

열 적외선 영상에서 기계를 위한 객체 기반 압축 기법

이예지 김신 윤경로† *임한신 *추현곤 *정원식

건국대학교 *한국전자통신연구원

yoonk@konkuk.ac.kr†

Object-based Compression Method for Machine Vision
in Thermal Infrared Image

Yegi Lee Shin Kim Kyoungro Yoon† *Hanshin Lim *Hyon-Gon Choo *Won-Sik Cheong

Konkuk University, *Electronics and Telecommunications Research Institute

요약

최근 딥러닝 기술에 발전으로 스마트 시티, 자율주행 자동차, 감시, 사물인터넷 등 다양한 분야에서 활용이 되고 있으며, 이에 따라 기계를 위한 영상 압축에 대한 필요성이 대두되고 있다. 본 논문에서는 열 적외선 영상에서 기계 소비를 위한 객체 기반 압축 기법을 제안한다. 신경망의 객체 탐지 결과와 객체 크기에 따라 이미지를 객체 부분과 배경 부분으로 나누어 서로 다른 압축률로 인/디코딩 한 후, 나뉜진 이미지를 다시 하나의 이미지로 합치는 기법을 사용하여 압축하였으며, 이는 압축효율은 높으면서 객체 탐지 성능을 높게 유지한다. 실험 결과, 제안하는 방법이 Pareto mAP에서 BD-rate가 -28.92%로 FLIR anchor 결과와 비교했을 때 압축효율이 뛰어나다는 것을 확인할 수 있다.

1. 서론

최근 딥러닝 기술에 발전으로 스마트 시티, 자율주행 자동차, 감시, 모니터링 등 다양한 분야에서 딥러닝 기술이 연구되고 있으며, 실제 산업에 적용되는 사례가 증가하고 있다. 이에 따라 기계 소비를 위한 영상 데이터의 양이 증가하여, HEVC(H.265, High Efficiency Video Coding), VVC(H.266, Versatile Video Coding)와 같은 기존 사용자 기반 영상 압축 방식이 아닌 기계를 위한 효율적인 영상 압축 및 전송에 대한 필요성이 대두되고 있다. 이러한 필요성을 충족시키기 위해 국제 표준화 기구인 MPEG에서는 기계 소비를 위한 영상 압축 기술에 대한 표준화 진행하기 위해 VCM을 구성하였다. 지난 134차 회의에서는 CFE에 대한 응답 기술에 대한 논의를 하였다.

본 논문에서는 VCM CFE 응답에 대한 하나의 후보기술로 열 적외선 영상 압축을 위한 객체 기반 압축 방법을 제안한다. 제안하는 방법은 MPEG-VCM에서 객체 탐지 평가 신경망으로 사용되는 Faster R-CNN X101-FPN[3]을 사용하여 객체 탐지를 수행한다. 그리고 탐지 결과인 물체의 위치 정보(Bounding Box)와 객체 크기에 따라 열적외선 영상을 객체 부분과 배경 부분으로 나누어 VVC 코덱으로 서로 다른 압축률로 인/디코딩 후 다시 하나의 이미지로 만든다.

본 논문에서 제안하는 기법은 기계 소비를 위한 영상 압축 방식에 초점을 두고 있다. 기계가 관심이 있는 객체 부분은 낮은 압축률로 인코딩하고, 그 외 배경부분은 큰 압축률로 인코딩하여 효율을 높이면서 객체 탐지 성능은 높게 유지할 수 있다. 실험 결과, 이미지 전체를 동일한 압축률로 인코딩을 수행 했을 때 보다 Pareto mAP에서 BD-rate가 -28.92%로 압축효율이 향상되는 것을 확인할 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 실험 환경 및 객체 기

반 압축 기법에 대해 서술하며, 제 3장에서는 열 적외선 영상으로 객체 기반 압축을 수행한 결과에 대해 기술한다. 마지막으로 제 4장에서는 본 논문의 결론을 짓는다.

2. 실험 환경 및 객체기반 압축 기법

2-1. 실험 환경

본 논문에서 실험에 사용된 데이터 세트는 FLIR Thermal 데이터 세트[4]로, VCM에서 anchor 데이터 세트로 채택된 데이터 세트이다. 지난 MPEG 134차 회의에서 업데이트 된 FLIR anchor는 VTM(VVC Test Model) 8.2를 사용했다. 하지만 본 논문에서는 VTM 12.0으로 실험한 FLIR anchor 값을 계산하여, 기계를 위한 객체 압축 결과에 대한 비교군으로 사용하였다. 실험에 사용된 객체 탐지 신경망은 Faster R-CNN X101 FPN[3]이며, FLIR 데이터 세트로 재학습한 신경망을 사용하였다. 그 외의 실험 환경은 MPEG-VCM에 평가체제[1] 문서를 따랐으며, 요약하면 아래와 같다.

- 영상 코덱: VTM 12.0
- QP(Quantization Parameter): 22, 27, 32, 37, 42, 47
- bitDepth: 10
- 영상 처리(포맷 및 해상도 변환): FFMPEG 4.2.2
- Encoding pixel format: yuv420p
- Decoding pixel format: yuv420p10le

2-2. 객체 기반 압축 기법

기계를 위한 객체 기반 압축 기법의 주된 영감은 영상 내 정보의 중요성에 따라 압축률을 다르게 하는 것이다. 본 논문에서는 기계가 관심이 있고 중요한 부분이라고 생각이 되는 객체 부분은 낮은 QP로 압축을 수행하고, 상대적으로 덜 중요한 부분으로 여겨지는 배경 부분은 높은 QP로 압축을 수행한다. 이는 압축 효율을 높이면서, 다른 기계가 디코딩된 영상으로 객체 탐지를 수행하게 되면 높은 객체 탐지 성능을 가질 수 있다.

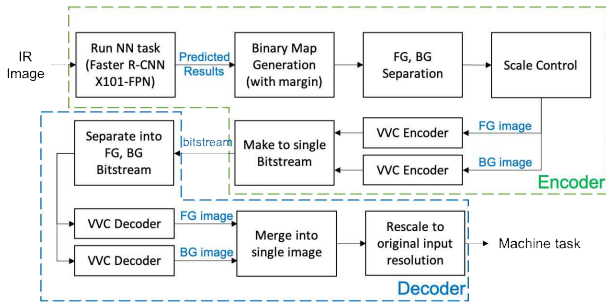


그림 1. 객체 기반 압축 수행 과정

그림 1은 객체 기반 압축 기법 수행 과정을 보여준다. 수행 과정은 다음과 같다. 인코더는 객체 부분과 배경 부분의 분리를 위해 객체 탐지 신경망을 실행하여 객체 탐지 결과를 얻는다. 객체 탐지 신경망은 VCM에서 평가 네트워크로 사용되는 Detectron2의 Faster R-CNN X101-FPN[3]을 사용한다. 그 후 객체 탐지 결과인 위치 정보로 이진 맵을 생성하며, 이진 맵은 그림 2의 알고리즘과 같이 객체 위치 정보의 마진을 더하여 생성된다. 마진 없이 이진 맵을 만들게 되면, 객체 주변의 아티팩트(artifact)의 여파로 인해 임무 수행 결과에 악영향을 미칠 수 있기 때문에 그림 2와 같은 방법을 사용한다.

Algorithm 1 Binary Map Generation Algorithm

```

Input : 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 = {obj1, obj2, ..., objn} using neural network
2: for i = 1, ..., n do
3:   if (h × 2 < w or w × 2 < h) and h × w < 64 × 64 then
4:     pw ← (h+w) / 2, ph ← (h+w) / 2
5:   else if h ≤ w then
6:     pw ← h, ph ← h
7:   else
8:     pw ← w, ph ← w
9:   if w × h < 32 × 32 then
10:    x_margin ← pw × 1.6, y_margin ← ph × 1.6
11:   else if w × h < 96 × 96 then
12:    x_margin ← pw × 2, y_margin ← ph × 2
13:   else
14:    x_margin ← pw × 1.4, y_margin ← ph × 1.4
15:   adjusted_obji[0] ← obji[0] - int(x_margin / 2)
16:   adjusted_obji[1] ← obji[1] - int(y_margin / 2)
17:   adjusted_obji[2] ← obji[2] + int(x_margin / 2)
18:   adjusted_obji[3] ← obji[3] + int(y_margin / 2)
19: for i = 1, ..., n do
20:   create foreground map with extended adjusted_obji by 4 pixels
21:   create background map with reduced adjusted_obji by 4 pixels
    
```

그림 2 이진 맵 생성 알고리즘

그 후, 생성된 이진 맵을 기반으로 비트연산자를 사용하여 객체 이

미지(Foreground image)와 배경 이미지(Background image)으로 나눈다. 압축 효율을 높이기 위해 VVC 압축 전 객체 이미지와 배경 이미지는 down-scaling 될 수 있으며, down-scaling 수행 후 배경 이미지 해상도는 객체 이미지 해상도보다 항상 작거나 같게 된다. 다음으로 두 이미지는 각각 VVC 인코더를 사용하여 압축을 수행한다. VVC 인코딩을 수행 시 객체 이미지의 QP는 배경 이미지의 QP보다 항상 작다. 마지막으로 VVC 인코딩 수행 후 만들어진 두 개의 비트스트림(bitstream)은 하나로 만들어 다른 기계에게 전송이 된다.

그림 1의 디코더는 하나의 비트스트림을 받아 두 개의 비트스트림으로 나누어지며, VVC 디코더에 의해 객체 이미지와 배경 이미지로 디코딩 된다. 분리된 두 이미지는 보간법(interpolation)등을 통해 다시 하나의 이미지로 만들어지며, 만약 출력 영상이 원본 영상 해상도가 아닌 경우에는 FFmpeg으로 up-scaling된다.

3. 실험 결과

본 논문에서는 VTM 12.0으로 계산한 FLIR anchor 값을 계산하여, 객체 기반 압축 기법의 성능 비교를 하고자 한다. 실험은 총 2가지로 진행이 되었다. 실험 1은 객체 이미지 해상도 100%, 배경 이미지 해상도 100%로 진행하였으며, 실험 2는 객체 이미지 해상도 75%, 배경 이미지 해상도 50%로 객체 기반 압축 실험을 하였다. 그림 3은 FLIR anchor 결과와 실험 1, 실험 2 결과를 기반으로 Pareto-Front Curve를 그린 결과이다. 그림 3을 보면, FLIR anchor 보다 본 논문에서 제안한 방법이 근사한 bpp(bits per pixel)에서 더 높은 mAP(mean Average Precision, mAP@[0.5:0.95])를 가지는 것을 확인할 수 있다.

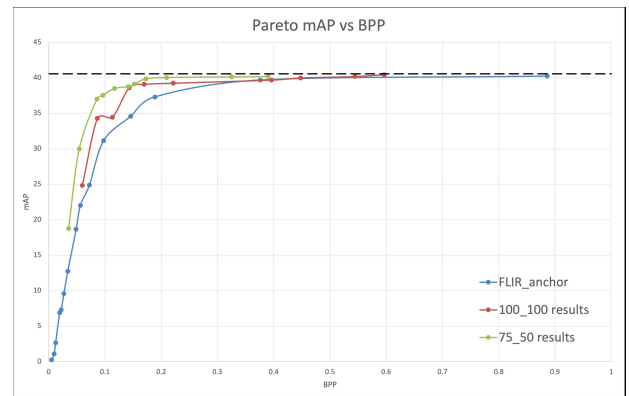


그림 3 Pareto-Front curve 결과(파란색 선: FLIR anchor, 빨간색 선: FG 해상도 100%, BG 해상도 100% 결과, 초록색 선: FG 해상도 75%, BG 해상도 50% 결과)

또한 템플릿 문서[2]를 기반으로 BD-rate를 계산한 결과는 표 1과 같다. FLIR anchor는 각 해상도별 QP 22~47까지 6개의 결과가 존재하기 때문에 본 논문에서는 결과 비교를 위해 실험 1과 2의 결과 중에 Pareto mAP 결과를 기반으로 각 6개의 값을 선별하여 BD-rate 값을 구하였다. 실험 결과, FLIR anchor보다 본 논문에서 제안한 객체 기반 압축 기법이 압축 효율이 뛰어나다는 것을 확인할 수 있다.

표 1 BD-rate 결과

mAP	100%	-30.65%
	75%	-41.65%
	Average mAP	-36.15%
Pareto mAP		-28.92%

4. 결론

본 논문에서는 열 적외선 영상에서 기계 소비를 위한 객체 기반 압축 기법을 제안하였다. 제안하는 방법은 기계가 관심이 있는 객체 부분은 압축을 적게, 그 외의 배경 부분은 압축을 크게 하여 압축 효율을 높이면서 객체 탐지 성능 또한 높게 유지한다. 실험 결과 VTM 12.0으로 계산된 FLIR anchor보다 Pareto mAP에서 BD-rate가 -28.92%로 압축효율이 높은 것을 확인 할 수 있었다

본 논문에서 제안하는 방법은 열 적외선 영상뿐만 아니라 RGB 영상에서도 충분히 좋은 성능을 보일 것이라고 생각되며, 향후에는 RGB 영상으로 실험을 진행 할 예정이다.

감사의 글

본 연구 논문은 과학기술정보통신부 및 정보통신기획 평가원의 출연금으로 수행되고 있는 "기계를 위한 영상 부호화 기술(No.2020-0-00011)" 과제의 연구결과입니다.

참고문헌

- [1] ISO/IEC JCT1/SC29/WG2, "Evaluation Framework for Video Coding for Machine", N78, Apr. 2021.
- [2] ISO/IEC JCT1/SC29/WG2, "[VCM]Updated Evaluation Methodology and Reporting Template", m56962, Apr. 2021.
- [3] Detectron2,
<https://github.com/facebookresearch/detectron2> (accessed May, 24, 2021).
- [4] Free FLIR Thermal dataset,
<https://www.flir.com/oem/adas/dataset/> (accessed May, 24, 2021).