

특집논문 (Special Paper)

방송공학회논문지 제27권 제3호, 2022년 5월 (JBE Vol.27, No.3, May 2022)

<https://doi.org/10.5909/JBE.2022.27.3.318>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

피쳐 프레임 구성 방법에 따른 피쳐 맵 압축 효율 및 머신 태스크 성능 분석

이 성 배^{a)}, 이 민 석^{a)}, 김 규 현^{a)†}

Analysis of Feature Map Compression Efficiency and Machine Task Performance According to Feature Frame Configuration Method

Seongbae Rhee^{a)}, Minseok Lee^{a)}, and Kyuheon Kim^{a)†}

요 약

최근 하드웨어 연산 장치와 소프트웨어 기반 프레임워크의 발전으로 딥러닝 네트워크를 활용한 머신 태스크가 다양한 산업 분야 및 개인 IoT 장비에서의 활용이 기대되고 있다. 그러나 딥러닝 네트워크를 구동하기 위한 장치의 고비용 문제와 서버에서 머신 태스크 결과만을 전송받을 때 사용자가 요구하는 결과를 받지 못할 수 있다는 제한 사항을 극복하기 위하여 Collaborative Intelligence (CI)에서는 피쳐 맵의 전송을 그 해결 방법으로 제시하였다. 본 논문에서는 CI 패러다임을 지원하기 위하여 방대한 데이터 크기를 갖는 피쳐 맵의 효율적인 압축 방법을 실험을 통해 분석 및 제시하였다. 해당 방법은 전통적인 비디오 코덱에서의 압축 효율을 높이기 위하여 피쳐 맵의 재정렬을 적용하여 중복성을 높였으며, 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에 활용하여 압축 효율을 높이고 머신 태스크의 성능을 유지하는 피쳐 맵 방법을 제시하였다. 본 논문에서는 이와 같은 방법의 분석을 통해 MPEG-VCM의 피쳐 압축 앵커 대비 BPP와 mAP의 BD-rate에서 14.29%의 성능이 향상됨을 검증하였다.

Abstract

With the recent development of hardware computing devices and software based frameworks, machine tasks using deep learning networks are expected to be utilized in various industrial fields and personal IoT devices. However, in order to overcome the limitations of high cost device for utilizing the deep learning network and that the user may not receive the results requested when only the machine task results are transmitted from the server, Collaborative Intelligence (CI) proposed the transmission of feature maps as a solution. In this paper, an efficient compression method for feature maps with vast data sizes to support the CI paradigm was analyzed and presented through experiments. This method increases redundancy by applying feature map reordering to improve compression efficiency in traditional video codecs, and proposes a feature map method that improves compression efficiency and maintains the performance of machine tasks by simultaneously utilizing image compression format and video compression format. As a result of the experiment, the proposed method shows 14.29% gain in BD-rate of BPP and mAP compared to the feature compression anchor of MPEG-VCM.

Keyword : Deep Learning, Video Coding for Machine, Feature Frame, Feature Map Compression, Moving Picture Expert Group

Copyright © 2022 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

1. 서론

최근 하드웨어 연산 장치와 소프트웨어 기반 프레임워크의 발전으로 딥러닝 (Deep Learning) 네트워크를 활용한 데이터 처리 연구가 빠르게 진행되고 있다. 특히, 영상 처리 분야에서는 입력 영상의 특징 (Feature)을 서술하는 합성곱 신경망 (CNN, Convolutional Neural Network)을^[1] 활용하여 Object Detection, Instance Segmentation, Pose Estimation, Super Resolution 등 다양한 머신 태스크 (Machine Task)에 대한 연구가 활발히 진행되고 있다. 또한, 딥러닝 네트워크 기반의 머신 태스크는 의료, 군사, 보안, 제조 등 다양한 전문 산업 분야에서 활용이 시도되고 있으며, 더 나아가 개인 IoT (Internet of Things) 장비에서의 활용 또한 기대되고 있다. 그러나 딥러닝 네트워크 기반의 영상 처리 기술에는 수많은 파라미터와 연산 과정이 필요하기 때문에 고성능의 장치가 요구되며, 이는 전문 산업 분야와 개인 IoT 장비에서 머신 태스크를 활용함에 있어 고비용 문제가 발생할 수 있다.

이와 같이 머신 태스크를 위한 장치의 고비용 문제를 극복하기 위하여 딥러닝 기술의 서비스 형태는 고사양의 중앙 디바이스에 딥러닝 네트워크를 위치시키고, 머신 태스크의 결과만을 개인/단말 디바이스에 전송하는 구조를 취하기도 한다^[2]. 그러나 이러한 서비스 구조에서 딥러닝 네트워크는 오직 설계자가 정의한 결과에 대해서만 모델 학습이 진행되기 때문에 사용자가 원하는 머신 태스크 결과를 얻지 못하는 문제가 발생할 수 있으며, 사용자가 요구하는 결과에 대한 딥러닝 네트워크의 재학습 또한 제한될 수 있다.

이에 딥러닝 네트워크가 개인/단말 디바이스에 위치하는 경우 발생할 수 있는 장비의 고비용 문제와 머신 태스크 결과만을 전송 받는 경우에 발생할 수 있는 비정의 문제를 극복하기 위하여 딥러닝 네트워크를 고사양의 디바이스와 비교적 저사양의 디바이스에 분할시켜 머신 태스크를 수행하는 Collaborative Intelligence (CI)가 제안되었다^[3]. CI 구조에서 고성능의 디바이스는 딥러닝 네트워크의 컨볼루션 연산을 진행하고, 그 결과 값인 피쳐 맵을 저사양의 디바이스에 전송한다. 이에 피쳐 맵을 전달받은 디바이스에서는 해당 피쳐 맵이 추출된 위치까지의 딥러닝 네트워크의 컨볼루션 연산을 진행하지 않아도 되기 때문에 비교적 저사양 디바이스에서도 머신 태스크를 수행할 수 있다.

그러나 딥러닝 네트워크에서 추출되는 피쳐 맵은 입력 영상 한 장에 대하여 2차원 형태의 피쳐 맵이 해당 레이어의 채널 수 만큼 생성되는 거대한 데이터이기 때문에 전송 비용이 비싸다는 제한 사항이 있다. 한편, 이와 같은 피쳐 맵의 전송 비용 문제는 국제표준화기구 MPEG (Moving Picture Expert Group)의 기계를 위한 부호화 (VCM, Video Coding for Machine) 그룹에서 논의되고 있으며, 머신 태스크의 성능을 일정 수준 확보하면서 피쳐 맵을 효율적으로 압축하는 기술 표준 개발을 목표로 진행하고 있다^[4]. MPEG-VCM에서는 전통적인 비디오 코덱인 VVC^[5] (VTM-12.0)를 통해 모든 피쳐 맵을 하나의 정지 영상에 패킹하여 압축하는 것을 피쳐 코딩의 anchor로 정의하였으며^[4], 제안된 피쳐 코딩 방법으로는 비디오 코덱의 특성을 활용한 피쳐 맵 압축 방법과 PCA (Principal Component Analysis)^[6], Resizing^[7]을 통해 피쳐 맵의 차원을 축소한 이후 압축하는 방법 등이 대표적이다.

앞선 MPEG-VCM에서 제안된 피쳐 코딩 기술 중에서 피쳐 맵의 차원을 축소하는 PCA 기반과 Resizing 기반의 피쳐 맵 압축 방법이 전통적인 비디오 코덱의 특성만을 활용한 피쳐 맵 압축 방법보다 압축률 대비 머신 태스크의 성능이 높게 나타나고 있다. 그러나 전통적인 비디오 코덱만을 활용한 피쳐 맵 압축 방법은 설계가 간단하고 기존에 설치된 비디오 압축 환경만으로 구현이 가능하다는 장점이 존재한다. 이에, 본 논문에서는 전통적인 비디오 코덱만을 활용하여 피쳐 맵의 중복성을 높인 피쳐 프레임 (Feature Frame) 구성 방법을 통해서 효율적인 피쳐 맵 압축 및 안정

a) 경희대학교 전자정보융합공학과(Graduate School of Electronic Information Convergence Engineering)

‡ Corresponding Author : 김규현(Kyuheon Kim)

E-mail: kyuheonkim@khu.ac.kr

Tel: +82-31-201-2442

ORCID: <https://orcid.org/>

※ 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00011, (전문연구실)기계를 위한 영상부호화 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 18, 2022; Revised May 13, 2022; Accepted May 13, 2022.

적인 머신 태스크를 수행하는 방법을 제안하고자 한다. 이에 2장에서는 기존 전통적인 비디오 코덱의 특성을 활용한 피쳐 맵 압축 방법을 분석하고, 이어지는 3장에서는 피쳐 프레임 간의 중복성을 높이기 위한 피쳐 프레임 구성 방법에 따른 피쳐 맵 압축 효율성과 머신 태스크 안정성의 상관 관계를 분석하였다. 4장에서는 이와 같은 분석을 통해 효율적인 피쳐 프레임 구성 방안에 대한 기준을 제시하였다.

II. 배경 기술

MPEG-VCM에서 전통적인 비디오 코덱만을 활용하여 딥러닝 네트워크의 피쳐 맵을 압축하는 절차는 그림 1에서 나타나는 바와 같이, 피쳐 맵 압축 Encoder의 경우 Feature Extraction, Feature Conversion, Video Encoder로 구성되며, 피쳐 맵 압축 Decoder의 경우는 Video Decoder, Feature Conversion, Machine Analysis로 구성된다^[8]. 그림 1에서 나타나는 첫 번째 단계와 마지막 단계인 Feature Extraction과 Machine Analysis는 딥러닝 네트워크의 내부에서 진행되는 단계로써, 각각 입력 영상에 대한 피쳐 맵 추출 과정과 전송받은 피쳐 맵을 통한 머신 태스크 진행 단계에 해당한다. 또한, Video Encoder와 Video Decoder 단계는 각각 VVC (VTM-12.0)을^[9] 통해서 피쳐 맵 데이터

를 비트스트림 (Bitstream)으로 부호화하고 복호화하는 단계를 나타낸다. 그리고 피쳐 맵 압축 구조에서 Encoder와 Decoder 양쪽에 위치한 Feature Conversion 단계는 전통적인 비디오 코덱의 특성을 활용해 압축 효율을 높이는 모든 방법들을 지칭한다. 이때 Feature Conversion으로 제안된 방법으로는 피쳐 맵의 구조를 변화하여 압축을 진행하는 방법, 피쳐 맵의 순서를 재정렬 (Reordering)하여 압축하는 방법, 정지 영상 압축과 동영상 압축 방법을 적용한 방법으로 구분할 수 있으며, 각각에 대한 설명은 다음과 같다.

1. 피쳐 맵 구조 변화

Feature Conversion으로 제안된 기술 중 피쳐 맵의 구조를 변화하여 압축을 진행하는 방법은 퀴팅 (Quilting)[10]이 대표적이다. 퀴팅은 한 채널에 존재하는 모든 피쳐 맵에서 동일한 픽셀 위치의 값들을 군집 단위로 타일링 (Tiling)하는 기술이다. 일례로, 모든 피쳐 맵을 하나의 정지 영상에 패킹한 그림 2의 (a)에 대하여 퀴팅을 적용한 예시는 그림 2의 (b)와 같이 나타난다.

일반 카메라를 통해서 촬영된 동영상에 대하여 N장의 프레임을 퀴팅하는 경우에는 동영상 프레임 간 동일한 픽셀 위치에서의 성분 값이 유사하기 때문에 중복성이 높은 퀴

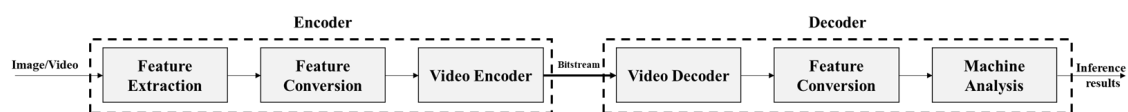
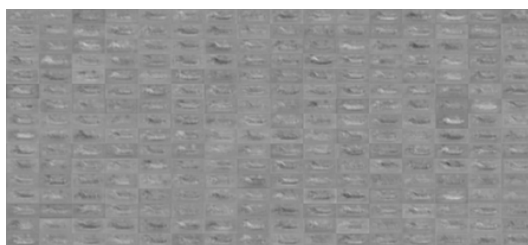
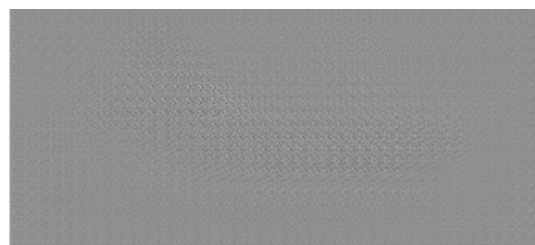


그림 1. MPEG-VCM의 피쳐 맵 압축 구조 예시

Fig. 1. Examples of feature coding structure in MPEG-VCM



(a) Tiling



(b) Quilting

그림 2. 피쳐 맵 패킹 (a) 타일링 (Tiling) (b) 퀴팅 (Quilting)

Fig. 2. Packing feature maps into image by (a) Tiling (b) Quilting

팅 영상이 생성되어 압축 효율이 발생할 수 있지만, 피쳐 맵의 경우에는 채널마다 서로 다른 커널에 대한 컨볼루션 결과이기 때문에 피쳐 맵들의 동일 픽셀 위치일지라도 값들의 유사성이 작아 그림 2의 (b)와 같이 복잡한 패턴 영상이 나타나는 것을 확인할 수 있다. 이로 인하여, 피쳐 맵에 킬링을 적용하여 압축하는 경우에는 압축 성능이 좋지 못한 것으로 나타난다^[10].

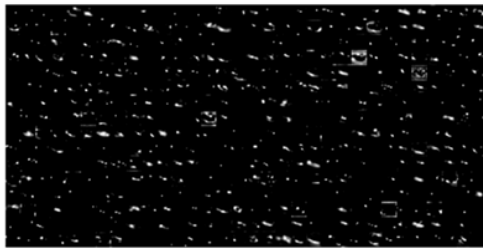
2. 피쳐 맵 재정렬

피쳐 맵을 재정렬하여 압축하는 방식^[11,12,13]은 특정한 정렬 기준을 설정하여 피쳐 맵에 순서를 부여하고, 피쳐 맵이 패킹된 정지 영상의 중복성을 높여 압축 효율을 높이는 방법이다. 이때 정렬 기준은 단순히 밝기 정보를 활용하여 정렬할 수 있고, 가장 밝은 피쳐 맵을 정의한 이후 MSE (Mean Squared Error)^[14], SSIM (Structure Similarity Index Measure)^[15] 등의 영상 평가 방법을 활용하여 구조적 정렬

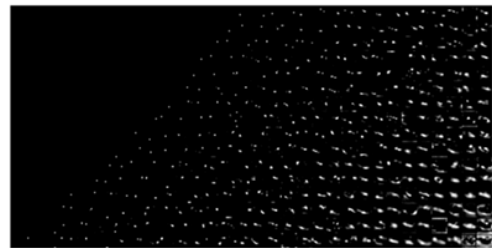
을 추가로 진행할 수 있다. 일례로, 그림 3의 (a)와 같이 나타나는 피쳐 맵에 대하여 밝기 정렬로 재정렬하는 경우, 그림 3의 (b)와 같이 나타난다. 이와 같이 피쳐 맵을 재정렬하여 중복성을 높여 압축하는 경우 압축 성능이 약간 향상함에도 머신 태스크의 성능이 유지되는 것으로 나타난다^[11,12,13].

3. 정지 영상 기반 피쳐 맵 압축과 동영상 기반 피쳐 맵 압축

전통적인 비디오 코덱을 활용하여 피쳐 맵을 압축하는 경우에는 그림 4의 (a)에서 나타나는 바와 같이 하나의 정지 영상에 타일링 패킹하여 정지 영상 압축 포맷으로 진행할 수 있으며, 이는 AI (All-Intra) 모드로 부호화된다. 또한, 그림 4의 (b)에서 나타나는 바와 같이 하나의 피쳐 맵을 하나의 동영상 프레임으로 간주하여 피쳐 맵 채널 수 만큼의 프레임이 구성된 피쳐 맵 동영상에 대하여 동영상 압축 포

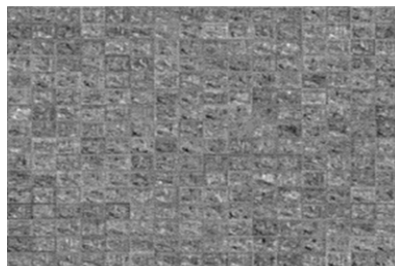


(a) Tiling

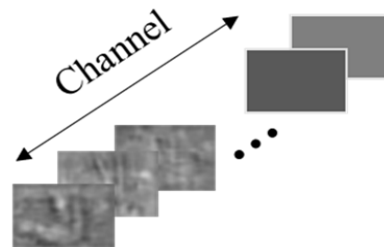


(b) Reordering

그림 3. 피쳐 맵 재정렬 예시 (a) Tiling (b) Reordering
Fig. 3. Examples of feature map reordering (a) Tiling (b) Reordering



(a) Image Compression Format



(b) Video Compression Format

그림 4. 피쳐 맵 압축 포맷 예시 (a) 정지 영상 압축 포맷 (b) 동영상 압축 포맷
Fig. 4. Examples of feature map compression format (a) Image Compression Format (b) Video Compression Format

맷으로 진행할 수 있으며, 이는 RA (Random-Access) 모드로 부호화된다. 이때 정지 영상 압축 포맷으로 피쳐 맵을 압축하는 것은 동영상 압축 포맷으로 피쳐 맵을 압축하는 것보다 머신 태스크의 성능이 높게 유지되는 것으로 나타났다지만, 압축 효율이 적은 것으로 나타났다^[16].

이에, Suzuki et al.^[16]은 모든 피쳐 맵으로 2장의 피쳐 프레임을 구성함으로써, 정지 영상 압축 포맷과 동영상 압축 포맷이 결합된 피쳐 맵 압축 방법을 제안하였다. 제안 방법의 실험에서는 ImageNet으로 사전 학습된 VGG-16^[17] 네트워크를 활용하여 ImageNet-2012 validation set에서 임의로 선택한 10,000장의 정지 영상에 대하여 8번째 레이어의 피쳐 맵을 8비트로 양자화하여 QP (Quantization Parameter) $\in \{22, 27, 32, 37\}$ 에 대하여 RA 모드로 부호화하였고, KBPI (Kbytes/images) 대비 Top-1 accuracy를 비교하였으며, 그 결과는 그림 5에서 나타나는 바와 같다. 이때, 정지 영상 압축 포맷과 동영상 압축 포맷의 압축은 각각 AI 모드와 RA 모드로 부호화하였다. 또한, 같은 실험 조건으로 VGG-16의 11번째 피쳐 맵과 ResNet-18^[22]의 13번째와 15번째 피쳐 맵 각각에 대한 압축 실험한 결과는 표 1과 같이 나타났다. 표 1은 제안 방법을 정지 영상 압축 포맷과 동영상 압축 포맷 각각에 대하여 KBPI와 Image Recognition Accuracy의 BD-rate^[23] 관점에서 비교 결과로써, 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에 활용하는 경우 정지 영상 압축 포맷에 비하여 약간의 BD-rate 성능 향상이 나타나는 것을 확인할 수 있다.

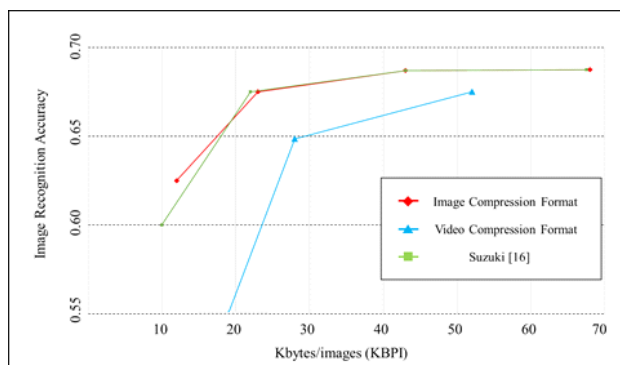


그림 5. 피쳐 맵 압축 포맷에 대한 압축 성능 및 머신 태스크 성능 비교
Fig. 5. Comparison results of compression and machine task performance according to feature map compression format

표 1. 정지 영상 압축 포맷, 동영상 압축 포맷, Suzuki[16] BD-rate 비교 결과

Table 1. BD-rate results of Suzuki[16] with Image Compression Format and Video Compression Format

Networt	Layer	vs. Image Compression Format	vs. Video Compression Format
VGG	L8	-1.80%	-13.1%
	L11	-1.37%	-27.8%
ResNet	L13	-1.39%	-28.1%
	L13	-3.57%	-50.7%

상기의 제안 기술들을 통해서 피쳐 맵을 재정렬하는 것과 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에 활용하는 경우 피쳐 맵의 압축과 머신 태스크 성능에서 이득이 있음을 확인하였다. 한편, 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에 활용한 실험에서^[16] 정지 영상 압축 포맷의 Image Recognition Accuracy는 높게 나타나지만 KBPI 또한 높아 압축률이 적게 나타나는 반면, 동영상 압축 포맷의 Image Recognition Accuracy는 낮지만 KBPI는 작아 압축률이 크게 나타나는 것을 통해 압축률과 머신 태스크의 성능이 트레이드오프 관계임을 확인할 수 있다. 또한, 해당 실험에서 단일 계층에 존재하는 모든 피쳐 맵을 단 2장의 피쳐 프레임에 패킹하여 RA 부호화한 결과에서 KBPI와 Image Recognition Accuracy의 BD-rate 성능이 향상된 것을 고려할 때, 피쳐 프레임에 할당하는 피쳐 맵의 개수를 달리하여 RA 부호화를 진행함으로써 피쳐 맵의 압축 성능과 머신 태스크의 성능이 최적인 지점을 확인할 필요가 있다.

이에 본 논문에서는 피쳐 맵의 재정렬을 적용하여 비디오 코덱에서의 압축 효율을 높이고, 피쳐 프레임에 할당하는 피쳐 맵의 개수를 다양하게 설정하여 실험함으로써, 피쳐 맵의 압축 성능과 머신 태스크의 성능이 최적인 지점을 확인하여 효율적인 피쳐 프레임 기반 피쳐 맵 압축 방법을 제시하고자 한다. 또한, 단일 계층으로 구성된 딥러닝 네트워크의 피쳐 맵 압축뿐만 아니라, 다계층으로 구성된 딥러닝 네트워크의 피쳐 맵을 압축하는 방법을 이어지는 3장에서 설명하고자 한다.

III. 피쳐 프레임 구성 방법에 따른 압축 효율 및 머신 태스크 성능 분석

본 논문에서는 전통적인 비디오 코덱만을 활용하여 설계가 간단하고 기존에 설비된 비디오 압축 환경만으로도 구현이 가능한 딥러닝 네트워크의 피쳐 맵 압축 방법을 압축 효율과 머신 태스크 성능 결과를 토대로 분석하고자 한다. 앞서 설명하였듯이, 본 논문에서 제시하는 방법은 단일 계층으로 구성된 딥러닝 네트워크의 피쳐 맵 압축뿐만 아니라, 다계층으로 구성된 딥러닝 네트워크의 피쳐 맵 압축 또한 목표하기 때문에 FPN (Feature Pyramid Network) 구조를 백본 (Backbone) 네트워크로 사용하는 딥러닝 네트워크를 활용하였다. 그 중 Object Detection에 대한 머신 태스크를 수행하는 Faster R-CNN X101-FPN^[18]로부터 추출된 P-layer의 피쳐 맵에 대하여 실험을 진행하였다.

본 실험에서는 피쳐 맵의 중복성을 높이기 위한 방법으로 피쳐 맵의 재정렬을 진행하며, 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에 활용할 경우 피쳐 맵의 압축 성능과 머신 태스크의 성능이 최적인 지점을 실험을 통해 확인함으로써 피쳐 프레임의 구성 방법을 정의하고자 한다. 또한, 재정렬된 피쳐 맵을 피쳐 프레임에 할당할 때 Spatial-priority 할당 방법과 Temporal-priority 할당 방법 중에서 어떤 방법이 적합한 할당 방법인지를 실험을 통해서 확인하고자 한다.

이에, 피쳐 맵의 압축 성능과 머신 태스크 성능에 대한 경향성 확인에는 COCO2017 validation set^[19]에서 임의로 선택한 100장의 테스트 영상으로 실험하였으며, MPEG-VCM에서 정의한 압축 코덱인 VTM-12.0의 부호화 시간이 오래 걸리는 점을 감안하여 VVenC^[20]를 압축 코덱으로 활용하였다. 또한, 실험을 통해 확인한 경향성을 검증하기 위하여, 마지막 실험에서는 MPEG-VCM에서 정의한 실험 환경인 OpenImageV6 validation set^[21] 중 5,000장의 테스트 영상에 대하여 VTM-12.0으로 피쳐 프레임을 압축하고 그 결과를 MPEG-VCM의 Feature compression anchor와 비교하였다.

1. 피쳐 맵 재정렬 실험

본 실험은 피쳐 맵의 재정렬을 통하여 피쳐 프레임을 구

성하였을 때 피쳐 맵의 압축 성능과 머신 태스크의 성능의 정도를 확인하기 위한 실험이지만, 아직 피쳐 프레임에 대한 정의를 하지 않았기 때문에 모든 피쳐 맵을 하나의 정지 영상에 타일링 패킹한 경우에 대해서 재정렬 실험을 진행하였으며, 피쳐 맵 재정렬을 피쳐 프레임에 적용한 실험 결과는 본 논문 3장의 3절에서 설명하고자 한다.

백본 네트워크의 P2, P3, P4, P5 계층에서 추출되는 피쳐 맵은 각 계층에 대하여 독립적으로 재정렬하였으며, 이때의 재정렬은 8비트로 양자화된 피쳐 맵을 흑백 영상으로 간주하여 한 피쳐 맵별로 평균 밝기 값을 계산하고, 평균 밝기의 크기를 순서로 피쳐 맵을 재정렬하였다. 일례로, P2에 존재하는 256개의 피쳐 맵에 대한 재정렬은 256번의 평균 밝기 계산이 필요하며, 256개의 피쳐 맵은 각 피쳐 맵의 평균 밝기 크기를 기준으로 재정렬된다. COCO2017 validation set에서 임의로 선택한 100장의 테스트 영상에 대하여 재정렬하여 타일링 패킹한 결과는 표 2와 같이 나타난다. 이때 각 P2, P3, P4, P5 계층마다 존재하는 256개의 피쳐 맵을 가로에 16개, 세로에 16개를 할당하여 피쳐 정지 영상을 구성하여 총 4장의 피쳐 정지 영상을 구성한다. 구성된 피쳐 정지 영상은 각각 $QP \in \{22, 27, 32, 37, 42, 47\}$ 에 대하여 VVenC 코덱을 통해 압축하였다. 또한, 평균 밝기를 통해 재정렬된 피쳐 맵은 피쳐 정지 영상에 할당될 때 Raster-scan 순서로 할당하였다. 각 계층마다 생성된 피쳐 정지 영상의 압축은 AI 모드를 통해서 압축되어야 하지만, VVenC에서 지원되는 모드는 RA이므로 RA-medium Configuration을 활용하였다.

표 2에서 나타나는 Original feature map은 재정렬이 적용되지 않은 원본의 피쳐 맵을 딥러닝 네트워크의 채널 인덱스 순서에 따라 타일링하여 압축한 경우이고, Reordered feature map은 밝기 값 크기를 기준으로 재정렬된 피쳐 맵을 Raster-scan 순서로 타일링하여 압축한 경우이다. 이때 BPP는 피쳐 맵의 부호화된 비트스트림 용량에 대하여 입력 영상의 가로와 세로 픽셀 크기로 나눈 값을 비트(Bit) 단위로 표현한 값이며, 재정렬된 피쳐 맵 압축의 경우 피쳐 맵을 원래의 순서로 복원하기 위한 리스트 값으로 입력된 정지 영상 당 8192 비트를 추가하였으며, 이는 P2, P3, P4, P5에 대한 계층 수 4, 한 계층마다 존재하는 피쳐 맵의 채널 수 256, 0-255 사이의 인덱스를 복원하기 위한 용량 1 바이트

표 2. 피쳐 맵의 재정렬 실험 결과 (a) Original feature map (b) Reordered feature map

Table 2. Experiment results of feature map reordering (a) Original feature map (b) Reordered feature map

Original feature map compression	QP	mAP	BPP	Reordered feature map compression	QP	mAP	BPP
Uncompressed	-	44.13	-	Uncompressed	-	44.13	-
Compressed	27	44.13	49.74	Compressed	27	44.13	49.58
	32	43.08	25.63		32	43.18	25.48
	37	40.98	11.77		37	41.14	11.61
	42	37.23	4.83		42	37.08	4.66
	47	27.10	1.83		47	27.44	1.65
BD-rate	- 4.86%						

(Byte), 바이트를 비트로 변환을 위한 8을 모두 곱한 값이다. 표2에서 나타나는 바와 같이, 재정렬된 경우와 재정렬이 적용하지 않은 경우의 BPP와 mAP (Mean Average Precision)의 BD-rate에서 재정렬된 피쳐 맵 압축에서 4.86%의 이득이 있음을 확인할 수 있고, 이는 피쳐 맵의 재정렬이 전통적인 비디오 코덱을 통해서 피쳐 맵을 압축할 때 필요한 절차임을 확인할 수 있다.

2. 피쳐 프레임 구성 실험

피쳐 프레임 구성에 대한 실험은 앞서 설명한 정지 영상 압축 포맷과 동영상 압축 포맷을 결합하여 사용할 때 한 피쳐 프레임에 할당하는 피쳐 맵의 개수에 따라 피쳐 맵 압축 효율과 머신 태스크 성능이 달라질 수 있기에, 프레임을 구성하는 피쳐 맵과 태스크 성능에 대한 최적화된 지점

의 존재를 파악하는 것이 요구된다. 일례로, 피쳐 프레임을 구성하는 것은 그림 6에서 나타나는 바와 같으며, 딥러닝 네트워크에 존재하는 각각의 P2, P3, P4, P5 계층에 대하여 독립적으로 일정 개수의 피쳐 맵을 묶어 피쳐 프레임을 구성한다.

이에, 백본 네트워크의 계층마다 존재하는 피쳐 맵의 개수는 256개로, 압축 코덱의 Configuration을 고려하여 2의 n승의 개의 피쳐 맵이 한 피쳐 프레임을 구성한다. 이에 할당되는 피쳐 맵의 개수를 (가로 할당개수, 세로 할당 개수) 형태로 표현하였을 때 생성되는 피쳐 프레임은 각각 (16, 16)의 피쳐 맵으로 1장의 프레임, (8, 8)의 피쳐 맵으로 4장의 피쳐 프레임, (4, 4)의 피쳐 맵으로 16장의 피쳐 프레임, (2, 2)의 피쳐 맵으로 64장의 피쳐 프레임, (1, 1)의 피쳐 맵으로 256장의 프레임이 생성된다. 이때 VVenC에서 AI 모드 부호화를 지원하지 않으므로 모든 프레임은 RA-me-

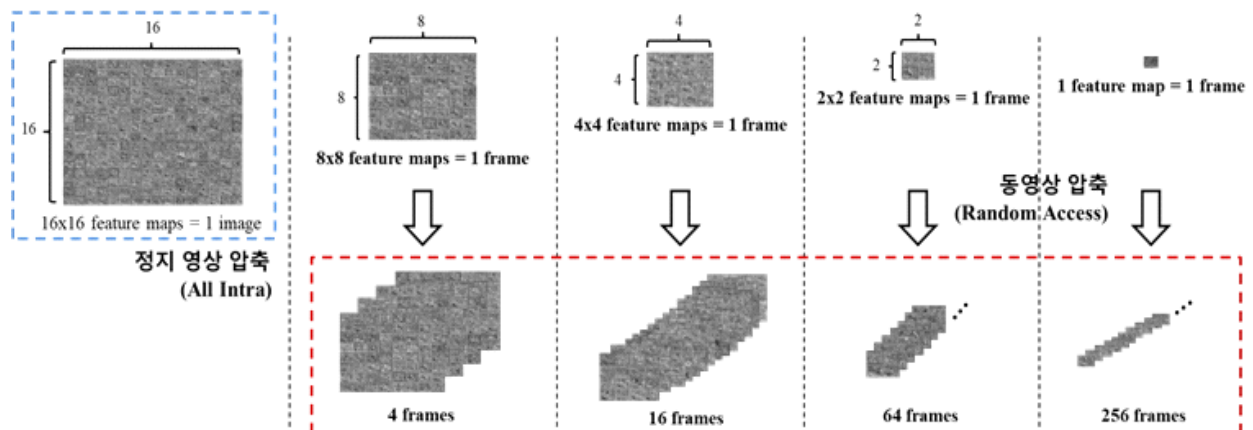


그림 6. 피쳐 프레임 예시

Fig. 6. Example of feature frame

dium Configuration을 통해 $QP \in \{22, 27, 32, 37, 42, 47\}$ 에 대하여 압축을 진행하였고, 이때 피처 맵의 양자화 크기는 8 비트이고 GOP 크기는 디폴트 값인 32를 사용하였다. 또한, 해당 실험은 피처 프레임에 할당되는 피처 맵 개수에 따른 BPP 대비 mAP 실험이므로 재배열은 적용하지 않았다.

앞선 실험 조건을 기반으로 COCO2017 validation set에서 임의로 선택한 100장의 테스트 영상에 대하여 피처 프레임을 구성하는 피처 맵 할당 개수에 따른 피처 맵 압축 성능을 실험한 결과는 표 3과 같이 나타나고, 머신 태스크 성능

을 실험한 결과는 표 4와 같이 나타난다. 먼저 표 3에서 나타나는 바와 같이, 피처 프레임을 구성하는 피처 맵의 할당 개수가 많을수록 BPP가 높게 나타나 압축 효율이 좋지 않은 것을 확인할 수 있고, 피처 프레임을 구성하는 피처 맵의 개수가 다름에도 (1, 1), (2, 2), (4, 4)개의 피처 맵으로 피처 프레임을 구성하는 경우 BPP가 비슷하게 나타나는 것을 확인할 수 있다.

한편, 피처 프레임을 구성하는 피처 맵의 할당 개수에 따라 달라지는 피처 맵의 머신 태스크 성능은 표 4와 같이 나타나며, 이때 피처 프레임을 구성하는 피처 맵의 할당 개

표 3. 피처 프레임 구성 방법에 따른 압축 성능 결과
Table 3. Compression performance results according to feature frame construction

Feature frame compression with VVenC (BPP)						
QP	-	feature frame (1,1)	feature frame (2,2)	feature frame (4,4)	feature frame (8,8)	feature frame (16,16)
22	Byte	1312665	1278044	1292249	1556970	3204944
	BPP	34.18	33.28	33.65	40.54	83.46
27	Byte	542249	515028.4	528706.5	729346.8	1910024
	BPP	14.12	13.41	13.76	18.99	49.74
32	Byte	232261.9	205257.2	212126.2	332612.1	984565.5
	BPP	6.04	5.34	5.52	8.66	25.63
37	Byte	111576.5	85075.56	86166.67	143281.5	452031.7
	BPP	2.90	2.21	2.24	3.73	11.77
42	Byte	61669.36	36165.26	33366.42	57010.35	185479.4
	BPP	1.60	0.94	0.86	1.48	4.83
47	Byte	42337.97	17998.58	13519.64	22151.45	70279.73
	BPP	1.10	0.46	0.35	0.57	1.83

표 4. 피처 프레임 구성 방법에 따른 머신 태스크 성능 결과
Table 4. Machine task performance results according to feature frame construction

Feature frame compression with VVenC (mAP)						
QP	feature frame (1,1)	feature frame (2,2)	feature frame (4,4)	feature frame (8,8)	feature frame (16,16)	Uncompressed
22	0.4413	0.4413	0.4413	0.4413	0.4413	0.44
27	0.4083	0.4077	0.4090	0.4165	0.4413	
32	0.3625	0.3520	0.3703	0.3773	0.4308	
37	0.2588	0.2573	0.2747	0.2893	0.4098	
42	0.0696	0.0620	0.0518	0.1277	0.372	
47	0.0006	0	0.0005	0.0045	0.2710	

수가 많을수록 mAP가 높게 나타나는 것을 확인할 수 있다. 이에, 피쳐 프레임을 구성하는 피쳐 맵의 할당 개수가 많을수록 압축 효율은 떨어지지만, 머신 태스크의 성능이 올라가는 트레이드오프 관계임을 확인할 수 있다.

이에, 피쳐 프레임을 구성하는 피쳐 맵의 개수에 따라 압축 및 머신 태스크 성능의 최적인 지점을 찾기 위하여 표 3과 표 4의 결과를 시각화하면 그림 7과 같이 나타난다. 그림 7에서 나타는 바와 같이, 가장 left-upper인 그래프는 (16, 16)개의 피쳐 맵으로 한 장의 프레임을 생성하여 정지 영상 압축한 것으로써 가장 효율적인 피쳐 맵 압축 방법으로 간주할 수 있지만, 해당 실험의 결과는 각각의 P2, P3, P4, P5 계층에 대하여 모두 동일한 피쳐 맵 할당 개수로 진행한 실험이다. 그러나, 각 계층이 제공하는 피쳐 맵의 특성이 다르기에 각각에 적합한 프레임 구성 피쳐 맵 할당 개수를 일괄적으로 적용할지 또는 개별적으로 정의하여야 할지에 대한 검토가 요구된다.

각 계층별로 서로 다른 개수의 피쳐 맵을 할당하여 피쳐 프레임을 구성하는 실험은 식 1에서 나타는 바와 같이 진행되었다. 가로 n 개, 세로 n 개의 피쳐 맵으로 구성된 피쳐 프

레이م $fr(n, n)$ 에 대하여 $n = 1, 2, 4, 8, 16$ 에 대하여 각각의 계층에 대하여 진행하였다. 각각의 P2, P3, P4, P5에 대하여 가능한 조합을 모두 실험한 결과, BPP 대비 mAP 그래프가 left-upper에서 나타는 조합은 {각 계층: 할당 개수}로 표현할 때 {P2: (4, 4), P3: (4, 4), P4: (16, 16), P5: (16, 16)}로 나타났다.

이는 그림 7에서 모든 계층에 대하여 동일하게 (4, 4)의 피쳐 맵으로 피쳐 프레임을 구성하는 경우가 (16, 16)의 피쳐 맵으로 피쳐 프레임을 구성하는 경우를 제외하였을 때 left-upper에 위치하며, 표 5에서 나타나는 바와 같이 P2와 P3의 경우 (4, 4)와 (16, 16)의 BPP 차이가 크기에 압축 이득이 머신 태스크 손실보다 더 크게 나타날 수 있지만, P4와 P5의 경우 (4, 4)와 (16, 16)의 BPP 차이가 비교적 작기 때문에 압축 이득보다 머신 태스크의 손실이 크게 작용하는 것으로 확인할 수 있다. 이에 각 계층별로 {P2: (4, 4), P3: (4, 4), P4: (16, 16), P5: (16, 16)}개의 피쳐 맵을 할당하여 구성한 피쳐 프레임에 대한 실험은 MPEG-VCM 실험 환경을 통해 이후 4절에서 다루도록 한다.

$$mAP(fr_{p2}(i,i), fr_{p3}(j,j), fr_{p4}(k,k), fr_{p5}(l,l)) \quad \{\text{for } i, j, k, l \in (1, 2, 4, 8, 16)\} \quad (1)$$

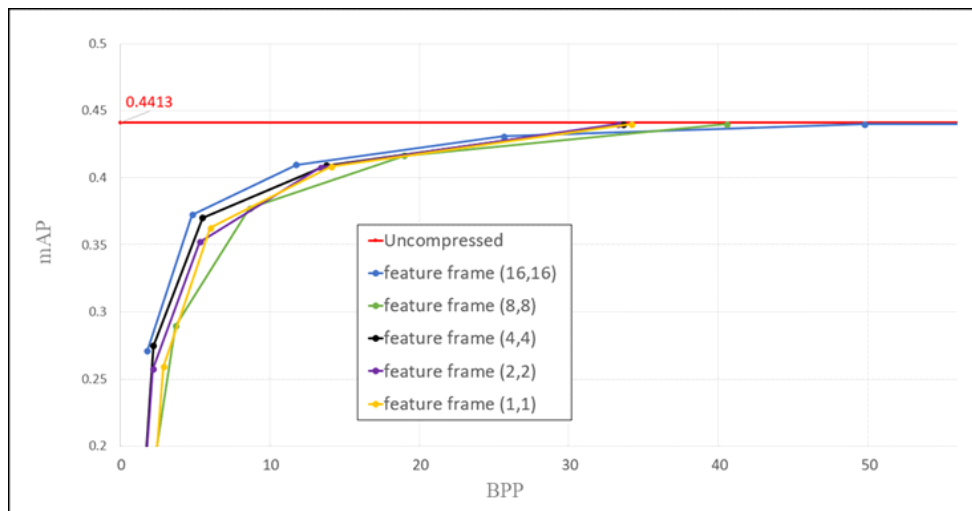


그림 7. 피쳐 맵 할당 개수에 따른 피쳐 프레임 압축 및 머신 태스크 성능 결과

Fig. 7. Feature frame compression and machine task performance results according to the number of feature map allocations

표 5. 계층별 피쳐 프레임 압축 결과

Table 5. Feature frame compression results according to hierarchy

Feature frame compression with VVenC (BPP)					
QP	feature frame (1,1)	feature frame (2,2)	feature frame (4,4)	feature frame (8,8)	feature frame (16,16)
P2 layer feature frame compression with VVenC (BPP)					
22	24.4136	24.3381	24.8180	30.1987	63.3426
27	9.4663	9.4049	9.8148	13.8579	37.2728
32	3.6641	3.5506	3.7936	6.1684	18.7456
37	1.5097	1.3780	1.4898	2.5860	8.3211
42	0.6606	0.5327	0.5598	1.0022	3.2917
P3 layer feature frame compression with VVenC (BPP)					
22	6.0601	5.8222	5.8700	6.9797	13.9332
27	2.6980	2.5266	2.5752	3.4053	8.4369
32	1.2518	1.0930	1.1159	1.6351	4.5599
37	0.6279	0.4809	0.4823	0.7476	2.2528
42	0.3429	0.2053	0.1920	0.3130	1.0050
P4 layer feature frame compression with VVenC (BPP)					
22	2.2171	1.9935	1.9464	2.2553	4.2409
27	1.0903	0.9162	0.8915	1.1442	2.7077
32	0.5802	0.4203	0.3974	0.5653	1.5377
37	0.3490	0.1972	0.1725	0.2595	0.7831
42	0.2383	0.0945	0.0683	0.1086	0.3550
P5 layer feature frame compression with VVenC (BPP)					
22	0.9316	0.7296	0.6754	0.7550	1.3581
27	0.5220	0.3617	0.3236	0.3969	0.8979
32	0.3188	0.1824	0.1487	0.2008	0.5296
37	0.2268	0.0993	0.0669	0.0944	0.2787
42	0.1883	0.0621	0.0303	0.0416	0.1277

3. 피쳐 프레임에 대한 재정렬된 피쳐 맵 할당 실험

재정렬된 피쳐 맵으로 다양한 프레임의 구성이 다양할 수 있기에, 본 3절에서는 재정렬된 피쳐 맵을 피쳐 프레임에 할당하는 방법에 대한 실험을 진행하였다. 해당 방법은

그림 8에서 나타나는 바와 같이 Spatial-priority와 Temporal-priority 할당 방법이 있다. Spatial-priority 할당 방식은 재정렬된 피쳐 맵을 정렬 순서에 따라 하나의 프레임을 먼저 채우고, 다음 프레임을 채우는 형식으로 진행되고, Temporal-priority 할당 방법은 각 피쳐 프레임마다 동일한 위치를 먼저 채우는 형식으로 진행된다.

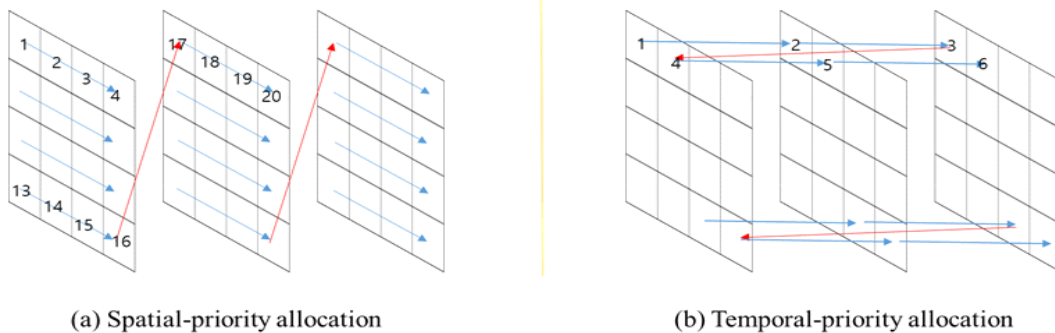


그림 8. 피쳐 프레임에 대한 피쳐 맵 할당 방법 (a) Spatial-priority (b) Temporal-priority
Fig. 8. Allocation methods of reordered feature maps into feature frame

표 6. 우선 할당 방법에 따른 압축 결과

Table 6. Compression results according to priority allocation

Feature frame (4,4) compression with VVenC (BPP)			
QP	Original (Unsorted Tiling)	Spatial-priority allocation	Temporal-priority allocation
22	33.6523	33.0733	33.0060
27	13.7684	13.6863	13.6270
32	5.5241	5.5172	5.4832
37	2.2439	2.2098	2.1985
42	0.8689	0.8383	0.8348
47	0.3521	0.3241	0.3220

이에, 실험에서는 재정렬된 피쳐 맵을 (4, 4)개의 피쳐 맵으로 구성된 피쳐 프레임 16장에 대하여 Spatial-priority와 Temporal-priority 각각을 적용하였으며 그 결과는 표 6과 같이 나타나며, 이때 Original은 정렬하지 않은 피쳐 맵을 타일링 패킹하여 압축하는 경우에 해당한다. 표 6에 나타나는 바와 같이, 정렬된 피쳐 맵에 대하여 (4, 4) 피쳐 프레임을 생성할 때 BPP 측면에서는 Temporal-priority 할당이 Spatial-priority 할당보다 좋은 것을 확인할 수 있다.

4. 피쳐 프레임 기반 피쳐 맵 압축 실험

본 4절에서는 앞서 설명한 방법들을 동시에 적용하였을 때의 피쳐 맵 압축 성능과 머신 태스크 성능을 확인하고자 한다. 이에 최종적으로 생성된 피쳐 맵은 Faster R-CNN X101-FPN의 P2, P3, P4, P5 각각의 계층에 대하여 서로 다른 개수의 피쳐 맵으로 {P2: (4, 4), P3: (4, 4), P4: (16, 16), P5: (16, 16)} 조합으로 피쳐 프레임을 구성하였으며, 이때 재정렬된 피쳐 맵은 Temporal-priority 할당 방식으로 피쳐 프레임에 할당하였다.

본 실험에서 사용된 실험 데이터는 MPEG-VCM에서 정의한 OpenImageV6 validation set의 테스트 영상 5,000장이고, 압축에 활용한 비디오 코덱은 VTM-12.0을 사용하였다. 이때 MPEG-VCM에서 정의한 방법에 따라서 피쳐 맵을 10 비트로 양자화하였으며, YUV 4:0:0로 변환하여 VTM-12.0을 통해 압축하였다. Image compression anchor와 Feature compression anchor는 AI 모드를 통해 부호화하였으며, 제안하는 피쳐 프레임은 P2와 P3에서 생성된 피쳐

프레임에 대해서는 RA 모드로 부호화를 진행하였고, P4와 P5에 대해서는 AI 모드로 부호화하였다. 또한, 제안하는 피쳐 프레임 방법에서는 재정렬된 피쳐 맵을 원래의 순서로 복원하기 위한 메시지 비트를 고려하기 위하여 입력된 정지 영상 당 8192 비트를 추가하였으며, 이는 앞서 설명한 바와 마찬가지로 P2, P3, P4, P5에 대한 계층 수 4, 한 계층마다 존재하는 피쳐 맵의 채널 수 256, 0-255 사이의 인덱스를 복원하기 위한 1 바이트, 바이트를 비트로 변환을 위한 8을 모두 곱한 값이다. 이때 Window 10, i7-11700K의 CPU로 구성된 실험 환경에서 파이썬(Python) 언어를 사용하여 아나콘다(Anaconda) 환경의 주피터 노트북(Jupyter notebook)을 통해 입력 영상 한 장에 대한 P2, P3, P4, P5 계층에 존재하는 모든 피쳐 맵 1,024개를 재정렬하는데 소요되는 시간은 평균 0.0623초로 나타났다.

제안하는 피쳐 프레임의 실험 결과와 MPEG-VCM의 피쳐 맵 압축 앵커의 비교 결과는 표 7과 그림 9와 같이 나타난다. 그림 9에서 나타나는 Image compression anchor는 일반적인 정지 영상을 VTM-12.0을 통해 압축하고 딥러닝 네트워크에 입력한 경우를 나타내며, Feature compression anchor는 모든 피쳐 맵을 하나의 정지 영상에 타일링 패킹하여 압축한 경우를 나타낸다. 그림 9에서 나타나는 바와 같이, 앞서 설명한 방법들을 동시에 적용하였을 때의 그래프 곡선이 Feature compression anchor의 그래프 곡선보다 left-upper에 형성되어 성능이 더 좋은 것을 확인할 수 있으며, 구체적으로 feature compression anchor에 비하여 14.29%의 BD-rate 이득이 있음을 표 7에서 확인할 수 있다. 따라서 전통적인 비디오 코덱만을 활용하여 딥러닝 네트워

표 7. 피쳐 프레임 기반 피쳐 맵 압축 방법의 BD-rate

Table 7. BD-rate of feature frame based feature map compression method

Feature frame compression with VTM-12.0 (mAP vs BPP)							
Feature compression anchor	QP	mAP	BPP	Proposed method	QP	mAP	BPP
Uncompressed	-	78.93					
Compressed	39	77.05	0.64	Compressed	35	76.38	0.60
	41	74.75	0.44		37	74.01	0.42
	43	69.65	0.29		39	70.63	0.29
	45	59.72	0.19		41	65.07	0.21
	47	44.40	0.13		43	59.86	0.14
	49	25.00	0.09		45	51.90	0.10
BD-rate	-14.29%						

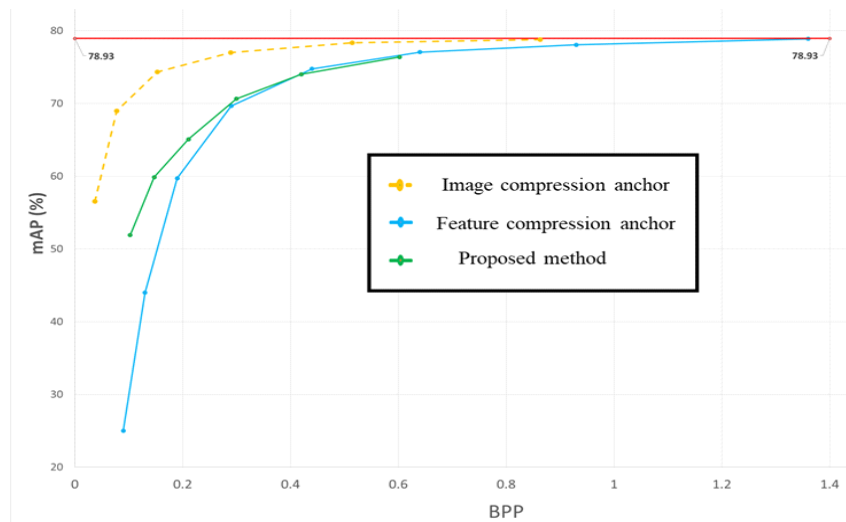


그림 9. 피쳐 프레임 기반 피쳐 맵 압축의 rate-precision 곡선

Fig. 9. Rate-precision curves for feature frame based feature map compression

크의 피쳐 맵 압축할 때 본 논문에서 정의하는 피쳐 프레임 기반 피쳐 맵 압축 방법을 통하여 보다 효율적인 피쳐 맵 압축이 가능한 것을 확인할 수 있다.

IV. 결 론

본 논문에서는 전통적인 비디오 코덱만을 사용하여 다계층으로 구성된 딥러닝 네트워크의 피쳐 맵을 보다 효율적으로 압축하는 방법을 제시하였다. 해당 방법은 피쳐 맵 간의 중복성을 높이기 위하여 피쳐 맵의 재정렬을 진행하였으며, 정지 영상 압축 포맷과 동영상 압축 포맷을 동시에

활용하기 위하여 다양한 개수의 피쳐 맵으로 피쳐 프레임을 구성하였다. 또한, 재정렬된 피쳐 맵을 피쳐 프레임에 할당함에 있어서 Spatial-priority 할당보다 압축 성능이 높게 나타난 Temporal-priority 할당 방식을 활용하였다.

객관적인 분석 결과 도출을 위하여 본 논문에서는 Open-ImageV6 validation set에 대하여 객체 검출 성능을 실험하였으며, 그 결과는 MPEG-VCM 피쳐 맵 압축 앵커 대비 BPP와 mAP의 BD-rate 관점에서 14.29% 만큼의 압축 이득을 보여 피쳐 프레임 기반의 피쳐 맵 압축 방법의 효율성을 검증하였다.

또한, 본 논문에서 설명하는 피쳐 프레임 기반 피쳐 맵 압축 방법은 전통적인 비디오 코덱만을 활용하기에 설계가

간단하고 기존에 설비된 비디오 압축 환경만으로 구현이 가능하다는 장점이 존재하기에 객체 검출뿐만 아니라 다양한 머신 태스크에서도 효율적인 피쳐 맵의 압축이 가능하면서 머신 태스크의 성능을 유지하는지 추후 연구에서 확인할 필요성이 있다.

참 고 문 헌 (References)

- [1] LECUN, Yann, et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989, 1.4: 541-551
doi: <https://doi.org/10.1162/neco.1989.1.4.541>
- [2] KANG, Yiping, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 2017, 45.1: 615-629.
doi: <https://doi.org/10.1145/3093337.3037698>
- [3] CHEN, Zhuo, et al. Toward intelligent sensing: Intermediate deep feature compression. *IEEE Transactions on Image Processing*, 2019, 29: 2230-2243.
doi: <https://doi.org/10.1109/TIP.2019.2941660>
- [4] M. Rafie, Y. Zhang, and S. Liu, "[VCM] Evaluation Framework for Video Coding for Machines," ISO/IEC JTC 1/SC 29/WG 2, m58385, Online, Oct. 2021.
- [5] Minhun Lee, Hansol Choi et al. "[VCM track 1] EE1.2 P-layer feature map anchor generation for object detection on OpenImageV6 dataset", ISO/IEC JTC 1/SC 29/WG 2, m58786, Online, Jan. 2022.
- [6] Minhun Lee, Hansol Choi et al. "[VCM track 1] Advanced feature map compression based on optimal transformation with VVC and DeepCABAC", ISO/IEC JTC 1/SC 29/WG 2, m58787, Online, Jan. 2022.
- [7] Jung Heum Kang, Hye Won Jeong et al. "[VCM track 1] Feature Compression with resize in feature domain", ISO/IEC JTC 1/SC 29/WG 2, m58867, Online, Jan. 2022.
- [8] C. Rosewarne, Y. Kim et al. "[VCM] EE1 summary report", ISO/IEC JTC 1/SC 29/WG 2, m58793, Online, Jan. 2022.
- [9] VTM12.0, https://vcgit.hhi.fraunhofer.de/jvet/VVCSsoftware_VTM/-/tree/VTM-12.0 (accessed Jan. 11, 2022).
- [10] CHOI, Hyomin; BAJIĆ, Ivan V. Deep feature compression for collaborative object detection. In: 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018. p. 3743-3747.
doi: <https://doi.org/10.1109/ICIP.2018.8451100>
- [11] HAN, Heeji, et al. Feature map channel reordering and compression for Neural Network feature map coding. In: *Proceedings of the Korean Society of Broadcast Engineers Conference*. The Korean Institute of Broadcast and Media Engineers, 2021. p. 39-42.
- [12] Heeji Han, Haechul Choi et al. "[VCM] Investigation on feature map channel reordering and compression for object detection", ISO/IEC JTC 1/SC 29/WG 2, m56653, Online, Apr. 2021.
- [13] Yong-Uk Yoon, Dongha Kim et al. "[VCM] Compression of reordered feature sequences based on channel means for object detection", ISO/IEC JTC 1/SC 29/WG 2, m57497, Online, Jul. 2021.
- [14] TUCHLER, Michael; SINGER, Andrew C.; KOETTER, Ralf. Minimum mean squared error equalization using a priori information. *IEEE Transactions on Signal processing*, 2002, 50.3: 673-683.
doi: <https://doi.org/10.1109/78.984761>
- [15] WANG, Zhou, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004, 13.4: 600-612.
doi: <https://doi.org/10.1109/TIP.2003.819861>
- [16] SUZUKI, Satoshi, et al. Deep Feature Compression With Spatio-Temporal Arranging for Collaborative Intelligence. In: 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020. p. 3099-3103.
doi: <https://doi.org/10.1109/ICIP40778.2020.9190933>
- [17] SENGUPTA, Abhronil, et al. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, 2019, 13: 95.
doi: <https://doi.org/10.3389/fnins.2019.00095>
- [18] PHAM, Vung; PHAM, Chau; DANG, Tommy. Road damage detection and classification with detectron2 and faster r-cnn. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020. p. 5592-5601.
doi: <https://doi.org/10.1109/BigData50022.2020.9378027>
- [19] COCO2017 validation set, <https://cocodataset.org/#download> (accessed Apr. 11, 2022)
- [20] WIECKOWSKI, Adam, et al. VVenC: An open and optimized VVC encoder implementation. In: 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2021. p. 1-2.
doi: <https://doi.org/10.1109/ICMEW53276.2021.9455944>
- [21] OpenImageV6, <https://storage.googleapis.com/openimages/web/download.html> (accessed Apr. 11, 2022).
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
doi: <https://doi.org/10.1109/cvpr.2016.90>
- [23] G. Bjontegaard, "Calculation of average PSNR differences between RDcurves," Tech. Rep. VCEGM33, Video Coding Experts Group (VCEG), 2001.

저 자 소 개



이 성 배

- 2019년 2월 : 경희대학교 전자공학과 공학사
- 2021년 2월 : 경희대학교 전자정보융합공학과 공학석사
- 2021년 3월 ~ 현재 : 경희대학교 전자정보융합공학과 박사과정
- ORCID : <https://orcid.org/0000-0002-7392-6870>
- 주관심분야 : 딥러닝, 영상처리, 멀티미디어 시스템



이 민 석

- 2021년 8월 : 경희대학교 전자공학과 공학사, 컴퓨터공학과 공학사
- 2021년 9월 ~ 현재 : 경희대학교 전자정보융합공학과 석사과정
- ORCID : <http://orcid.org/0000-0001-6500-7156>
- 주관심분야 : 딥러닝모델 경량화, 영상처리, 멀티미디어 시스템



김 규 현

- 1989년 2월 : 한양대학교 전자공학과 공학사
- 1992년 9월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학석사
- 1996년 7월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학박사
- 1996년 ~ 1997년 : 영국 University of Sheffield, Research Fellow
- 1997년 ~ 2006년 : 한국전자통신연구원 대화형미디어연구팀장
- 2006년 ~ 현재 : 경희대학교 전자정보대학 교수
- ORCID : <http://orcid.org/0000-0003-1553-936X>
- 주관심분야 : 디지털 방송, 영상처리, 멀티미디어 통신, 디지털 대화형 방송