

## NNR 비트스트림 기반 ISOBMFF 캡슐화 방안

\*이민석 \*\*이성배 \*\*\*남귀중 \*\*\*\*김규현

경희대학교

\*qsibmini@khu.ac.kr \*\*rhee@khu.ac.kr \*\*\*nkj0427@khu.ac.kr

\*\*\*\*kyuheonkim@khu.ac.kr

## ISOBMFF encapsulation method based on NNR bitstream

\*Lee, Minseok \*\*Rhee, Seongbae \*\*\*Nam, Kwijung \*\*\*\*Kim, Kyuheon

Kyunghee University

## 요약

최근 딥러닝(Deep Learning) 기술이 다양한 분야에서 활용되고 있으며, 사전 학습된 딥러닝 모델에 대한 압축과 전송 방안에 관한 연구 또한 활발히 진행되고 있다. 이와 관련하여, 국제 표준화 기구인 ISO/IEC 산하 MPEG(Moving Picture Expert Group)에서는 인공지능경망 모델을 다양한 딥러닝 프레임워크(Deep Learning Framework)에서 상호운용 가능한 포맷으로 압축 표현할 수 있는 NNC(Compression of Neural Network for Multimedia Content Description and Analysis) 표준화를 진행하고 있다. 압축된 딥러닝 모델의 데이터를 효과적으로 저장하여 전송 및 사용하기 위해서는 ISOBMFF(ISO based Media File Format) 캡슐화 과정이 필요하다. 본 논문에서는 MPEG의 NNC 표준에 따라 사전 학습된 딥러닝 모델을 압축한 후 이를 통해 생성된 비트스트림(bitstream)을 ISOBMFF로 캡슐화하기 위한 기술을 제안 및 실험한다. 또한, 실험에 대한 검증을 위하여 생성된 ISOBMFF 데이터를 비트스트림으로 복원한 뒤 복호화하여 입력 비트스트림과 차이가 없음을 확인한다.

## 1. 서론

최근 딥러닝 네트워크의 성능이 발전함에 따라 개인 디바이스에서 딥러닝 네트워크를 저장, 학습, 실행하고자 하는 니즈(needs)가 증가하고 있다. 딥러닝 네트워크를 개인 디바이스에서 활용하기 위해서는 사전에 학습된 딥러닝 모델을 네트워크를 통해 전송 받아야 하지만, 딥러닝 네트워크는 데이터 크기가 크기 때문에 전송비용이 비싸다는 제한 사항이 있다. 또한, 딥러닝 네트워크를 구성하기 위해서는 상당히 많은 파라미터와 연산 과정이 필요하기에 메모리와 연산 능력이 부족한 개인 디바이스에서 딥러닝 네트워크를 저장, 학습, 실행하기에는 어려움이 있다[1]. 이를 해결하고자, 국제 표준화 기구인 ISO/IEC 산하 MPEG(Moving Picture Expert Group)에서는 사전 학습된 딥러닝 네트워크에 대하여 경량화 및 압축을 수행함으로써 효율적인 전송 및 저장을 가능케 하는 NNC(Compression of Neural Networks for Multimedia Content Description and Analysis) 표준을 제정하고 있다[2].

NNC 부호화기는 사전 학습된 딥러닝 네트워크를 입력으로 받아 이를 압축하여 NNR(Compression of Neural Networks for Multimedia Content Description and Analysis) 비트스트림(bitstream)을 출력한다. NNR은 NNC의 이전 명칭으로써, 비트스트림 관련 유닛에 대해서는 이전 명칭인 NNR이 사용된다. NNR 비트스트림은 압축 효율이 뛰어나다는 장점이 있으나, 인터넷 기반 전송 프로토콜에 최적화되어있지 않다는 단점이 있다. 개인 디바이스에서의 딥러닝 네트워크 실행에 대한 수요가 늘어남에 따라 서버에서 안정적으로 딥러닝 네트워크 데이터를 전송해

야할 필요성은 커졌으며, NNR 비트스트림은 이에 부적합할 수 있다. 그러나 NNR 비트스트림을 ISOBMFF(ISO based Media File Format)의 형태로 캡슐화(Encapsulate)하면 이러한 문제를 해결할 수 있다. ISOBMFF는 멀티미디어 컨텐츠의 저장, 교환, 및 점진적 다운로드(Progressive Download)를 위한 기술로 개발되었으며, 현재 인터넷상에서 동영상 등 수많은 미디어 데이터는 ISOBMFF의 형태로 저장 및 전송되고 있다[3]. 이러한 점을 고려하였을 때, NNR 비트스트림의 안정적·효율적 전송을 위해서는 ISOBMFF 캡슐화 과정이 필요하다.

이에 본 논문의 2장에서 NNC 및 ISOBMFF 표준에 대해 설명하고, 3장에서는 본 논문에서 제안하는 ISOBMFF 기반 NNR 비트스트림의 캡슐화 기술을 제안하고자 한다. 이어지는 4장에서는 제안한 캡슐화 기술의 실험 결과를 기술하며, 5장에서는 마지막으로 제안한 기술에 대한 결론을 맺는다.

## 2. 배경 기술

본 장에서는 본 논문에서 제안하는 방안의 배경이 되는 MPEG NNR 비트스트림 및 ISOBMFF의 구조에 대해 설명한다.

## 2.1 NNR 비트스트림

NNR 비트스트림은 그림 1과 같이 인공지능경망의 데이터 혹은 관

런 메타데이터(metadata)를 담은 구조체인 NNR 유닛(NNR unit)으로 이루어져 있다. NNR 유닛은 크기 정보, 헤더(header), 그리고 페이로드(payload)로 구성되어 있으며, NNR 유닛의 종류는 표 1과 같이 총 7가지이다. NNR\_STR은 NNR 비트스트림의 시작을 알리는 역할을 하며, NNR\_MPS는 인공지능경망의 메타데이터 및 부가 정보를 담고 있다. NNR\_LPS는 인공지능경망의 계층(layer)별 혹은 부분별 메타데이터를 가지며, NNR\_TPL은 딥러닝 프레임워크(Deep Learning Framework)의 종류를 나타낸다. NNR\_QNT는 인공지능경망의 압축 과정에서 활용한 양자화(quantization) 정보를 담고, NNR\_NDU는 압축된 인공지능경망 데이터를 담으며, NNR\_AGG는 여러 개의 NNR 유닛을 담을 수 있는 구조이다.

식별자 (identifier)	NNR Unit Type
NNR_STR	NNR start unit
NNR_MPS	NNR model parameter set data unit
NNR_LPS	NNR layer parameter set data unit
NNR_TPL	NNR topology data unit
NNR_QNT	NNR quantization data unit
NNR_NDU	NNR compressed data unit
NNR_AGG	NNR aggregate unit

표 1. NNR 유닛 종류

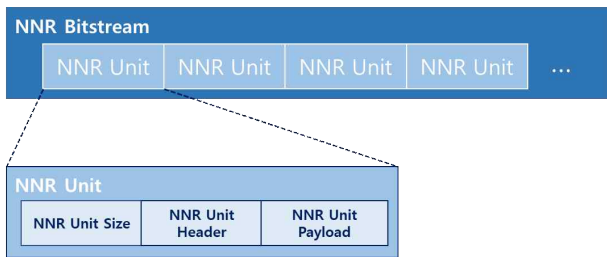


그림 1. NNR 비트스트림 구조

## 2.2 ISOBMFF

ISOBMFF는 MPEG에서 미디어 데이터 및 그와 관련된 메타데이터의 소비를 지원하기 위해 제정한 파일 포맷(file format) 표준이다. 이는 유연하게 확장이 가능한 파일 포맷으로, 미디어의 상호 작용 및 표현 등에 유용하다. ISOBMFF는 네트워크 프로토콜(network protocol)의 종류에 영향을 받지 않도록 설계되었으며, 이러한 강점은 인터넷 프로토콜에서 ISOBMFF가 대중적으로 쓰이게 된 계기이다. ISOBMFF를 이루는 기본적인 단위는 박스(box)이며, 이는 박스의 식별자(type identifier)와 길이에 의해 정의되는 객체 지향형 구조체이다. 박스의 종류는 데이터를 담은 풀 박스(full box)와 서로 다른 박스를 담은 컨테이너 박스(container box)가 있으며, 이러한 박스들이 모여 ISOBMFF를 이룬다. ISOBMFF는 그림 2와 같이 기본적으로 “ftyp”, “moov”, 그리고 “mdat”으로 구성되어 있다. ftyp(file type)는 파일의 종류에 관한 정보를 담고 있으며 moov(movie)에는 미디어의 메타데이터가, mdat(media data)에는 압축된 미디어 데이터가 입력된다. 그림 3은 대표적인 ISOBMFF인 MP4 파일이 어떠한

박스들로 구성되어 있는지 나타낸다. 그림 3에서 확인할 수 있듯 moov 박스 내부에 계층적으로 여러 종류의 박스들이 존재하며, 이러한 박스들에 입력된 메타데이터를 통해 mdat 박스의 미디어 데이터가 재생된다.

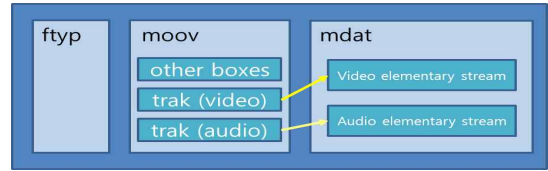


그림 2. ISOBMFF 구조

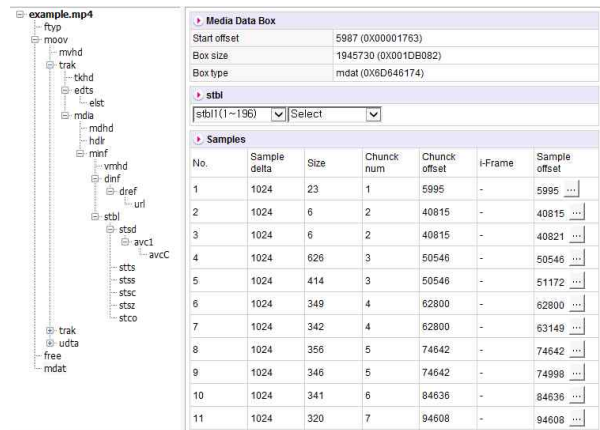


그림 3. MP4 파일 내부 구성

## 3. ISOBMFF 기반 NNR 비트스트림의 캡슐화

2장에서 기술된 배경 기술을 기반으로, 본 장에서는 NNR 비트스트림을 ISOBMFF에서 정의한 박스 구조로 캡슐화하는 방안을 제안한다.

ISOBMFF 캡슐화는 크게 단일 트랙(track) 캡슐화와 멀티 트랙 캡슐화로 나누어지며, 이는 moov 박스 하위의 trak(track) 박스의 개수에 따라 구별된다. 일반적인 MP4 파일은 비디오의 메타데이터와 오디오의 메타데이터를 각기 다른 trak 박스에서 관리하기 때문에 멀티 트랙 구조이다. 그러나 본 논문에서 제안하는 NNR 비트스트림의 ISOBMFF 캡슐화는 단일 트랙 캡슐화에 해당하며, 이는 인공지능경망 내부 데이터의 성질이 유사하여 여러 개의 trak 박스를 만들 필요성이 떨어지기 때문이다.

본 논문에서 제안하는 캡슐화는 그림 4와 같이 구성된다. ISOBMFF 표준에 맞게 ftyp, moov, 그리고 mdat 박스를 먼저 만들어주었으며, moov 박스 안에 trak 박스를 추가하였다. 그림 4에는 생략되어 있으나 trak 박스 내부에 ISOBMFF 표준에서 요구하는 박스를 전부 생성하였으며, 이후 stsd 박스 하위에 NNR 비트스트림의 메타데이터를 담기 위한 sample entry 박스를 추가하였다. Sample entry 박스의 명칭은 “nnr1”이며, nnr1 하위에는 nnrC(NNR Configuration) 박스를 생성하여 NNR\_NDU를 제외한 모든 종류의 NNR 유닛을 담았다. 이는 NNR\_MPS 등의 NNR 유닛이 압축된 인

공신경망의 메타데이터를 담고 있기 때문이며, 압축된 인공신경망 데이터에 해당하는 NNR\_NDU는 moov 박스가 아닌 mdat 박스 내부에 추가하였다. 비디오 데이터에서 한 장의 프레임이 주로 하나의 샘플(sample)이라면, 제안 방안에서는 하나의 NNR\_NDU 유닛을 하나의 샘플로 정의하였다. 각 샘플의 크기는 moov 박스 내부에 저장되어 있기에 개별적으로 각 NNR\_NDU 유닛에 접근할 수 있다.

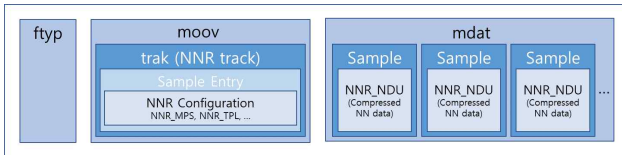


그림 4. NNR 비트스트림 단일 트랙 캡슐화 파일 구조

#### 4. 실험 결과

본 장에서는 앞선 3장에서 제안한 NNR 비트스트림의 ISOBMFF 캡슐화 방안을 실제로 구현하여 실험한 결과를 확인한다. 실험은 Ubuntu 20.04 환경에서 C++ 및 Python을 통해 진행되었으며, 입력 비트스트림은 슈퍼레졸루션(Super Resolution) 모델 중 하나인 EDSR(Enhanced Deep Super Resolution)[4] 네트워크를 MPEG NNC version1 표준의 레퍼런스 소프트웨어(Reference Software)인 NCTM(Network Coding Test Model) 6.1을 통해 압축한 비트스트림이다.

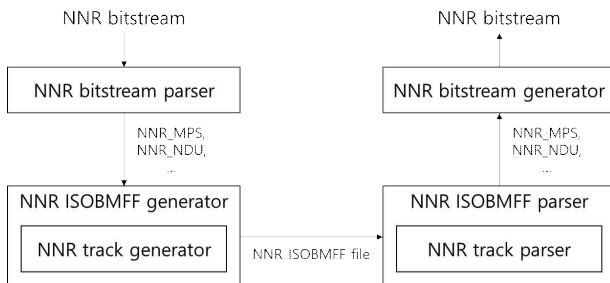


그림 5. NNR ISOBMFF 생성 및 검증 구조도

본 논문에서 제안한 NNR 비트스트림의 ISOBMFF 캡슐화 및 검증 구조도는 그림 5와 같다. ISOBMFF 파일을 생성하기 위해 NNR 비트스트림을 먼저 NNR 유닛 단위로 파싱(parsing)한 뒤, 유닛들을 3장에서 제안한 방식과 같이 배치하여 캡슐화를 진행하였다. ISOBMFF 캡슐화 과정에서 데이터의 손실이 없음을 검증하기 위해 NNR ISOBMFF 파서(parser)를 활용하여 비트스트림을 복원한 뒤 입력 비트스트림과 비교하였다.

생성된 NNR ISOBMFF의 내부 구성은 그림 6과 같으며, MP4 리더(MP4 reader)를 통해 정상적으로 읽혔으므로 ISOBMFF 표준에 맞게 설계되었음을 또한 확인할 수 있다. 그림 6에서는 nnr1 하위의 nnrC 박스가 보이지 않는데, 이는 nnr1과 nnrC가 ISOBMFF의 필수 박스가 아니기 때문이다.

No.	Sample delta	Size	Chunk num	Chunk offset	I-Frame	Sample offset
1	0	8	1	2056	-	2056
2	0	6	2	2064	-	2064
3	0	40	3	2070	-	2070
4	0	40	4	2110	-	2110
5	0	40	5	2150	-	2150
6	0	40	6	2190	-	2190
7	0	588	7	2230	-	2230
8	0	31	8	2818	-	2818
9	0	88	9	2849	-	2849
10	0	34	10	2937	-	2937
11	0	88	11	2971	-	2971

그림 6. NNR ISOBMFF 내부 구성



그림 7. EDSR 슈퍼레졸루션 태스크 수행 결과

검증 실험 결과는 그림 7과 같다. 복원된 NNR 비트스트림이 정성적으로 복호화되어 슈퍼레졸루션 태스크(task)를 문제없이 수행하는 것을 확인할 수 있다. 또한 슈퍼레졸루션 결과물들의 PSNR(Peak Signal-to-Noise Ratio) 및 SSIM(Structural Similarity Index Measure)[5]이 동일한 것을 통해 캡슐화 과정에서 데이터 손실이 없었음을 알 수 있다.

#### 5. 결론

본 논문에서는 MPEG NNC 부호화기를 통해 생성한 NNR 비트스트림 기반의 ISOBMFF 캡슐화 방안을 제안하고, 이를 실제로 구현하여 실험을 진행하였다. 또한, 실험 결과를 검증하기 위하여 캡슐화된 파일을 NNR 비트스트림으로 복원한 뒤, 복호화 후 태스크 수행을 통해 데이터의 손실 없이 캡슐화가 진행되었음을 확인하였다.

#### ACKNOWLEDGMENT

본 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신 기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00802, 속성을 유지하는 지능적 미디어 화면비 변환 기술 개발)  
This research was supported by the This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2021-0-02046) supervised by the IITP and the Institute of Information & communications Technology Planning & evaluation (IITP) (Grant number: 2020-0-00452).

---

## 참 고 문 헌

- [1] Seongbae Rhee, Kyuheon Kim, "A Study on the Application of Super Resolution in a 360-degree Video based Surveillance System" Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Korea, pp.292-293, 2022
- [2] H. Kirchhoffer et al., "Overview of the Neural Network Compression and Representation (NNR) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 5, pp. 3203-3216, May 2022
- [3] ISO/IEC 14496-12, "Coding of audio-visual objects – Part 12: ISO base media file format", February 2015
- [4] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," 2nd NTIRE: New Trends in Image Restoration and Enhancement workshop and challenge on image super-resolution in conjunction with CVPR 2017
- [5] Chaofeng Li et al., "Content-partitioned structural similarity index for image quality assessment," Signal Processing: Image Communication Volume 25, Issue 7, pp.517-526, 2010