Journal of Korea Multimedia Society Vol. 25, No. 6, June 2022(pp. 783-792)
https://doi.org/10.9717/kmms.2022.25.6.783

다목적 비디오 부/복호화를 위한 다층 퍼셉트론 기반 삼항 트리 분할 결정 방법

이태식†, 전동산††

Multi-Layer Perceptron Based Ternary Tree Partitioning Decision Method for Versatile Video Coding

Taesik Lee†, Dongsan Jun††

ABSTRACT

Versatile Video Coding (VVC) is the latest video coding standard, which had been developed by the Joint Video Experts Team (JVET) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) in 2020. Although VVC can provide powerful coding performance, it requires tremendous computational complexity to determine the optimal block structures during the encoding process. In this paper, we propose a fast ternary tree decision method using two neural networks with 7 nodes as input vector based on the multi-layer perceptron structure, names STH-NN and STV-NN. As a training result of neural network, the STH-NN and STV-NN achieved accuracies of 85% and 91%, respectively. Experimental results show that the proposed method reduces the encoding complexity up to 25% with unnoticeable coding loss compared to the VVC test model (VTM).

Key words: Block Structure, Fast Encoding, Multi-Layer Perceptron, Neural Network, Ternary Tree, Versatile Video Coding

1. 서 론

최근 비디오 콘텐츠의 다양화 및 온라인 동영상 서비스 수요가 증대됨에 따라, 유무선 네트워크상에서의 동영상 스트리밍을 위한 데이터 트래픽이 증가하는 추세이다. 원본 영상 대비 주관적 화질을 최대한 유지하면서 주어진 네트워크 대역폭 내에서 전송에 요구되는 원본 데이터 량을 줄이기 위해서는 비디오 압축 기술이 필수적으로 요구된다. 현재 최신 비디오 압축 코덱인 다목적 비디오 코딩(Versatile Video Coding, VVC)[1]은 ISO/IEC Moving Picture Experts Group(MPEG)과 ITU-T Video Coding Experts Group(VCEG)에서 공동으로 결성한 Joint Video Experts Group(JVET)에 의해 2020년 표준 개발이 완료되었다.

VVC는 이전 비디오 코딩 표준인 고효율 비디오 코딩(High Efficiency Video Coding, HEVC)[2] 보다 약 2배의 높은 압축 성능을 달성한 반면, 새로운 부호화 기술들이 대거 채택되면서 부호화기의 복잡도가 HEVC 대비 최대 약 27배 증가하였다[3]. VVC에 채택된 기술들은, Quaternary Tree plus Multi-Type Tree(QTMTT), Position Dependent intra Prediction Combination(PDPC)[4], Cross Component Linear Model intra prediction(CCLM)[5], Wide

※ Corresponding Author : Dongsan Jun, Address: (49315) 37, Nakdong-daero 550 beon-gil, Saha-gu, Busan, Korea, TEL : +82-51-200-5823, FAX : +82-53-200-7783, E-mail : dsjun@dau.ac.kr
Receipt date : Mar. 28, 2022, Revision date : May 22, 2022
Approval date : Jun. 13, 2022

† Department of Computer Engineering, Dong-A University (E-mail : tslee@donga.ac.kr)
†† Department of Computer Engineering, Dong-A University
※ This work was supported by the Dong-A University research fund.

Angular Intra Prediction(WAIP)[6], Intra Sub-Partitioning(ISP)[7], Matrix weighted Intra Prediction(MIP)[8], 그리고 다양한 화면 간 예측 기술[9] 등이 있다. 일반적으로 연산성능 및 메모리 사양이 제한적인 저복잡도 환경의 모바일 디바이스에서 대용량의 고해상도 영상을 실시간 또는 고속으로 부호화 및 스트리밍 서비스하기 위해서는 비디오 코덱의 부호화기 복잡도를 줄이는 것이 필수적으로 요구된다.

본 논문에서는 다층 퍼셉트론(Multi-layer Perceptron, MLP) 구조에 기반한 7개의 노드를 입력층으로 가지는 두 개의 신경망(Neural Network, NN)을 설계하여, VVC 표준에서 지원하는 삼항 트리(Ternary Tree, TT) 수행 여부에 대한 조기 결정 방법을 제안한다. 본 제안 방법은 VVC 참조 소프트웨어인 VVC Test Model(VTM)에 구현하였으며, JVET 공통실험조건(Common Test Condition, CTC)에 따라 부호화기 복잡도 감소와 이에 따른 손실 왜곡을 평가하였다[10].

본 논문의 구성은 다음과 같다. 2장에서 VVC 블록 분할 구조 및 부호화 절차에 대해 소개한 후, 3장에서 제안하는 심층신경망 기반 삼항 트리 분할 결정 기법에 대하여 설명한다. 4장에서 제안 방법의 실험 결과를 보여준 후, 5장에서 결론을 맺는다.

2. 기존 방법

본 장에서는 영상 압축 기술 방법 중 하나인 부호화 블록 구조(Block Structure)의 분할 구조 및 절차에 대해 기술한다. 영상은 다수의 픽처로 구성되어 있으며, 하나의 픽처는 가로 및 세로 크기가 동일한 Coding Tree Unit(CTU)으로 분할되어 부/복호화가 수행된다. HEVC 표준에서 제정된 최대 및 최소 CTU 크기는 64×64 및 8×8을 지원하며, 이때 최소 CU 크기는 4×4로 제한되어 있다. HEVC에서 하나의 CTU는 쿼드 트리(Quad Tree, QT)구조에 따라 재귀적으로 정사각형 형태로 분할되는데, 이때 최종 분할된 하나의 블록을 코딩 유닛(Coding Unit, CU)이라 명명하고 예측 및 변환을 독립적으로 수행하게 된다. 하나의 CU가 예측 및 변환 과정을 수행하게 될 때, 현재 부호화 되는 CU 내에서 예측 유닛(Prediction Unit, PU)과 변환 유닛(Transform Unit, TU)으로 각각 분할 될 수 있으며, 율-왜곡 관점에서 예측과 변환에 대한 최적의 분할 구조를 결정할 수 있도록 부호화 과정을 수행한다.

VVC 표준에서는 최대 CTU 크기는 128×128, 최소 CTU 크기는 16×16을 지원하며, 이때 최소 CU 크기는 HEVC와 동일하게 4×4로 제한된다. HEVC 대비 VVC는 하나의 CTU가 정사각형 형태의 QT 분할 뿐만 아니라 이진 트리(Binary Tree, BT) 혹은 삼항 트리(Ternary Tree, TT)구조와 같은 직사각형 형태로 분할이 가능한 멀티타입 트리(Multi-type Tree, MTT) 구조를 지원한다. Fig. 1은 VVC CU 분할 구조를 나타내며, 이때 하나의 CU는 쿼드 트리 분할(SPLIT_QT), 이진 수직 분할(SPLIT_BT_VER), 이진 수평 분할(SPLIT_BT_HOR), 삼항 수직 분할

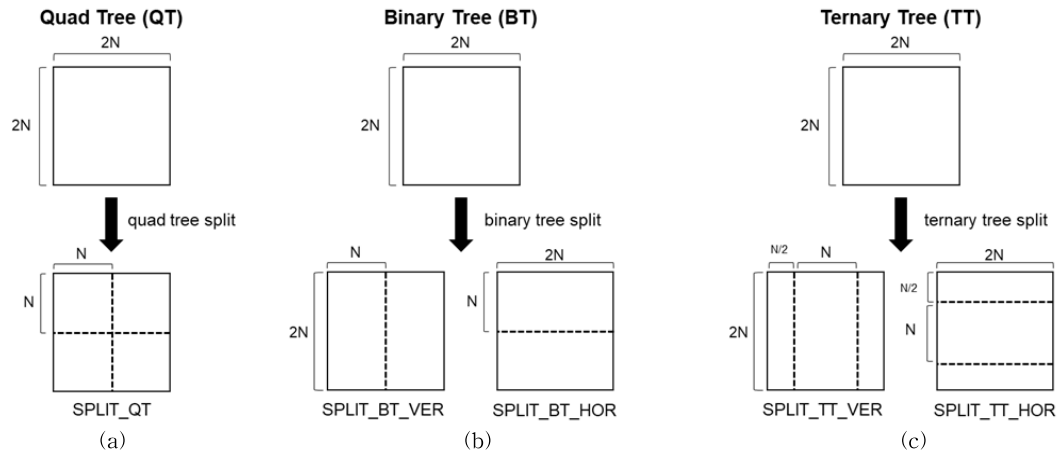

Fig. 1. VVC block structure consisting of a quad tree and a multi-type tree. (a) QT split, (b) BT split, and (c) TT split.

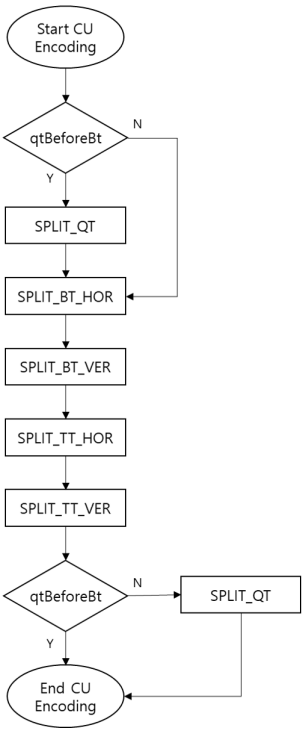

Fig. 2. The CU encoding process of VVC.

(SPLIT_TT_VER) 및 삼항 수평 분할(SPLIT_TT_HOR) 구조에 따라 독립적인 부/복호화가 가능하도록 표준에서 정의하고 있다.

Fig. 2는 VTM에 구현된 CU 최적 분할구조를 결정하기 위한 부호화 과정을 보여주고 있으며, VTM에 설정된 qtBeforeBt 플래그에 따라 최적 블록 분할 결정을 위한 부호화 과정이 달라지게 구현되어 있다. qtBeforeBt 플래그는 현재 부호화되는 CU를 기준으로 부/복호화가 종료된 상단 및 좌측 CU의 QT 분할 깊이가 현재 CU의 QT 분할 깊이보다 깊은 경우 및 부호화가 진행되는 CU의 넓이가 16보다 큰 경우를 모두 만족할 때 해당 플래그가 참으로 설정되며, 이때 QT 구조 분할을 먼저 수행하게 된다.

3. 제안하는 방법

VVC에서 현재 부호화되는 CU는 최적의 블록 구조를 찾기 위해서 Fig. 1에 도시된 모든 분할 구조에 대한 율-왜곡 최적화(Rate Distortion Optimization, RDO) 과정을 수행한다. 본 논문에서는 VVC 블록 분할 과정 중 생기는 부호화기 복잡도를 줄이기 위해, 수평 및 수직 방향 TT 분할에 대한 RDO 과정을 조기 결정할 수 있는 MLP 기반 2개의 신경망을 제안하였으며, 수평 및 수직 방향 TT 분할을 위한 신경망을 SPLIT_TT_HOR-Neural Network(STH-NN) 및 SPLIT_TT_VER-Neural Network(STV-NN)으로 명명하였다. 제안된 신경망은 0과 1사이의 값을 최종 출력하도록 설계하였으며, 이때 0에 가까운 값을 가질수록 해당 TT 분할에 대한 부호화 과정을 생략(Bypass) 하도록 설정하였다.

제안하는 신경망의 입력 특성을 도출하기 위해, CU간 계층구조를 구분하기 위한 Parent 및 Current CU를 Fig. 3에서 정의하였다. 본 논문에서 정의한 Parent CU는 정사각형 형태의 QT 노드이거나 또는

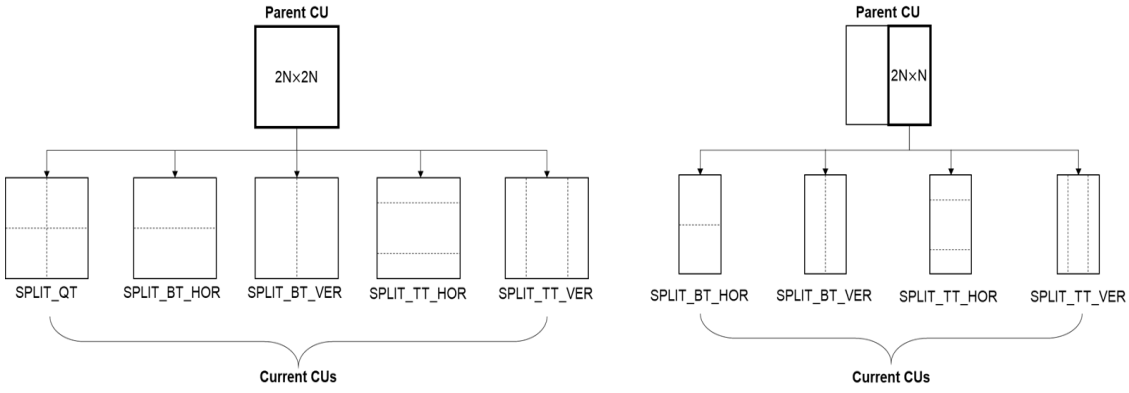

Fig. 3. Relation between parent CU and current CU. (a) Example of QT as a parent CU and (b) Example of MTT as a parent CU.

Current CU 영역을 포함하는 정사각형 혹은 직사각형 형태의 MTT 노드일 수 있다. 예를 들어, Fig. 3(a)와 같이 분할된 Current QT, BT 및 TT CU는 2N×2N의 동일한 Parent CU를 가지며, 이와 유사하게 Fig. 3(b)와 같이 분할된 BT 및 TT CU는 해당 CU를 포함하는 2N×N의 Parent CU를 가질 수 있도록 정의하였다.

Algorithm 1은 제안하는 신경망에서 입력으로 사용되는 7가지 입력 특성(Input feature)의 추출과정

Algorithm 1. Input Feature Extraction

```
FeatureExtraction
{
IPC = intra prediction mode / 66
if CU split == TT_H then
    RBS = Height / (Width + Height)
    RND = horizontal / (horizontal+vertical)
    if Optimal CU == BT_H then OBD = 1
    else OBD = 0
    endif
    if QT Cost > BT_H Cost then TTI += 0.5
    endif
    if QT Cost > BT_V Cost then TTI += 0.5
    endif
    if ISP split == horizontal then IEP = 1
    else IEP = 0
    endif
    if Left CU split == horizontal then DNB += 0.5
    endif
    if Above CU split == horizontal then DNB += 0.5
    endif
    if CU split == TT_V then
    RBS = Width / (Width + Height)
    RND = vertical / (horizontal+vertical)
    if Optimal CU == BT_V then OBD = 1
    else OBD = 0
    endif
    if QT Cost > BT_H Cost then TTI += 0.25
    endif
    if QT Cost > BT_V Cost then TTI += 0.25
    endif
    if QT Cost > TT_H Cost then TTI += 0.5
    endif
    if ISP split == vertical then IEP = 1
    else IEP = 0
    endif
    if Left CU split == vertical then DNB += 0.5
    endif
    if Above CU split == vertical then DNB += 0.5
endif
}
```

에 대한 수도코드를 보여준다. 이때 제안된 7가지 입력 특성들은 Parent CU에서 유도되는 정보 및 부호화 과정 중에서 획득할 수 있는 정보를 0과 1사이의 실숫값으로 정량화하였으며, 하나의 입력 벡터로 구성한 후 제안하는 신경망의 입력으로 사용하였다. Ratio of Block Size(RBS)는 Parent CU의 넓이와 높이 비를 값으로 정량화하여 입력 특성으로 사용하며, 현재 CU의 높이가 넓이보다 큰 경우 수평 방향 블록 분할에 대한 가중치를 주기 위해 STH-NN에 대한 입력 특성 값이 1에 가까워지도록 설계하였다. Optimal BT Direction(OBD)은 부호화 과정 중 BT 최적 분할 방향이 TT 분할 방향과 동일한지 여부를 부울 값으로 정량화한 입력 특성을 나타내며, BT 최적 분할 방향이 수평으로 분할되었을 경우 STH-NN에 대한 입력 특성이 1의 값을 가지도록 설계하였다. Ratio of the Number of Directions(RND)는 TT 분할 전 최적의 분할 구조에 포함되어 있는 CU들의 수평 및 수직 방향 발생 횟수에 대한 비(Ratio)를 나타내며, 수평 방향에 대한 분할이 많은 경우 STH-NN에 대한 입력 특성 값이 1에 근접한 값을 가질 수 있도록 정의하였다. TT split Indication(TTI)은 부호화 과정 중 얻어지는 QT, BT 및 TT 수평 방향 율-왜곡 비용(RD Cost)을 비교하여 신경망의 입력 특성 값을 설정하였으며, 이때 TT 수평 및 수직 분할 방향에 따라 비교되는 분할 구조는 다르게 설계하였다. STH-NN은 BT와 QT RD Cost를 비교하여 BT RD Cost가 QT 보다 낮은 경우 입력 특성 값이 0.5씩 증가되도록 설정하였고, STV-NN은 BT 및 TT 수평 방향과 QT RD Cost를 비교하여 BT RD Cost가 QT 보다 낮은 경우 입력 특성 값이 0.25씩 증가되고 TT 수평 방향 RD Cost가 QT 보다 낮은 경우 입력 특성 값은 0.5 증가되도록 설정하였다. ISP in the Encoding Process(IEP)는 Parent CU의 화면내 예측 과정에서 ISP 분할 방향이 TT 분할 방향과 동일한지 여부를 확인하는 지표로서, ISP의 분할 방향과 TT 분할 방향이 동일한 경우 입력 특성 값을 1로 정량화하였다. 참고로, ISP는 CU 크기에 따라 2개 혹은 4개의 균일한 크기를 가지는 수직 또는 수평 방향으로 분할하여 독립적인 변환과 복원 과정을 수행하는 화면내 예측 모드이다. Direction of the Neighboring Block(DNB)은 Parent CU 기준으로 상단과 좌측 CU 분할 방향 정보를 유도하여, 수평 방향으로 분할

되어 있을 경우 STH-NN에 대한 입력 특성 값은 0.5씩 더해지도록 설정하였다. 마지막으로 IPC는 Parent CU의 화면내 예측 모드(Intra Prediction Mode, IPM) 값을 66으로 나누어 신경망의 입력으로 사용하여, 부호화 과정에서 유도되는 화면내 예측 모드 정보를 TT 분할 여부에 대한 결정 과정에서 반영될 수 있도록 제안하였다.

Fig. 4는 TT 수평 및 수직 방향에 대한 부호화 수행 여부를 결정하기 위해 MLP를 이용하여 설계한 신경망을 도시하고 있으며, 이때 STH-NN 및 STV-NN은 가중치 행렬을 기준으로 총 3개의 전결합 층으로 구성하였다. STH-NN 및 STV-NN 신경망은 입력층, 첫 번째 은닉층, 출력층 노드를 각각 7개, 60개, 1개로 동일하게 설계한 반면, STH-NN 및 STV-NN에 대한 두 번째 은닉층 노드의 개수는 60개, 75개로 서로 다르게 설계하였다. 신경망의 최종 출력은 0과 1사이의 실숫값을 가질 수 있으며, 설정한 임곗값에 따라 TT 분할에 대한 수행 여부를 결정한다. 본 논문에서 사용된 임곗값은 0.5로 설정하여 출력 값이 1인 경우 TT 방향 분할에 대한 부호화를 수행하고 반대로 0인 경우 조기종료 하도록 학습시켰다.

Fig. 5는 VTM 내에 제안 방법이 적용된 CU 부호화 과정을 보여주며, 수평 및 수직방향 TT 분할 조기결정을 위한 모듈은 회색 블록으로 표현하고 있다. 제안 방법은 SPLIT_BT_VER 부호화 과정이 종료된 이후 진행되며, SPLIT_TT_HOR 수행여부를 결정한 후 SPLIT_TT_VER 수행여부를 결정한다. 먼저, STH-NN의 입력층으로 사용되기 위한 7가지 입력 특성들을 추출한 후, STH-NN을 통해 0과 1사이 최종 출력값을 획득한다. 그런 다음, 설정한 임곗값과 출력값을 비교하여 SPLIT_TT_HOR 수행여부를 결정한다. 이어서, SPLIT_TT_VER 수행여부를 결정하기 위한 과정이 진행되며, 이 과정은 SPLIT_TT_HOR 수행여부를 결정하는 과정과 동일하게 진행된다.

Table 1. Training environment of the proposed network.

Settings	Options
Training sequence	BVI-DVC
Number of sequence	19
Resolution of sequence	3840×2160
Number of training datasets for STH-NN	533,670
Number of training datasets for STV-NN	552,775
Quantization parameter	30
Framework	Keras 2.2.4

4. 실험 결과

제안 방법은 Python 3.6.8에 기반한 Keras 2.2.4버전의 라이브러리를 사용하여 VTM 13.0[11]에구현하였으며, Table 1은 구체적인 실험환경을 보여주고 있다. 제안하는 2개의 신경망을 학습하기 위해, 3840

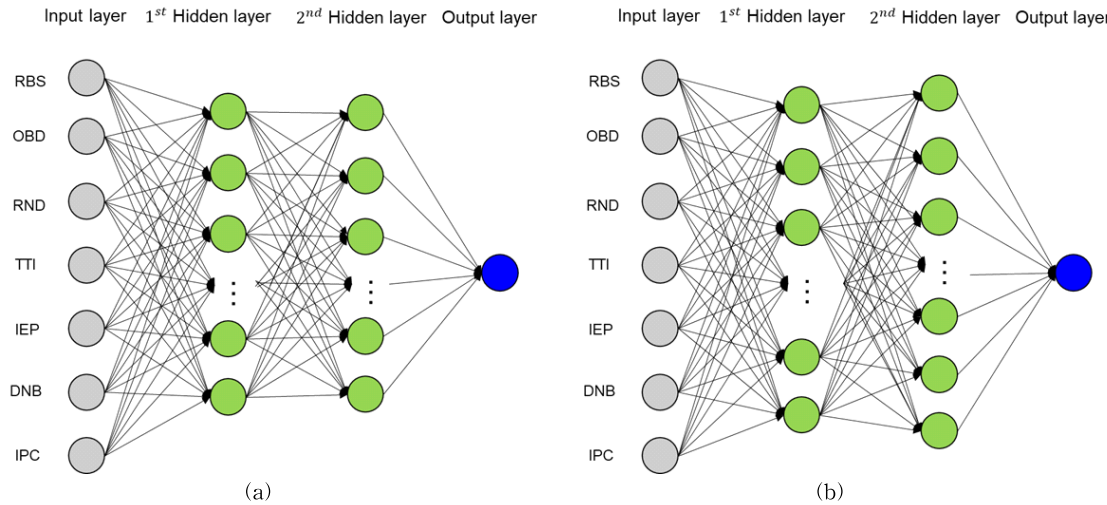

Fig. 4. Proposed network architectures. (a) STH-NN and (b) STV-NN.

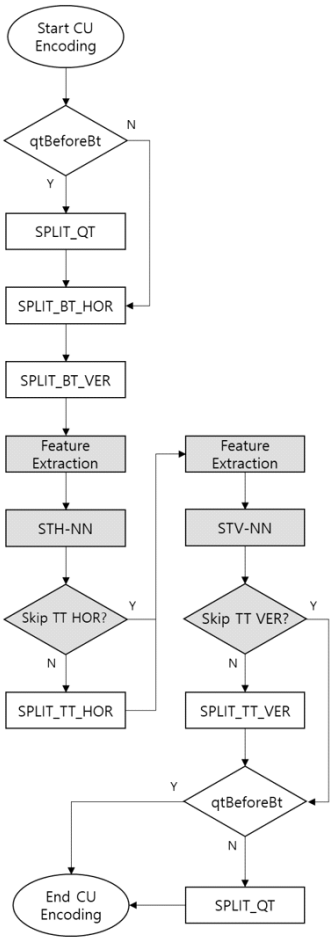

Fig. 5. Proposed CU encoding procedures.

×2160의 해상도를 가지는 19개의 영상으로 구성된 BVI-DVC[12] 데이터셋을 사용하였으며, JVET CTC에 명시된 실험 환경과 차이를 주기 위해 양자화 매개변수를 30으로 설정하였다. 제안하는 STH-NN 및 STV-NN 신경망을 학습하기 위해, 훈련용 데이터셋의 부호화 과정에서 다양한 CU 크기를 가지는 블록을 수집하였으며, 이때 STH-NN 및 STV-NN 학습을 위해 각각 533,670개 및 552,775개의 CU 블록들을 사용하였다. 참고로 훈련용 데이터 수집 시, 각 데이터 세트의 정답 비율은 TT 분할이 되는 경우 및 분할이 되지 않는 경우에 대하여 50%의 비율로 조정하여 추출하였으며, 검증용 데이터 세트는 훈련용 데이터 세트에서 무작위로 20%를 추출하여 사용하였다.

Table 2는 제안한 신경망의 훈련 시 사용된 하이퍼 파라미터를 보여준다. 최종 출력층을 제외한 모든 은닉층의 활성화 함수는 ReLU를 사용하였고, 최종 출력층은 0과 1사이의 실숫값을 출력하기 위해 Sigmoid 함수를 사용하였다. 이때 배치(Mini-batch) 크기, 학습률, 최적화 기법은 각각 128, 0.01, 확률적 경사 하강법으로 설정하였으며, 초기 네트워크 가중치는 Xavier[13] 초기화하였다. 최적화된 모델의 매개변수를 찾기 위해 500번의 에폭(epoch) 만큼 학습을 반복하였으며, 학습 과정 중 평균 제곱 오차(Mean Squared Error, MSE)를 손실 함수로 사용하였다.

Table 2. Hyper parameters of the proposed network.

Hyper Parameters	Options
Optimizer	Stochastic Gradient Descent
Activation function	ReLU to Sigmoid
Loss function	Mean Squared Error
Learning rate	0.01
Initial weight	Xavier
Number of epochs	500
Batch size	128

Fig. 6은 제안한 신경망의 훈련 결과와 검증 결과를 나타내며, 각각 정확도와 손실함수를 사용해 측정하였다. 참고로 검증 결과는 부호화기 복잡도 및 손실 왜곡에 영향을 미치므로 높은 정확도와 낮은 손실을 달성할 수 있도록 신경망을 설계하였다. 제안하는 STH-NN의 검증 정확도는 85%, 검증 손실은 0.10을 달성하였으며, STV-NN의 검증 정확도는 91%, 검증 손실은 0.06을 달성하였다.

최종 제안하는 신경망을 도출하기 위해, 입력 특성의 개수, 은닉층의 수, 그리고 각 은닉층의 노드 수에 대한 Tool-off 실험을 진행하였다. Table 3은 제안하는 신경망의 은닉층의 수 그리고 은닉층의 노드 수를 변경하여 진행한 실험 결과를 보여주며, STH-NN의 경우 단일 신경망을 사용 하였을 때 검증 정확도와 검증 손실은 각각 84.48%와 0.106의 성능을 나타낸 반면 제안하는 신경망은 검증 정확도와 검증 손실에서 84.98%와 0.102의 성능을 달성하였다. STV-NN 또한 제안하는 신경망 구조가 검증 정확도와 검증 손실 측면에서 가장 좋은 성능을 달성하였다. Table 4는 7개의 입력 특성을 사용하였을 때 정확도와 손실에 대하여 측정한 다음 7개 입력 특성 중 하나를 생략하는 방식으로 진행한 실험 결과를 보여준다. 실험 결과 7개의 입력 특성을 모두 사용하였을 때 가장

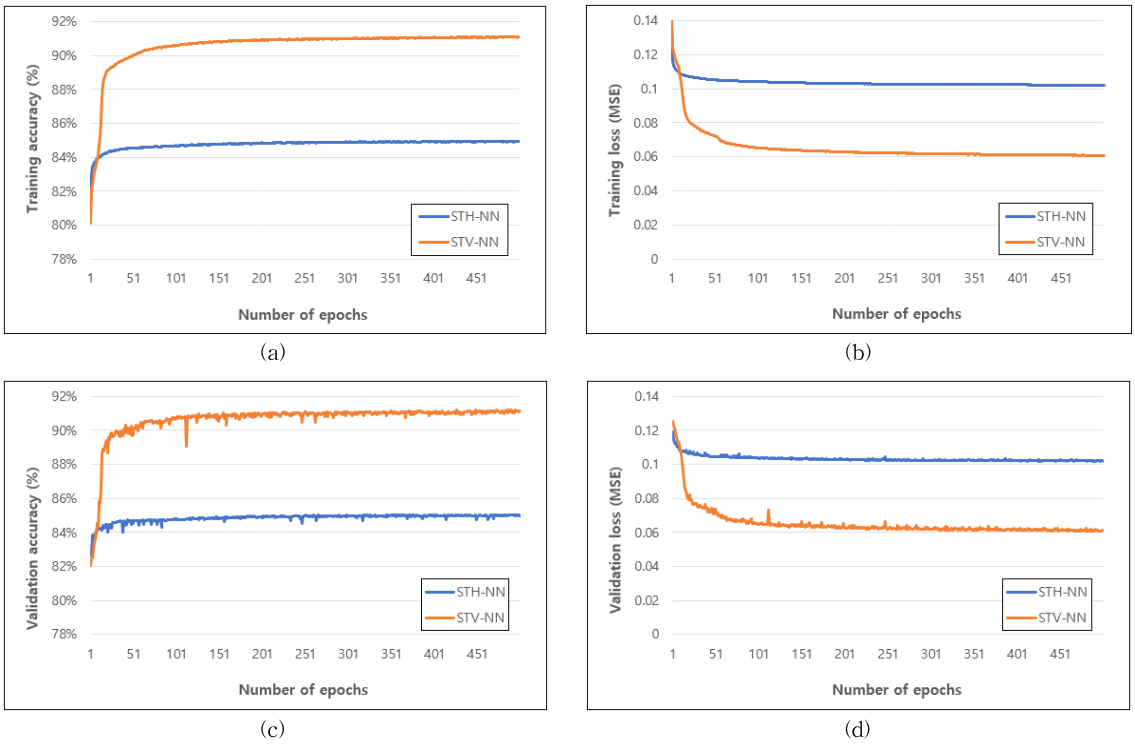

Fig. 6. Training results of proposed network. (a) Training accuracy, (b) Training loss, (c) Validation accuracy, and (d) Validation loss.

Table 3. Verification of the numbers of hidden layers and nodes in the proposed network.

Category	STH-NN (7×60×60×1)				STV-NN (7×60×75×1)			
	Training datasets		Validation datasets		Training datasets		Validation datasets	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
7×60×1	0.106	84.41%	0.106	84.48%	0.077	89.47%	0.079	89.31%
7×60×15×1	0.103	84.81%	0.103	84.76%	0.066	90.70%	0.066	90.75%
7×60×30×1	0.102	84.88%	0.104	84.72%	0.062	90.99%	0.062	90.96%
7×60×45×1	0.102	84.86%	0.103	84.63%	0.063	90.88%	0.063	90.96%
7×60×60×1	0.102	84.94%	0.102	84.98%	0.061	91.17%	0.063	90.86%
7×60×75×1	0.102	84.92%	0.102	84.96%	0.061	91.10%	0.061	91.14%

높은 검증 정확도와 낮은 검증 손실을 달성하였으며, 특히 RND 입력 특성이 제안한 신경망 구조에서 효과적인 입력 특성으로 사용되는 것을 확인하였다.

테스트 단계에서 제안하는 신경망은 Intel Xeon Gold 6138 40-cores 2.00 GHz, 256 GB RAM, 64-bit Window server 2016 환경에서 수행되었다. 최종 성능실험에 사용된 영상은 JVET CTC에 정의된 클래스 A(3,840×2,160) 6종 및 B(1,920×1,080) 5종을 사용하였고, All Intra(AI) 환경에서 VTM 13.0과 비교하여 복잡도 및 화질 열하를 측정하였다. 손실 왜곡을 평가하기 위해 JVET 표준화에서 사용하는 Bjontegaard Delta Bit Rate(BDBR) 측정값을 동일하게 측정하였으며, 이때 BDBR의 증가는 코딩 손실을 의미하고 반대인 경우 코딩 이득을 의미한다. 식 (1)은 JVET CTC에서 권장하는 테스트 영상에 대한 Y, U, 및 V 성분에 대한 가중치가 부여된 평균 BDBR을

Table 4. Tool-off tests on the proposed network input features.

Category	STH-NN (7×60×60×1)				STV-NN (7×60×75×1)			
	Training datasets		Validation datasets		Training datasets		Validation datasets	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
All features	0.102	84.94%	0.102	84.98%	0.061	91.10%	0.061	91.14%
DNB tool-off	0.102	84.90%	0.104	84.68%	0.061	91.16%	0.061	91.02%
IEP tool-off	0.103	84.73%	0.102	84.88%	0.063	90.85%	0.063	90.79%
IPC tool-off	0.104	84.76%	0.103	84.83%	0.063	90.97%	0.063	90.97%
OBD tool-off	0.104	84.74%	0.104	84.63%	0.062	91.03%	0.062	91.00%
RBS tool-off	0.119	82.88%	0.119	82.90%	0.072	89.75%	0.073	89.55%
RND tool-off	0.171	74.79%	0.170	74.95%	0.173	74.14%	0.174	74.17%
TTI tool-off	0.109	83.64%	0.109	83.72%	0.073	89.69%	0.075	89.44%

나타내며, 가중치에 따라 Y 성분의 중요도가 크게 반영된다는 것을 알 수 있다. 식 (2)는 부호화기 복잡도 감소를 측정하기 위해 JVET CTC 상 명시된 4개의 양자화 파라미터에서 측정한 평균 복잡도 감소율을 나타내며, 여기서 T_{org} 및 $T_{proposed}$는 VTM 13.0 및 제안 방법이 적용된 부호화 시간을 측정한 값이다.

$$BDBR_{YUV} = \frac{6 \times BDBR_Y + BDBR_U + BDBR_V}{8} \quad (1)$$

$$T_{enc} = \frac{1}{4} \sum_{QP_i \in 22,27,32,37} \frac{T_{org}(QP_i) - T_{proposed}(QP_i)}{T_{org}(QP_i)} \quad (2)$$

Table 5는 식 (1) 및 식 (2)를 통해 제안 방법의 실험 결과를 보여주고 있으며, VTM 13.0 대비 $BDBR_{YUV}$ 측면에서 평균 0.32%의 손실 왜곡 및 부호화기 복잡도가 평균 25% 감소되는 것을 보여준다. 특히, 클래스 A의 경우 클래스 B보다 부호화기 복잡도가 평균 1.2% 개선된 것을 보여주고 있으며, 이는 제안하는 신경망 훈련 시 사용한 데이터 세트의 해상도와 연관성이 있음을 알 수 있다. 또한, Table 6은 제안하는 방법과 관련된 연구의 실험 결과이며, 관련 연구는 신경망을 사용하지 않고 BT와 TT사이의 분할 방향 정보에 따른 상관관계를 이용하여 컨텍스트 기반 빠른 TT 결정 방법을 제안한 연구이다[14]. 제안하는 방법은 관련 연구에 비해 부호화기 복잡도를 평균 11% 더 감소 시켰으며, 신경망을 사용하였을 때 더 좋은 복잡도 감소효과를 나타내는 것을 확인하였다.

Table 5. Coding performance of proposed method over VTM 13.0.

Class	Sequences	$BDBR_Y$	$BDBR_U$	$BDBR_V$	$BDBR_{YUV}$	T_{enc}
A	Tango2	0.26%	-0.12%	0.07%	0.19%	28%
	FoodMarket4	0.28%	0.15%	0.03%	0.23%	23%
	Campfire	0.33%	0.05%	0.16%	0.28%	24%
	CatRobot	0.45%	0.21%	0.19%	0.38%	23%
	DaylightRoad2	0.52%	0.39%	0.29%	0.48%	26%
	ParkRunning3	0.24%	0.18%	0.18%	0.23%	26%
B	MarketPlace	0.29%	0.16%	0.04%	0.24%	26%
	RitualDance	0.44%	0.25%	0.21%	0.39%	24%
	Cactus	0.43%	0.18%	0.34%	0.39%	26%
	BasketballDrive	0.42%	0.17%	0.12%	0.35%	23%
	BQTerrace	0.37%	0.38%	0.28%	0.36%	20%
Average		0.37%	0.18%	0.17%	0.32%	25%

Table 6. Coding performance of related work over VTM 13.0.

Class	Sequences	$BDBR_Y$	$BDBR_U$	$BDBR_V$	$BDBR_{YUV}$	T_{enc}
A	Tango2	0.17%	0.09%	0.30%	0.17%	14%
	FoodMarket4	0.16%	0.14%	0.14%	0.15%	16%
	Campfire	0.27%	0.35%	0.58%	0.32%	13%
	CatRobot	0.36%	0.56%	0.49%	0.40%	15%
	DaylightRoad2	0.44%	0.99%	0.77%	0.55%	14%
	ParkRunning3	0.15%	0.35%	0.32%	0.19%	13%
B	MarketPlace	0.18%	0.44%	0.28%	0.23%	15%
	RitualDance	0.33%	0.56%	0.41%	0.37%	15%
	Cactus	0.38%	0.49%	0.71%	0.44%	15%
	BasketballDrive	0.39%	0.66%	0.60%	0.45%	14%
	BQTerrace	0.34%	0.89%	0.94%	0.48%	14%
Average		0.29%	0.50%	0.50%	0.34%	14%

5. 결 론

본 논문에서는 VVC 부호화기의 복잡도를 줄이기 위해 다층 퍼셉트론 구조에 기반한 7개의 노드를 입력 벡터로 가지는 두 개의 신경망을 설계하여, VVC 부호화 과정에서 TT 분할 여부에 대해 조기 결정하는 방법을 제안하였다. 제안된 2개의 신경망 각각은 Parent CU에서 도출되는 정보 및 부호화 과정 중에서 획득할 수 있는 부호화 정보들을 활용하여 7개 입력 특성들을 하나의 입력 벡터로 정의한 후, TT 분할 수행 여부를 결정하도록 신경망을 설계하였다. 신경망 훈련 결과 STH-NN 및 STV-NN 신경망은 수평방향 및 수직방향 분할 여부에 대해 각각 85% 및 91%의 정확도를 달성하였으며, 실험 결과 제안한 방법은 VTM 13.0의 AI 환경에서 평균 0.32%의 손실 왜곡으로 부호화기 복잡도를 평균 25% 감소시키는 것을 확인하였다.

REFERENCE

[1] B. Bross, Y.K. Wang, Y. Ye, S. Liu, J. Chen, and G.J. Sullivan, et al., "Overview of the Versatile Video Coding (VVC) Standard and Its Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3736-3764, 2021.

[2] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649-1668, 2012.

[3] JVET, *AHG Report: Test Model Software Development (AHG3)*, JVET-W0003, v.1, 2021.

[4] JVET, *CE3: Simplified PDPC (Test 2.4.1)*, JVET-K0063, v.2, 2018.

[5] JVET, *CE:3 Tests of Cross-Component Linear Model in BMS1.0 (Test 4.1.8, 4.1.9, 4.1.10, 4.1.11)*, JVET-K0190, v.1, 2018.

[6] JVET, *CE3-Related: Wide-Angle Intra Prediction for Non-Square Blocks*, JVET-K0500, v.4, 2018.

[7] JVET, *CE3: Intra Sub-Partitions Coding Mode (Test 1.1.1 and 1.1.2)*, JVET-M0102, v.5, 2019.

[8] JVET, *CE3: Affine Linear Weighted Intra Prediction (CE3-4.1, CE3-4.2)*, JVET-N0217, v.1, 2019.

[9] Y. Choi and B. Kim, "A Review on Motion Estimation and Compensation for Versatile Video Coding Technology (VVC)," *Journal of Korea Multimedia Society*, Vol. 22, No. 7, pp. 770-779, 2019.

[10] JVET, *VTM Common Test Conditions and*

Software Reference Configurations for SDR Video, JVET-T2010, v.1, 2020.
[11] VTM 13.0, https://vcgit.hhi.fraunhofer.de/jvet/VVC Software_VTM/-/tags/VTM-13.0 (accessed March, 16, 2022).
[12] D. Ma, F. Zhang, and D. Bull, "BVI-DVC: A Training Database for Deep Video Compression," *arXiv Preprint*, arXiv:2003.13552, 2020.
[13] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," *Proceedings of Machine Learning Research*, pp. 249-256, 2010.
[14] S. Park and J. Kang, "Context-Based Ternary Tree Decision Method in Versatile Video Coding for Fast Intra Coding," *IEEE Access*, Vol. 7, pp. 172597-172605, 2019.

이 태 식

2022년 2월 경남대학교 정보통신공학과(공학사)
2022년 3월~현재 동아대학교 컴퓨터공학과(석사과정)
관심분야: 영상압축, 딥러닝 기반 영상처리, 딥러닝/머신러닝

전 동 산

2002년 2월 부산대학교 전자컴퓨터공학부(공학사)
2004년 2월 KAIST 전기및전자공학과(공학석사)
2011년 2월 KAIST 전기및전자공학과(공학박사)
2004년 5월 한국전자통신연구원(책임연구원)
2018년 3월 경남대학교 정보통신공학과 조교수
2021년~현재 동아대학교 컴퓨터공학과 조교수
관심분야: 영상압축, 지능형 영상처리, 머신러닝/딥러닝 등