TU 블록 크기에 따른 CNN기반 인루프필터

*김양우 +정세윤 ++조승현 **이영렬

*, **세종대학교 컴퓨터공학과 +, ++한국전자통신연구소

*ywkim@sju.ac.kr +jsy@etri.re.kr ++shcho@etri.re.kr **yllee@sejong.ac.kr

CNN-based In-loop Filter on TU Block

*Yang-Woo Kim +Seyoon Jeong ++Seunghyun Cho **Yung-Lyul Lee

*, **Sejong University +, ++ ETRI

요약

VVC(Versatile Video Coding)는 입력된 영상을 CTU(Coding Tree Unit) 단위로 분할하여 코딩하며, 이를 다시 QTBTT(Quadtree plus binary tree and triple tree)로 분할하고, TU(Transform Unit)도 이와 같은 단위로 분할된다. 따라서 TU의 크기는 4x4, 4x8, 4x16, 4x32, 8x4, 16x4, 32x4, 8x8, 8x16, 8x32, 16x8, 32x8, 16x16, 16x32, 32x16, 32x32, 64x64의 17가지 종류가 있다. 기존의 VVC 참조 Software인 VTM에서는 디블록킹필터와 SAO(Sample Adaptive Offset)로 이루어진 인루프필터를 이용하여 에러를 복원하는데, 본 논문은 TU 크기에 따라서 원본블록과 복원블록의 차이(에러)가 통계적으로 다름을 이용하여 서로 다른 CNN(Convolution Neural Network)을 구축하고 에러를 복원하는 방법으로 VTM의 인루프 필터를 대체한다. 복원영상의 에러를 감소시키기 위하여 TU 블록크기에 따라 DenseNet의 Dense Block기반 CNN을 구성하고, Hyper Parameter와 복잡도의 감소를 위해 네트워크 간에 일부 가중치를 공유하는 모양의 Network를 구성하였다.

1. 서론

VVC(Versatile Video Coding)은 ITU-T VCEG와 ISO/IEC MPEG이 JVET(Joint Video Exploration Team)을 2015년 10월에 결성하고, 2018년 4월부터 HEVC(High Efficiency Video Coing)이후 새로운 비디오 코딩 표준화를 목표로 작업을 시작했으며, 기술 검증을 위한 참조 Software인 VTM(VVC Test Model)을 공개하였다. VTM은 HEVC와 유사한 디블록킹 필터와 SAO(Sample Adaptive Offset)를 포함한 인루프필터로 구성되어 있다.

VTM은 블록단위 비디오 코딩을 위하여 128x128 크기의 CTU(Coding Tree Unit)를 정의하고, 최소 4x4크기부터 최대 64x64 크기의 CU(Coding Unit)을 정의하였다. 각 영상의 프레임은 CTU 단위로 분할되고, CTU는 다시 64x64크기의 CU로 분할된 후, 각 CU는 재귀적으로 QTBTT(Quadtree plus binary tree and triple tree) 분할에 의해 최적의 CU로 분할된다. VTM에서 CU의 개념은 HEVC에서의 잔차 신호의 블록 단위 주파수 변환 및 양자화를 위한 TU(변환단위, Transform Unit)와 PU(예측단위 , Prediction Unit)를 포함한다. 따라서 VTM에서 TU의 크기는 4x4, 4x8, 4x16, 4x32, 8x4, 16x4, 32x4, 8x8, 8x16, 8x32, 16x8, 32x8, 16x16, 16x32, 32x16, 32x32, 64x64의 17가지 종류가 있다.

그림 1은 832x480 해상도인 BasketballDrill영상의 TU 분할구조에 따른 픽셀단위의 원본 영상과 복원 영상의 MSE(Mean Squared Error)를 보여준다. 오른쪽 그림에서 흰색은 MSE가 큰 값, 검정색은 MSE가 작은 값을 보인다. 그림에서 TU블럭의 크기가 작을수록 MSE가 큰 경향을 관찰 할 수 있는데, 이는 VTM은 DCT(Discrete Cosine Transform)을 사용하기 때문에, TU가 분할될 때 비교적 텍스쳐(Texture)한 영역이 확률적으로 보다 더많이 분할 되기 때문이다.

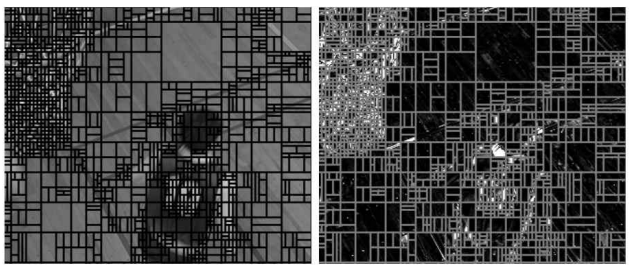

그림 1. VTM 1.1에서 TU분할의 예시와(왼쪽) TU 블록 크기에 따른 에러의 양(오른쪽)

최근 컴퓨터 비전, 영상 압축등 영상처리 분야에서 딥러닝(Deep Learning)을 이용한 많은 연구들이 진행 되었다. 딥러닝은 기존의 신경망 네트워크(Neural Network)에서 은닉층(Hidden Layer)가 2개 이상인 네트워크를 의미하는데, 이는 활성화함수(Activation Function)를 이용한 비선형성으로 인간의 두뇌를 모방하여 여러 영상처리 분야에서 기존의 방법들보다 좋은 성능을 보이고 있다. 이중에서 CNN(Convolution Neural Network)은 각 layer별로 Convolution Kernel을 이용, 입력영상의 지역적인 특징을 추출하여 다음 layer로 보내는 방법으로, 딥러닝을 이용한 영상처리 분야에서 뛰어난 성능을 보이고 있다.

2. 블록 크기에 따른 CNN 인루프 필터

2.1 네트워크 구조

제안하는 네트워크 구조는 그림2와 같다. 원본영상은 VTM1.1에서 기존의 인루프 필터를 거치지 않은 YUV 4:2:0 영상이, 예측영상과

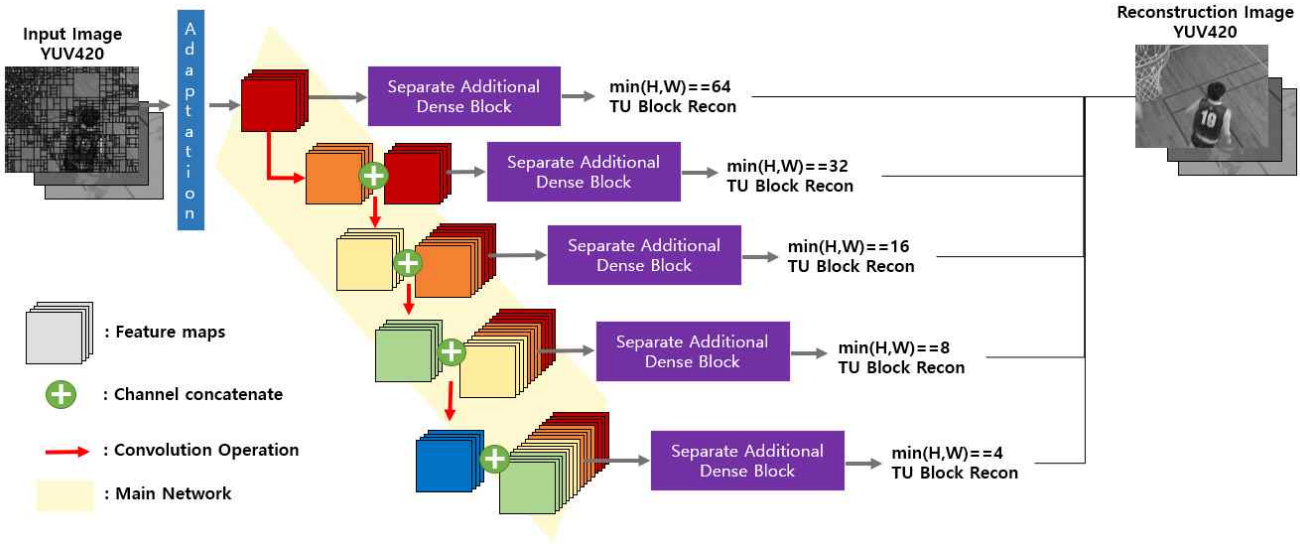

그림 2. 제안하는 전체 네트워크 구조

역양자화를 거친 잔차신호 형태로 구성되어 있다. 이를 TU 블록 분할 정보를 이용하여 TU 블록을 순차적으로 CNN의 입력으로 넣게 된다. 이때, 인루프 필터링을 하는 시점에서 현재 TU의 주변 픽셀 정보를 사용할 수 있으므로, Luma성분의 경우 2픽셀, Croma성분의 경우 1픽셀의 주변정보를 동시에 획득한다. TU Block은 본격적인 네트워크에 입력으로 들어가기 전에, Block Boundary 부분에 대한 정보 불균형을 해결하고, YUV 채널 간의 크기 차이를 처리하기 위해 Adaptation이라는 선행처리를 거친다. Adaptation 과정을 거친 영상은, 기존에 Luma성분 채널의 가로, 세로 크기에서 절반에 해당하는 크기의 특징 맵 18개 채널로 변환된다.

크기	PSNR	크기	PSNR
4x4	33.39	16x16	38.32
4x8	34.38	16x32	39.62
4x16	36.24	32x32	39.72
4x32	38.87	64x64	41.76
8x8	35.38		
8x16	37.00		
8x32	39.04		

표 1. VTM1.1 QP32에서 TU 블록 크기에 따른 PSNR

표 1과 그림 1에 따르면 VTM1.1로 분할된 TU들은 블록 크기가 작을수록 에러가 적으며, 블록 크기가 커질수록 에러가 커지는 경향이 있다. 이를 이용하여, 블록 크기가 작은 TU들은 좀 더 깊은 네트워크로 처리하고, 블록 크기가 큰 TU들은 복원 할 에러가 비교적 별로 없기 때문에 얕은 네트워크로 처리한다. 이때, 블록 크기마다 각각의 네트워크를 구성하면, 네트워크가 지나치게 복잡해지고, 계산복잡도가 커지기 때문에, 네트워크에서 일부 가중치를 공유하고, 큰 TU들은 네트워크를 비교적 빨리 빠져나오는 “Shared Network”를 구성하였다. 이는 각각의 TU마다 별도의 네트워크를 구성하는 “Seperate Network”보다 의미있는 차이가 나지 않았다. “Main Network”는 DenseNet[1]의 Dense Block과 유사한 구조를 가진다. “Shared Network”에서 각 TU들은 크기에 맞게 Main 네트워크를 빠져나온후, 각각 추가적인 학습을 거친 Seperate Additional Dense Block을 지나 최종적으로 잔차신호 형태로 구성된다. 이 재구성된 신호를 원래의 예측신호와 픽셀 단위로 더하여 출력한다.

2.2 학습 방법

제안하는 Shared Network 구조를 학습시키기 위하여 표 2의 비디오 스퀀스에서 VTM1.1을 이용하여 원본영상, 기존의 인루프필터를 거치지 않은 복원영상의 예측신호와 잔차신호를 TU 블록 크기마다 표 3와 같은 수를 획득하였다. TU 블록 크기별 데이터 수의 비율은 실제 VTM1.1에서의 TU 블록 크기별 비율과 유사하다. 그리고 원본영상과 네트워크를 거친 복원영상에 대한 MAE(Mean Absolute Error)를 이용하여 네트워크를 Adam Optimizer로 학습하였다. 네트워크의 가중치를 공유하는 부분에 대해서 TU크기별로 재학습되어 가중치가 변화하기 때문에, 충분한 학습을 거친후에 가중치를 공유하는 Main Network부분을 더 이상 학습하지 않고 각각의 블록마다 별도로 구성된 Seperate Additional Dense Block만을 학습하는 방법을 통해 네트워크를 추가적으로 최적화 하였다.

학습데이터는 VVC의 실험모델 VTM1.1을 사용하여 획득하고, Network를 학습시키기 위해 Tensorflow Python API를 사용하였다.

3 실험결과

본 논문은 기존의 VTM1.1의 인루프 필터를 대체하기 위해서 CNN을 이용한 TU블록 크기에 따른 새로운 인루프필터를 제안하였다. TU블록간에 따른 별도의 네트워크를 가중치 공유를 통하여 계산복잡도를 개선하였다. 제안하는 Shared Network구조는 기존의 VTM1.1 화면내예측 모드에서 기존 인루프필터 대비 약 0.2%의 PSNR 향상을 보였다.

클래스	시퀀스	해상도	비트 심도	프레임 율
B	Kimono	1920x1080	8	24fps
B	ParkScene	1920x1080	8	24fps
B	ParkScene	1920x1080	8	24fps
B	Cactus	1920x1080	8	24fps
B	BasketballDrive	1920x1080	8	50fps
B	BQTerrace	1920x1080	8	60fps
C	BasketballDrill	832x480	8	50fps
C	BQMall	832x480	8	60fps
C	PartyScene	832x480	8	50fps
C	RaceHorses	832x480	8	30fps
D	BasketballPass	416x240	8	50fps
D	BQSquare	416x240	8	60fps
D	BlowingBubbles	416x240	8	50fps
D	RaceHorses	416x240	8	30fps
E	FourPeople	1280x720	8	60fps
E	Johnny	1280x720	8	60fps

표 2. 학습데이터 추출 시퀀스

TU 블록 크기	학습 데이터 개수
4x4	88428
4x8 또는 8x4	124590
4x16 또는 16x4	47386
4x32 또는 32x4	11017
8x8	129913
8x16 또는 16x8	108835
8x32 또는 32x8	32245
16x16	127090
16x32 또는 32x16	61402
32x32	93276
64x64	25681

표 3. TU 블록 크기 별 사용 학습 데이터 개수

4 결론

제안하는 CNN 네트워크를 이용한 TU 블록 크기에 따른 인루프 필터는 기존의 VTM1.1 대비 평균적으로 0.2% PSNR 향상을 보인다.

감사의 글

이 논문은 일부 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-00072, 초실감 테라미디어를 위한 AV 부호화 및 LF 미디어 원천기술 개발)

참고문헌

[1]Huang Gao, Liu Zhuang, Laurens van der Maaten, Q. Weinberger Kilian, "Densely connected convolutional networks", 2017 IEEE Conference on Computer Vision and Pattern Recognition CVPR, pp. 2261-2269, 2017.

[2] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, "High efficiency video coding (HEVC) text specification draft 7", document JCTVC-I1003, Jul. 2012 K. D. Hong and K. J. Lim, "A study on image understanding," IEEE Trans. Image Processing, vol. 3, no. 2, pp. 1-10, 2007.

[3] D. P. Kingma, J. L. Ba, "Adam: a Method for Stochastic Optimization", International Conference on Learning Representations, pp. 1-13, 2015.

[4] Zhang, K., Zuo, W., Chen, Y., et al.: 'Beyond a Gaussian denoiser: residual learning of deep Cnn for image denoising', IEEE Trans. Image Process., 2017, 26, (7), pp. 3142 - 3155.