Neural Network-based Video Coding (NNVC) 개요

ㅁ 조현동, 김휘용 / 경희대학교

요 약

본 고에서는 JVET에서 진행 중인 차세대 비디오 코딩 기술 탐색 작업 중, 신경망 기반 비디오 코딩(Neural Network-based Video Coding, NNVC)의 최신 동향을 기술한다. 특히 NNVC Software-15.0에 채택된 Tool-based 접근법의 핵심 기술인 신경망 기반 인루프 필터(NNLF) 및 적응적 NNLF, 화면 내 예측(NN-Intra), 화면 간 예측(NN-Inter), 그리고 초해상화(NNSR)의 구조와 동작 원리를 상세히 설명한다. 또한, JVET의 공통 테스트 조건(CTC)과 SADL 라이브러리 환경 하에서 각 기술이 VTM 대비 보여주는 부호화 효율(BD-Rate)과 연산 복잡도(kMAC/pixel, Encoding/Decoding Time)를 정량적으로 분석함으로써, 향후 차세대 비디오 코딩 표준화를 위한 NNVC 기술의 가능성과 실용적 제약사항을 고찰한다.

I. 서 론

비디오 압축 기술은 고품질 비디오 콘텐츠를 효율적으로 저장 및 전송하기 위해 지속적으로 고도화되어 왔다. 2020년 7월, ISO/IEC JTC 1/SC 29 MPEG과 ITU-T SG16/Q6 VCEG가 공동으로 설립한 Joint Video Experts Team(JVET)은 차세대 비디오 코딩 표준인 Versatile Video Coding(VVC)[1]을 제정하였다. 이후 2025년 10월에는 VVC 이후의 차세대 비디오 코딩 표준 후보 기술의 타당성을 검토하기 위한 Call for Evidence(CfE)가 JVET 주관으로 시행되었다. JVET의 차세대 비디오 코딩 기술 탐색은 크게 두 범주로 구분된다. 첫째, 신호 처리 기반 접근법으로서 AVC/H.264-HEVC/H.265-VVC/H.266의 계보를 잇는 Enhanced Compression Model(ECM)이 있다. ECM은 VVC 대비 향상된 압축 효율을 달성하기 위해 보다 정교한 블록 분할, 예측 및 부호화 기법을 탐구한다. 둘째, Neural Network-based Video Coding(NNVC) 접근법은 VVC 코덱의 일부 구성 요소를 신경망 기반 기법으로 대체함으로써 성능 향상을 도모한다.

NNVC는 대규모 데이터로부터 일반적인 영상의 분포를 학습할 수 있다는 점에서 다양한 비디오 콘텐츠에 대

<표 1> NNVC Software-15.0 Tool-based 최신 기술

카테고리		Anchor
신경망 기반 인루프 필터 Neural Network-based In-Loop Filter (**NNLF**)	고복잡도 [5] High Operation Point (**HOP**)	X (software adoption)
	저복잡도 [3] Low Operation Point (**LOP**)	O
	초저복잡도 [4] Very Low Operation Point (**VLOP**)	X (software adoption)
신경망 기반 화면 내 예측 [6] Neural Network-based Intra Prediction (**NN-Intra**)		O
신경망 기반 화면 간 예측 [7] Neural Network-based Inter Prediction (**NN-Inter**)		X (software adoption)
신경망 기반 초해상화 필터 [8] Neural Network-based Super Resolution (**NNSR**)		X (software adoption)
적응적 신경망 기반 인루프 필터 [9] Adaptive Neural Network-based Loop Filter (**Adaptive NNLF**)		X (software adoption)

한 높은 적응성을 제공한다. NNVC 연구는 종단간(End-to-End) 방식과 도구 기반(Tool-based) 방식으로 다시 구분될 수 있다. End-to-End 방식은 Learned Image Compression[2]을 기반으로 화면 내 예측을 수행하고, 이를 Multi-Layer Coding 등의 형태로 확장하여 화면 간 예측에 활용하는 방향으로 전개되고 있다. 반면, Tool-based 방식은 개별 요소를 신경망 기반 도구로 교체하는 접근으로, 비교적 오랫동안 연구가 되어 왔다.

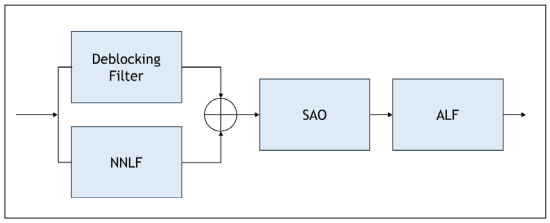
본 고에서는 NNVC Software-15.0[1]에 채택된 Tool-based 기술들을 소개한다. 대표적인 Tool-based 기술로는 신경망 기반 인루프 필터(Neural Network-based In-Loop Filter, **NNLF**)[3,4,5], 신경망 기반 화면 내 예측(Neural Network-based Intra Prediction, **NN-Intra**)[6], 신경망 기반 화면 간 예측(Neural Network-based Inter Prediction, **NN-Inter**)[7], 신경망 기반 초해상화 필터(Neural Network-based Super Resolution, **NNSR**)[8], 적응적 신경망 기반 필터(Adaptive Neural Network-based In-Loop Filter, **Adaptive NNLF**)[9,10]가 포함된다.

II. NNVC Tool-based 기술 소개

본 장에서는 NNVC Software-15.0에서 사용되는 Tool-based 기술을 설명한다. 〈표 1〉에 보인 바와 같이 NNVC Software-15.0은 저복잡도(Low Operation Point, LOP) NNLF와 NN-Intra를 Anchor로 사용하며, 그 외 기술들은 소프트웨어에만 채택되어 있다.

1. 신경망 기반 인루프 필터(NNLF)

인루프 필터는 복원된 프레임에 존재하는 블로킹(blocking), 링잉(ringing), 블러(blur) 등 코딩 아티팩트를 제거하기 위해 설계되었다. 인루프 필터가 적용된 복원 프레임은 최종 출력 영상으로 사용될 뿐만 아니라 이후 프레임 예측을 위한 참조 프레임으로도 활용되므로, 인루프 필터의 성능은 전체 부호화 효율에 직접적인 영향을 미친다. 전통적인 인루프 필터는 사전 정의된 규칙 기반(rule-based) 설계에 의존하는 반면, 신경망 기반 인루프 필터(NNLF)는 대규모 데이터셋 학습을 통해 다양한 콘텐츠



<그림 1> 기존 인루프 필터와 결합된 NNLF의 동작 순서

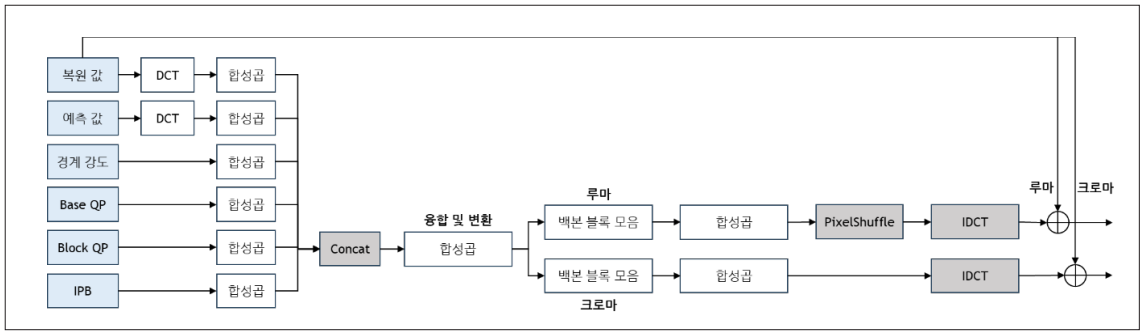
특성과 왜곡 패턴에 보다 유연하게 대응할 수 있다는 장점을 갖는다. 다만 NNLF는 일반적으로 모델 복잡도의 증가를 수반하며, 이는 연산량 증가 및 처리 지연으로 이어질 수 있다. 한편, NNVC Software-15.0에서 NNLF의 출력은 〈그림 1〉과 같이 디블로킹 필터(deblocking filter)의 출력과 가중합 형태로 결합하여 최종 출력을 생성하며, SAO와 ALF를 추가적으로 수행한다.

NNVC Software-15.0에 포함된 NNLF는 복잡도(파라미터 수, kMAC/pixel)에 따라 초저복잡도(Very Low Operation Point, VLOP), 저복잡도(Low Operation Point, LOP), 고복잡도(High Operation Point, HOP)로 구분된다. 또한 JVET 회의 진행되는 과정에서 구조 및 설계가 개선되며, 개선된 버전은 숫자를 증가시키는 방식으로 표기한다. 현재 NNVC Software-15.0에서는 HOP5 NNLF, LOP6 NNLF, VLOP4 NNLF가 각 복잡도에서 최신 버전으로 사용된다.

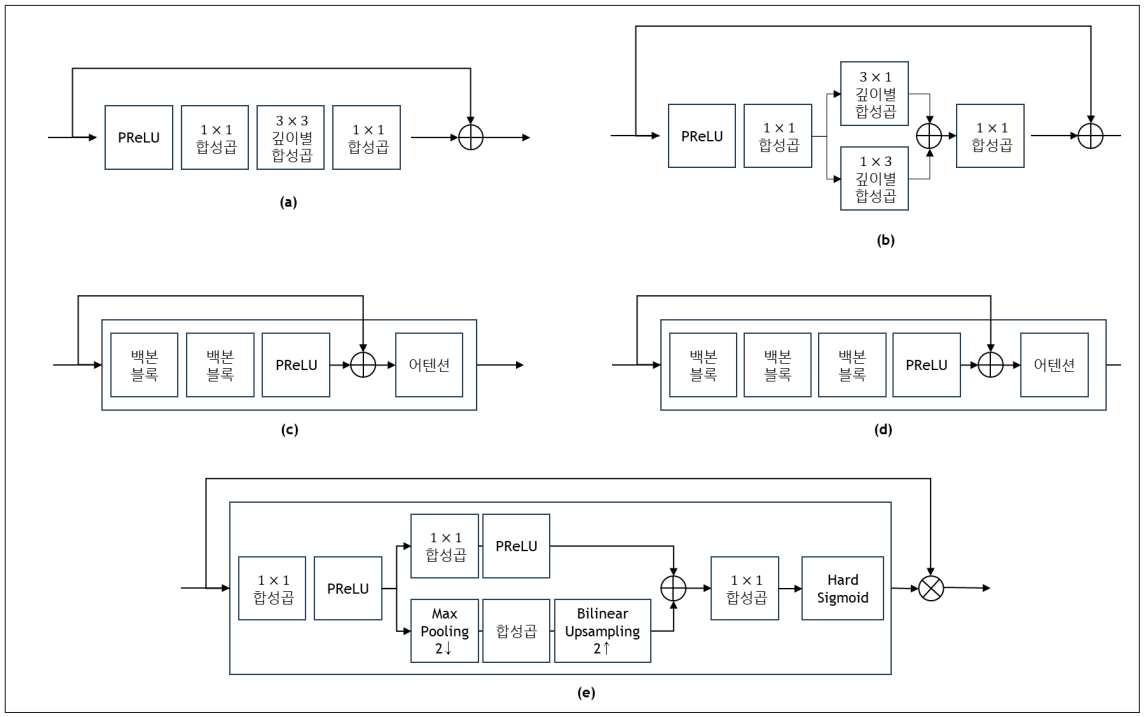
NNLF는 일반적으로 좌·우·상·하 방향으로 각각 8픽셀 패딩을 포함한 144×144 크기의 입력 패치를 사용한다. 입력은 (i) 필터링 전 복호화 영상, (ii) 예측 영상, (iii) 경계 강도(boundary strength) 값, (iv) BaseQP, (v) BlockQP, (vi) 프레임 유형(IPB: I/P/B 프레임) 정보로 구성된다. 이러한 입력들을 채널 방향으로 결합(concatenation)한 뒤, 전처리와 백본 블록을 거쳐 처리하며, 마지막에 후처리를 통해 복원 블록과 동일한 크기의 출력을 생성한다.

〈그림 2〉와 같이 VLOP NNLF와 LOP NNLF는 전반적으로 유사한 구조를 공유한다. 다만 LOP NNLF는 VLOP NNLF에 비해 많은 백본 블록의 개수를 사용하여 모델 용량과 연산량이 증가하며, 일부 백본 블록에서 병렬 구조를 추가로 도입한다는 점에서 차이가 있다. 여러 입력 중 복호화 영상과 예측 영상의 루마(luma) 패치는 2×2 DCT-2 변환을 적용하여 (H×W×1) 형태의 입력을 ((H/2)×(W/2)×4) 형태의 주파수 도메인 표현으로 변환한 후, 크로마(UV) 입력과 결합한다. 이후 각 입력에 3×3 합성곱(convolution)을 수행하고 결합하여, 결합된 특징을 3×3 및 1×1 합성곱으로 융합 및 비선형 변환을 수행한다. 이후 네트워크는 루마 분기와 크로마 분기로 분리되며, 각 분기에서 일련의 백본 블록을 통해 비선형 변환을 추가적으로 수행한다. 출력 단계에서는 3×3 합성곱과 역 DCT 변환, 그리고 루마 경로의 경우 PixelShuffle을 사용하여 채널 및 공간 크기를 복원한다. 이 과정의 결과로 출력은 복원 대상 블록과 동일한 크기를 갖도록 구성된다. 각 백본 블록은 (i) PReLU 비선형 함수, (ii) 1×1 합성곱, (iii) 3×3 커널 기반의 깊이별 합성곱(depth-wise convolution), (iv) 1×1 합성곱 순서로 구성된다. 또한 출력 크기를 맞추기 위해 크롭(crop) 대신 일부 블록에서 패딩을 사용하지 않는 합성곱(padding-free convolution)을 사용하여 중간 특징 맵의 공간 크기를 축소한다. 또한 백본 블록 여러 개를 쌓고, 어텐션(attention) 모듈을 사용하여 트윈 블록(TwinBlock)과 트리플 블록(TripleBlock)을 구성하였고, LOP NNLF는 병렬 구조 백본 블록을 추가적으로 도입하였다. 구체적으로, PReLU와 1×1 합성곱 이후의 채널 특징 맵을 1×3 경로와 3×1 경로로 분기한 뒤, 각 경로에서 깊이별 분리형 합성곱(depth-wise separable convolution)을 적용한다. 두 경로의 출력을 원소별 합으로 결합한 후, 마지막 1×1 합성곱을 통해 출력을 생성한다. 반면 크로마 분기는 어텐션 모듈 없이 백본 블록만으로 구성되어 있으며, 루마 특징을 추출하여 크로마 특징과 결합한 뒤 후속 처리를 수행한다.

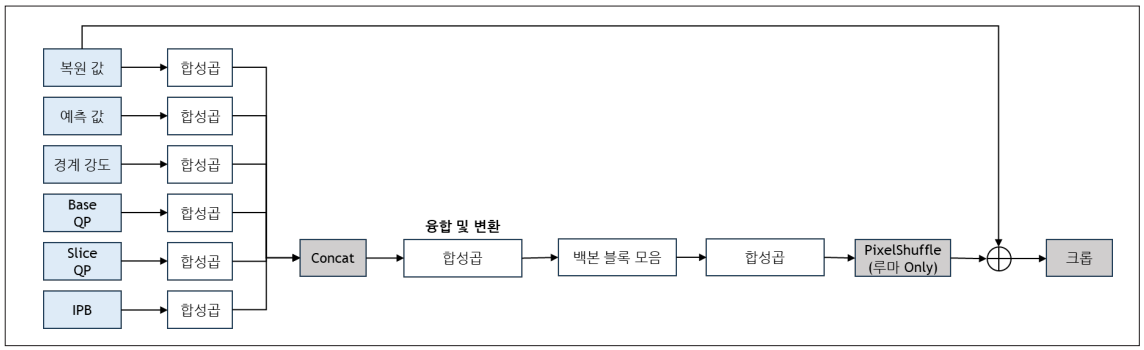
HOP NNLF는 〈그림 4〉와 같이 VLOP NNLF와 LOP NNLF와 달리 YUV 정보를 분기하지 않고 통합적으로



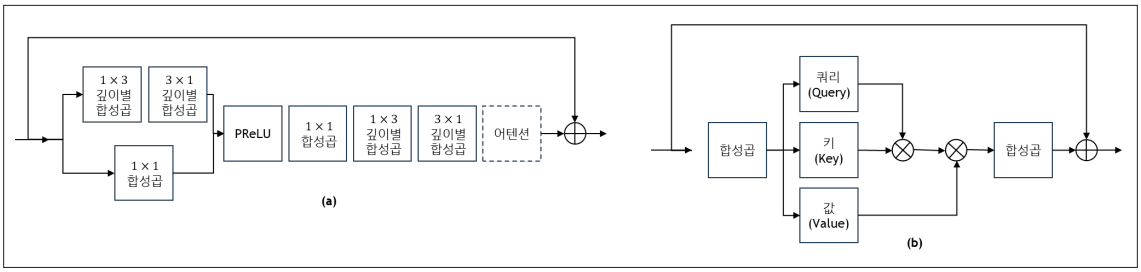
<그림 2> VLOP NNLF와 LOP NNLF의 모델 구조



<그림 3> VLOP NNLF와 LOP NNLF에서 사용되는 블록 모음. (a) 백본 블록, (b) 병렬 구조 백본 블록, (c) 트윈 블록, (d) 트리플 블록, (e) 어텐션



<그림 4> HOP NNLF의 모델 구조



<그림 5> HOP NNLF에서 사용되는 블록 모음. (a) 백본 블록, (b) 어텐션

처리하는 구조를 갖는다. 또한 HOP NNLF의 백본 블록의 어텐션 모듈은 트랜스포머(transformer)[12]에서 제안된 쿼리-키-값(query-key-value) 기반 어텐션 메커니즘을 변형한 구조를 채택하여, 고성능을 목표로 설계되었다.

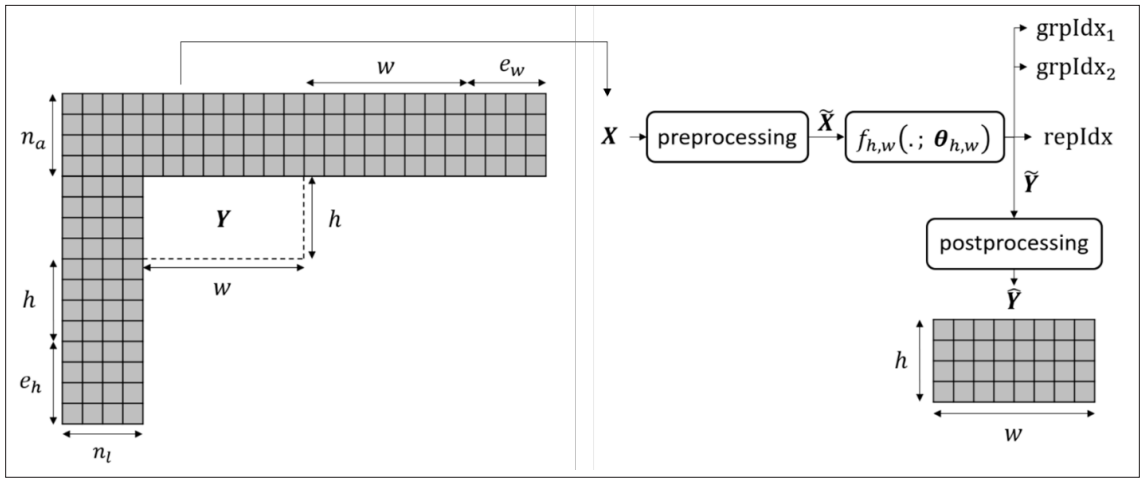
NNLF를 사용하면 인코더와 디코더의 복잡도가 증가할 수 있으므로, 복잡도-성능 균형을 고려하여 NNLF를 적응적으로 제어하기 위한 다양한 알고리즘이 채택되었다. [14]에서는 디코더 복잡도를 줄이기 위해 경계 강도(boundary strength) 분할 정보를 이용한 필터링 제어 방법을 제안하였다. 경계 강도 분할은 경계 강도 에지(edge)로 둘러싸인 영역을 의미하며, 한 블록 내 경계 강도 분할 수가 많을수록 해당 블록이 NNLF로 개선될 가능성이 높다고 가정한다. 이에 따라 인코더는 블록별 경계 강도 분할 수를 계산하고, 이를 임계값(threshold)과 비교하여 NNLF 적용 여부를 결정한다. 이때 슬라이스 단위 비용 함수 Cost= $D+\lambda\cdot N_{\text{filt}}$를 최소화하는 임계값이 선택되며, 여기서 D는 원본 샘플과 최종 복원 샘플 간 왜곡, N_{filt}는 NNLF가 적용된 블록의 개수를 의미한다. 또한 λ는 슬라이스 QP에 따라 설정 가능한 가중치로, 왜곡 감소와 복잡도 증가간의 균형을 조절한다. 이를 통해 필터링 이득이 적은 블록에 대한 불필요한 연산을 생략함으로써 디코딩 시간을 단축할 수 있다. 또한 [15]에서는 GOP 내 계층적 깊이(hierarchical depth)에 따라 NNLF의 활성화를 프레임 수준에서 제한하는 필터링 제어 방법을 제안하였다.

2. 신경망 기반 화면 내 예측(NN-Intra)

화면 내 예측은 현재 프레임에서 이미 복호화된 인접 블록의 참조 샘플을 이용하여 현재 부호화 대상 블록을 예측하고, 예측 오차(잔차)를 부호화하는 방식이다. 신경망 기반 화면 내 예측(NN-Intra)은 참조 샘플을 학습된 신경망의 입력으로 하여 예측을 수행하는 방식이다.

NN-Intra는 사전 정의된 블록 크기 집합 T ={4×4,8×4,16×4,32×4,8×8,16×8,16×16}에 대해 각각 학습된 7개의 신경망을 사용한다. 추가적으로 부호화 블록의 (h, w)가 T에 속하지 않더라도, 〈표 2〉에 정의된 규칙에 따라 참조 샘플을 업샘플링/다운샘플링 및 전치(transposition)하여 T에 속하는 형태로 변환함으로써 NN-Intra를 적용할 수 있다. 〈그림 6〉의 예와 같이, 참조 샘플을 전처리한 후 MLP(Multi-Layer Perceptron) 기반 신경망 $f_{(h,w)}(\cdot;\theta_{(h,w)})$에 입력하고, 신경망에 의해 LFNST 커널 선택을 위한 인덱스 grpIdx1, grpIdx2와 예측 결과 $\tilde{Y}$를 출력한다. $\tilde{Y}$는 후처리를 거쳐 최종 예측 블록 $\hat{Y}$를 생성한다. 또한 신경망은 $\hat{Y}$를 가장 잘 근사하는 VVC 화면 내 예측 모드(PLANAR, DC, 또는 방향성 모드)의 인덱스 repIdx ∈ ⟦0,66⟧도 함께 출력한다.

참조 샘플의 개수는 현재 부호화 블록의 크기에 따라 달라지며, 〈그림 7〉 (a)에 기술된 규칙에 따라 구성된다. 또한 신경망의 학습환경과 동일하게 처리하기 위해 전처리 단계에서는 〈그림 7〉 (b)의 예시에서 참조 가능 샘플 $\bar{X}$를 $\bar{X}$의 평균과의 차분으로 대체하고, 참조 불가능 샘플



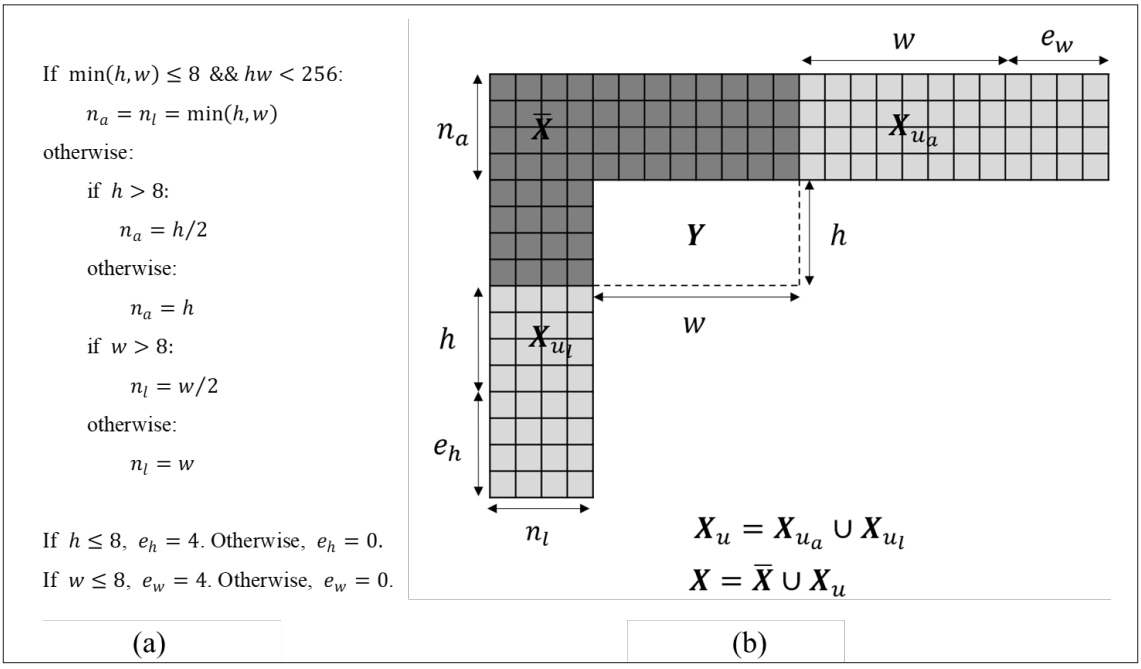
<그림 6> NN-Intra 수행 절차(예: h=4, w=8)[13]

<표 2> 부호화 블록 크기에 따른 사용 신경망 표. T에 속하지 않은 경우 γ,δ를 이용하여 업샘플링 및 다운샘플링을 수행하며, 필요 시 전치(transposition)를 적용[13]

height and width of the block to be predicted (h, w)	γ	δ	transposition	neural network used for prediction
(4, 4)	1	1	no	$f_{4,4}(.,\boldsymbol{\theta}_{4,4})$
(4, 8)	1	1	no	$f_{4,8}(.,\boldsymbol{\theta}_{4,8})$
(8, 4)	1	1	yes	$f_{4,8}(.,\boldsymbol{\theta}_{4,8})$
(4, 16)	1	1	no	$f_{4,16}(.,\boldsymbol{\theta}_{4,16})$
(16, 4)	1	1	yes	$f_{4,16}(.,\boldsymbol{\theta}_{4,16})$
(4, 32)	1	1	no	$f_{4,32}(.,\boldsymbol{\theta}_{4,32})$
(32, 4)	1	1	yes	$f_{4,32}(.,\boldsymbol{\theta}_{4,32})$
(8, 8)	1	1	no	$f_{8,8}(.,\boldsymbol{\theta}_{8,8})$
(8, 16)	1	1	no	$f_{8,16}(.,\boldsymbol{\theta}_{8,16})$
(16, 8)	1	1	yes	$f_{8,16}(.,\boldsymbol{\theta}_{8,16})$
(8, 32)	2	1	no	$f_{8,16}(.,\boldsymbol{\theta}_{8,16})$
(32, 8)	1	2	yes	$f_{8,16}(.,\boldsymbol{\theta}_{8,16})$
(16, 16)	1	1	no	$f_{16,16}(.,\boldsymbol{\theta}_{16,16})$
(16, 32)	2	1	no	$f_{16,16}(.,\boldsymbol{\theta}_{16,16})$
(32, 16)	1	2	no	$f_{16,16}(.,\boldsymbol{\theta}_{16,16})$
(32, 32)	2	2	no	$f_{16,16}(.,\boldsymbol{\theta}_{16,16})$
(64, 64)	4	4	no	$f_{16,16}(.,\boldsymbol{\theta}_{16,16})$

X_u는 0으로 채운다. 이후 부동소수점(float) 모델의 경우 $1/2^{(b-8)}$을, 정수(integer) 모델의 경우 $1/2^{(Q_{in}-b+8)}$을 곱한다. 여기서 b는 internal bitdepth, Q_{in}은 입력 양자화 계수를 의미한다. 후처리 단계에서는 전처리에서 사용된 값 ($\bar{X}$의 평균, $1/2^{(b-8)}$ 또는 $1/2^{(Q_{in}-b+8)}$)을 이용하여 예측 샘플 값을 역변환하여 최종 예측 값으로 출력한다.

NN-Intra의 루마 시그널링의 경우, 현재 부호화 블록의 크기 (h, w)가 블록 크기 집합 T에 속하면 nnFlag가 신호화되며, nnFlag가 1인 경우 해당 루마 블록은 NN-Intra로 예측된다. 반대로 nnFlag가 0이면 NN-Intra는 사용되지 않으며, 기존 VVC의 화면 내 예측 모드 시그널링 절차에 따라 예측된다. 한편 $(h, w) \notin T$인 경우에는 nnFlag



<그림 7> (a) 블록 크기에 따른 참조 샘플 개수 결정 규칙, (b) h=4,w=8 블록에서의 참조 샘플 선택 예시[13]

자체가 신호화되지 않고, 항상 기존 화면 내 예측 모드로 처리된다.

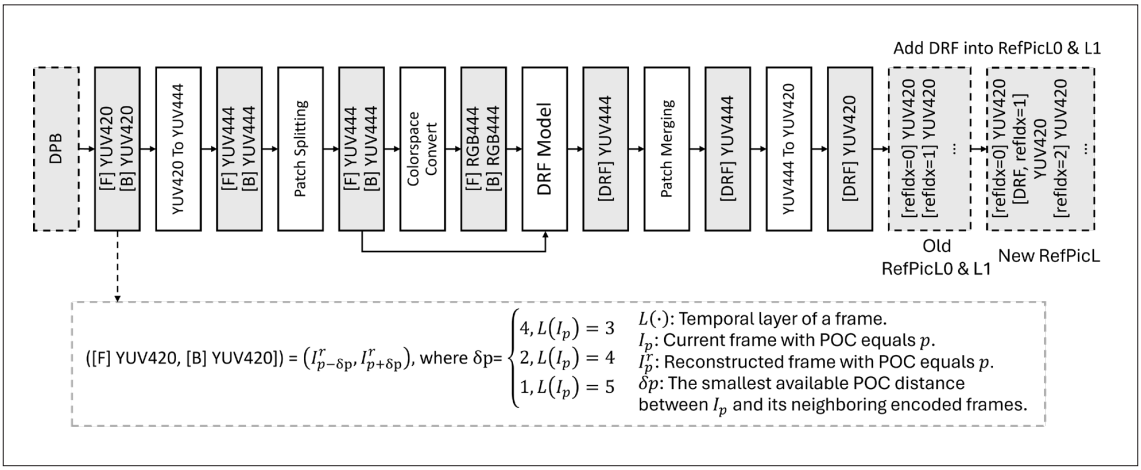
또한 루마 CB의 MPM(Most Probable Mode) 리스트를 구성하는 과정에서도 NN-Intra의 결과가 반영될 수 있다. 구체적으로, 좌측 루마 CB가 NN-Intra로 예측된 경우 해당 블록의 모드 인덱스는 NN-Intra 과정에서 산출된 repIdx로 대체될 수 있으며, 이 repIdx는 현재 블록의 MPM 리스트에 포함되는 후보 인덱스로 사용된다. 상단 루마 CB가 NN-Intra로 예측된 경우에도 동일하게, 상단 블록에서 산출된 repIdx를 후보 인덱스로 사용하여 MPM 리스트에 삽입할 수 있다. 즉, NN-Intra를 위해 새로운 모드 인덱스를 정의하기보다, 기존 VVC 인트라 모드 인덱스 범위 내의 값을 활용하는 방식이다

3. 신경망 기반 화면 간 예측(NN-Inter)

화면 간 예측은 시간적으로 인접한 프레임들(참조 프레임)로부터 현재 프레임을 예측하고, 예측 오차를 부호화하여 압축 효율을 높이는 기술이다. 신경망 기반 화면 간 예측(NN-Inter)은 Random Access의 양방향(bi-directional) 예측에만 사용되며, 양방향 참조 프레임들을 입력으로 받아 신경망이 새로운 참조 프레임을 합성하고, 이를 참조 프레임 리스트(Reference Picture List, RPL)에 삽입하여 양방향 예측 성능을 향상시키는 것을 목적으로 한다.

Random Access 환경에서 양방향 예측을 수행할 때 신경망 네트워크는 시간적으로 대칭이며 POC(Picture Order Count) 거리가 서로 동일하고 그 중에서도 가장 작은 거리를 갖는 두 개의 복원 참조 프레임을 입력으로 사용하며, 참조 프레임보다 높은 temporal layer에 속하는 프레임에만 적용되는데 실제로는 주로 temporal layer 3, 4, 5에 속한 프레임에서 사용되고 계층적 GOP 구조에 따라 입력 프레임 조합이 달라질 수 있다.

신경망에 의해 생성된 프레임은 RPL0과 RPL1 모두

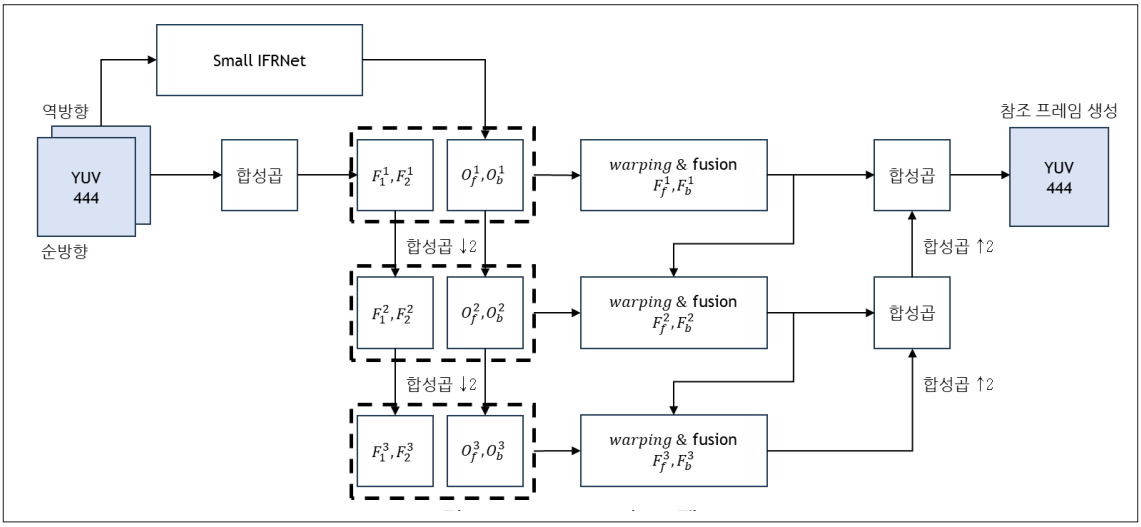


<그림 8> NN-Inter 추론 과정. 순방향(Forward, F) 프레임, 역방향(Backward, B) 프레임을 YUV420->YUV444->RGB444로 변환 및 패치로 분할(Patch Splitting)하여, DRF(Deep Reference Frame)를 패치단위 수행 후, 출력 패치들을 병합(Merging)하여 참조 프레임을 생성한다. 이때 현재 POC의 p번째 프레임 I_p의 temporal level $L(I_p)$에 따라, POC 거리 δ_p가 가장 작은 두 개의 참조 프레임을 입력으로 선택한다[13].

의 두 번째 위치에 삽입되며, 생성 프레임의 POC는 현재 인코딩 중인 프레임과 동일하게 설정되어 현재 프레임과 같은 시점에 존재하는 가상의 참조 프레임처럼 취급된다.

추론 과정에서는 〈그림 8〉과 같이 240×240 크기의 패치로 분할한 뒤, 입력 프레임의 경계를 고려하여 각 패치를 상·하·좌·우로 8픽셀씩 확장한 후, 신경망 모델에 입력하고, 출력 패치를 입력 패치의 원본 영역에 맞춰 잘라 이어 붙여 RPL에 삽입한다.

아래 〈그림 9〉의 NN-Inter 네트워크는 두 개의 입력 프레임(순방향(forward), 역방향(backward))에 대해 각각 독립적으로 일련의 합성곱 연산을 적용하여 다중 스케일 특징 F_1^l과 F_2^l $(l \in \{1,2,3\})$을 생성한다. 또한 두 입력 프레임 사이의 optical flow는 다중 스케일에서 추정



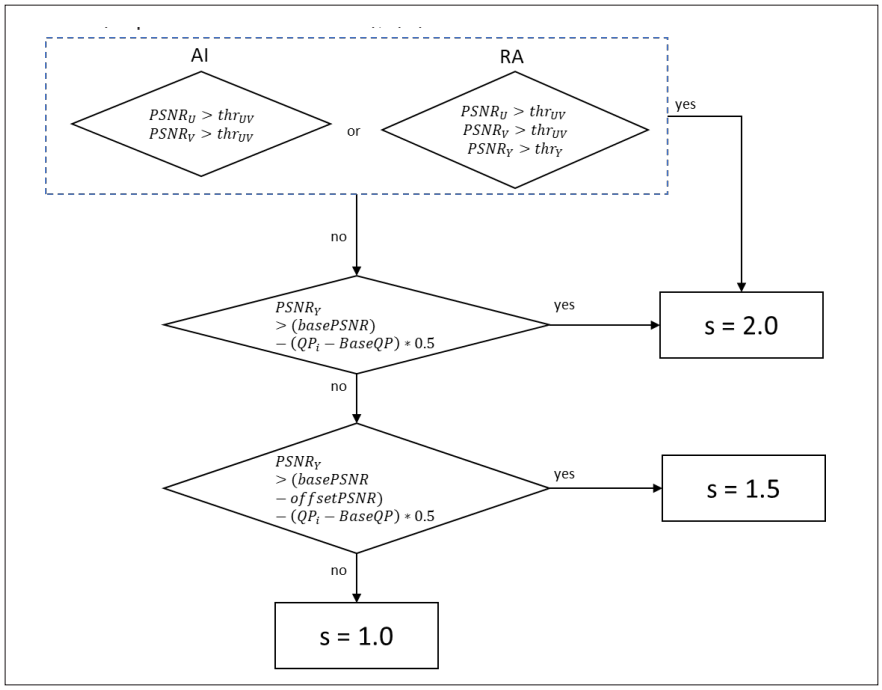
<그림 9> NN-Inter의 모델 구조

되며, 구체적으로 Small IFRNet을 사용하여 순방향 및 역방향 optical flow를 추정하고, 얻어진 optical flow 특징은 추가 다운샘플링을 통해 다중 스케일 optical flow O_f^l과 $O_b^l\,(l \in \{1,2,3\})$를 생성한다. 이후 다중 스케일 특징과 optical flow를 스케일별로 warping을 수행하여, F_f^l과 F_b^l $(l \in \{1,2,3\})$를 생성하고, 이를 융합(fusion)한다. 다중 스케일에서 얻은 융합된 특징들은 참조 프레임 생성을 위한 업샘플링 과정에서 함께 연산되며, 최종적으로 하나의 참조 프레임을 생성한다.

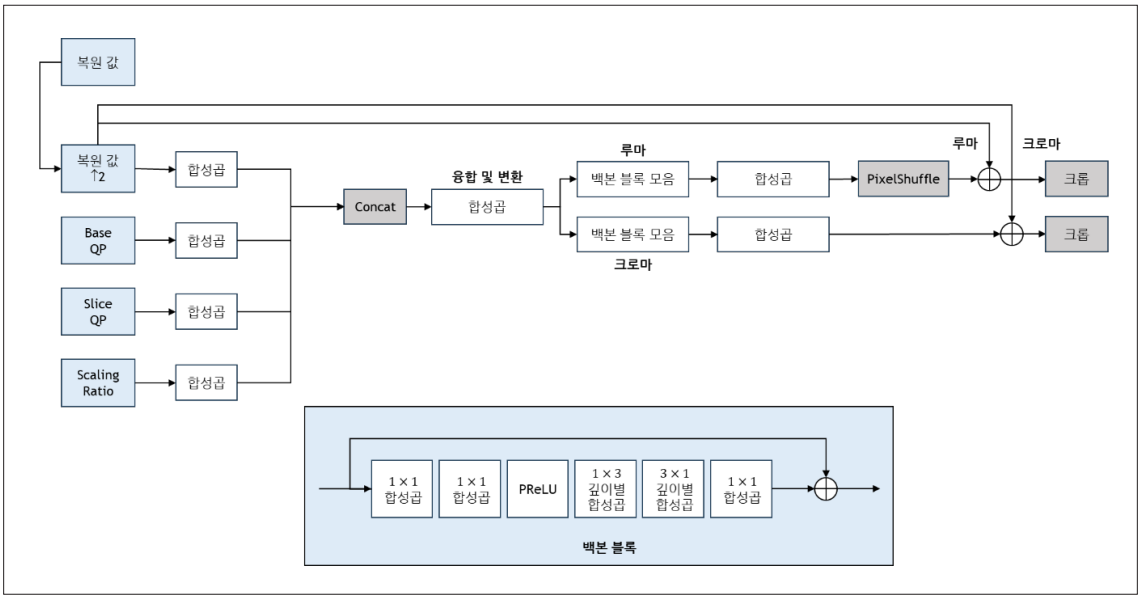
다만 생성 프레임에는 움직임 벡터 정보가 존재하지 않으므로, 현재 프레임과 동일 위치(collocated) 프레임 및 각 참조 프레임 간 POC 거리를 이용해 동일 위치 CU(Coding Unit)의 움직임 벡터를 스케일링하여 유도하는 TMVP(Temporal Motion Vector Prediction)는 사용할 수 없으며, 따라서 현재 프레임의 참조 프레임이 신경망으로 예측했거나 collocated 프레임의 참조 프레임이 신경망으로 예측한 경우 TMVP 후보를 생성하지 않는다.

4. 신경망 기반 초해상화(NNSR)

신경망 기반 초해상화 기술은 입력 영상의 해상도를 낮춰 공간적 중복성을 줄인 상태로 부호화한 뒤, 디코더에서 신경망 기반 초해상화(NNSR)를 통해 원본 해상도로 복원하는 방법이다. 해상도를 낮추면 필요한 비트량이 감소하고 공간적 중복성이 완화되므로, 적절한 복원 기법을 사용할 경우 해상도를 유지한 채 부호화하는 방식 대비 복원 화질 저하를 방지할 수 있다. NNVC에서는 다운샘플링에 Reference Picture Resampling(RPR)을 사용하고 업샘플링에는 NNSR를 적용하며, 인코더는 GOP 단위로 해상도 변경 여부와 샘플링 비율을 결정한다. 구체적으로 $QP \geq 32$에서 각 GOP의 첫 프레임을 대상으로 RPR 기반 다운-업샘플링을 수행한 뒤 원본 대비 각 색차 성분에 대한 $PSNR_Y$, $PSNR_U$, $PSNR_V$를 측정하여 샘플링 비율(1.0, 1.5, 2.0)을 선택하는데, RA에서는 $PSNR_U$와 $PSNR_V$가 thr_{UV}를, $PSNR_Y$가 thr_Y를 모두 초과



<그림 10> GOP 수준 인코딩 해상도 결정 알고리즘[13]



<그림 11> NNSR 네트워크 구조

하면 2.0을 선택하고, AI에서는 $PSNR_U$와 $PSNR_V$가 thr_{UV}를 초과하면 2.0을 선택한다. 위 조건을 만족하지 않는 경우에는 루마 PSNR 기준으로 $PSNR_Y$〉 *basePSNR* - (QP_i - *baseQP*)·0.5이면 2.0, 그렇지 않고 $PSNR_Y$〉(*basePSNR*-*offsetPSNR*) - (QP_i - *baseQP*)·0.5이면 1.5, 그 외에는 1.0을 선택하며, *basePSNR*과 *offsetPSNR*은 사전 정의된 임계값(단, *offsetPSNR*은 RA/AI에 따라 상이함)을 의미한다. 또한 선택된 스케일링 비율이 2.0 또는 1.5인 경우에는 해당 비율에 대응하는 QP 오프셋을 추가로 설정하여 부호화를 수행한다.

〈그림 11〉은 NNVC Software-15.0에서 사용하는 NNSR 모델 구조이다. 입력으로는 BaseQP, SliceQP, Scaling Ratio, 복원 값을 사용하며, 복원 값을 먼저 RPR로 업샘플링을 수행한 후, 모델의 입력으로 사용한다. NNSR의 백본 블록은 (i) PReLU 비선형 함수, (ii) 1×1 합성곱, (iii) 3×3 커널 기반의 깊이별 합성곱(depth-wise convolution), (iv) 1×1 합성곱 순서로 구성된다.

5. 적응적 신경망 기반 인루프 필터 (Adaptive NNLF)

적응적 신경망 기반 인루프 필터(adaptive NNLF)는 NNLF의 기본 네트워크 구조는 유지하되, 합성곱 출력에 곱셈 계수(multiplier) *m*을 도입하여 $\sigma((W*x+b)\cdot m)$형태로 활성값(activation)을 콘텐츠 특성에 맞게 스케일링함으로써 필터 출력을 조정하는 방식이다. 인코더는 시퀀스의 RA 세그먼트 단위로 *m*을 오버피팅(overfitting)하고, 기본 곱셈 계수 대비 변화량(업데이트)을 Neural Network Compression and Representation(NNR) 방식으로 압축한 뒤 RA 세그먼트 내 첫 번째 B-프레임의 Neural Network Filter Update Adaptation Parameter Set(APS)에 탑재하여 비트스트림으로 전송한다. 또한 전송·저장해야 하는 파라미터 수를 추가로 줄이기 위해, 곱셈 계수를 저랭크(low-rank) 형태로 분해하여 학습하는 DeComposed Adaptive(DCA) 방식도 제안되었다. 마지막으로 인코더는 각 RA 세그먼트에 대해 기본 NNLF 적용

결과와 adaptive NNLF 적용 결과를 비교하여(⊿PSNR) 세그먼트별 adaptive NNLF 사용 여부를 독립적으로 결정하며, 인접 RA 세그먼트 간 중첩되는 I-프레임은 adaptive NNLF 대신 기존 NNLF로 필터링한다.

III. NNVC Tool-based 기술의 성능 및 복잡도 분석

본 장에서는 NNVC의 Tool-based 기술들이 제공하는 압축 성능 향상과 그에 수반되는 복잡도 증가(연산량 및 처리 시간)를 정량적으로 분석한다. NNVC 기술이 제안 및 채택되기 위해서는 JVET에서 정의한 공통 테스트 조건(Common Test Conditions, CTC)[16]을 준수해야 하며, 제안 도구는 외부 의존성 없이 순수 C++ 환경에서 VVC 코덱 소프트웨어에 직접 통합 가능하도록 제공되는 SADL(Small Ad-hoc Deep-learning Library)[17] 기반 구현을 요구한다.

1. 공통 테스트 조건(Common Test Condition, CTC)

[16]에서는 NNVC 실험이 공통적으로 준수해야 하는 실험 환경, 절차, 평가지표 등을 규정한다. 실험은 4가지 코딩 조건(All Intra, Random Access, Low Delay B, Low Delay P)에서 수행되어야 하며, 결과 보고 시 Random Access(RA) 또는 Low Delay(LD) 계열 중 최소 하나 이상의 조건에 대한 성능 결과를 필수로 포함해야 한다. 테스트 시퀀스는 해상도 기준으로 Class A, B, C, D로 구분되며, 콘텐츠 특성에 따라 Class E, F, H2 등의 추가 클래스로 구성된다. 평가 지표로는 객관적 화질 지표인 PSNR을 필수로 사용하고, MS-SSIM은 권장 지표로 제시한다. 또한 이러한 화질 지표를 기반으로 BD-Rate, CPU 인코딩 시간(Encoding Time, EncT), CPU 디코딩 시간(Decoding Time, DecT)을 함께 보고하도록 한다. 더불어 신경망 기반 도구의 복잡도 평가를 위해 kMAC/pixel, 파라미터 수 등 모델의 복잡도 지표를 포함하며, 학습이 수행된 경우에는 에포크(epoch), 배치 사이즈(batch size), 학습 시간, 손실 함수 등을 함께 기술하도록 권고한다. 마지막으로 추론 크로스체크를 요구하며, 학습 기반 방법의 경우 상이한 플랫폼에서도 학습 결과가 재현될 수 있도록 학습 크로스체크를 포함한다.

2. Small Ad-hoc Deep-learning Library(SADL)

SADL은 PyTorch나 TensorFlow로 학습한 모델을 C++ 환경에서 돌리기 위해 사용하는 경량 라이브러리이다. 학습한 모델을 ONNX 포맷으로 변환 후, SADL 전용 포맷으로 변환하여 사용한다. 외부 의존성 없이 C++로만 되어 있기 때문에 VVC 코덱 소프트웨어에 바로 붙이기 용이하고, 하드웨어 구현을 고려하여 Integer 추론을 지원한다.

3. 성능 및 복잡도 분석

〈표 3〉에서는 NNVC에서 사용하고 있는 기술들의 성능 및 복잡도를 비교하여 제시한다. VTM(VVC Test model) 대비 Anchor(LOP6+NN-Intra)의 Y BD-Rate 성능은 RA에서 약 8.01%, AI에서 약 8.59%, LDB에서 약 6.09% 개선된다. 또한 LOP6를 다른 복잡도의 NNLF로 대체할 경우, VLOP4는 VTM 대비 RA 약 6.07%, AI 약 7.14%, LDB 약 3.9% 개선을 보이며, HOP5는 VTM 대비 RA 약 14.07%, AI 약 12.96%, LDB 약 10.4%로 큰 성능 향상을 보인다. 다만 복잡도가 증가함에 따라 인코딩 시간 및 디코딩 시간이 증가하며, 특히 HOP5의 경우 고복잡도 연산으로 인해 디코딩 시간이 급격히 증가하는 경향을 보인다.

〈표 4〉에서는 Anchor 기술에 추가적으로 NN-Inter 및 NNSR를 적용했을 때의 성능을 보인다. NN-Inter는 RA

조건에서만 사용하며, Anchor 기술 대비 성능 개선을 보이지만 높은 연산량으로 인해 디코딩 시간이 Anchor 대비 약 22.6배로 증가한다. NNSR은 RA 및 AI 조건에서만 사용되며, Class A1과 A2 (4K 해상도) 시퀀스에 한해 적용한다. 〈표 4〉의 결과는 전체 Class 평균에 대한 결과이며, Class A 평균 기준으로 NNSR 적용 시 Y BD-Rate는

<표 3> VTM 대비 NN-Intra+NNLF 기술의 성능 (NN-Intra 파라미터 수: 1.3M, kMAC/pixel: 4.8k)

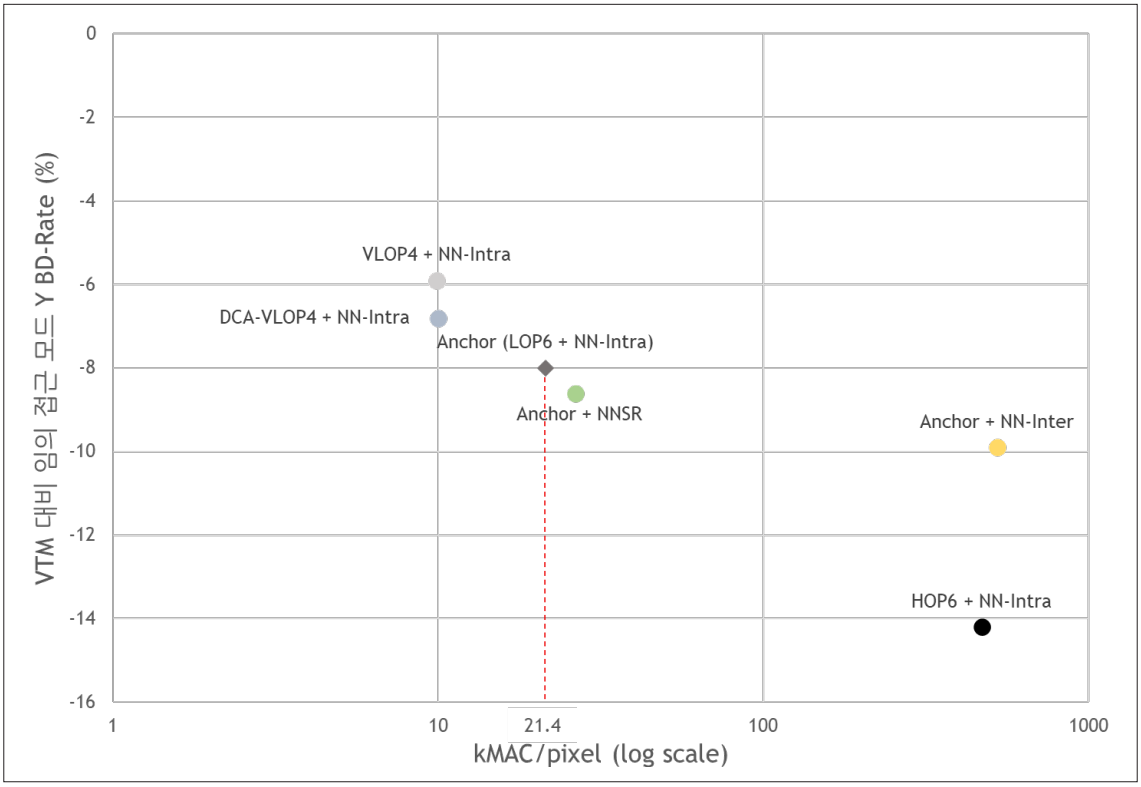
카테고리	복잡도		VTM-23.11 대비 성능 (YUV BD-Rate, Encoding Time, Decoding Time)															
	파라미터 수	MAC/ pixel	Random Access					All Intra					Low Delay B					
			Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	
Anchor (LOP6+NN-Intra)	1.5M	21.4k	-8.01%	-14.92%	-13.53%	116%	2678%	-8.59%	-15.19%	-15.17%	160%	2204%	-6.09%	-9.71%	-9.59%	109%	2998%	
VLOP4+NN-Intra	1.4M	9.9k	-6.07%	-7.52%	-6.03%	113%	1373%	-7.14%	-8.92%	-8.60%	158%	1265%	-3.90%	-7.09%	-3.42%	105%	1578%	
HOP5+NN-Intra	2.7M	471k	-14.07%	-19.84%	-20.13%	340%	117985%	-12.96%	-15.45%	-16.79%	272%	83613%	-10.40%	-10.99%	-6.37%	385%	123979%	

<표 4> Anchor 대비 NN-Inter와 NNSR 기술의 성능

카테고리	복잡도		Anchor(LOP6+NN-Intra) 대비 성능 (YUV BD-Rate, Encoding Time, Decoding Time)															
	파라미터 수	MAC/ pixel	Random Access					All Intra					Low Delay B					
			Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	
Anchor+NN-Inter	5.3M	525k	-1.91%	-1.17%	-0.81%	145%	2256%	NA					NA					
Anchor+NNSR	1.5M	26.4k	-0.6%	1.3%	0.9%	97%	90%	-0.8%	3.1%	3%	100%	81%	NA					

<표 5> VLOP4 대비 Adaptive VLOP4 기술의 성능

카테고리	복잡도		VLOP4+NN-Intra 대비 성능 (YUV BD-Rate, Encoding Time, Decoding Time)															
	파라미터 수	MAC/ pixel	Random Access					All Intra					Low Delay B					
			Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT	
DCA-VLOP4+NN-Intra	1.4M	10k	-0.9%	-1.1%	-1.2%	200%	97%	NA					NA					



<그림 12> kMAC/pixel을 고려했을 때, VTM 대비 NNVC 기술의 RA Y BD-Rate 성능

Anchor 대비 RA에서 약 1.5%, AI에서 약 2.5% 개선되나, UV BD-Rate는 RA에서 약 3%, AI에서 약 9% 성능 저하가 관찰된다. NNSR은 해상도를 줄여서 부호화하는 구조적 특성으로 인해 Anchor 대비 인코딩 시간 및 디코딩 시간이 감소한다.

〈표 5〉에서는 VLOP4+NN-Intra 대비 DeComposed Adaptive(DCA) 기술을 적용했을 때의 성능을 보인다. 제안 기술이 테스트 시퀀스에 대해 오버피팅을 수행했기 때문에, 적은 복잡도 증가에도 불구하고 RA에서 Y BD-Rate가 약 0.9% 개선된다.

마지막으로 〈그림 12〉는 상기 기술들에 대해 VTM 대비 RA 조건의 Y BD-Rate 성능을 kMAC/pixel(log scale)을 함께 보인 것으로, 성능 향상과 연산 복잡도 간의 trade-off를 시각적으로 비교한다.

NNVC Tool-based 기술은 VTM 대비 유의미한 BD-Rate 개선을 제공하지만, 일반적으로 성능 향상폭이 커질수록 파라미터 수와 kMAC/pixel이 증가하며 이에 따른 처리 시간(특히 디코딩 시간) 부담이 동반된다. 예를 들어 HOP 계열은 가장 큰 성능 이득을 보이지만 디코딩 시간이 급격히 증가하여 적용 가능한 사용 시나리오가 제한될 수 있다. LOP 및 VLOP은 상대적으로 인코딩 시간 증가가 크지 않아 적용 가능성이 높으나, 여전히 디코딩 시간이 적지 않다. NN-Inter 역시 Anchor 대비 추가적인 성능 향상을 제공하지만 디코딩 시간 증가 폭이 매우 커 HOP과 유사하게 적용 범위가 제한될 가능성이 크다. 반면 NNSR은 해상도 축소 기반 구조로 인해 인코딩 및 디코딩 시간이 감소하는 장점이 있으나, 다른 도구들에 비해 BD-Rate 개선 폭은 상대적으로 제한적이다.

DCA(VLOP4)는 비교적 낮은 복잡도 증가로 성능 향상을 보였으나, 시퀀스(또는 세그먼트) 단위 오버피팅이 필요하다는 점에서 실사용 시나리오가 제한적이다. 신경망 추론을 위해 GPU 또는 NPU 기반 가속을 활용하는 것은 하나의 대안이 될 수 있으나, [18]에서 지적하듯 CPU-가속기 간 데이터 이동 오버헤드가 발생할 수 있어 근본적인 해결책으로 보기 어렵다. 따라서 NNVC tool-based 기술의 실용적 적용을 위해서는 파라미터 수와 kMAC/pixel 등 모델 복잡도를 체계적으로 관리하는 동시에, 인코딩/디코딩 시간을 절감할 수 있는 저복잡도 설계 및 구현 최적화가 필수적이다.

IV. 결 론

본 고에서는 NNVC Software-15.0에 채택된 Tool-based 기술들의 구조와 성능을 분석하였다. NNVC Tool-based 기술들은 기존 VVC 표준 대비 높은 부호화 효율 향상을 입증하며 차세대 비디오 코딩 표준 후보 기술로서의 잠재력을 보여주었다.

Anchor 기술은 21.4 kMAC/pixel의 연산량으로 VTM 대비 RA Y BD-Rate 약 8%의 성능 향상을 달성하였으며, VLOP4+NN-Intra 구성의 경우 Anchor 대비 연산량을 줄이면서도 VTM 대비 약 6%의 성능 향상을 기록하였다. 이는 kMAC/pixel을 고려할 때 유의미한 효율 개선이라 할 수 있다.

그러나 가장 낮은 복잡도를 갖는 VLOP NNLF조차 디코딩 시간이 VTM 대비 약 13배 소요되는 한계가 확인되었다. 따라서 향후 연구는 압축 성능의 고도화뿐만 아니라, 모델 경량화 및 하드웨어 친화적 설계를 통해 연산 복잡도를 획기적으로 낮추고 실용적인 구현 가능성을 높이는 방향으로 전개되어야 할 것이다.

참 고 문 헌

[1] Recommendation ITU-T H.266 | ISO/IEC 23090-3, “Versatile video coding,” 2020.
[2] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” 2016.
[3] T. Shao, P. Yin, S. McCarthy (Dolby), J. N. Shingala, A. Shyam, A. Suneja, S. P. Badya (Ittiam), “EE1-1.2: LOP5 improvement with parallel 1x3/3x1 Backbone”, document JVET-AL0084, 38th Meeting, teleconference, 26 March-04 April 2025.
[4] Y. Li, M. Coban, M. Karczewicz, L. Kerofsky (Qualcomm), “EE1-2.1: Improved VLOP Attention with SIMD acceleration”, document JVET-AM0135, 39th Meeting, Daejeon, KR, 26 June - 4 July 2025.
[5] Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm), F. Galpin (Interdigital), Y. Li, J. Li, C. Lin, K. Zhang, L. Zhang (Bytedance), “EE1 (AhG 11): On the complexity adjustment of HOP4”, document JVET-AI0172, 35th Meeting, Sapporo, JP, 12-19 July 2024.
[6] T. Dumas, F. Galpin, P. Bordes (Interdigital), “EE1-5.1: combination of the neural network-based intra prediction mode and ISP”, document JVET-AI0130, 35th Meeting, Sapporo, JP, 12-19 July 2024.
[7] X. Chen, N. Fu, W. Zhang, J. Zhang, D. Ding, W. Ma, Z. Chen (Wuhan Univ.), “EE1-3.2: Deep Reference Frame Generation for Inter Prediction Enhancement”, document JVET-AM0175, 39th Meeting, Daejeon, KR, 26 June - 4 July 2025.
[8] J. Ye, Q. Liu (HUST), Z. Lv (vivo), “EE1-4.1 - Wavelet transform for super-resolution loss function”, document JVET-AJ0056, 36th Meeting, Kemer, TR, 01-08 November 2024.
[9] Z. Xu, J. Konieczny, A. Filippov, C. Hollmann, V. Rufitskiy, T. Dong (TCL), “EE1-1.3 Dimension-wise decomposed representation of multiplier for content-adaptive loop filtering”, document JVET-AL0169, 38th Meeting, teleconference, 26 March - 04 April 2025.
[10] Z. Xu, J. Konieczny, A. Filippov, C. Hollmann, V. Rufitskiy, T. Dong, H. Qin (TCL), “EE1-2.2: Decomposed Content-Adaptive VLOP4”, document JVET-AN0153, 40th Meeting, Geneva, CH, 03 - 12 October 2025.
[11] F. Galpin, Yue Li, Yun Li, D. Rusanovskyy, T. Shao, J. Ström, L. Wang, “Description of algorithms version 13 and software version 15 in neural network-based video coding (NNVC)”, document JVET-AN2019, 40th Meeting, Geneva, CH, 03 - 12 October 2025.
[12] Vaswani, Ashish, et al. “Attention is all you need.” Advances in neural information processing systems 30 (2017).
[13] Franck Galpin, Yue Li, Yun Li, Dmytro Rusanovskyy, Tong Shao, Jacob Ström, Liqiang Wang, “Description of algorithms version 13 and software version 15 in neural network-based video coding (NNVC)”, document JVET-AN2019, 40th Meeting: Geneva, CH, 3-12 October 2025.
[14] H. Kwon, J. Seo, H. Ko (HYU), D. Kim, S.-C. Lim (ETRI), “EE1-1.5: Adaptive skip of LOP filtering based on boundary strength partitions”, document JVET-AK0093, 37th Meeting, Geneva, CH, 14-22 January 2025.
[15] F. Galpin, E. François (InterDigital), “AhG11: Filter restriction depending on hierarchical depth”, document JVET-AN0215, 40th Meeting, Geneva, CH, 03 - 12 October 2025.
[16] E. Alshina, F. Galpin, R.-L. Liao, S. Liu, A. Segall, “Common test conditions and evaluation procedures for neural network-based video coding technology”, document JVET-AJ2016, 36th Meeting, Kemer, TR, 01-08 November 2024.
[17] https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/sadl
[18] T. Hsieh, W.-J. Chien, V. Seregin, M. Karczewicz (Qualcomm), “Neural Network Analysis for Next-Generation Video Compression Standard”, document JVET-AK0323, 37th Meeting, Geneva, CH, 14-22 January 2025.

저 자 소 개

조 현 동

- 2023년 8월 : 경희대학교 소프트웨어융합 학사
- 2023년 8월 ~ 2025년 8월 : 경희대학교 컴퓨터공학부 석사
- 2025년 9월 ~ 현재 : 경희대학교 컴퓨터공학부 박사 과정
- 주관심분야 : 비디오 부호화, 딥러닝, 컴퓨터비전, 표준화

김 휘 용

- 1994년 8월 : KAIST 전기및전자공학과 공학사
- 1998년 2월 : KAIST 전기및전자공학과 공학석사
- 2004년 2월 : KAIST 전기및전자공학과 공학박사
- 2003년 8월 ~ 2005년 10월 : ㈜애드팍테크놀러지 멀티미디어팀 팀장
- 2005년 11월 ~ 2019년 8월 : 한국전자통신연구원(ETRI) 실감 AV연구그룹 그룹장
- 2013년 9월 ~ 2014년 8월 : Univ. of Southern Califonia (ISC) Visiting Scholar
- 2019년 9월 ~ 2020년 2월 : 숙명여자대학교 전자공학전공 부교수
- 2020년 3월 ~ 현재 : 경희대학교 컴퓨터공학과 정교수
- 주관심분야 : 비디오 부호화, 딥러닝 영상처리, 디지털 홀로그램