특집논문 (Special Paper)
방송공학회논문지 제27권 제3호, 2022년 5월 (JBE Vol.27, No.3, May 2022)
https://doi.org/10.5909/JBE.2022.27.3.308
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

물체탐색과 전경영상을 이용한 인공지능 멀티태스크 성능 비교

정 민 혁[a], 김 상 균[b)‡], 이 진 영[c], 추 현 곤[c], 이 희 경[c], 정 원 식[c]

Comparison of Artificial Intelligence Multitask Performance using Object Detection and Foreground Image

Min Hyuk Jeong[a], Sang-Kyun Kim[b)‡], Jin Young Lee[c], Hyon-Gon Choo[c], HeeKyung Lee[c], and Won-Sik Cheong[c]

요 약

딥러닝 기반 머신 비전 기술을 이용한 영상분석 과정에서 전송되고 저장되는 방대한 양의 동영상 데이터의 용량을 효율적으로 줄이기 위한 연구들이 진행 중이다. MPEG(Moving Picture Expert Group)은 VCM(Video Coding for Machine)이라는 표준화 프로젝트를 신설해 인간을 위한 동영상 부호화가 아닌 기계를 위한 동영상 부호화에 대한 연구를 진행 중이다. 그 중 한 번의 영상 입력으로 여러 가지 태스크를 수행하는 멀티태스크에 대한 연구를 진행하고 있다. 본 논문에서는 효율적인 멀티태스크를 위한 파이프라인을 제안한다. 제안하는 파이프라인은 물체탐지를 선행해야 하는 각 태스크들의 물체탐지를 모두 수행하지 않고 한번만 선행하여 그 결과를 각 태스크의 입력으로 사용한다. 제안하는 멀티태스크 파이프라인의 효율성을 알아보기 위해 입력영상의 압축효율, 수행시간, 그리고 결과 정확도에 대한 비교 실험을 수행한다. 실험 결과 입력 영상의 용량이 97.5% 이상 감소한데 반해 결과 정확도는 소폭 감소하여 멀티태스크에 대한 효율적인 수행 가능성을 확인할 수 있었다.

Abstract

Researches are underway to efficiently reduce the size of video data transmitted and stored in the image analysis process using deep learning-based machine vision technology. MPEG (Moving Picture Expert Group) has newly established a standardization project called VCM (Video Coding for Machine) and is conducting research on video encoding for machines rather than video encoding for humans. We are researching a multitask that performs various tasks with one image input. The proposed pipeline does not perform all object detection of each task that should precede object detection, but precedes it only once and uses the result as an input for each task. In this paper, we propose a pipeline for efficient multitasking and perform comparative experiments on compression efficiency, execution time, and result accuracy of the input image to check the efficiency. As a result of the experiment, the capacity of the input image decreased by more than 97.5%, while the accuracy of the result decreased slightly, confirming the possibility of efficient multitasking.

Keywords : Video coding for machine, Object detection, Object tracking, Pose estimation, MPEG

Ⅰ. 서 론

딥러닝 기반의 머신 비전 기술이 활발히 연구 개발되면서, 추론의 정확도와 처리속도가 높아지며 기존 컴퓨터 비전 기반의 기술을 대체하여 산업 전반에 자리잡고 있다. 또한 자율주행자동차, 지능화된 방범카메라 등 다양한 영상분석 기반 서비스들이 보편화 되면서 기계를 이용한 영상분석에 대한 수요 또한 증가하고있다[1]. 이러한 머신 비전 기반 기술을 이용한 영상분석을 수행하는 과정에서 기존의 동영상 코딩 방식으로 인코딩하여 영상을 전송하거나 저장하는 것은 용량면에서나, 영상분석 결과의 정확도면에서 효율적이지 못할 가능성이 높다. 이에 멀티미디어 관련 표준 기술을 개발하는 MPEG에서는 video coding for machine(이하: VCM)이라는 프로젝트를 신설하여 기계를 위한 동영상 코딩에 대한 표준화를 진행 중이다.

MPEG-VCM은 머신 비전의 성능 향상 및 압축 효율을 위한 동영상 부호화를 목표로 하고있다. 동영상 뿐만 아니라 인공지능 네트워크 중간단계에서 추출되는 특징(feature)을 부호화하는 연구나[2] 추론결과를 부호화하는 연구들[3] 또한 이루어지고 있다. VCM은 파이프라인을 정의하고 물체탐지(i.e., object detection), 물체추적(i.e., object tracking), 물체영역분할(i.e., object segmentation) 등 머신 태스크를 위한 평가 데이터셋과 인공지능 네트워크를 설정하여 여러가지 실험과 평가를 진행중이다. 또한 한 번에 한 가지 태스크만을 수행하는 싱글태스크 뿐만 아니라, 두가지 이상의 태스크를 수행하는 멀티태스크에 수행을 위한 탐사실험(Exploration Experiment: EE)을 수립하여 진행 중이다. 본 논문은 멀티태스크 파이프라인을 제안하고 수행시간, 입력영상의 압축 효율, 추론 결과의 정확도를 측정하기 위한 실험을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 멀티태스크를 위한 파이프라인을 제안한다. 3장에서는 실험 간 교환되는 데이터에 대해 설명한다. 4장에서는 실험 환경을 소개하고 실험의 결과에 대해 설명한다. 마지막으로 5장에서는 본 논문에 대한 결론을 맺는다.

Ⅱ. 아키텍처

본 논문에서는 자세추론(i.e., pose estimation)이나 물체추적등 물체탐지가 선행되어야 하는 태스크들의 물체탐지를 각각 수행하지 않고 한 번만 수행하여, 추출된 물체탐지 결과와 탐지된 물체만을 표현하는 전경영상을 각 태스크들의 입력으로 사용하는 멀티태스크 파이프라인을 제안한다. 그림 1은 복수의 인공지능 태스크를 동시 처리하기 위해 본 논문에서 제안하는 기계(인공지능)를 위한 비디오 압축(Video Coding for Machines: 이하 VCM) 멀티태스크 파이프라인을 도식화한 것이다.

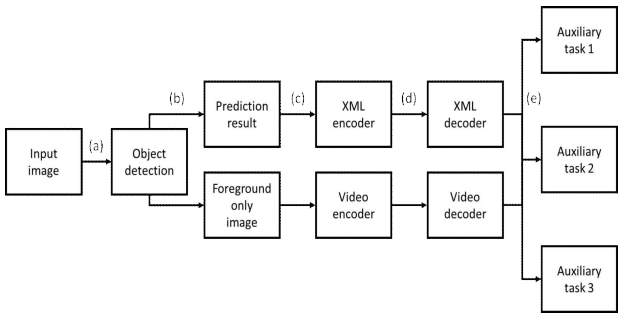

그림 1. 멀티태스크를 위한 파이프라인
Fig 1. Multi-task pipeline

영상을 물체탐지 네트워크에 입력하여(그림 1-(a)), 물체의 종류 및 위치를 표현하는 XML 파일과 탐지된 물체만을 표현하는 영상(i.e., 전경영상)을 추출한다(그림 1-(b)). 추출된 XML 파일과 전경영상을 각각 인코딩 후 디코딩하고(그림 1-(c)와 (d)), 이를 후처리 인공지능 네트워크(예: 자세추론과 물체추적)에 입력하여 각 태스크를 동시에 수행한다(그림 1-(e)).

a) 명지대학교 컴퓨터공학과(Dept. of Computer Engineering, Myongji University)
b) 명지대학교 융합소프트웨어학부(Dept. of Software Convergent, Myongji University)
c) 한국전자통신연구원(ETRI)
‡ Corresponding Author : 김상균(Sang-Kyun Kim)
E-mail: goldmunt@gmail.com
Tel: +82-10-6406-8163
ORCID:https://orcid.org/0000-0002-2359-8709
※본 논문은 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2020-0-00011, 기계를 위 한 영상 부호화 기술).
This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(No. 2020-0-00011, Video Coding for Machine).
· Manuscript April 12, 2022; Revised May 16, 2022; Accepted May 16, 2022.

그림 2. 물체탐지 결과 XML(a)와 탐지된 물체만 표현하는 전경영상(b)
Fig 2. Object detection results: (a) XML and (b) foreground only image

Ⅲ. 실험 간 교환되는 데이터

1. 물체탐지 결과

자세추론과 물체추적 인공지능 네트워크 내에 전처리로서 공통으로 이루어지는 물체탐지를 선행하여 수행한다. 물체탐지 부분과 자세추론 및 물체추적 부분이 물리적으로 나눠져있다고 가정한다. 따라서 물체탐지 결과를 자세추론과 물체추적 네트워크의 입력으로 전달하기 위하여 파일 형태로 저장하는 과정이 필요하다. 이 과정에서 저장되는 물체탐지 결과 XML과 전경영상의 예는 그림 2와 같다.

그림 2-(a)는 물체탐지 과정에서 추출된 물체탐지 결과를 XML로 표현한 예시이다. 탐지된 물체의 분류, 신뢰도 점수(confidence score), 그리고 물체의 위치를 표현하는 박스의 좌표를 나타낸다. 신뢰도 점수는 해당 물체가 어떠한 클래스일 확률에 IoU(Intersection over Union)을 곱하여 계산되며, 해당 물체의 클래스와 위치를 모두 고려한다[4]. 그림 2-(b)는 영상에서 탐지된 물체만을 표현하는 전경영상이다. 압축효율을 높이기 위하여 탐지된 물체를 제외한 모든 부분을 검정색으로 표현하였다. 사람을 제외하고 배경을 검정색으로 만들고 VVC 인코딩 했을 때 VVC 인코딩 한 원본영상에 비해 약 10퍼센트에서 20퍼센트 까지 용량이 감소한다[5]. 배경을 검정색으로 표현할 때 어떤 클래스만을 표현하는가에 따라 효율의 차이를 보일 수 있다. 또한 표현하고자 하는 클래스에 해당하는 물체의 수에 따라 효율이 달라질 수 있다.

추출된 물체탐지 결과 XML은 zip 파일로 압축되고 전경영상은 VVC(Versatile Video Coding) 인코더로 압축되어 후처리 인공지능 네트워크들의 입력부로 전달된다. 후처리 인공지능 네트워크의 입력부에서는 전달받은 zip파일과 압축된 영상을 디코딩하여 각 네트워크의 입력으로 사용한다.

2. 물체탐지 네트워크

물체탐지 네트워크는 물체탐지 네트워크인 faster R-CNN X101-FPN을 사용하였다. 물체탐지는 탐지된 물체가 해당 클래스일 확률을 나타내는 점수인 신뢰도 점수가 0.6 이상인 물체만 추출하였다.

그림 3은 물체탐지 시 설정한 신뢰도 점수에 따른 물체탐지 추출결과를 보여준다. 그림 3-(a)는 신뢰도 점수를 0.4로 설정했을 때의 결과이다. 신뢰도 점수가 낮게 설정되어 사람이 아닌 물체까지 사람으로 인식하였다. 그림 3-(d)는 신뢰도 점수를 0.7로 설정했을 때의 결과이다. 신뢰도 점수가 높게 설정되어 사람임에도 사람으로 검출되지 않은 경우가 발생하였다.

본 논문에서 사용한 HiEve(Human in Events)[6] 데이터셋은 작은 크기의 사람과 큰 크기의 사람이 골고루 등장한다. 이러한 데이터셋의 특성을 고려했을 때 가장 좋은 결과를 추출할 수 있을 것으로 생각되는 신뢰도 점수인 0.6으로 설정하고 실험을 수행하였다.

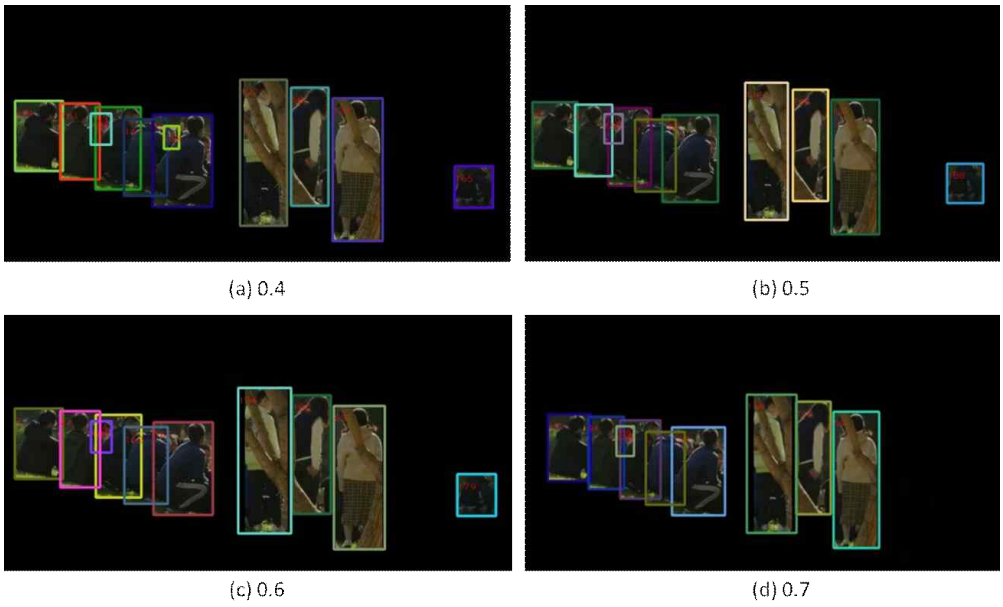

그림 3. 신뢰도 점수 별 물체탐지 추출결과 영상
Fig. 3. Image of object detection extraction result by confidence score

Ⅳ. 실험 환경 및 결과

1. 실험 환경

실험에 사용된 하드웨어 및 소프트웨어 환경은 표 1과 같다.

표 1. 실험의 하드웨어 및 소프트웨어 환경
Table 1. Hardware and software environment of the experiment

Hardware	
CPU	XEON W-2235
GPU	RTX 2080 SUPER (4ea)
Software	
CUDA	10.1
Ubuntu	20.04.01
Python	3.8.5
Tensorflow	2.3.1
PyTorch	1.5.0
Detectron2	0.1.3
VTM	12.0
ffmpeg	4.2.2

물체탐지를 위해 Detectron 2[7]를 사용하였고, 물체탐지의 결과로 추출된 전경영상을 압축하기 위해 ffmpeg 4.2.2[8]와 VTM 12.0[9]이 사용됐다. 동영상의 각 프레임을 png로 저장하고 ffmpeg을 사용하여 yuv420으로 인코딩한 후 VTM 인코더로 yuv파일을 vvc 파일로 인코딩한다.

표 2. 실험에 사용된 네트워크
Table 2. Networks used in the experiment

Task	Network
Object detection	Faster R-CNN X101-FPN
Pose estimation	pose_hrnet_w32_256x256 (Faster R-CNN R50)
Object tracking	JDE 1088x608 (YOLO v3 1088x608)

표 2는 실험에 사용된 각 태스크의 인공지능 네트워크이다. 후처리 인공지능 네트워크의 입력으로 사용되는 물체탐지 결과 추출을 위해서 Faster R-CNN X101-FPN이 사용되었다. 자세추론 네트워크로 HRNet[10]을 사용했으며 자세추론 네트워크의 물체탐지부는 기본으로 제공되는 Faster R-CNN R50이 사용됐다. 물체추적 네트워크로는 JDE 1088x608[11]이 사용되었으며 물체추적 네트워크의 물체탐지부는 기본으로 제공되는 YOLO v3 1088x608이 사용됐다.

표 3. 실험에 사용된 데이터
Table 3. Data used in the experiment

Video	Frames	Resolution
2.mp4	4819	1280x720
4.mp4	952	1660x1080
7.mp4	911	1920x1080
16.mp4	700	1920x1080

표 3은 실험에 사용된 데이터이다. HiEve 데이터셋의 중 2,4,7, 그리고 16번 동영상이 사용되었다. 사람의 크기가 큰 실내 영상, 사람 크기가 작은 실내 영상, 사람 크기가 큰 실외 영상, 그리고 사람 크기가 작은 실외 영상을 한 개씩 선정하여 총 네 개의 동영상으로 실험을 수행했다.

2. 실험 방법

실험은 본 논문에서 제안하는 물체탐지를 한 번만 선실행하고 결과를 후처리 인공지능 네트워크들의 입력으로 사용하는 멀티태스크 프로세스와(그림 4-(a)), 기존의 방식인 각자의 물체탐지 과정을 포함한 자세추론 및 물체추적을 수행하는 싱글태스크 프로세스(그림 4-(b)) 두 가지를 비교하였다. 비교항목으로는 수행시간, 입력영상의 용량 그리고 각 태스크의 추론결과를 비교하였다. 입력영상의 용량은 픽셀당 비트를 나타내는 BPP(Bits Per Pixel)를 사용하여 비교했고, 자세추론 결과는 mAP(mean Average Precision), 물체추적 결과는 MOTA(Multi-Object Tracking Accuracy)로 비교하였다.

$$BPP = \frac{total\ bits}{total\ pixels(width * height * numOfFrames)} \quad (1)$$

(수식1)은 BPP를 구하는 계산식이다. BPP는 압축된 파일의 크기를 압축 전 전체 픽셀수로 나눈 값이다. 전체 픽셀수는 해상도의 가로, 세로, 그리고 프레임 수를 곱한 값이다.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (2)$$

자세추론의 mAP는 각 신체부위의 AP(Average Precision)의 평균값으로 계산한다. (수식2)의 는 신체부위의 AP를 뜻하며 은 신체부위의 개수이다.

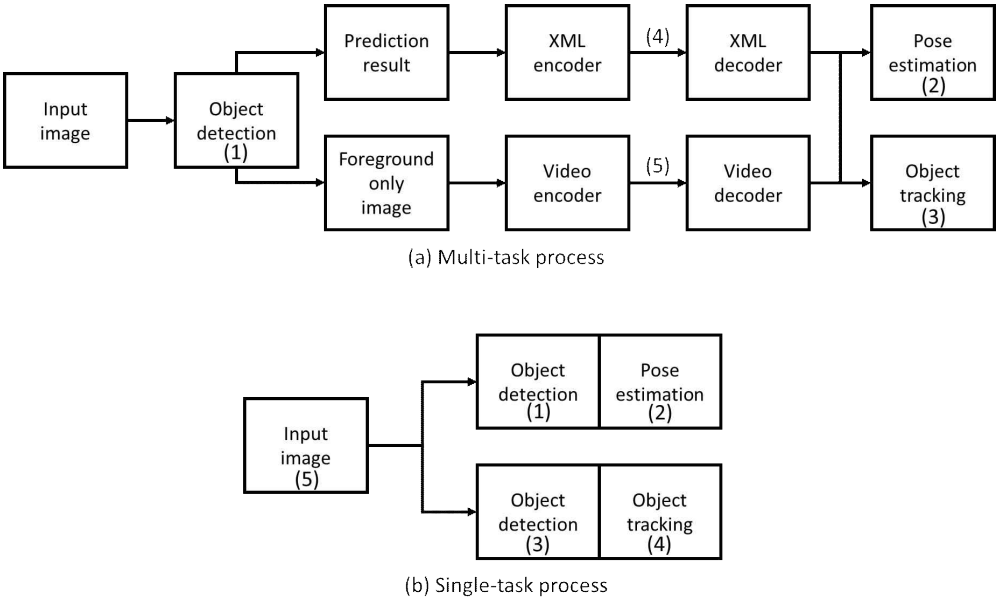

그림 4. (a) 멀티태스크 프로세스, (b) 싱글태스크 프로세스
Fig 4. (a) Multi-task process, (b) Single-task process

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (3)$$

(수식3)의 m_t는 false negative, fp_t는 false positive, 그리고 mme_t는 물체추적 과정에서 아이디가 변경된 횟수이며 MOTA 값은 이 세가지를 더하여 Ground truth 개수로 나눈 값을 1에서 뺀 값이다.

수행시간 측정 시 본 논문에서 제안하는 멀티태스크 파이프라인은 선행되는 물체탐지의 소요시간(그림 4-(a)-(1))과 각 후처리 인공지능 네트워크들의 소요시간(그림 4-(a)-(2), 그림 4-(a)-(3))을 더해 총 소요시간을 계산했고, 싱글태스크 파이프라인은 각자의 물체탐지 소요시간(그림 4. (b)-(1), 그림 4-(b)-(3))과 자세추론과 물체추적 네트워크에 각각 소요되는 시간(그림 4-(b)-(2), 그림 4-(b)-(4))을 더하여 총 소요시간을 계산하여 비교하였다.

입력영상의 용량 비교는 멀티태스크 프로세스의 물체추적 결과파일을 zip으로 압축한 것(그림 4-(a)-(4))과 전경영상을 인코딩한 것(그림 4-(a)-(5))의 용량을 더한 값과, 싱글태스크 프로세스의 압축되지 않은 원본영상의 용량(그림 4-(b)-(5))을 비교하였다. 전경영상의 압축은 QP(Quantization Parameter)를 6개(22, 27, 32, 37, 42, 47)로 설정하고 각 동영상을 6개의 QP로 VTM 인코딩하였다.

3. 실험 결과

후처리 인공지능 네트워크로 전달되는 추론결과의 압축 효율을 확인하기위해 멀티태스크 프로세스에서 물체탐지 시 생성되는 전경영상을 각 QP별로 압축한 용량과 추론결과 XML파일을 zip으로 압축한 용량을 더하여 어떠한 처리도 하지 않은 원본영상과의 용량을 BPP로 비교하였다.

표 4. 각 동영상의 QP별 BPP
Table 4. BPP by QP for each video

(unit : BPP)

	2.mp4	4.mp4	7.mp4	16.mp4
Original input images	10.072	7.561	8.490	11.568
qp22+xml zip	0.384	0.097	0.074	0.374
qp27+xml zip	0.250	0.064	0.048	0.227
qp32+xml zip	0.159	0.044	0.033	0.142
qp37+xml zip	0.099	0.03	0.023	0.090
qp42+xml zip	0.063	0.021	0.016	0.056
qp47+xml zip	0.041	0.015	0.012	0.034

QP22로 압축했을 경우 추론결과 zip파일을 포함한 동영상 4개의 평균용량이 원본대비 97.5% 감소했고, QP47로 압축했을 경우 추론결과 zip파일을 포함한 용량이 원본대비 99.7% 감소했다. 이는 관심 물체가 포함된 전경영상과

표 5. 자세추론의 멀티태스크와 싱글태스크의 QP별 mAP 비교
Table 5. Comparison of mAP by QP of the multitask and single task of pose estimation

(unit : mAP)

QP22	Single	Multi
2.mp4	77.46	65.58
4.mp4	20.64	19.55
7.mp4	14.54	56.8
16.mp4	29.67	35.3
Average	35.57	44.30

(a) QP22

QP27	Single	Multi
2.mp4	77.31	65.40
4.mp4	20.79	19.52
7.mp4	14.10	56.14
16.mp4	29.24	35.33
Average	35.36	44.09

(b)QP27

QP32	Single	Multi
2.mp4	76.79	65.04
4.mp4	19.93	19.45
7.mp4	13.18	55.66
16.mp4	28.33	35.07
Average	34.55	43.80

(c) QP32

QP37	Single	Multi
2.mp4	76.14	64.50
4.mp4	19.16	19.15
7.mp4	11.30	54.71
16.mp4	27.81	34.5
Average	33.60	43.21

(d) QP37

QP42	Single	Multi
2.mp4	74.81	63.55
4.mp4	18.50	18.9
7.mp4	10.69	53.20
16.mp4	26.16	33.69
Average	32.54	42.33

(e) QP42

QP47	Single	Multi
2.mp4	72.37	62.25
4.mp4	16.22	18.21
7.mp4	7.53	51.25
16.mp4	22.31	31.81
Average	29.60	40.88

(f) QP47

추론결과만을 압축함으로써 원본 영상 대비 큰 압축 효율을 보이는 것이다.

표 5는 멀티태스크 프로세스로 자세추론을 수행했을 때 QP별 mAP의 감소율을 확인하기 위해 각 QP별 mAP를 계산한 표이다. 싱글태스크 프로세스와의 비교를 위해 싱글태스크의 입력 또한 원본영상을 인코딩하고 태스크를 수행하여 멀티태스크 프로세스와 mAP결과를 비교하였다. mAP 비교는 4개 동영상의 mAP값의 평균으로 계산하였다.

싱글태스크와 멀티태스크를 위한 물체탐지 인공지능 네트워크가 서로 달라 동영상 마다 서로 다른 mAP 결과가 추출됐다. 싱글태스크 프로세스로 자세추론을 수행했을 때 QP22에 대비 QP47의 mAP는 약 16.8% 하락하였고, 멀티태스크 프로세스로 자세추론을 수행했을 때 QP47에 7.73% 하락하였다.

물체탐지 네트워크가 서로 달라 mAP의 절대적인 수치는 다르지만 QP에 따른 품질열화는 그림 5와 같이 비슷한 양상을 보인다.

표 6은 멀티태스크 및 싱글태스크로 물체추적을 수행했을 때 QP별 MOTA를 계산한 표이다. 멀티태스크 프로세스로 물체추적을 수행할 때 본 논문에서 사용한 물체추적 인공지능 네트워크의 특성 상 물체탐지 추론결과의 물체 좌표의 IoU만을 가지고 물체추적을 수행한다. 물체탐지의 결과는 압축하기 전에 추출하여 QP에 상관없이 동일하기 때

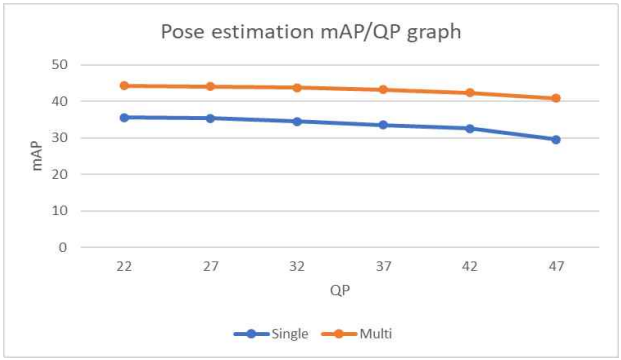

그림 5. 싱글태스크와 멀티태스크의 QP별 자세추론 mAP 결과 그래프
Fig. 5. Pose estimation mAP result graph by QP of single task and multi-task

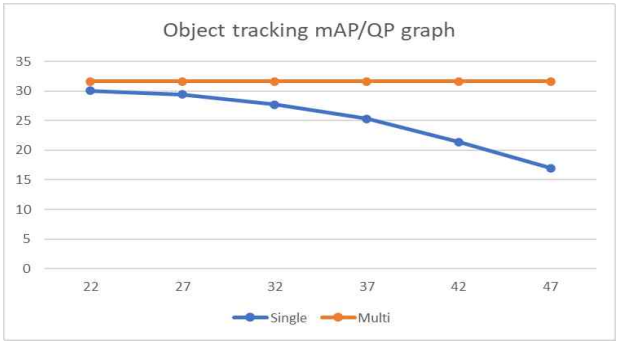

그림 6. 싱글태스크와 멀티태스크의 QP별 물체탐지 mAP 결과 그래프
Fig. 6. Object tracking mAP result graph by QP of single task and multi-task

표 6. 물체추적의 멀티태스크와 싱글태스크의 QP별 MOTA 비교
Table 6. Comparison of MOTA by QP of the multitask and single task of object tracking

(unit : MOTA)

QP22	Single	Multi
2.mp4	51.0	45.34
4.mp4	17.3	15.10
7.mp4	33.2	54.19
16.mp4	18.7	11.99
Average	30.05	31.65

(a) QP22

QP27	Single	Multi
2.mp4	50.6	45.34
4.mp4	16.7	15.10
7.mp4	32.0	54.19
16.mp4	18.5	11.99
Average	29.45	31.65

(b) QP27

QP32	Single	Multi
2.mp4	50.1	45.34
4.mp4	15.3	15.10
7.mp4	27.5	54.19
16.mp4	18.0	11.99
Average	27.72	31.65

(c) QP32

QP37	Single	Multi
2.mp4	49.6	45.34
4.mp4	12.3	15.10
7.mp4	22.2	54.19
16.mp4	17.3	11.99
Average	25.35	31.65

(d) QP37

QP42	Single	Multi
2.mp4	48.0	45.34
4.mp4	6.8	15.10
7.mp4	15.0	54.19
16.mp4	15.9	11.99
Average	21.42	31.65

(e) QP42

QP47	Single	Multi
2.mp4	45.1	45.34
4.mp4	3.6	15.10
7.mp4	6.4	54.19
16.mp4	12.9	11.99
Average	17.00	31.65

(f) QP47

문에 그림 6과 같이 모든 QP에서 같은 결과를 보인다. 반면 싱글태스크로 물체추적을 할 때 물체탐지 수행 시 QP의 영향을 받기 때문에 싱글 태스크로 물체추적을 한 결과 QP가 높아질 수록 MOTA가 감소했다. 싱글태스크로 물체추적을 수행한 결과 QP22 대비 QP47의 MOTA 결과는 약 43.4% 감소하였다. 물체추적 또한 자세추론과 마찬가지로 싱글태스크와 멀티태스크의 물체탐지 네트워크가 다르기 때문에 동영상 별로 MOTA에 차이를 보였다.

표 7은 멀티태스크와 싱글태스크 프로세스의 총 수행시간 비교 표이다. 멀티태스크 프로세스는 물체탐지를 한번만 수행하여 그 추론결과를 후처리 인공지능 네트워크의 입력으로 사용한다. 따라서 멀티태스크 프로세스의 총 소요시간은 물체탐지 시간, 물체탐지를 하지않는 자세추론 시간, 그리고 물체탐지를 하지않는 물체추적 시간을 더하여 계산했다. 싱글태스크 프로세스는 각 태스크의 물체탐지를 각각 수행해야하기 때문에 물체탐지를 포함한 자세추론 시간에 물체탐지를 포함한 물체추적 시간을 더해 총 수행시간을 계산했다. 싱글태스크 프로세스에서는 영상의 부복호화를 수행하지 않기 때문에 멀티태스크 프로세스에서 수행하는 영상 부복호화 시간은 제외했다.

표 7. 멀티태스크와 싱글태스크의 소요시간 비교
Table 7. Comparison of time taken for the multitask and single task

(unit : sec)

	Object detection	Pose estimation (w/o OD)	Object tracking (w/o OD)	Entire time (OD+PE+OT)
2.mp4	1200	1552	297	3049
4.mp4	274	554	91	919
7.mp4	252	709	91	1052
16.mp4	286	474	88	848

(a) Taken time for multi-task process

(unit : sec)

	Pose estimation (w/OD)	Object tracking (w/OD)	Entire time (PE+OT)
2.mp4	2066	394	2460
4.mp4	725	118	843
7.mp4	830	119	949
16.mp4	592	95	687

(b) Taken time for single-task process

멀티태스크 프로세스의 수행시간이 싱글태스크의 수행시간보다 전반적으로 더 오래 걸렸다. 멀티태스크 프로세스의 수행시간이 긴 이유는 물체탐지 시에 전경영상 생성을 위한 연산 시간 및 추론결과 파일과 전경영상을 저장할 때 발생하는 파일입출력 시간이 추가되어 더 긴 시간이 소요됐기 때문이다. 또한 물체추적 태스크에서 사용되는 물체탐지 인공지능 네트워크인 YOLO v3 1088x608이 멀티태스크 프로세스에서 사용하는 Faster R-CNN X101-FPN 보다 약 3배가량 빠른 연산능력을 보이기 때문에[12] 물체탐지 인공지능 네트워크의 성능차이로 인한 총 소요시간이 차이 또한 발생한다.

V. 결 론

실험결과 물체탐지 추론결과 파일과 전경영상을 압축했을 때 BPP가 크게 감소하였다. BPP가 크게 감소하는 것에 비해 자세추론의 mAP 결과는 소폭 감소하였고, 물체추적의 MOTA 결과는 감소하지 않았다. 멀티태스크 프로세스의 수행시간은 싱글태스크 프로세스에 비해 상대적으로 더 많은 시간이 걸렸다. 후처리 태스크의 숫자가 많아지면 멀티태스크 프로세스의 수행시간 효율이 높아질 수 있다.

멀티태스크 프로세스에서 자세추론과 물체추적 각각의 물체탐지부를 수행하지 않고 물체탐지를 한번만 선수행하여 각 네트워크의 입력으로 대체할 수 있을지 알아보기 위해 싱글태스크 프로세스에서 각 태스크의 물체탐지 네트워크를 변경하지 않고 실험을 수행했다. 각 태스크의 물체탐지 인공지능 네트워크가 서로 달라 정확한 mAP 및 MOTA 비교가 어려웠다. 향후 모든 태스크의 물체탐지 네트워크를 일치시켜 본 논문에서 제안하는 멀티태스크 프로세스의 효율을 알아보는 실험이 필요하다. mAP 및 MOTA는 물체탐지 인공지능 네트워크의 설정을 어떻게 하느냐에 따라, ground truth가 어떻게 만들어졌는지에 따라 결과가 달라질 수 있기 때문에 ground truth 작성 시 유의해야 한다.

향후 계획으로는 현재 VCM 앵커의 ground truth와 비슷한 결과를 도출하는 물체 탐색 신뢰도 점수를 구하고, 이를 통해 추출되는 영상과 추론 결과를 압축하여 좀 더 객관적인 실험결과를 도출할 예정이다. 아울러 추론결과의 XML

에 대한 효율적인 표현 방법(예: 이진화)을 연구하여, 압축 효율 증진을 위한 연구를 진행할 것이다. 또한 행동인식(i.e., action recognition)과 같은 후처리 인공지능 네트워크 태스크를 추가하여 제안된 방법의 성능 비교를 수행할 예정이다.

참 고 문 헌 (References)

[1] Y. Jang, D. Chung, "Technology Trend for Image Analysis Based on Deep Learning," Current Industrial and Technological Trends in Aerospace, vol.17, No.1, pp.113-122, July 2019.

[2] M. Jeong, S. Kim, H. Jin, H. Lee, H. Choo, H. Lim, and J. Seo, "Experiment on the Effect of Feature Map Encoding on CNN Performance Evaluation," JOURNAL OF BROADCAST ENGINEERING, vol.25, No.7, pp.1081-1094, December 2020.
doi: https://doi.org/10.5909/JBE.2020.25.7.1081

[3] H. Jin, M. Jeong, D. Yoo, S. Kim, J. Lee, H. Lee, and W. Cheong, "Compression of CNN Inference Results Using MPEG-7 Descriptor Binarization," Proceedings of the Korean Society of Broadcast Engineers Conference, pp.36-38, June 2021.

[4] S. Wenkel, K. Alhazmi, T. Liiv, S. Alrshoud, M. Simon, "Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation," Sensors, Vol. 21, No.13: 4350, 2021, (accessed May. 3, 2022).
doi: https://doi.org/10.3390/s21134350.

[5] H. Lee, J. Lee, H. Choo, W. Cheong, J. Seo, "[VCM] Object of interest based VCM for multi-task," ISO/IEC JTC1/SC29/WG02 m58846, Online, January 2022.

[6] W. Lin, K. Dong, R. Yang, T. Wang, A. Zhang and D. Liu, "[VCM] Anchor generation for HiEve(object tracking)," ISO/IEC JTC1/SC29/WG02 m55761, Online, December 2020.

[7] Github - facebookresearch/detectron2, https://github.com/facebookresearch/detectron2 (accessed Apr. 10, 2022).

[8] FFmpeg, https://ffmpeg.org/ (accessed Apr. 10, 2022).

[9] VTM-12.0 jvet/VVCSoftware, https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-12.0 (accessed Apr.10, 2022).

[10] Github - leoxiaobin/deep-high-resolution-net.pytorch, https://github.com/leoxiaobin/deep-high-resolution-net.pytorch(accessed Apr.10, 2022).

[11] Github - Zhongdao/Towards-Realtime-MOT, https://github.com/Zhongdao/Towards-Realtime-MOT (accessed Apr.10, 2022).

[12] J. Redmon and A. Farhadi, "YOLO v3: An Incremental Improvement", Computer Vision and Pattern Recognition, 2018. (accessed Apr. 10, 2022).
doi: https://doi.org/10.48550/arXiv.1804.02767

저 자 소 개

정 민 혁

- 2009년 ~ 2016년 : 명지대학교 컴퓨터공학과 학사
- 2016년 ~ 2018년 : 명지대학교 일반대학원 컴퓨터공학과 석사
- 2018년 ~ 현재 : 명지대학교 일반대학원 컴퓨터공학과 박사과정
- ORCID : https://orcid.org/0000-0001-6487-9219
- 주관심분야 : Internet of Things, Virtual Reality, 4D media

김 상 균

- 1997년 : 아이오와대(U of Iowa) 전산과학, BS(1991), MS(1995), PhD
- 1997년 3월 ~ 2007년 2월 : 삼성종합기술원 멀티미디어랩 전문연구원
- 2007년 3월 ~ 2016년 2월 : 명지대학교 컴퓨터공학과 교수
- 2007년 3월 ~ 현재 : 명지대학교 융합소프트웨어학부 데이터테크놀로지전공 교수
- ORCID : https://orcid.org/0000-0002-2359-8709
- 주관심분야 : digital content analysis and management, fast image search and indexing, color adaptation, 4D media, sensors and actuators, VR, Internet of Things, and multimedia standardization

저 자 소 개

이 진 영

- 1998년 : 미시간주립대(Michigan State University) 학사
- 1999년 : 미시간주립대 석사
- 2008년 : 미시간주립대 박사
- 2004년 ~ 현재 : 한국전자통신연구원
- ORCID : https://orcid.org/0000-0002-8718-1961
- 주관심분야 : 디지털통신/방송시스템, 3DTV, MMT

추 현 곤

- 1998년 2월 : 한양대학교 전자공학과 (공학사)
- 2000년 2월 : 한양대학교 전자공학과 (공학석사)
- 2005년 2월 : 한양대학교 전자통신전파공학과 (공학박사)
- 2005년 2월 ~ 현재 : 한국전자통신연구원 선임연구원
- 2015년 1월 ~ 2017년 1월 : 한국전자통신연구원 디지털홀로그래피연구실장
- 2017년 9월 ~ 2018년 8월 : Warsaw University of Technology, Poland 방문연구원
- ORCID : https://orcid.org/0000-0002-0742-5429
- 주관심분야 : Computer vision, 3D imaging and holography, 3D depth imaging, 3D broadcasting system

이 희 경

- 1999년 2월 : 영남대학교 공과대학 컴퓨터공학과 공학사
- 2002년 2월 : KAIST-ICC 정보통신공학부 공학석사
- 2002년 ~ 현재 : 한국전자통신연구원 실감미디어연구실 책임연구원
- ORCID : https://orcid.org/0000-0002-1502-561X
- 주관심분야 : 디지털방송 HCI, Gaze Tracking, VR/AR/MR

정 원 식

- 1992년 2월 : 경북대학교 전자공학과 (공학사)
- 1994년 2월 : 경북대학교 대학원 전자공학과 (공학석사)
- 2000년 2월 : 경북대학교 대학원 전자공학과 (공학박사)
- 2000년 5월 ~ 현재 : 한국전자통신연구원 책임연구원
- ORCID : https://orcid.org/0000-0001-5430-2969
- 주관심분야 : 3DTV 방송 시스템, 라이트필드 이미징, 영상부호화, 딥러닝기반 신호처리, 멀티미디어 표준화