

특집논문 (Special Paper)

방송공학회논문지 제26권 제6호, 2021년 11월 (JBE Vol.26, No.6, November 2021)

<https://doi.org/10.5909/JBE.2021.26.6.738>

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

머신 비전을 위한 열 적외선 영상의 객체 기반 압축 기법

이 예 지^{a)}, 김 신^{a)}, 임 한 신^{b)}, 추 현 곤^{b)}, 정 원 식^{b)}, 서 정 일^{b)}, 윤 경 로^{a)†}

Object-based Compression of Thermal Infrared Images for Machine Vision

Yegi Lee^{a)}, Shin Kim^{a)}, Hanshin Lim^{b)}, Hyon-Gon Choo^{b)}, Won-Sik Cheong^{b)}, Jeongil Seo^{b)}, and Kyoungro Yoon^{a)†}

요 약

오늘날 딥러닝 기술의 향상으로 영상 분류, 객체 탐지, 객체 분할, 객체 추적 등 컴퓨터 비전 분야 또한 큰 발전을 이루고 있다. 지능적 감시, 로봇, 사물 인터넷, 자율주행 자동차 등 딥러닝 기술이 결합된 다양한 응용 기술들은 실제 산업에 적용되고 있으며, 이에 따라 사람의 소비를 위한 영상 데이터 뿐만 아니라 머신 비전을 위한 영상 데이터의 효율적인 압축 방식에 대한 필요성이 대두되고 있다. 본 논문에서는 머신 비전을 위한 열 적외선 영상의 객체 기반 압축 기법을 제안한다. 효율적인 영상 압축과 신경망의 좋은 성능을 유지하기 위해 본 논문에서는 신경망의 객체 탐지 결과와 객체 크기에 따라 입력 영상을 객체 부분과 배경 부분으로 나누어 서로 다른 압축률로 부호화를 수행하는 방법을 제안한다. 제안하는 방법은 VVC로 영상 전체를 압축하는 방식보다 BD-rate 값이 최대 -19.83%로 압축 효율이 뛰어나다는 것을 확인할 수 있다.

Abstract

Today, with the improvement of deep learning technology, computer vision areas such as image classification, object detection, object segmentation, and object tracking have shown remarkable improvements. Various applications such as intelligent surveillance, robots, Internet of Things, and autonomous vehicles in combination with deep learning technology are being applied to actual industries. Accordingly, the requirement of an efficient compression method for video data is necessary for machine consumption as well as for human consumption. In this paper, we propose an object-based compression of thermal infrared images for machine vision. The input image is divided into object and background parts based on the object detection results to achieve efficient image compression and high neural network performance. The separated images are encoded in different compression ratios. The experimental result shows that the proposed method has superior compression efficiency with a maximum BD-rate value of -19.83% to the whole image compression done with VVC.

Keyword : VCM(Video Coding for Machines), Object Detection, Thermal Infrared Image, Image Compression, Machine Vision

I. 서 론

오늘날 딥러닝 기법은 자연어 처리, 컴퓨터 비전, 음성 인식 등 다양한 분야에서 매우 우수한 성능을 보여주고 있으며, 그 중에서도 특히 컴퓨터 비전 분야에서 큰 두각을 나타내고 있다. 이러한 기술 발전으로 인하여 이미지 분류, 객체 탐지, 객체 분할, 객체 추적 등과 같은 대부분의 컴퓨터 비전 분야는 CNN(Convolutional Neural Network)을 기반으로 높은 성능을 보여 주고 있으며, 지능형 감시, 로봇, 사물 인터넷, 자율주행 자동차 등 딥러닝 기법이 결합된 컴퓨터 비전 응용 기술들이 실제 산업에서 사용되고 있다. 따라서 머신 비전을 위한 영상 소비는 급속하게 증가하고 있다. 이에 따라 기존 HEVC(H.265, High Efficiency Video Coding) 및 VVC(H.266, Versatile Video Coding)와 같은 인지 기반 영상 압축 방식이 아닌 머신 중심의 영상 처리, 압축 및 전송 기술에 대한 연구 필요성이 대두되고 있다.

국제 표준화 기구인 MPEG(Moving Picture Experts Group)은 머신 비전을 위한 영상 압축 및 전송 기술에 대한 요구사항을 분석하고 표준화를 진행하기 위해 VCM(Video Coding for Machines) 그룹을 구성하였다. VCM은 효율적인 영상 압축 및 특징 추출에 중점을 두고 있으며, 머신 비전 작업을 효율적이고 정확하게 수행하는 것을 목표로 한다.

현재까지 제안된 VCM 기술들은 크게 특징 맵 압축 방식과 영상 압축 방식으로 나뉜다. 특징 맵 압축 방식은 신경망 내 특징 맵을 추출하고, 추출된 특징 맵의 압축을 수행하는 것을 의미한다. 특징 맵 압축 방식은 특징 맵 내 채널들을 하나의 이미지로 패킹한 후 VVC나 HEVC와 같은 고성능

인코더를 이용해 인트라 모드로 압축하는 방식^[1,2,3,9]과 채널들을 마치 비디오 시퀀스와 같이 만들어 압축하는 방식^[4,6]이 있다. 일반적으로 특정 맵 압축 방법은 데이터 크기가 원래의 영상보다 훨씬 크기 때문에 영상을 압축하는 것보다 압축 효율이 떨어진다. 이러한 단점을 개선하기 위해 유사한 채널들을 끌어 재정렬하는 방식^[4,6]이나 푸르닝(Pruning)을 통해 유사한 채널이나 정보량이 적은 채널들을 제거하는 방식^[5,7], PCA나 양자화를 통해 채널의 데이터 크기를 줄이는 방식^[8,10] 등이 제안되고 있다.

영상 압축 방식은 신경망 기반 압축^[11,12], VVC 최적화^[13,14,15], 객체 기반 압축 방법^[16] 등이 있다. 신경망 기반 영상 압축 방법의 경우 VAE(Variational AutoEncoder) architecture에 기반한 Hyperprior Model^[19] 및 GMM and Attention 모델^[20]이 제안되었으며^[11,12], 이러한 방식의 경우 기존의 VVC와 비슷한 압축 성능을 보이면서, 머신 비전의 성능을 향상시킬 수 있음을 보여준다. VVC 최적화 방식은 VVC 내 다양한 코딩 도구를 조정하거나 Rate-Distortion Optimization 등을 통해 머신 비전 작업을 위한 최적화를 수행하는 방법이다^[13,14,15]. 마지막으로 객체 기반 압축 방식은 영상 내 정보의 중요성에 따라 압축률을 다르게 하는 방식이다^[16].

본 논문에서는 머신 비전을 위한 영상 압축 방식에 대한 하나의 후보 기술로 열 적외선 영상을 대상으로 하는 객체 기반 압축 방법을 제안한다. 머신이 효율적으로 영상 분석을 분석하고 활용하기 위해서는 머신 비전의 관점에서 영상의 중요한 정보를 탐색해야 한다. 따라서 머신 비전을 위해 중요한 정보에 따라 압축률을 다르게 할당할 필요성이 있다. 각기 다른 압축률로 압축하기 위해 객체 탐지 신경망 수행 후 신경망의 예측 결과인 물체 위치 정보와 객체 크기에 따라 영상을 객체 부분과 배경 부분으로 구별한다. 나뉘진 영상들은 서로 다른 압축률로 각각 부호화 된 후 하나의 스트림으로 합쳐 진다. 이러한 방법은 이미지 내 정보의 중요성 초점을 두고 있다. 영상 내 중요한 정보이면서, 머신 비전 관점에서 관심이 있는 객체 부분은 낮은 압축률로 부호화하고, 상대적으로 덜 중요한 배경 부분은 큰 압축률로 부호화하여 압축 효율을 높이면서 뛰어난 탐지 성능을 보여준다. 제안하는 객체 기반 압축 기법은 데이터의 크기가 큰 특징 맵 압축 방식보다 압축 성능이 우수하며, 별도로

a) 건국대학교 컴퓨터공학부(Dept. of Computer Science and Engineering, Konkuk University)

a) 한국전자통신연구원(ETRI)

‡ Corresponding Author : 윤경로(Kyoungro Yoon)

E-mail: yoonk@konkuk.ac.kr

Tel: +82-2-910-5394

ORCID: <https://orcid.org/0000-0002-1153-4038>

※ 이 논문의 연구 결과 중 일부는 한국방송·미디어공학회 2021년 학계학술대회에서 발표한 바 있음.

※ 본 연구 논문은 과학기술정보통신부 및 정보통신기획평가원의 출연금으로 수행되고 있는 “기계를 위한 영상 부호화 기술 (No, 2020-0-00011)” 과제의 연구결과입니다.

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 received September 6, 2021; Revised November 17, 2021; Accepted November 17, 2021.

신경망 학습을 진행하지 않아도 되기 때문에 다른 신경망 기반 압축 방식들 보다 적용하기 편리하다는 장점이 있다.

본 논문의 구성은 다음과 같다. 제 2장 1절에서는 객체 기반 압축 기법에 대해 서술하며, 제 2장 2절에서는 객체 영상과 배경 영상 분리를 위한 이진 맵 생성 알고리즘에 대해서 기술한다. 제 3장에서는 1절에서는 데이터 세트 및 실험환경에 대하여 서술하고, 제 3장 2절에서는 객체 기반 압축 기법 실험 결과를 제시하고, 일반적인 VVC 기반 영상 압축 방식과 제안하는 방법과의 결과 비교 및 분석을 수행하며 마지막으로 제 4장에서 결론을 짓는다.

II. 본 론

1. 객체 기반 압축 기법

사람이 보기 위한 영상과 다르게 머신 비전을 위한 영상 압축 방식은 전반적인 양질의 영상을 보장하지 않더라도 객체 인식, 객체 추적 등 머신 비전 작업에 대한 성능은 우수하게 유지할 수 있게끔 해야한다. 따라서 본 논문에서는 영상 내 정보의 중요도에 따라 압축률을 다르게 하는 방법을 제안한다. 대부분의 컴퓨터 비전 기반 신경망은 합성곱 연산을 통해 특징을 추출한다. 신경망의 특징들은 입력 영상의 모든 픽셀과 연결되는 것이 아니라 수용 영역 (Receptive Field) 안에 있는 픽셀들을 기반으로 특징을 추출하게 된다. 신경망 내 특징은 특정 위치와 그 주변에 있는 일부 픽셀들하고만 연관성이 높고, 거리가 멀수록 그 영향이 감소하게 된다. 따라서 본 논문에서는 이러한 신경망의

특성을 반영하여, 압축하기 전에 객체 텁지 신경망을 수행하여, 객체 부분과 배경 부분으로 영상을 분리한다. 영상 내 중요한 정보인 객체 부분과 그 주변 픽셀들은 낮은 QP (Quantization Parameter)로 압축을 수행하고, 상대적으로 신경망의 성능에 덜 영향을 미치는 배경 부분은 높은 QP로 압축을 수행한다. 이는 기존 VVC와 같은 압축 방식보다 압축 효율을 높이면서, 신경망의 성능은 뛰어나게 유지할 수 있기 때문에 머신 비전을 위한 영상 압축 방식에 적합하다.

그림 1은 객체 기반 압축 기법의 실험 과정을 보여준다. 객체 기반 압축 수행 과정은 다음과 같다. 부호화기 부분에서 객체 부분과 배경 부분을 나누기 위해 Detectron2의 Faster R-CNN X101-FPN^[22]을 사용하여 영상에서 중요한 부분인 객체 부분을 추출한다. 부호화 전 어떠한 객체 텁지 신경망을 사용하는 것에 따라서 부호화 하려는 객체가 달라질 수 있으나 본 논문에서는 VCM에서 anchor로 채택된 Faster R-CNN X101-FPN^[22]을 사용하여 객체 예측 결과를 얻는다. 이러한 결과로부터 그림 2의 알고리즘을 사용해 이진 맵을 생성하며, 비트 연산을 통해 객체 영상과 배경 영상으로 나눈다. 또한, VVC로 압축을 수행하기 전, 압축 효율을 높이기 위해 객체 영상과 배경 영상은 원본 영상 해상도 보다 낮은 해상도로 재조정(Rescale, 여기서는 Down-sampling) 될 수 있다. 재조정 과정을 거치는 경우 배경 영상 해상도는 객체 영상 해상도 보다 항상 작거나 같게 된다. 다음으로 두 영상은 VVC 부호화기를 사용하여 각기 서로 다른 QP로 압축된다. 부호화 시 객체 영상의 QP는 배경 영상의 QP보다 항상 작다. 마지막으로 부호화 수행 후 만들어진 두 개의 비트스트림(Bitstream)은 하나로 합하여 전송된다. 두 개의 비트스트림을 하나로 합칠 때, 객체 영상과

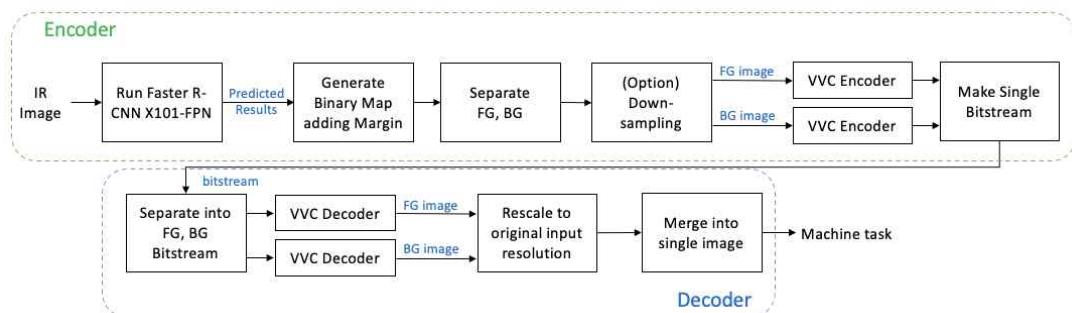


그림 1. 객체 기반 부호화 및 복호화 과정
Fig. 1. Pipeline of Object based Compression Method

배경 영상의 해상도와 합쳐진 비트스트림에서 배경 영상의 비트스트림 내 시작 위치 정보를 복호화기에게 전달하기 위해 3byte의 비트가 추가로 할당된다.

이렇게 압축된 비트스트림은 실험을 위하여 다시 복호화되게 되는데, 복호화기 부분에서는 하나의 비트스트림을 다시 두 개의 비트스트림으로 분할하고, VVC 복호화기를 통해 객체 영상과 배경 영상으로 각각 복호화된다. 그 후 복호화 된 두 영상 중 해상도가 원본 해상도와 다른 경우는, 원본 영상 해상도로 재조정(Rescale, 여기서는 Up-sampling)된다. 마지막으로 두 영상은 보간법(Interpolation) 등을 통해 다시 하나의 영상으로 만들어져 머신 비전의 성능을 비교 해 보게 된다.

2. 이진 맵 생성 알고리즘

본 절에서는 객체 영상과 배경 영상 분리를 위한 이진 맵

생성 알고리즘에 대해 서술한다. 마진 없이 이진 맵을 생성할 경우 충분한 수용 영역이 없기 때문에 머신 비전을 위한 임무 수행 결과에 악영향을 미칠 수 있다. 따라서 우수한 객체 탐지 성능을 유지하기 위해 그림 2의 이진 맵 생성 알고리즘으로 객체 위치 정보에 마진을 더하여 이진 맵을 생성한다. 영상 내 객체의 크기와 비율이 다르기 때문에 제안하는 방법은 객체의 비율과 크기에 따라 마진을 다르게 주는 방법을 사용하였다. 먼저 객체의 가로 세로 비율에 따라 와를 다르게 설정하였다(line 3-9). 그리고 객체 탐지 결과와 객체 크기에 따라 영상 내 32x32보다 작은 객체에 대해서는 α 배, 32x32부터 96x96사이의 객체에 대해서는 β 배, 96x96보다 큰 객체에 대해서는 γ 배 만큼 마진을 주었다(line 10-20). 객체 크기는 COCO기반 평가 중 크기에 따른 AP 평가 방법에서 정의한 객체 크기에 따라 나누었다. 또한, 복호화 이후 객체 영상과 배경 영상의 경계에서의 아티팩트(artifact)를 줄이기 위해 각각의 와에 대해 4픽셀 만큼 확장/축소하였다. 최적의 값을

Algorithm 1 Binary Map Generation Algorithm

Input : Predicted bboxes, IR image
Output : Binary Map Images

```

1: Obtain predicted bounding boxes, ( $x, y$  of top-left corner and  $x, y$  of bottom-right corner),  $BBox = \{obj_1, obj_2, \dots, obj_n\}$  from Object Detection Network
2: for  $i = 1, \dots, n$  do
3:   if  $(h \times 2 < w \text{ or } w \times 2 < h)$  and  $h \times w < 64 \times 64$  then
4:      $pw \leftarrow (h+w) / 2, ph \leftarrow (h+w) / 2$ 
5:   else if  $h \leq w$  then
6:      $pw \leftarrow h, ph \leftarrow h$ 
7:   else
8:      $pw \leftarrow w, ph \leftarrow w$ 
9:   end if
10:  if  $w \times h < 32 \times 32$  then
11:     $x\_margin \leftarrow pw \times 1.6, y\_margin \leftarrow ph \times \alpha$ 
12:  else if  $w \times h < 96 \times 96$  then
13:     $x\_margin \leftarrow pw \times 2, y\_margin \leftarrow ph \times \beta$ 
14:  else
15:     $x\_margin \leftarrow pw \times 1.4, y\_margin \leftarrow ph \times \gamma$ 
16:  end if
17:   $adjusted\_obj_i[0] \leftarrow obj_i[0] - \text{int}(x\_margin / 2)$ 
18:   $adjusted\_obj_i[1] \leftarrow obj_i[1] - \text{int}(y\_margin / 2)$ 
19:   $adjusted\_obj_i[2] \leftarrow obj_i[2] + \text{int}(x\_margin / 2)$ 
20:   $adjusted\_obj_i[3] \leftarrow obj_i[3] + \text{int}(y\_margin / 2)$ 
21: end for
22: for  $i = 1, \dots, n$  do
23:   create foreground binary map with extended  $adjusted\_obj_i$  by 4 pixels
24:   create background binary map with reduced  $adjusted\_obj_i$  by 4 pixels
25: end for

```

그림 2. 이진 맵 생성 알고리즘

Fig. 2. Binary Map Generation Algorithm

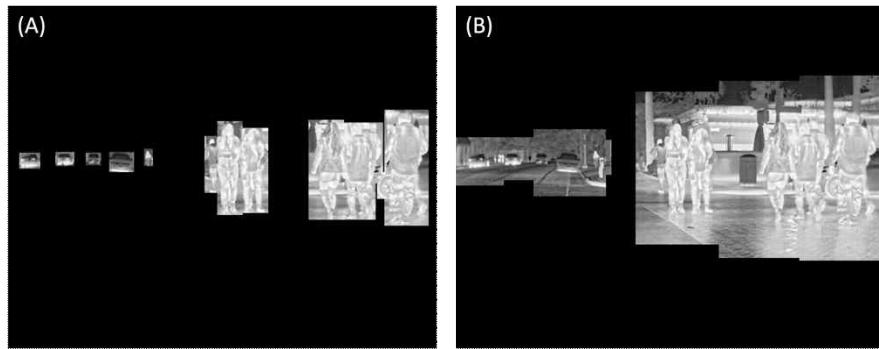


그림 3. 마진에 따른 객체 영상 샘플
Fig. 3. Object Image Sample according to Margin

구하기 위해 본 논문에서는 여러 번의 파라미터 튜닝 과정을 통해 원본 영상의 객체 탐지 결과와 거의 비슷한 성능을 낼 수 있는 $\alpha=1.4$, $\beta=2$, $\gamma=1.6$ 라는 값을 얻었다.

표 1. 마진 따른 객체 탐지 수행 결과
Table 1. Object Detection Results according to Margin

	No Margin	Adding Margin($\alpha=1.4$, $\beta=2$, $\gamma=1.6$)
AP	23.518	39.290
AP50	42.957	75.835
AP75	22.850	35.595
APs	13.190	29.583
APm	31.162	46.581
API	50.289	64.755

그림 3의 (A)는 마진 없이, (B)는 그림 2의 알고리즘에 따라 마진을 추가하여 객체 영상을 생성했을 때 샘플 영상을 보여준다. (B)의 경우 영상 내 객체를 중심으로 마진을 준 것을 확인할 수 있다. 표 1은 마진 없는 객체 영상에 기반하여 객체 탐지를 수행 했을 때와 이진 맵 생성 알고리즘을 통해 마진을 할당한 객체 영상에 기반하여, 객체 탐지를 수행 했을 때의 결과를 보여준다. 원본 영상의 AP(Average Precision)는 40.557이다. 마진 없이 설정한 객체 영상을 기반하여 객체 탐지를 수행한 경우 23.518로 AP가 크게 떨어지지만 이진 맵 생성 알고리즘을 통해 마진을 할당한 객체 영상을 기반으로 객체 탐지를 수행하면 AP가 39.290로 원본 영상 결과와 비교해 약 1%만 감소한 것을 볼 수 있다. 따라서 충분한 마진을 주어 객체 영상과 배경 영상을 나누는 것은 압축률 상승에 따른 객체 탐지 성능의 저하를 방지한다고 할 수 있다.

III. 실험 결과 및 분석

1. 데이터 세트 및 실험 환경

실험에 사용된 데이터 세트는 VCM Anchor 데이터 세트로 채택된 데이터 세트 중 하나인 FLIR Thermal 데이터 세트^[21]이다. FLIR Thermal 데이터 세트는 자동차 위에 열 적외선 카메라를 설치하여 도로 위를 쪽은 열 적외선 영상으로, 열 센서를 통해 취득한 복사열 정보를 기반으로 영상을 생성하기 때문에 RGB 영상 데이터 세트에 비해 환경의 영향을 덜 받는다. 따라서 열 적외선 영상은 감시, 로봇, 자율주행 자동차 등 실제 산업에서 RGB 영상보다 멀티 비전 환경에 적합하다고 할 수 있다. 표 2는 영상 압축 및 신경망 수행을 위한 실험 환경을 보여준다.

표 2. 실험 환경
Table 2. Experimental Environment

For Running Neural Network		For Image Compression	
OS	Ubuntu 18.04 LTS	OS	Mac OS Big Sur 11.5
GPU	RTX 2080 Ti	CPU	Apple M1 chip
CUDA	10.2	FFMPEG	4.2.2
Pytorch	1.5.1	VTM	12.0

실험에 사용된 객체 탐지 신경망은 Detectron2에 구현되어 있는 Faster R-CNN X101-FPN^[22]이며, VCM에서 객체 탐지 신경망으로 채택된 신경망이다. Detectron2의 Faster R-CNN X101-FPN의 경우 RGB 영상으로 사전 학습된 신경

망이기 때문에 열 적외선 영상의 특징을 인지할 수 있도록 FLIR 데이터 세트로 재 학습하여 사용하였다. 영상 압축 기술은 가장 최신 기술인 VVC를 사용하였으며, VTM (VVC Test Model) 12.0을 통해 부호화를 수행하였다. FLIR 데이터 세트는 bitDepth가 8bit의 영상이지만 VTM을 사용해 영상을 압축할 때 bitDepth 파라미터는 10, Conformance WindowMode는 1인 All Intra 모드로 압축을 수행하였다. 그 외 영상 크기 조정, 영상 포맷 변환 등의 기술은 FFmpeg 4.2.2를 사용하였다. FFmpeg으로 PNG에서 YUV로 변경하는 경우에는 yuv420p를, 복호화된 YUV를 PNG로 변경하는 경우에는 yuv420p10le를 사용하였다. 모든 실험 환경은 VCM의 평가 체제 문서^[17]를 따랐다.

2. 실험 결과

VTM 12.0^[24]으로 전체 이미지를 압축 한 결과와 제안하는 객체 기반 압축을 한 결과 및 성능 비교는 다음과 같다.. 표 3은 VTM 12.0으로 전체 영상을 4개의 해상도(100%,

표 3. FLIR Anchor 결과
 Table 3. FLIR Anchor Results

Scale	QP	bpp	mAP
original			40.557
100%	22	1.892	39.317
	27	1.325	39.323
	32	0.376	39.685
	37	0.146	34.578
	42	0.072	24.888
	47	0.034	12.746
75%	22	0.886	40.340
	27	0.399	39.641
	32	0.189	36.626
	37	0.098	30.878
	42	0.049	19.025
	47	0.022	7.601
50%	22	0.350	35.607
	27	0.193	35.143
	32	0.107	29.875
	37	0.056	22.369
	42	0.027	10.043
	47	0.012	2.567
25%	22	0.099	18.439
	27	0.063	15.966
	32	0.037	12.012
	37	0.020	6.271
	42	0.010	1.142
	47	0.005	0.252

75%, 50%, 25%) 및 6개의 QP(22, 27, 32, 37, 42, 47)로 압축을 수행한 FLIR Anchor 결과이며, 표 4는 제안하는 방법을 적용한 결과를 보여준다. 본 논문에서는 객체 기반 압축을 수행할 때 객체 영상 해상도 100% 배경 영상 해상도 100%와 객체 영상 해상도 75% 배경 영상 해상도 50%에서 서로 다른 QP로 압축을 수행하였다.

표 4. 객체 기반 압축 수행 결과

Table 4. Result of Object Detection based Compression

FG Scale	BG Scale	FG QP	BG QP	bpp	mAP
100%	100%	22	27	1.540	39.477
		22	32	0.779	40.244
		22	37	0.596	40.445
		22	42	0.544	40.190
		22	47	0.518	39.656
		27	32	0.630	39.353
		27	37	0.447	39.981
		27	42	0.396	39.686
		27	47	0.369	39.169
		32	37	0.221	39.248
		32	42	0.169	39.095
		32	47	0.143	38.606
		37	42	0.113	34.437
		37	47	0.087	34.275
75%	50%	42	47	0.060	24.846
		22	27	0.390	40.188
		22	32	0.325	40.131
		22	37	0.288	39.667
		22	42	0.268	39.438
		22	47	0.257	38.892
		27	32	0.209	40.045
		27	37	0.173	39.846
		27	42	0.152	39.116
		27	47	0.142	38.766
		32	37	0.117	38.489
		32	42	0.096	37.522
		32	47	0.086	37.001
		37	42	0.065	29.945
		37	47	0.054	29.983
		42	47	0.035	18.763

표 3과 4를 비교해보면 유사한 bpp(bits per pixel)에서 표 4가 더 높은 mAP(mean Average Precision)^[23]를 가지는 것을 볼 수 있다. 표 3의 100% 해상도, QP 22의 bpp값은 1.892이지만 표 4의 객체 영상 해상도 100%, QP 22, 배경 영상 해상도 100%, QP 47에 대한 bpp 값은 0.518이다. 두 결과를 비교하면 유사한 mAP에서 bpp값은 3배 이상 작아진 것을 확인 할 수 있다. 객체 영상과 배경 영상을 분리 시 객체 영상에 충분한 마진을 주었기 때문에 배경 영상의

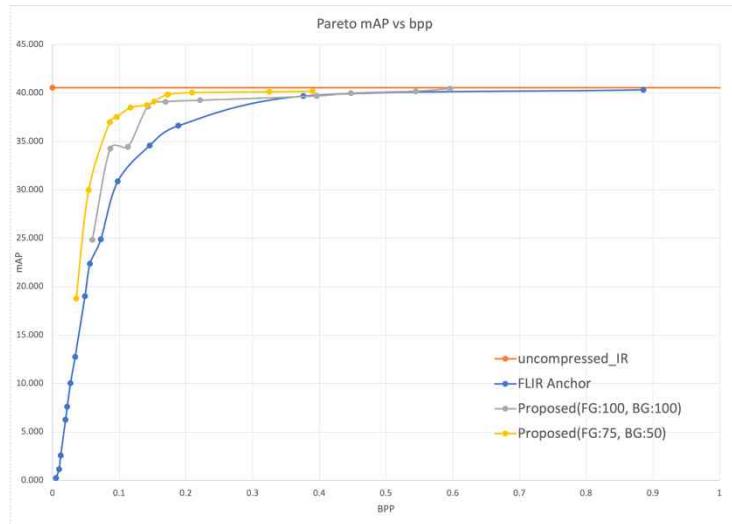


그림 4. Pareto-Front Curve 결과
Fig. 4. Pareto-Front Curve Result

압축률이 객체 영상 압축률과 큰 차이가 있음에도 불구하고 mAP값은 크게 떨어지지 않는다.

그림 4는 표 3과 4의 결과를 기반으로 VCM에서 제공하는 템플릿 문서^[18] Pareto-Front Curve 결과를 보여준다. 그림 4를 보면 제안하는 방법이 FLIR Anchor 커브보다 좌측 상단에 그려지는 것을 확인 할 수 있다. 그림 5는 VVC로 전체 영상을 하나의 QP로 부/복호화한 결과 (a)와 논문에서 제안하는 방법으로 객체 기반 압축을 수행 한 결과를 복호

화한 결과(b)를 보여주며, (a), (b) 모두 복호화 된 영상에 대해 객체 탐지를 수행한 결과이다. 그림 5의 (b)를 통해 객체 부분에 대해서 (a)와 동일하게 높은 품질을 유지하는 것을 알 수 있다. 배경 부분은 비교적 낮은 품질을 가지고 있지만 영상 내 중요한 부분인 객체 부분에 대한 영상 품질이 높기 때문에 주관적인 품질도 크게 나쁘지 않은 것을 확인할 수 있다. 또한 두 영상에 대해 객체 탐지를 수행한 결과는 대등한 것을 알 수 있다. 따라서 본 논문에서 제안하

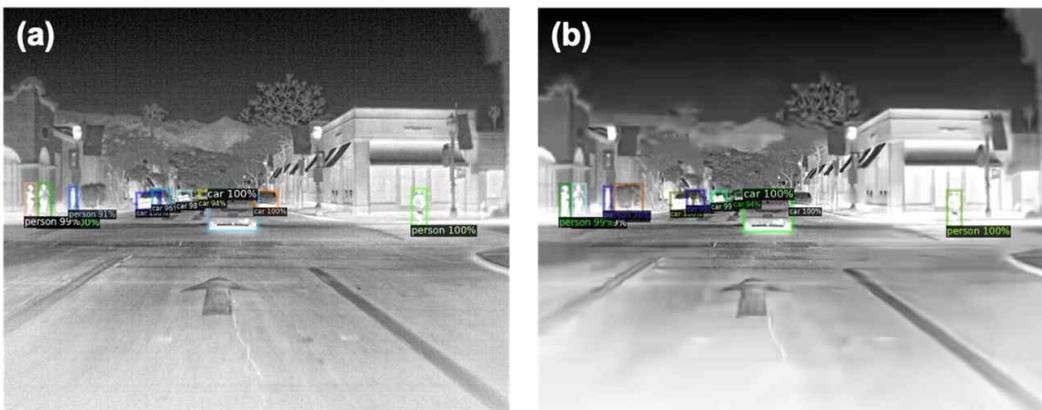


그림 5. 영상 복호화 후 객체 탐지 수행 결과
(a: 해상도: 100%, QP: 22, b: FG 해상도: 100%, BG 해상도: 100%, FG QP 22, BG QP: 47)
Table 5. Object detection result after decoding
(a: resolution: 100%, QP: 22, b: FG resolution: 100%, BG resolution: 100%, FG QP 22, BG QP: 47)

는 방식이 높은 영상 압축률을 가질 뿐만이 아니라 머신 비전 관점에서도 높은 객체 탐지 성능을 보여준다.

표 5. FLIR Anchor와 객체 기반 압축 기법 결과 비교
Table 5. Comparison of FLIR Anchor and Object-Based Compression Results

FLIR Anchor			Proposed Method(select 6 results)		
scale	bpp	mAP	scale	bpp	mAP
100%	1.892	39.317	FG: 100, BG: 100	0.596	40.445
	1.325	39.323		0.544	40.190
	0.376	39.685		0.447	39.981
	0.146	34.578		0.221	39.248
	0.072	24.888		0.113	34.437
	0.034	12.746		0.060	24.846
75%	0.886	40.340	FG: 75, BG: 50	0.390	40.188
	0.399	39.641		0.325	40.131
	0.189	36.626		0.173	39.846
	0.098	30.878		0.117	38.489
	0.049	19.025		0.054	29.983
	0.022	7.601		0.035	18.763

표 6. BD-rate 결과
Table 6. BD-rate result

Pareto-mAP(None)	-19.83%
100%	-31.30%
75%	-42.92%

압축 성능을 비교하기 위해 표 4에서 객체 영상 해상도에 따라 pareto point를 기반으로 6개의 값을 선택하여 FLIR anchor와 실험 결과를 비교한 것은 표 5와 같으며, 이를 기반으로 BD-rate를 계산한 결과는 표 6과 같다. BD-rate 결과는 6개의 값을 선택하는 기준에 따라 달라지며, 본 논문에서는 객체 영상의 QP가 22인 경우에는 2개, 나머지는 각 QP별로 1개씩 선택하여 FLIR Anchor와 비교하였다. 그 결과 제안하는 방법은 VVC로 영상 전체를 압축하는 방식보다 Pareto-mAP에서 BD-rate 값이 최대 -19.83%로 압축 효율이 뛰어나다는 것을 확인할 수 있다.

IV. 결 론

본 논문에서는 머신 비전을 위한 객체 기반 압축 방법을

제안하였다. 제안하는 방법은 영상 내 정보의 중요성에 따라 머신 비전 성능에 큰 영향을 주는 객체 부분은 압축률을 작게, 상대적으로 덜 영향을 주는 배경 부분은 압축률을 크게 하면서 압축 효율을 높였다. 또한 객체 영상과 배경 영상 분리 과정에서 객체 탐지 결과에 마진을 주어 신경망의 성능을 높게 유지하였다. 실험 결과에 따르면 제안하는 방법은 VVC로 전체 영상을 압축하는 방법보다 BD-rate를 기반으로 볼 때 19.83% 만큼 압축 효율이 높은 것을 확인할 수 있었다.

이러한 객체 기반 압축 방법은 열 적외선뿐만 아니라 RGB 영상에서도 좋은 결과를 보일 것으로 예상되므로, 향후 연구로서 RGB 영상 데이터 세트로 객체 기반 영상 압축 실험을 진행할 예정이다. 또한 최근 딥러닝의 발전으로 신경망 기반 영상 압축^[11,12] 방식에 대한 연구가 다양하게 진행되고 있다. 따라서 VVC 외의 다양한 영상 압축 방법을 사용하여 결과를 비교할 예정이며, 부호화 전 객체 탐지 신경망의 종류에 따라 부호화 하려는 객체가 달라질 수 있기 때문에 이러한 부분에 대해서도 추가 실험을 진행할 예정이다. 또한 객체 탐지 뿐만 아니라 객체 분할 및 객체 추적 등 머신 비전을 위한 다양한 임무 수행을 할 수 있도록 향후 연구를 진행할 예정이다.

참 고 문 헌 (References)

- [1] E. Ke, S. Wang, C. Lin, C. Lin, Y. Nien, T. Li, D. Liu, “[VCM] Feature map compression for VCM”, the 130th MPEG meeting, Alpbach, April, 2020.
- [2] Y. K. Yoon, D. Park, S. Chun, J. Kim, “[VCM] Results of feature map coding for object segmentation on Cityscapes datasets”, the 132nd MPEG meeting, Online, October, 2020.
- [3] S. Kim, M. Jeong, H. Jin, H. Lee, H. Choo, H. Lim, J. Seo, “[VCM] A report on intermediate feature coding for object detection and segmentation”, the 132nd MPEG meeting, Online, October, 2020.
- [4] H. Han, H. Choi, S. Kwak, J. Yun, W. Cheong, J. Seo, “[VCM] Investigation on feature map channel reordering and compression for object detection”, the 134th MPEG meeting, Online, April, 2021.
- [5] H. Wang, H. Wang, L. Wang, Y. Zhang, “[VCM] Feature prediction using Levene Test, KL-Divergence, and linear correlation and A Pipeline for Feature Coding”, the 135th MPEG meeting, Online, July, 2021.
- [6] Y. Yoon, D. Kim, J. Kim, “[VCM] Compression of reordered feature sequences based on channel means for object detection”, the 135th MPEG meeting, Online, July, 2021.

- [7] H. Wang, H. Wang, L. Wang, Y. Zhang, D. Yan, “[VCM] A Method of Intra-Frame Channel Prediction for Feature Coding”, the 134th MPEG meeting, Online, April, 2021.
- [8] H. Choi, M. Lee, J. Kim, K. Kim, Y. Lee, D. Sim, S. Oh, J. Do, H. Kwon, S. Jeong, “[VCM] A result of feature data reduction using PCA for object detection”, the 132nd MPEG meeting, Online, October, 2020.
- [9] S. Wang, Z. Wang, Y. Ye, S. Wang, “[VCM] Image or video format of feature map compression for object detection”, the 133rd MPEG meeting, Online, January, 2021.
- [10] S. Wang, Z. Wang, Y. Ye, S. Wang, “[VCM] Investigation on feature map quantization for object detection and compression”, the 133rd MPEG meeting, Online, January, 2021.
- [11] Y. Yoon, J. Kim, “[VCM] Evaluation results of object segmentation with deep learning-based image compression”, the 133rd MPEG meeting, Online, January, 2021.
- [12] B. Zhu, L. Yu, D. Li, “[VCM] Deep learning-based compression for machine vision”, the 135th MPEG meeting, Online, July, 2021.
- [13] S. Wang, C. Lin, C. Lin, T. Li, Y. Nie, “[VCM] Video codec optimization for VCM”, the 135th MPEG meeting, Online, July, 2021.
- [14] C. Hollmann, J. Ström, M. Damghanian, L. Litwic, “[VCM] VCM-based rate-distortion optimization for VVC”, the 135th MPEG meeting, Online, July, 2021.
- [15] W. Gao, X. Xu, S. Liu, “[VCM] Response to CfE: Investigation of VVC Codec for Video Coding for Machine”, the 134th MPEG meeting, Online, April, 2021.
- [16] Y. Lee, S. Kim, K. Yoon, H. Lim, H. Choo, W. Cheong, Je. Seo, “[VCM] Updated Evidence of object detection in FLIR using VTM 12.0”, the 135th MPEG meeting, Online, July, 2021.
- [17] WG 02 MPEG Technical requirements, “Evaluation Framework for Video Coding for Machines”, the 135th MPEG meeting, Online, July, 2021.
- [18] C. Hollmann, J. Ström, M. Damghanian, L. Litwic, “[VCM] VCM-based rate-distortion optimization for VVC”, the 135th MPEG meeting, Online, July, 2021.
- [19] D. Minnen, J. Ballé, and G. Toderici, “Joint Autoregressive and Hierarchical Priors for Learned Image Compression”, *arXiv:1809.02736*, Sep. 2018, <http://arxiv.org/abs/1809.02736>(accessed Aug. 30, 2021).
- [20] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules”, *arXiv:2001.01568*, Jan. 2020, <http://arxiv.org/abs/2001.01568> (accessed Aug. 30, 2021).
- [21] Free FLIR Thermal dataset, <https://www.flir.com/oem/adas/dataset/> (accessed Aug. 30, 2021).
- [22] Detectron2, <https://github.com/facebookresearch/detectron2> (accessed Aug. 30, 2021).
- [23] T Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla 'r, and C. Zitnick, “Microsoft coco: Common objects in context”, in *European conference on computer vision. Springer*, 2014, pp. 740 - 755.
- [24] WG 05 MPEG Joint Video Coding Team(s) with ITU-T SG 16, “Test Model 12 for Versatile Video Coding (VTM 12)”, the 133rd MPEG meeting, Online, January, 2021.

저자 소개

이 예 지



- 2018년 2월 : 극동대학교 스마트모바일학과 졸업(학사)
- 2020년 2월 : 건국대학교 스마트ICT융합과 졸업(석사)
- 2020년 3월 ~ 현재 : 건국대학교 컴퓨터공학과 박사과정
- ORCID : <https://orcid.org/0000-0002-0292-160X>
- 주관심분야 : 영상처리, 인공지능, 컴퓨터비전

김 신



- 2015년 8월 : 건국대학교 컴퓨터공학과 졸업(학사)
- 2017년 2월 : 건국대학교 컴퓨터공학과 졸업(석사)
- 2017년 3월 ~ 현재 : 건국대학교 컴퓨터공학과 박사과정
- ORCID : <https://orcid.org/0000-0001-8492-3758>
- 주관심분야 : 영상처리, 인공지능, 컴퓨터 비전

저자 소개

임한신



- 2004년 2월 : 연세대학교 전기전자공학부 (수학 부전공) (공학사)
- 2006년 2월 : 한국과학기술원 전기전자공학부 (공학석사)
- 2007년 9월 ~ 2007년 12월 : TU Berlin 방문연구원
- 2014년 2월 : 한국과학기술원 전기전자공학부 (공학박사)
- 2014년 3월 ~ Current : 한국전자통신연구원 선임연구원
- ORCID : <https://orcid.org/0000-0003-4829-2893>
- 주관심분야 : 2D/3D Image Processing, Computer Vision, 3D Reconstruction and Modeling, VR/AR Technology

추현곤



- 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

정원식



- 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

서정일



- 1994년 2월 : 경북대학교 전자공학과 (공학사)
- 1996년 2월 : 경북대학교 대학원 전자공학과 (공학석사)
- 2005년 8월 : 경북대학교 대학원 전자공학과 (공학박사)
- 1998년 2월 ~ 2000년 10월 : LG반도체 주임연구원
- 2010년 8월 ~ 2011년 7월 : 영국 Southampton University, ISVR 방문연구원
- 2000년 11월 ~ 현재 : 한국전자통신연구원 실감미디어연구실 실장
- ORCID : <http://orcid.org/0000-0001-5131-0939>
- 주관심분야 : 오디오 신호처리, 실감음향, 디지털방송, 멀티미디어 표준화

윤경로



- 1987년 2월 : 연세대학교 전자전산기공학과 졸업(학사)
- 1989년 12월 : University of Michigan, Ann Arbor, 전자공학과 졸업(석사)
- 1999년 5월 : Syracuse University, 전산과학과 졸업(박사)
- 1999년 6월 ~ 2003년 8월 : LG 전자기술원 책임연구원/그룹장
- 2003년 9월 ~ 현재 : 건국대학교 컴퓨터공학과/스마트ICT융합공학과 교수
- ORCID : <https://orcid.org/0000-0002-1153-4038>
- 주관심분야 : 스마트미디어시스템, 멀티미디어검색, 영상처리, 멀티미디어/메타데이터 처리