

일반논문 (Regular Paper)

방송공학회논문지 제27권 제3호, 2022년 5월 (JBE Vol.27, No.3, May 2022)

<https://doi.org/10.5909/JBE.2022.27.3.332>

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

## 문맥적응적 화면내 예측 모델 학습 및 부호화 성능분석

문기화<sup>a)</sup>, 박도현<sup>a)</sup>, 김재곤<sup>a)†</sup>

# Context-Adaptive Intra Prediction Model Training and Its Coding Performance Analysis

Gihwa Moon<sup>a)</sup>, Dohyeon Park<sup>a)</sup>, and Jae-Gon Kim<sup>a)†</sup>

### 요 약

최근 딥러닝을 적용하는 비디오 압축에 대한 연구가 활발히 진행되고 있다. 특히, 화면내 예측 부호화의 성능 한계를 극복할 수 있는 방안으로 딥러닝 기반의 화면내 예측 부호화 기술이 연구되고 있다. 본 논문은 신경망 기반 문맥적응적 화면내 예측 모델의 학습 방법과 그 부호화 성능분석을 제시한다. 즉, 본 논문에서는 주변 참조샘플의 문맥정보를 입력하여 현재블록을 예측하는 기존의 합성곱 신경망(CNN: Convolutional Neural network) 기반의 화면내 예측 모델을 학습한다. 학습된 화면내 예측 모델을 HEVC(High Efficiency Video Coding)의 참조 소프트웨어인 HM16.19에 추가적인 화면내 예측모드로 구현하고 그 부호화 성능을 분석하였다. 실험결과 학습한 예측 모델은 HEVC 대비 AI(All Intra) 모드에서 0.28% BD-rate 부호화 성능 향상을 보였다. 또한 비디오 부호화 블록 분할 구조를 고려하여 학습한 경우의 성능도 확인하였다.

### Abstract

Recently, with the development of deep learning and artificial neural network technologies, research on the application of neural network has been actively conducted in the field of video coding. In particular, deep learning-based intra prediction is being studied as a way to overcome the performance limitations of the existing intra prediction techniques. This paper presents a method of context-adaptive neural network-based intra prediction model training and its coding performance analysis. In other words, in this paper, we implement and train a known intra prediction model based on convolutional neural network (CNN) that predicts a current block using contextual information from reference blocks. Then, we integrate the trained model into HM16.19 as an additional intra prediction mode and evaluate the coding performance of the trained model. Experimental results show that the trained model gives 0.28% BD-rate bit saving over HEVC in All Intra (AI) coding mode. In addition, the coding performance change of training considering block partition is also presented.

Keyword: Context-adaptive, CNN, Intra prediction, Block partition, HEVC

## 1. 서론

딥러닝 기술은 컴퓨터 비전 및 영상처리 분야에 적용되어 뛰어난 성능을 보이고 있으며, 최근 딥러닝 기반의 비디오 부호화에 대한 연구도 활발히 진행되고 있다. HEVC (High Efficiency Video Coding)<sup>[1]</sup> 이후 ITU-T VCEG (Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)으로 구성된 JVET(Joint Video Experts Team)에서는 VVC(Versatile Video Coding) 표준을 완료하였으며, 최근 NNVC (Neural Network based Video Coding) AhG(Ad-hoc Group)을 두고 딥러닝 기반의 비디오 부호화 기술 탐색을 진행하고 있다<sup>[2][3]</sup>. 현재 NNVC에서 논의되고 있는 딥러닝 기반의 부호화 기술들은 기존 비디오 부호화 요소 기술을 대체하거나 추가적인 요소 기술로써 세부 인-루프 필터링, RPR (Reference Picture Resampling), 그리고 화면내 예측 등이 있다<sup>[4]</sup>.

특히, 복호화된 주변 샘플을 참조해 예측하고자 하는 블록을 미리 정의된 제한된 모드의 고정된 패턴으로 예측하는 화면내 예측 기술은 영상의 다양한 특징을 반영하기 어려워 부호화 효율의 한계가 있으며, 이와 같은 영상의 공간적 특성 및 참조샘플의 부족을 극복하고자 딥러닝 기반의 화면내 예측 기술의 중요성이 부각되고 있다<sup>[5][6]</sup>. 현재 JVET NNVC에서 논의되고 있는 화면내 예측 모델 중 문맥적응적(context-adaptive) 화면내 예측 모델은 합성곱 신경망(CNN: Convolutional Neural Network)과 완전 연결계층망(FCN: Fully-Connected Network)으로 구성되어 현재블록에 인접한 주변 복원블록들을 입력하여 예측블록을 생성한다.

본 논문에서는 NNVC에서 다루고 있는 화면내 예측 신경망인 문맥적응적 화면내 예측 모델의 구현 및 학습과 학습한 예측 모델의 부호화 성능분석을 제시한다. 즉, 학습된

문맥적응적 화면내 예측 모델을 HEVC참조 소프트웨어인 HM16.19에 추가적인 하나의 화면내 예측 모드로 구현하고 그 부호화 성능을 분석한다.

또한 비디오 부호화에서 블록분할을 고려한 학습과 그에 따른 부호화 성능변화를 확인한다. 블록분할은 율-왜곡(RD: Rate-Distortion) 측면에서 최적의 예측 모드와 블록 크기로 부호화 하도록 결정된다. 화면내 예측모델 학습에서 블록분할을 고려하지 않을 경우 학습 데이터를 구성할 때 특정 영상에서 모델이 현재 예측하려는 블록 사이즈에 맞게 무작위로 선택하여 데이터를 사용한다. 이는 RDO (RD-Optimization) 기반의 블록분할을 진행하는 비디오 코덱에는 적합하지 않을 수 있다. 따라서 블록분할 구조를 반영한 학습에서는 재귀적 블록분할에 따른 문맥적응적 예측 모델의 RD 성능을 비교하여 최적의 블록분할을 유도하고, 유도된 블록분할에서 결정된 크기의 테스트 데이터를 사용하여 블록 크기에 따른 문맥적응적 모델을 학습한다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 NNVC에서 논의되고 있는 문맥적응적 화면내 예측 모델에 대해 소개하며, 3장에서는 제안하는 화면내 예측 모델 학습기법을 설명한다. 4장에서는 학습된 문맥적응적 화면내 예측 모델을 HEVC에 적용한 부호화 성능을 제시하며, 마지막으로 5장에서 결론을 맺는다.

## II. 문맥적응적 신경망 기반 화면내 예측

JVET NNVC에서 논의되고 있는 신경망 기반 문맥적응적 화면내 예측 모델은 예측하고자 하는 현재블록의 크기에 따라 사용하는 네트워크를 다르게 한다<sup>[7][8]</sup>. 해당 화면내 예측 모델은 작은 블록에 대해 완전연결계층(FCL: FC Layer) 기반의 화면내 예측 모델을 사용하고, 큰 블록에 대해서는 합성곱계층(CL: Convolutional Layer) 및 FCL 기반의 화면내 예측 모델을 사용한다. 즉, 현재 예측블록이  $\min(h, w) \leq 8$ 의 조건을 만족하는 경우, 그림 1과 같은 FCL 기반의 네트워크 구조를 가진다. 여기서,  $w$ 와  $h$ 는 현재블록의 가로와 세로의 크기를 각각 의미한다. 현재 블록( $Y$ ) 주변의 참조샘플( $X$ )을 바로 3계층의 FCL에 입력하여

a) 한국항공대학교 항공전자정보공학과(Korea Aerospace University, School of Electronics and Information Engineering)

‡ Corresponding Author : 김재곤(Jae-Gon Kim)  
E-mail: jgkim@kau.ac.kr  
Tel: +82-2-300-0414  
ORCID: <https://orcid.org/0000-0003-3686-4786>

※ This work was supported by Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by Korea Government (MSIT) (2017-0-00486).

· Manuscript received August 23, 2021; Revised March 31, 2022; Accepted April 25, 2022.

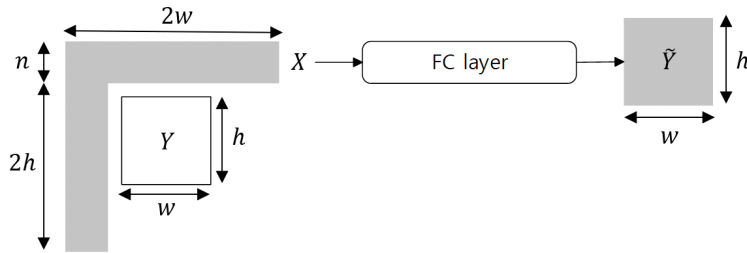


그림 1. 4x4, 8x8 블록에 대한 문맥적응적 신경망 기반 화면내 예측 네트워크  
 Fig. 1. Context-adaptive neural network-based intra prediction for 4x4, 8x8 blocks

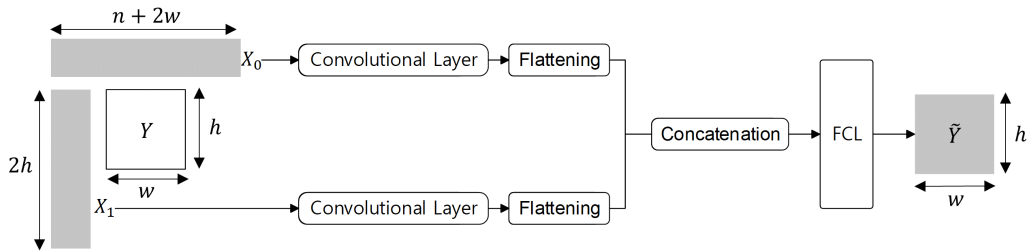


그림 2. 16x16, 32x32 블록에 대한 문맥적응적 신경망 기반 화면내 예측 네트워크  
 Fig. 2. Context-adaptive neural network-based intra prediction for 16x16, 32x32 blocks

최종적인 예측블록( $\tilde{Y}$ )을 생성한다. 반면, 예측하고자 하는 현재블록이  $\min(h, w) > 8$ 의 조건을 만족하는 경우, 그림 2와 같은 CL 및 FCL 기반의 네트워크 구조를 갖는다. 현재블록의 상단과 좌측에 대한 두개의 참조블록( $X_0, X_1$ )은 각각 4계층의 CNN 기반의 문맥분석 모듈에 입력된다.  $X_0$ 과  $X_1$ 을 입력으로 하는 두 개의 문맥분석 모듈 결과는 평탄화(flattening) 과정을 거쳐 하나로 연결된(concatenation) 후 2계층의 FCL 기반의 화면내 예측 모듈에 입력되며 최종 예측블록( $\tilde{Y}$ )을 생성한다. 이와 같이 현재블록 크기에 따라 사전에 매핑된 모델을 사용함으로써 추가적인 시그널링 없이 적합한 모델을 사용하여 현재 블록을 예측할 수 있도록 하고 있다. 본 논문에서의 참조샘플은 비디오 코딩의 부/복호화의 복잡도를 고려하여 모든 블록 크기에 대한 화면내 예측 모델에 네 개의 라인( $n=4$ )을 동일하게 사용하였다.

### III. 화면내 예측 모델 학습기법

본 장에서는 II 장에서 기술한 문맥적응적 신경망의 학습 기법을 제시한다. 먼저 개별 블록 크기에 따른 문맥적응적

화면내 예측 모델을 학습하고, 개별적으로 학습된 모든 블록 크기에 대한 화면내 예측 모델을 사용하여 블록분할 구조를 반영하는 학습방법으로 재학습(retraining) 한다.

#### 1. 화면내 예측 모델 학습

학습에 사용한 데이터셋은 CNN 기반의 비디오 압축을 위한 데이터 셋인 BVI-DVC<sup>[9]</sup>를 사용한다. 이는 전체 800개의 시퀀스로 구성되어있으며, 270p 부터 2160p까지 총 4개의 해상도로 가진다. 화면내 예측에서 사용되는 참조샘플은 이전에 복원된 샘플을 사용하므로 BVI-DVC 데이터를 VVC의 참조 S/W인 VTM 12.0을 이용하여 22, 27, 32, 37의 QP(Quantization Parameter)로 부호화하여 복원된 영상을 사용한다. 비디오 시퀀스의 시간적 중복성을 고려하여 8 프레임당 한 프레임을 추출하였으며 추출된 프레임을 1 CTU(Coding Tree Unit) 단위로 분할한 후 CTU의 분산값을 사용하여 낮은 밝기 변화량을 갖는 데이터들을 제외하여 약 540K개의 CTU를 데이터셋으로 구성하였다.

손실함수는 잔차신호의 왜곡을 최소화하는 동시에 비디오 부호화 효율을 극대화하도록 변환 및 양자화를 반영하

여 정의한다. 식 (1) 에서의  $T$ 는 2D DCT-2 변환 함수이고,  $O$ 와  $P$ 는 각각 원본블록과 예측블록을 의미한다. 즉,  $C$ 는 원본블록과 예측블록의 잔차블록에 대해 2D DCT 변환을 수행하여 얻은 변환계수를 의미한다. 각 변환계수에 대한 절대값의 합으로 하는 SATD(Sum of Absolute Transformed Difference)를 손실함수로 사용하여 학습을 진행한다.

$$C = T(O - P) \quad (1)$$

$4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$  네 가지의 블록 크기에 각각에 대해서 위와 같은 데이터셋 및 손실함수를 이용하여 학습하며 학습에는 Adam 최적화기를 사용한다. 또한 학습율(learning rate)은 초기 0.0005에서 5 에폭(Epoch) 마다 23%씩 감소하는 학습율 감소기법을 적용하고 총 100 에폭을 학습한다.

## 2. 블록분할을 반영한 화면내 예측 모델 학습

본 논문에서는 문맥적응적 화면내 예측 모델을 학습함에 있어 HEVC의 블록분할 구조를 반영하는 학습기법을 제시한다. HEVC에서는 최적의 부호화 효율을 갖는 부호화 블록 구조를 결정하기 위해 다양한 블록분할 및 분할 깊이에 대한 율-왜곡 최적화(RDO)를 수행한다. 특히, HEVC의 화면내 예측 모드에 대해서는 쿼드(quad) 분할만 가능하다. 이러한 블록분할 방식을 모델 학습에 반영하기 위하여 III-1 장의 설명과 같이 블록 크기에 따라서 개별적으로 학습한 네 개의 학습 모델을 이용하여 각 테스트 데이터의 최적의 블록분할을 결정한다. 최적의 블록분할은 그림 3과 같이  $2N \times 2N$  블록에 대한 율-왜곡 손실 값과 해당 블록이 분할된  $N \times N$  네 개 블록에 대한 손실 값의 합을 비교하여 결정된다. 즉,  $2N \times 2N$  블록에 대한 손실 값이 하위 깊이의 4개의  $N \times N$  블록들에 대한 손실 값의 합보다 작을 경우,  $2N \times 2N$  블록은 해당하는 블록크기에 대한 화면내 예측 모델을 학습하기 위한 데이터셋에 포함된다. 해당 과정을  $N$ 의 값이 16에서 4까지 재귀적으로 반복한다.  $N$ 이 4일 때,  $8 \times 8$  블록의 손실 값보다 하위 깊이의  $4 \times 4$  블록들의 손실 값의 합이 작을 경우 해당하는  $4 \times 4$  블록들은 해당하는 블록크기에 대한 화면내 예측 모델을 학습하기 위한 데이터

셋에 포함된다. 최종적으로, 입력된 CTU에서  $32 \times 32$  블록 단위로 위의 프로세스를 반복하며 블록분할 구조를 유도하고 유도된 블록분할을 기반으로 해당하는 블록 크기에 따른 화면내 예측 모델을 학습하기 위한 데이터 셋을 구성한다. 이와 같이 블록분할 결과에 따라 최적의 블록 크기로 결정된 해당 크기의 모델들에 대해서 학습함으로써 각각의 모델이 해당 블록 크기를 갖는 데이터들의 특징을 학습하게 된다. 그림 4는 개별적으로 학습된 모델을 이용하여 재귀적인 경정을 통해 블록분할이 결정된 예이다.

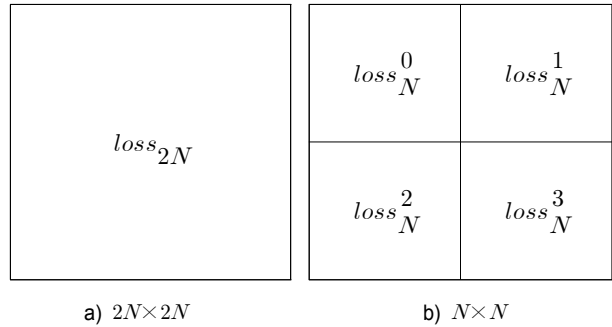


그림 3. 상위블록과 하위블록의 손실값  
 Fig. 3. Loss of upper and lower blocks

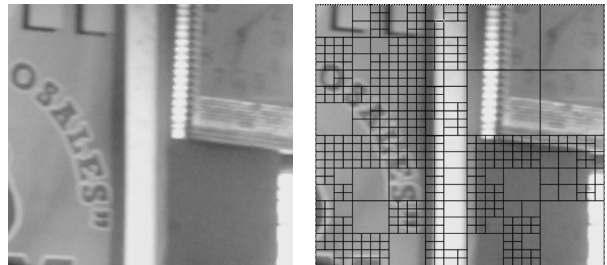


그림 4. 블록분할 구조 반영 시 블록분할 결과(좌: 원본영상, 우: 블록분할)  
 Fig. 4. Block partitioning results when reflecting block partitioning structure (Left: original CTU, Right: CTU with block partition)

비디오 부호화 과정에서는 많은 변환계수들의 값이 0으로 표현될수록 전송해야 하는 비트의 수가 줄어들므로 압축에서 이득을 얻게 된다. 이를 학습방법에 반영하기 위하여, 식 (2)와 같이 기존 SATD에 변환계수에 로지스틱(logistic) 함수를 적용하는 항을 추가하여 예측신호의 왜곡이 줄어들면서 동시에 변환계수들이 0에 더 빠르게 수렴할

수 있도록 손실함수를 정의한다. 그림 5는 식 (2)로 정의한 손실함수를 표현한 것으로  $\alpha$ ,  $\beta$ ,  $\gamma$ 는 정의된 손실함수를 구현하기 위한 파라미터이다. 이는 입력 샘플의 스케일에 따라서 다르게 조절이 가능하며 본 논문의 모델 학습에서는  $\alpha$ ,  $\beta$ ,  $\gamma$ 를 각각 25, 10, 1의 값으로 설정한다. 손실함수를 제외한 데이터셋 및 최적화기 등의 학습조건은 III-1장의 개별 학습 모델과 동일하게 진행한다.

$$L = \sum_i (abs(c_i) + \alpha \times g(\beta \times abs(c_i) - \gamma)) , \quad (2)$$

$$g(x) = 1 / (1 + e^{-x}) .$$

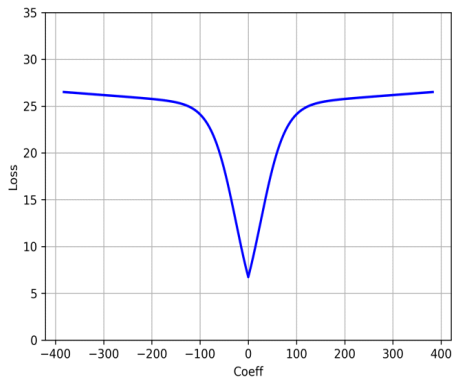


그림 5. 로지스틱 함수를 적용한 손실함수  
Fig. 5. Loss function with logistic function

#### IV. 실험결과

본 논문에서의 실험은 학습한 모델 성능을 평가하기 위하여 제안모델을 HMI6.19에 화면내 예측의 추가적인 예측 모드로 구현하였다. 최근 차세대 비디오 코덱(VVC)이 완료된 시점에서 HEVC 코덱을 사용하는 이유는 코덱 성능 분석을 위해 블록 크기 별 모델 학습이 필요한데 VVC는 현재 다양한 블록 크기를 허용하고 있다. 이는 데이터셋 구성과 모델 학습에 많은 시간이 필요하다. 따라서 쿼드 분할만을 허용하는 HEVC에 구현하여 제시한 학습 기법의 성능 향상을 확인하였다.

##### 1. HM 통합구현

비디오 부호화 실험을 위해 딥러닝 기반의 화면내 예측 모델을 HEVC에 추가적인 화면내 예측 모드로 통합하였으며 현재블록에 대한 딥러닝 기반의 화면내 예측 모드의 수행 여부를 지시하기 위해 시그널링을 추가하였다. 그림 6과 같이 가 '1'인 경우 현재 블록은 딥러닝 기반 화면내 예측 모드를 수행하며 '0'인 경우 기존의 화면내 예측 모드 시그널링 및 기존의 화면내 예측을 수행한다.

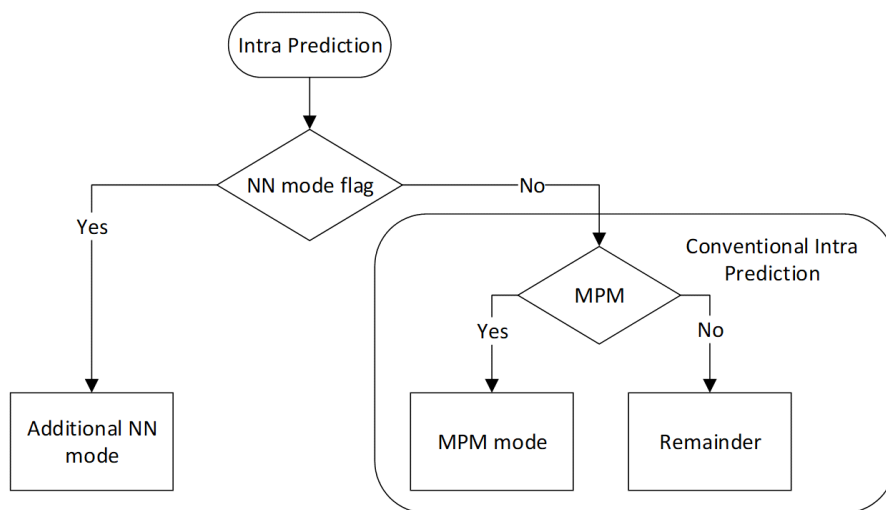


그림 6. HEVC 화면내 예측 부호화 모드로의 NN 모드 통합 구현  
Fig. 6. Integration of NN-based mode into intra prediction of HEVC

## 2. 객관적 성능

비디오 부호화 실험은 JVET CTC(Common Test Condition)<sup>[10]</sup>에 따라서 A, B, C, D, E 클래스의 테스트 시퀀스를 사용하였다. 각 시퀀스는 AI(All-intra) 부호화 모드에서 {22, 27, 32, 37}의 4개의 QP로 부호화 하였다. 각 클래스의 결과는 해당 클래스에 포함되는 시퀀스들의 BD-rate 평균값이다.

표 1은 네 개의 블록 크기에 대해서 독립적으로 학습한 모델을 HM16.19에 구현하였을 때 HM16.19 대비 부호화 성능이다. 휘도성분(Y)에서 0.28%의 BD-rate 비트를 절감의 성능 향상을 보였다. 표 2는 HM16.19 대비 블록분할을 적용한 학습모델에 대한 부호화 성능이며, Y에서 0.32% 성능향상을 보였다. 표 1과 표 2를 비교하여, 계층적 블록분할을 기반으로 학습한 모델이 개별적으로 학습한 모델 보

다 Y에서 0.03% 추가 성능 향상이 있음을 확인하였다. 다소 미미한 성능 향상이지만 블록분할 방식을 학습에 적용함으로써 성능이 향상될 수 있음을 확인할 수 있다.

표 3은 제안하는 기법이 제안기법의 추가적인 예측 모드 비율을 보여주는 것으로 JVET CTC 중 A, B, C, D, E 클래스별 한 개의 시퀀스를 사용하였다. 개별적으로 학습된 모델이 HM16.19에서 선택될 비율은 7.28%이다. 블록분할을 고려하여 학습한 모델이 HM16.19에서 선택될 비율은 8.38%로 이전 선택 비율 대비 제안하는 방법의 추가 모드(NN)의 선택 비율이 1.0% 높아짐을 확인할 수 있다.

복잡도의 경우, 본 논문의 실험에서는 학습된 모델을 PyTorch에서 제공하는 LibTorch 라이브러리를 이용하여 비디오 코덱에 통합하였다. LibTorch를 이용하여 신경망 모델을 처리하는 것은 실제 모델의 복잡도에 비해 많은 인코딩/디코딩 시간을 소요하기 때문에 표 2와 같이 약 각각

표 1. 블록 크기별 개별 학습 모델의 부호화 성능(Anchor: HM16.19)

Table 1. BD-rate coding performance of prediction models trained individually depending on block size (Anchor: HM16.19)

	All Intra Main10				
	Over HM16.19			EncT	DecT
	Y	U	V		
Class A1	-0.37%	-0.14%	-0.10%	9136%	1892%
Class A2	-0.22%	-0.01%	-0.02%	8343%	1998%
Class B	-0.28%	0.03%	0.07%	7589%	1597%
Class C	-0.22%	0.15%	0.09%	22507%	1324%
Class E	-0.35%	0.05%	-0.04%	6094%	1231%
Overall	-0.28%	0.02%	0.01%	9762%	1566%
Class D	-0.24%	0.11%	0.24%	58528%	1320%

표 2. 재귀적 블록분할을 고려한 학습 모델의 부호화 성능(Anchor: HM16.19)

Table 2. BD-rate coding performance of prediction models trained considering recurve block partition (Anchor: HM16.19)

	All Intra Main10				
	Over HM16.19			EncT	DecT
	Y	U	V		
Class A1	-0.39%	-0.18%	-0.08%	9790%	1960%
Class A2	-0.22%	0.09%	0.04%	8936%	2296%
Class B	-0.30%	0.05%	0.13%	17057%	3324%
Class C	-0.28%	-0.08%	0.06%	20919%	3271%
Class E	-0.42%	0.02%	-0.04%	22896%	2454%
Overall	-0.32%	-0.02%	0.04%	15344%	2711%
Class D	-0.29%	-0.08%	0.21%	51632%	2330%

표 3. 재귀적 블록분할을 고려한 학습 모델의 화면내 예측 모드 선택 비율(Anchor: HM16.19)

Table 3. Intra prediction mode selection rate of prediction models trained considering recurve block partition (Anchor: HM16.19)

Class	Sequence	Individually trained model (%)	Proposed method (%)
A	Tango	6.34	6.96
B	Cactus	8.75	9.90
C	BasketBallDrill	4.14	5.12
D	BlowingBubble	8.25	9.84
E	FourPeople	8.94	10.14
Average		7.28	8.38

150배, 27배에 달하는 시간 증가를 보였다. 실험에 사용한 PC 환경(AMD Ryzen 2950X 16-Core, 64GB RAM)이 다소 상이하여 표 1, 2에 제시된 인코딩/디코딩 시간은 다소 부정확할 수 있다.

### 3. 주관적 예측 성능

블록 크기별 개별 모델과 블록분할을 반영하여 학습한

모델의 화면내 예측 블록의 휘도 영상을 비교하여 주관적 예측 성능을 확인한다. 그림 7과 그림 8은 현재 블록의 크기가 32×32와 8×8일 때 예측 블록을 보여준다. (a)는 학습모델의 입력으로 사용된 참조샘플, (b)는 예측하려는 블록의 원본 영상이며, (c)와 (d)는 각각 개별 모델의 예측 블록 영상, 블록분할을 반영한 학습된 모델의 예측 블록 영상이다. 개별 학습된 모델(c)의 경우 예측 블록 영상의 전체적인 밝기가 주변 참조샘플의 밝기 값의 평균을 따르는 경향

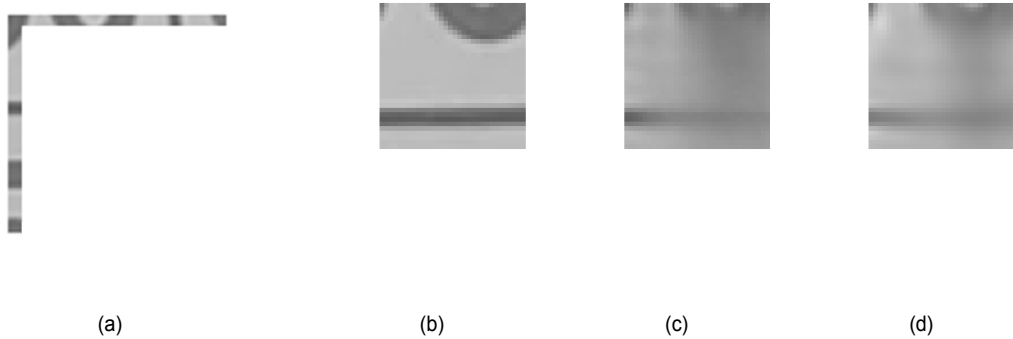


그림 7. 32x32에 대한 학습 모델의 예측블록 영상 결과 (a) 참조샘플 (b) 원본 영상 (c) 개별 학습 모델 (d) 블록분할 반영 모델

Fig. 7. Prediction block image results of the trained model for 32x32 blocks (a) Reference sample (b) Original image (c) Individual trained model (d) Model trained with block partition consideration

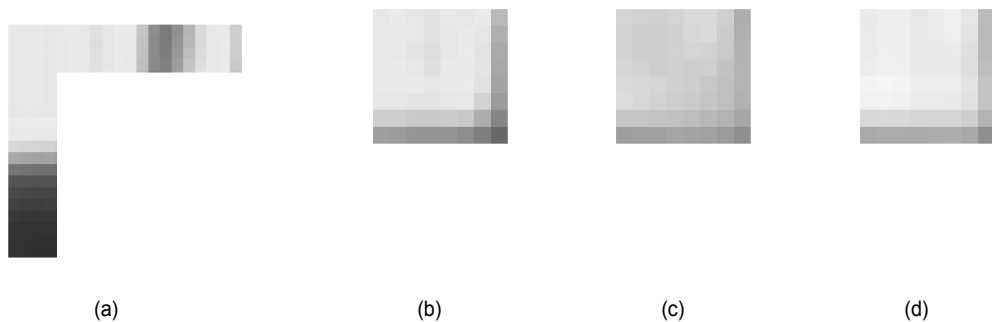


그림 8. 8x8에 대한 학습 모델의 예측블록 영상 결과 (a) 참조샘플 (b) 원본이미지 (c) 개별 학습 모델 (d) 블록분할 반영 모델

Fig. 8. Prediction block image results of the trained model for 8x8 blocks (a) Reference sample (b) Original image (c) Individual trained model (d) Model trained with block partition consideration

을 보인다. 반면, (d)의 경우 개별 학습 모델(c) 대비 원본과 유사한 밝기의 배경을 가지며 선명하게 예측함을 확인하였다.

## V. 결론

본 논문에서는 NNVC에서 논의 중인 문맥적응적 화면내 예측 신경망 모델의 학습 기법을 제시하고 코덱에서의 그 부호화 성능을 분석하였다. 문맥적응적 화면내 예측 신경망 모델을 최적의 블록분할 경쟁을 통해 학습하고, 이를 HEVC에 적용하여 부호화 성능이 향상될 수 있음을 확인하였다. 제시한 학습기법은 문맥적응적 예측 모델을  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ 의 블록 크기별로 각각 학습한 후 학습된 네 개의 모델을 HM16.19의 기존 화면내 예측 35가지 모드에 추가 모드로 구현하여 부호화 성능을 확인하였다. 또한, 부호화 블록분할을 학습에 고려하기 위해서 개별적으로 학습된 해당 모델들을 이용하여 재귀적 블록분할 경쟁을 통해 최적의 블록분할이 학습에 반영되도록 각 모델에 대한 재학습을 진행하였다. HM16.19 대비 개별 모델로 학습한 모델은 AI(All intra) 모드에서 0.28%, 블록분할 고려하여 학습한 모델은 0.32%의 성능개선을 보였다. 블록분할을 고려한 학습은 개별 모델 대비 0.03%의 추가 성능으로 미미하지만 성능개선의 여지가 있음을 보여 주었다. 큰 블록 데이터 대한 학습 데이터 확보 등 큰 블록을 보다 효과적으로 학습할 수 있는 방법과 코덱 내에서 최적의 시그널링 설계 등 문맥적응적 신경망 모델 기반의 화면내 예측의 부호화 성능 향상을 위한 추가 연구가 필요하다.

## 참고 문헌 (References)

- [1] High Efficiency Video Coding, Version 1, Rec. ITU-T H.265, ISO/IEC 23008-2, Jan. 2013.  
doi: 10.1007/978-3-319-06895-4
- [2] Versatile Video Coding, ISO/IEC FDIS 23090-3, Jul. 2020.
- [3] S. Liu, E. Alshina, J. Pfaff, M. Wien, P. Wu, and Y. Ye, "JVET AHG report: Neural-network-based video coding," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, JVET-V0011, Apr. 2021.
- [4] Alshina, S. Lui, W. Chen, F. Galpin, Y. Li, Z. Ma, H. Wang, "EE1: Summary of exploration experiments on neural network-based video coding," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, JVET-W0023, July. 2021.
- [5] "Use cases and requirements for deep neural networks based video coding," ISO/IEC JTC 1/SC 29/WG 2, N22, Oct. 2020.
- [6] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," IEEE Trans. Image Proc., vol. 27, no. 7, Mar. 2018.  
doi: 10.1109/TIP.2018.2817044
- [7] T. Dumas, A. Roumy, and C. Guillemot, "Context adaptive neural network based prediction for image compression," IEEE Trans. Image Proc., vol. 29, Aug. 2019.  
doi: 10.1109/TIP.2019.2934565
- [8] T. Dumas, F. Galpin, P. Bordes, and F. Leleannec (InterDigital), "AHG11: BD-rate gains vs complexity of NN-based intra prediction," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, JVET-W0081, July 2021.
- [9] D. Ma, F. Zhang and D. Bull, "BVI-DVC: A training database for deep video compression," 2020, arXiv:2003.13552.  
<https://data.bris.ac.uk/data/dataset/3hj4t64fkbrgn2ghwp9en4vhtn>
- [10] J. Boyce, K. Suehring, X. Li, and V. Seregin, "JVET common test conditions and software reference configurations," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Apr. 2018.



---

저 자 소 개

---



**문 기 화**

- 2021년 2월 : 한국항공대학교 소프트웨어학과 학사
- 2021년 3월 ~ 현재 : 한국항공대학교 항공전자정보공학부 석사과정
- ORCID : <https://orcid.org/0000-0001-6727-7790>
- 주관심분야 : 비디오 부호화, 영상처리, 딥러닝



**박 도 현**

- 2016년 2월 : 국립한밭대학교 멀티미디어공학 학사
- 2018년 2월 : 한국항공대학교 항공전자정보공학과 석사
- 2018년 3월 ~ 현재 : 한국항공대학교 항공전자정보공학과 박사과정
- ORCID : <https://orcid.org/0000-0002-5873-0132>
- 주관심분야 : 비디오 부호화, 영상처리, 360 VR 비디오, 머신러닝



**김 재 곤**

- 1990년 2월 : 경북대학교 전자공학과 학사
- 1992년 2월 : KAIST 전기 및 전자공학과 석사
- 2005년 2월 : KAIST 전기 및 전자공학과 박사
- 1992년 3월 ~ 2007년 2월 : 한국전자통신연구원(ETRI) 선임연구원/팀장
- 2001년 9월 ~ 2002년 7월 : Columbia University 연구원
- 2015년 12월 ~ 2016년 1월 : UC San Diego, Visiting Scholar
- 2007년 9월 ~ 현재 : 한국항공대학교 항공전자정보공학부 교수
- ORCID : <http://orcid.org/0000-0003-3686-4786>
- 주관심분야 : 비디오 부호화 표준, 비디오 신호처리, Immersive Video, Deep Learning