

## 피쳐 맵 리사이징과 보간법에 따른 압축 및 머신타스크 성능 분석

\*이성배 \*\*이민석 \*\*\*김규헌

경희대학교

\*rhee@khu.ac.kr \*\*qsibmini@khu.ac.kr \*\*\*kyuhyeonkim@khu.ac.kr

## Analysis of compression and machine task performance according to feature map resizing and interpolation

\*Rhee, Seong-bae \*\*Lee, Min-Seok \*\*\*Kim, Kyu-Heon

Kyunghee University

## 요약

최근 딥러닝 네트워크의 피쳐 맵을 활용하여 머신 태스크를 수행하는 Collaborative Intelligence에 대한 관심이 증가하고 있다. CI 구조는 피쳐 맵을 전송함에 따라서 저사양 디바이스에서 딥러닝 기반의 머신 태스크 수행을 가능하게 하여 다양한 산업에서 활용될 것으로 기대되고 있다. 그러나 CI 구조에서 전송되는 피쳐 맵은 데이터 크기가 방대하기 때문에 전송에 있어 효율적인 피쳐 맵 압축이 필요하다. 이에 본 논문에서는 MPEG-VCM에서 제안된 리사이징(resizing)과 보간법(interpolation)을 활용하여 피쳐 맵을 압축하는 Feature Coding 기술에 대하여, 다양한 리사이징 및 보간 방법을 조합하여 가장 우수한 압축 성능 대비 머신 태스크 성능을 나타내는 조합을 실험을 통해서 확인하고자 한다.

## 1. 서론

최근 딥러닝 네트워크의 피쳐 맵(Feature Map)을 활용하여 머신 태스크(Machine Task)를 수행하는 CI(Collaborative Intelligence)에 대한 관심이 증가하고 있다[1]. CI 과정에서는 고성능 디바이스에서 딥러닝 네트워크의 콘볼루션 연산을 수행하고, 그 결과 값인 피쳐 맵을 비교적 저사양 디바이스에 전송한다. 이에 피쳐 맵을 전송받은 디바이스에서는 해당 피쳐 맵이 생성되기까지의 콘볼루션 연산이 필요하지 않기 때문에 비교적 저사양의 디바이스임에도 딥러닝 네트워크의 머신 태스크 수행이 가능하다는 장점으로 다양한 산업에서 활용될 것으로 기대되고 있다.

그러나 CI 구조를 위해 전송되는 피쳐 맵의 데이터 크기는 입력 영상의 데이터 크기에 비하여 거대하기 때문에 전송비용이 비싸다는 제한 사항이 있다. 이에, 국제표준화기구 MPEG(Moving Picture Expert Group)의 기계를 위한 부호화(VCM, Video Coding for Machine) 그룹에서는 머신 태스크의 성능을 일정 수준 확보하면서 피쳐 맵을 효율적으로 압축하는 기술 표준 개발을 목표로 진행하고 있다[2].

MPEG-VCM에서 제안된 Feature Coding 기술 중에서는 피쳐 맵의 크기를 리사이징(resizing)하여 압축하고, 머신 태스크에 활용할 때 보간법(interpolation)을 통해 본래 크기의 피쳐 맵으로 복원함으로써, 피쳐 코딩 앵케[2] 대비 높은 압축률 대비 머신 태스크 성능을 보여주었다[3, 4]. 그러나 제안된 기술에서는 피쳐 맵의 크기를 줄여 압축 이득을 향상시킨 것이 중점인 실험이었기에, 어떤 리사이징 방법과 보간 방법이 피쳐 맵에 적합한 방법인지를 실험적으로 확인하지 않았다. 이에, 본 논문에서는 리사이징 기반의 피쳐 맵 압축 기술을 지원하기 위하여 다양한 리사이징 및 보간 방법 각각의 조합에 대하여 피쳐 맵의 압축률 대비 머신 태스크 성능을 확인함으로써, 피쳐 맵 압축에 적합한 리사이징 및 보간 방법을 확인하고자 한다.

## 2. 배경 기술

본 논문에서는 다양한 리사이징과 보간법을 통해서 피쳐 맵의 압축률 대비 머신 태스크 성능을 각각 실험함으로써, 어떤 조합이 최적의 압축률 대비 머신 태스크 성능을 보이는지 실험하고자 한다. 실험을 위해 활용될 리사이징과 보간 방법으로는 bilinear, nearest, bicubic, area 방법이다.

흔히 영상처리 분야에서 품질이 좋게 여겨지는 보간 방법은 bicubic 이다. 일례로, 그림 1에서 나타나는 original image에 대하여 bilinear, nearest, bicubic, area 각각의 보간 방법으로 8배를 확대한 결과는 그림 1의 (a), (b), (c), (d)로 나타난다. 이때, nearest와 area 보간 방법에서는 윤곽선 부분이 매끄럽지 않은 것을 확인할 수 있고, bilinear에서는 결과 영상이 다소 흐리게 나타나는 것을 확인할 수 있다. 이에, [5]에서는 bilinear, nearest, bicubic, area 보간 방법 중에서 bicubic 보간 방법이 사람의 인지적 평가에서 가장 좋은 보간 방법으로 평가되었다.

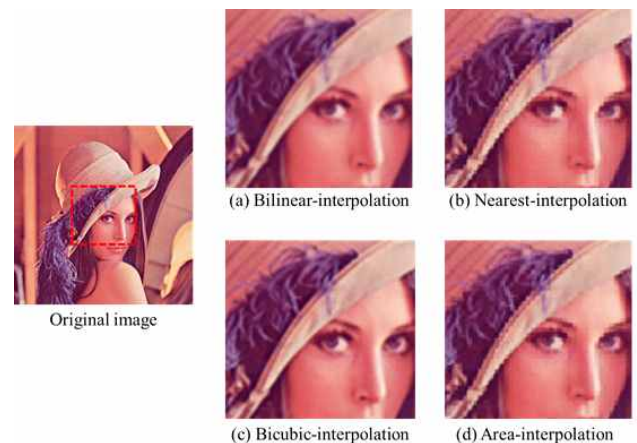


그림 1. 보간 방법에 따른 결과 예시

그러나 bicubic 보간법이 모든 영상에 대하여 뛰어난 복원력을 보이는 것은 아니다. 일례로, 그림 2의 original image처럼 나타나는 격자 형태의 영상에 대하여 각각의 보간법으로 8배를 확대하는 경우 bicubic의 결과 영상에서 윤곽선 부분이 흐리게 나타나는 반면, nearest와 area 보간 방법에서는 윤곽선이 비교적 선명하게 나타나는 것을 확인할 수 있다. 이처럼 영상의 특성에 따라서 적합한 보간 방법은 다를 수 있다.

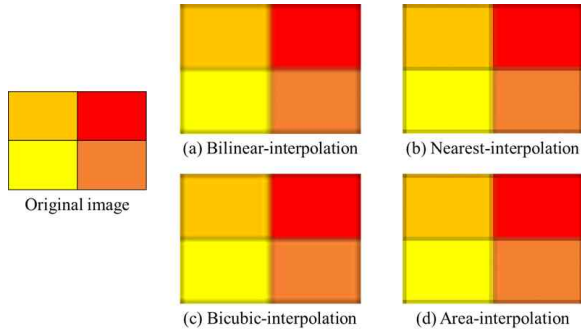


그림 2. 격자 형태의 영상에 대한 보간 결과 예시

이에 본 논문에서는 압축되지 않은 피쳐 맵에서 적합한 리사이징 및 보간 방법과 VVC (VTM-12.0)[8]를 통해 부호화 및 복호화 과정을 거친 피쳐 맵에 대하여 적합한 리사이징 및 보간 방법을 확인하는 실험 방법을 이어지는 3장에서 설명하고자 한다.

### 3. 피쳐 맵 리사이징 및 보간 실험 방법

본 논문에서 피쳐 맵의 리사이징 및 보간 실험은 피쳐 맵 리사이징 및 패킹, 피쳐 맵 압축, 피쳐 맵 복원 및 머신태스크 수행 과정으로 진행된다. 이때 활용된 피쳐 맵은 Faster R-CNN X101-FPN[6]의 P-layer에서 추출된 피쳐 맵을 활용하였다.

#### 3.1 피쳐 맵 리사이징 및 패킹

본 논문에서 진행하는 피쳐 맵 리사이징은 [4]에서 진행된 방법과 유사하게 Faster R-CNN X101-FPN의 P-layer P2, P3에 대하여 1/4로 리사이징하였으며, P4와 P5에 대해서는 리사이징을 적용하지 않았다. 이때 리사이징 방법은 각각의 리사이징 방법에 따라 압축률 대비 머신 태스크 성능을 확인하기 위하여 bilinear, nearest, bicubic, area 각각으로 진행하였다.

리사이징된 P2, P3의 피쳐 맵과 리사이징을 적용하지 않은 P4, P5 피쳐 맵을 하나의 피쳐 이미지로 패킹하기 위하여, 그림 3에서 나타나는 바와 같은 순서로 P2, P4, P3, P5의 피쳐 맵을 하나의 피쳐 이미지로 타일링(Tiling) 패킹하였다. 이때 P2와 P4의 피쳐 맵 타일링에서는 피쳐 이미지의 가로 영역과 세로 영역 각각에 16개씩 할당되었으며, P3와 P5의 피쳐 맵 타일링에는 가로에 32개, 세로에 8개의 피쳐 맵을 각각 할당하였다.

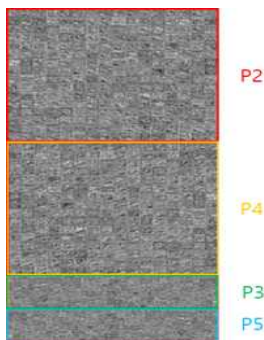


그림 3. 피쳐 맵 타일링 예시

### 3.2 피쳐 맵 압축

피쳐 맵 압축을 위하여 VTM-12.0을 활용하였으며, 앞선 절에서 생성한 피쳐 이미지를  $QP \in \{35, 37, 39, 41, 43, 45\}$ 에 대하여 AI (All-Intra) 모드로 부호화하였다. 이때 피쳐 맵의 양자화는 10bit로 진행되었으며, VTM-12.0에서 압축되기 위하여 YUV 4:0:0로 변환하였다.

### 3.3 피쳐 맵 복원 및 머신 태스크 수행

VTM-12.0을 통해서  $QP \in \{35, 37, 39, 41, 43, 45\}$ 에 대하여 AI 모드로 부호화된 비트스트림을 피쳐 맵으로 복호화하고, P2와 P3의 복원은 리사이징한 방법대로 복원하는 것이 아니라, bilinear, nearest, bicubic, area 각각의 보간 방법으로 복원하여 피쳐 맵 압축률 대비 머신 태스크 성능을 측정하였다. 이때 피쳐 맵의 보간법은 피쳐 맵 간의 간섭을 방지하기 위하여 단일 피쳐 맵 단위로 보간법을 적용하여 복원하였다.

## 4. 실험 결과

앞서 정의한 실험 방법을 요약하자면 P2, P3에 대하여 bilinear, nearest, bicubic, area 각각의 방법을 활용하여 1/4 크기로 리사이징하여 압축하고, 복원에서 bilinear, nearest, bicubic, area 각각의 보간 방법으로 4배를 확장하여 피쳐 맵의 압축률 대비 머신 태스크 성능을 측정한다. 이때 실험은 Windows 10 운영체제에서 AMD Ryzen Threadripper PRO 3975WX의 CPU로 구성된 환경에서 파이썬 언어를 통해 구현되었으며, 실험에 활용한 데이터는 MPEG-VCN에서 실험 데이터로 정의한 OpenImageV6 validation set[7] 내 5,000장의 테스트 영상을 활용하였다.

표 1은 uncompressed 피쳐 맵에 대한 mAP 결과로써, Faster R-CNN X101-FPN의 P-layer P2와 P3에 대하여 bilinear, nearest, bicubic, area 각각으로 리사이징을 진행하고, 다시 bilinear, nearest, bicubic, area 각각의 보간법으로 P2와 P3를 복원한 뒤 머신 태스크 성능인 mAP (mean Average Precision)를 측정하였다. 이때 리사이징이 area, 보간 방법이 bicubic인 경우에서 가장 높은 mAP 값이 나타났으며, 동일한 리사이징 및 보간 방법으로 진행하였을 때 가장 높은 mAP는 area일 때 나타났다. 반면, 가장 낮은 mAP를 나타낸 리사이징 방법은 nearest일 때 나타났다. 표 1에서 나타나는 피쳐 맵은 양자화나 압축이 진행되지 않은 텐서 (Tensor) 데이터이므로 BPP는 표기하지 않았다.

표 1. Uncompressed 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 측정 결과 (%)

Resizing method	Interpolation method			
	bilinear	nearest	bicubic	area
bilinear	78.46	78.26	78.01	78.26
nearest	76.85	76.62	76.85	76.62
bicubic	78.14	77.67	77.54	77.67
area	78.32	78.61	78.65	78.61

표 2, 3는 QP (Quantization Parameter)를 각각 35와 37로 설정하여 압축한 피쳐 맵에 대하여 BPP 대비 mAP를 측정한 결과를 나타내며, 이때 BPP는 압축된 피쳐 맵의 비트스트림 용량을 입력 영상의 가로와 세로 픽셀 크기로 나눈 값을 비트 (Bit) 단위로 변환한 값이다.

표 2, 3에서 나타나는 바와 같이, 두 실험 결과는 앞선 uncompressed의 실험 결과와 다르게 리사이징과 보간 방법이 bicubic일 때 mAP 값이 가장 높게 나타났다. 또한, 어떤 방법으로 리사이징했더라도 보간 방법이 bicubic일 때 가장 높은 mAP 값이 나타났으며, 보간 방법이 bilinear일 때 mAP가 가장 낮게 나타났다. 한편, 가장 작은 BPP는 area로 리사이징했을 때 나타났으며, 이때 평균적인 mAP도 가장 작게 나타났다.

표 2. QP = 35에서 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 및 BPP 결과

QP = 35		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	70.27	70.60	70.96	70.60
	BPP	0.3174			
nearest	mAP	70.41	70.90	71.15	70.90
	BPP	0.3228			
bicubic	mAP	71.16	71.47	71.99	71.47
	BPP	0.3312			
area	mAP	69.22	69.43	69.71	69.43
	BPP	0.2977			

표 3. QP =37에서 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 및 BPP 결과

QP = 37		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	66.40	66.61	66.72	66.61
	BPP	0.2164			
nearest	mAP	66.76	66.91	67.00	66.91
	BPP	0.2188			
bicubic	mAP	67.14	67.41	67.56	67.41
	BPP	0.2233			
area	mAP	66.12	66.20	66.28	66.20
	BPP	0.2061			

표 4는 QP를 39로 설정하여 압축한 피쳐 맵의 BPP 대비 mAP 결과로써, 앞선 표 2와 3의 결과와 동일하게 어떤 방법으로 리사이징했다더라도 bicubic 보간법을 적용하였을 때 가장 높은 mAP 측정값이 나타났으며, 가장 높은 mAP를 보인 조합은 리사이징에서 bilinear를 사용하고, 보간 방법으로 bicubic을 사용한 경우로 나타났다. 또한, 앞선 결과와 마찬가지로 가장 작은 BPP는 area로 리사이징했을 때 나타났지만, 앞선 결과와 달리 area로 리사이징 했을 때의 평균 mAP 값은 최소가 아닌 것으로 나타났다.

표 4. QP = 39에서 피쳐 맵에 대한 리사이징 및 보간 실험 결과 mAP 및 BPP 결과

QP = 39		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	63.80	63.82	63.92	63.82
	BPP	0.1450			
nearest	mAP	63.58	63.67	63.73	63.67
	BPP	0.1458			
bicubic	mAP	63.66	63.65	63.70	63.65
	BPP	0.1484			
area	mAP	63.85	63.87	63.88	63.87
	BPP	0.1397			

한편, 표 5, 6, 7은 QP를 각각 41, 43, 45로 설정하여 압축한 피쳐 맵을 통해 BPP 대비 mAP를 측정한 결과로써, QP가 커짐에 따라 bicubic 보간법보다 area 보간법을 활용하였을 때 더 높은 mAP 값이 나타내는 것을 확인할 수 있었으며, 앞선 결과와 마찬가지로 area 방법을 통해 리사이징했을 때 가장 작은 BPP를 보였다.

표 5. QP = 41에서 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 및 BPP 결과

QP = 41		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	61.06	61.12	61.06	61.12
	BPP	0.0949			
nearest	mAP	61.28	61.29	61.31	61.29
	BPP	0.0949			
bicubic	mAP	61.10	61.14	61.12	61.14
	BPP	0.0965			
area	mAP	61.32	61.30	61.32	61.30
	BPP	0.0923			

표 6. QP = 43에서 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 및 BPP 결과

QP = 43		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	57.71	57.66	57.72	57.66
	BPP	0.0602			
nearest	mAP	58.04	58.07	58.04	58.07
	BPP	0.0599			
bicubic	mAP	57.83	57.84	57.83	57.84
	BPP	0.0609			
area	mAP	57.82	57.86	57.82	57.86
	BPP	0.0589			

표 7. QP = 45에서 피쳐 맵에 대한 리사이징 및 보간 방법에 따른 mAP 및 BPP 결과

QP = 45		Interpolation method			
Resizing method		bilinear	nearest	bicubic	area
bilinear	mAP	51.67	51.67	51.68	51.67
	BPP	0.0372			
nearest	mAP	52.00	52.00	51.99	52.00
	BPP	0.0369			
bicubic	mAP	52.02	52.01	51.99	52.01
	BPP	0.0375			
area	mAP	52.02	52.03	52.02	52.03
	BPP	0.0365			

## 5. 결론

본 논문에서는 MPEG-VCM에서 리사이징을 활용한 피쳐 맵 압축 방법에서 보다 높은 피쳐 맵 압축 효율과 머신 태스크 성능을 얻기 위하여 리사이징과 보간 방법을 각각 bilinear, nearest, bicubic, area으로 실험을 진행하였고, 각각의 조합에 대한 BPP 대비 mAP 측정 결과를 확인하였다.

이때 uncompressed 피쳐 맵에 대하여 다양한 리사이징과 보간 방법에 대한 실험에서는 area 리사이징과 bicubic 보간법을 활용하였을 때 가장 높은 mAP가 나타났으며, QP를 35, 37로 설정하여 피쳐 맵을 압축한 경우에는 리사이징과 보간 방법이 모두 bicubic일 때 가장 높은 mAP 값을 나타냈다.

그러나 QP가 점점 커짐에 따라서 bicubic 보간 방법보다는 area 보간 방법이 더 높은 mAP를 보였으며, QP가 45일 때는 BPP 측면과 mAP 측면에서 리사이징과 보간법이 모두 area일 때 최적인 것을 확인할 수 있었다.

본 논문에서 실험한 리사이징 기반 피쳐 맵 압축 방법은 제한된 6개의 QP 지점에 대하여 실험을 진행하였기에 QP 크기별로 최적의 리사이징과 보간 방법이 어떻게 변하는지를 정확하게 확인하기 제한되었지만, 더 많은 QP 값에 대하여 추가적인 실험을 진행함으로써 리사이징 기반 피쳐 맵 압축에서 적합한 리사이징 및 보간 방법을 확인하는 추가적인 연구가 필요하다.

## ACKNOWLEDGMENT

\*이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00011, (전통 연구실)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).

## 참 고 문 헌 (References)

- [1] CHEN, Zhuo, et al. Toward intelligent sensing: Intermediate deep feature compression. *IEEE Transactions on Image Processing*, 2019, 29: 2230-2243.
- [2] 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.
- [3] 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.
- [4] Seongbae Rhee, Kyuheon Kim, "[VCM track 1] Report on feature map compression in terms of resizing and feature frame approaches", ISO/IEC JTC 1/SC 29/WG 2, m59589, Online, Apr. 2022.
- [5] HAN, Dianyuan. Comparison of commonly used image interpolation methods. In: *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. 2013.
- [6] 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.
- [7] OpenImageV6, <https://storage.googleapis.com/openimages/web/download.html> (accessed May. 20, 2022).
- [8] VTM12.0, [https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/-/tree/VTM-12.0](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-12.0) (accessed May. 20, 2022).