2017.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking", IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464-3468.

[5] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Kalenichenko, D., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference.", Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[6] Krishnamoorthi, R., "Quantizing Deep Convolutional Networks for Efficient Inference: A whitepaper.", arXiv preprint arXiv:1806.08342, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385, 2015.