DVC: An End-to-end Deep Video Compression Framework

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao

* Published in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15-20 June 2019

Presented by,
Tanni Das
Graduate Research Assistant
Video Processing Lab,
Dept. of AI.Software Engineering,
Gachon University, Korea.

Outline

Outline

- Proposed method
    - Training strategy
        - Loss function
        - Quantization
        - Bit rate estimation
        - Buffering previous frames
- Experiments
- Conclusion

Motivation

- Conventional video compression approaches use the predictive coding architecture
- Previous video compression systems are not end-to-end optimized
- Two problems to directly apply previous deep neural network based image compression
  - ✓ How to generate and compress motion information tailored for video compression (because still image has no motion)
  - ✓ How to build a DNN based video compression system by minimizing the rate distortion based optimization

Key Points

- First end-to-end deep video compression (DVC) model (according to Authors' claim)
- Motion estimation, motion compensation, residual compression, motion compression, quantization, bit rate estimation are implemented with end to end neural networks (among them I did not find the quantization network. Need further study)
- The key components are jointly optimized based on rate distortion trade-off through a single loss function (modified rate distortion optimization)
- One to one mapping between the conventional video compression techniques and the proposed DVC model (for each component in previous techniques, a corresponding neural network is proposed)

Related works (Image compression)

- Handcrafted techniques based image compression has a disadvantage that the modules are separately optimized and may not achieve optimal compression performance
- Maximum DNN based image compression did not consider the number of bits used for compression
- Generalized divisive normalization (GDN), multi scale image decomposition, adversarial training, importance map, intra prediction are proposed to improve the image compression performance

Related works (Video compression)

- Traditional video codecs use predictive coding architecture
- Previous DNN based approaches focused on one particular module to improve, not end-to-end method

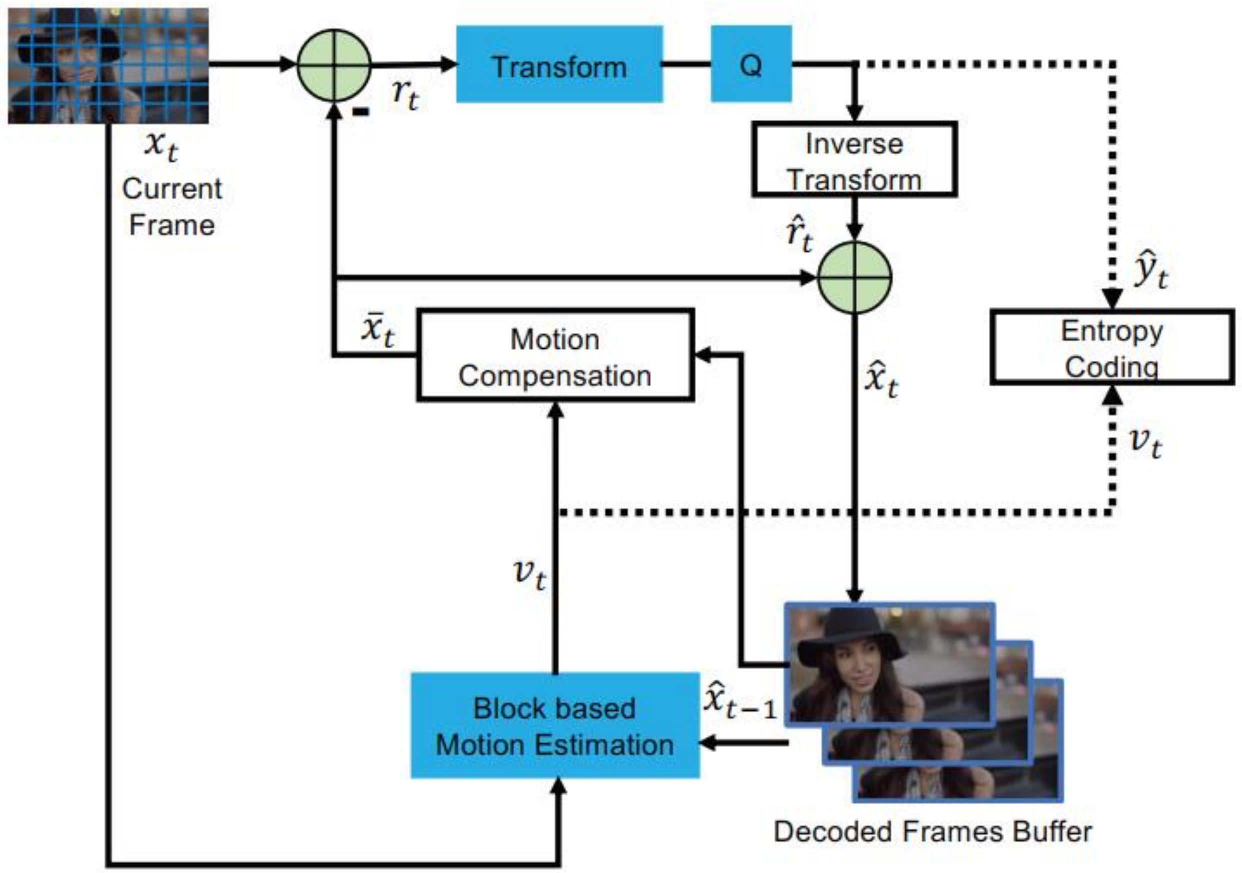
Related works (Motion estimation)

- Optical flow is widely used to exploit temporal relationship
- Traditional video codecs use block based motion estimation algorithm
- Accurate motion information at pixel level can be achieved through learning based optical flow method
- Much more bits are required to compress motion information if optical flow values are encoded by traditional video compression approaches

Proposed method (Introduction of notations)

Notation	Description
$v = \{x_1, x_2, x_3, \ldots\ldots, x_{t-1}, x_t, \ldots\}$	Video sequences
x_t	Frame at time stamp *t*
$\overline{x}_t$	Predicted frame
$\hat{x}_t$	Reconstructed/decoded frame
r_t	Residual between x_t and $\overline{x}_t$
$\hat{r}_t$	Reconstructed residual
v_t	Motion vector/optical flow value
$\hat{v}_t$	Reconstructed motion vector/optical flow
$r_t \rightarrow y_t, v_t \rightarrow m_t$	y_t is the output of residual encoder network
	Reason: Linear or non linear transform can improve compression efficiency

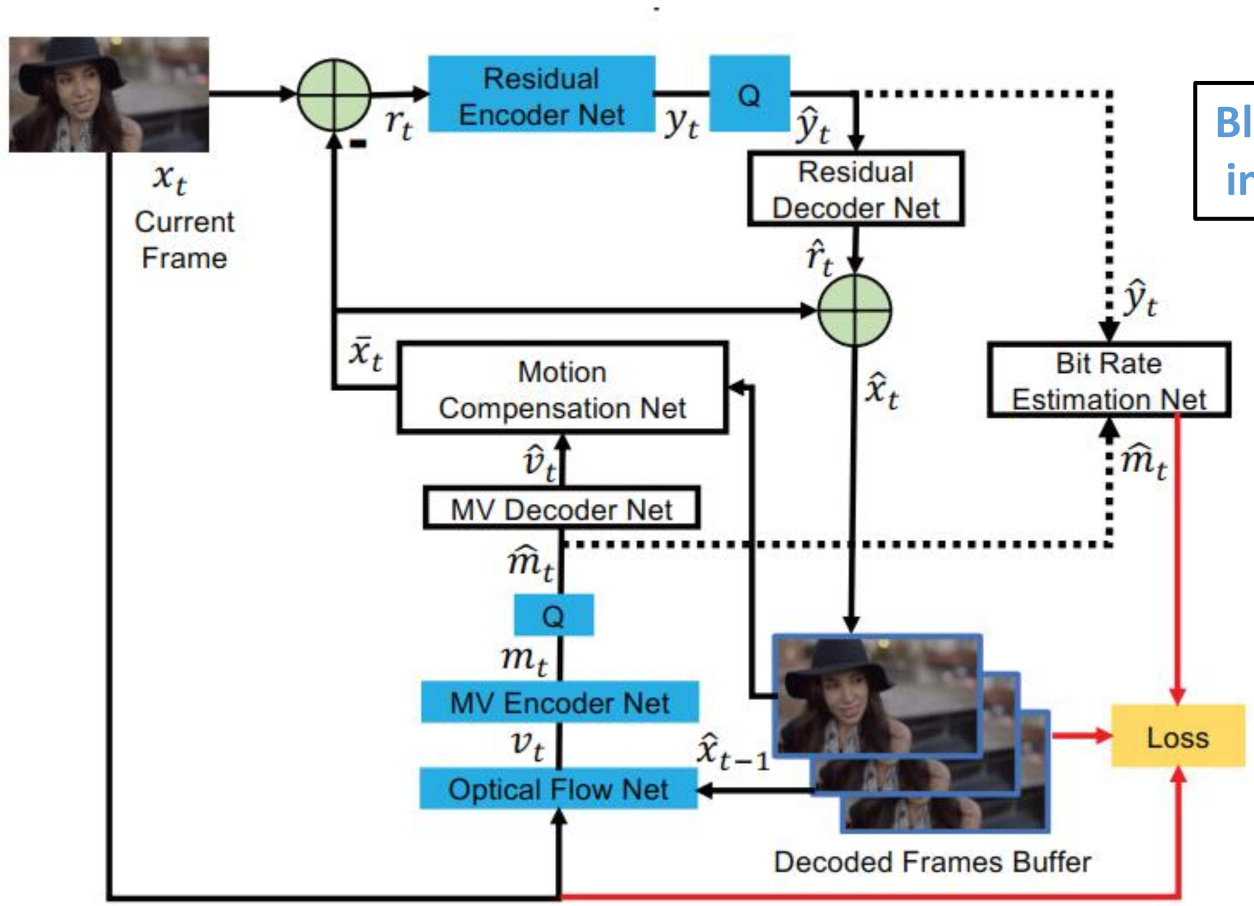
Proposed method (Brief introduction of video compression)



Blue modules are not included in the decoder side

Figure 1. The predictive coding architecture used by traditional video codec H.264 or H.265.

Proposed method (Overview of the proposed method)



Blue modules are not included in the decoder side

Figure 2. The proposed end-to-end video compression network.

Proposed method (Previous vs proposed technique)

Traditional approach	**Proposed approach (DVC)**
Step 1. Motion estimation Estimation of motion between x_t and $\hat{x}_{t-1}$. v_t is obtained.	**Step N1. Motion estimation and compression** $\hat{x}_{t-1}$ → optical flow net → v_t → MV encoder net → m_t → Quantization → $\hat{m}_t$ → MV decoder net → $\hat{v}_t$ * Without MV encoder-decoder network raw optical flow values require high bit stream
Step 2. Motion compensation $\overline{x}_t$ prediction based on v_t from step 1. Residual obtained as $r_t = x_t - \overline{x}_t$.	**Step N2. Motion compensation** A motion compensation network is designed to obtain the $\overline{x}_t$ based on the optical flow obtained in Step N1.
Step 3. Transform and quantization r_t → linear transform (DCT) → quantization → $\hat{y}_t$	**Step N3-N4. Transform, quantization, and inverse transform** r_t → Residual encoder network (against DCT) → y_t → quantization → $\hat{y}_t$ → Residual decoder network (against inverse transform) → $\hat{r}_t$
Step 4. Inverse transform $\hat{y}_t$ → inverse transform → $\hat{r}_t$	

Proposed method (Previous vs proposed technique) Contd.

Traditional approach	Proposed approach (DVC)
Step 5. Entropy coding v_t (from step 1) & $\hat{y}_t$ (from step 3) → bits (by entropy coding) → decoder	**Step N5. Entropy coding** At training stage, bit rate estimation net (a CNN) is used At testing stage, $\widehat{m}_t$ from step N1 & $\hat{y}_t$ from step N3 are **coded into bits (this is confusing. How to code? Traditional or the CNN used in training)**
Step 6. Frame reconstruction $\hat{x}_t = \hat{r}_t + \overline{x}_t$	**Step N6. Frame reconstruction** Same as traditional video codec

Proposed method (MV Encoder and Decoder Network)

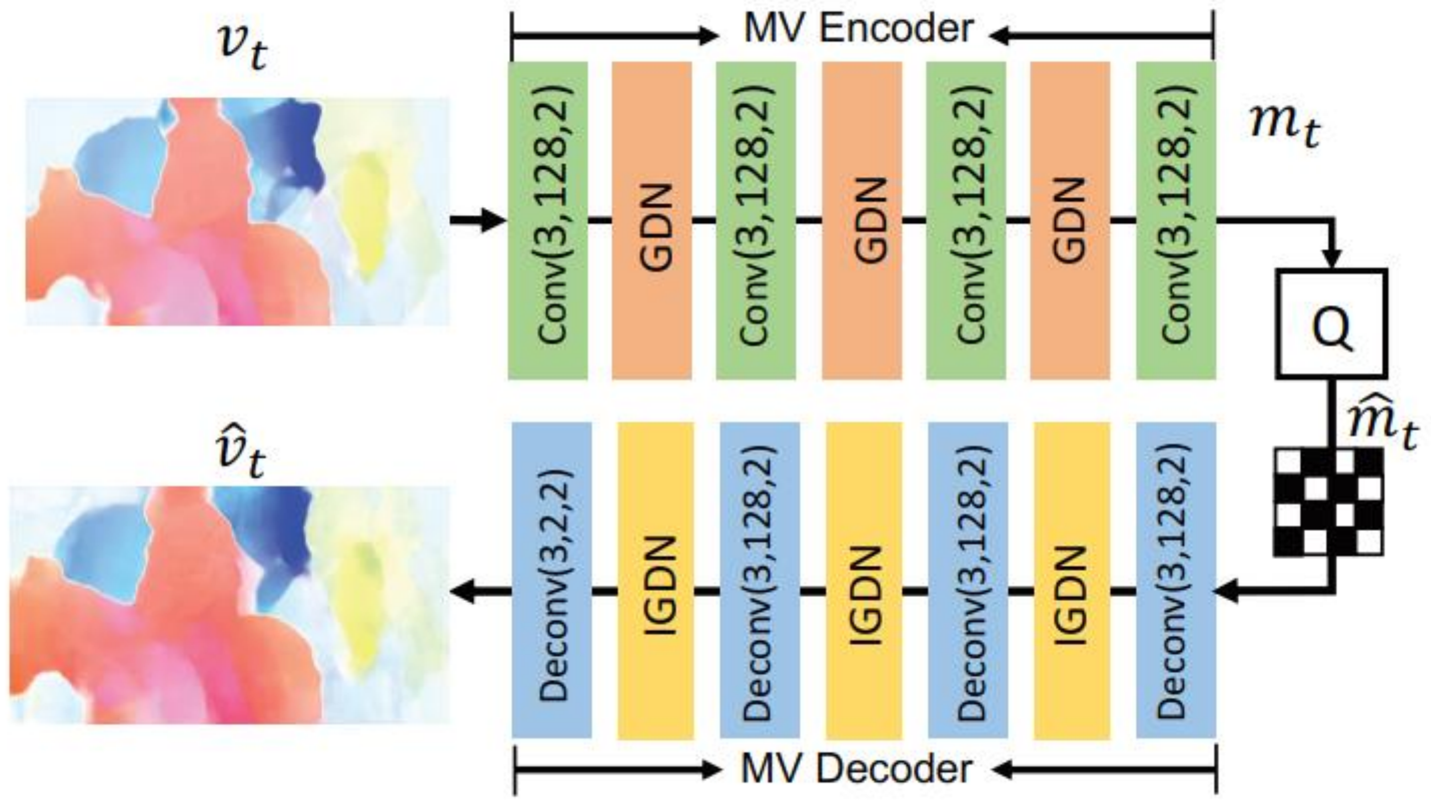


Figure 3. Our MV Encoder-decoder network. Conv (3,128,2) represents the convolution operation with the kernel size of 3x3, the output channel of 128 and the stride of 2. GDN/IGDN is the nonlinear transform function. The binary feature map is only used for illustration.

Proposed method (Motion compensation network)

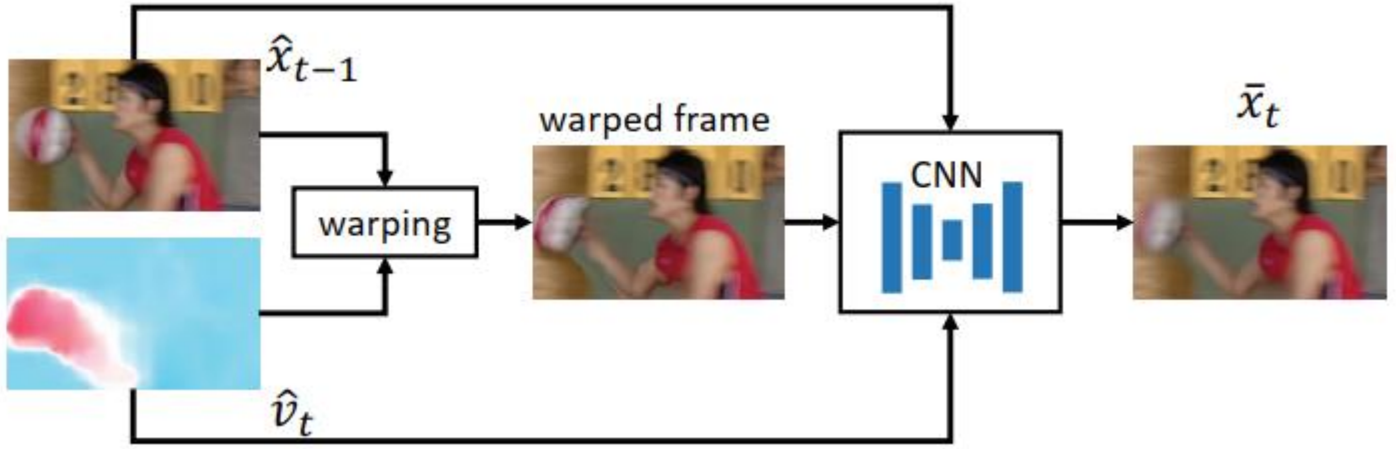


Figure 4. Motion compensation network.

- Detail description of CNN model was not found
- No need of loop filter / sample adaptive offset technique

Proposed method (Residual encoder and decoder network)

- Exploited the CNN from "*J. Balle, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436, 2018*"
- Achieve higher compression efficiency than DCT

Proposed method (Loss function)

- $\lambda D + R = \lambda d(x_t, \hat{x}_t) + \left(H(\widehat{m}_t) + H(\hat{y}_t)\right)$

where,

$d(x_t, \hat{x}_t)$ = distortion between x_t and $\hat{x}_t$ (here MSE is used)

$H(.)$ = number of bits used for encoding the representation

λ = Lagrange multiplier

D, R = not mentioned

Proposed method (Quantization)

- They used the method in *"J. Balle, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. arXiv preprint arXiv:1611.01704, 2016"*
- In training stage, $\hat{y}_t = y_t + \eta$; η = uniform noise
- In inference stage, $\hat{y}_t$ = *round(*y_t*)*

Proposed method (Bit rate estimation)

- Used the CNN from *“J. Balle, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436, 2018”*

Proposed method (Buffering previous frames)

- Online updating strategy is adopted
- Saving the previous frames in GPU is impossible when *t* is large
- Optimizing and storing one frame in each iteration is possible in this way

Experiments (Experimental Setup)

- **Vimeo-90k** dataset is used for training.
- **UVG** dataset and HEVC standard test sequence (**Class B**, **Class C**, **Class D**, **Class E**) are used for evaluating the proposed algorithm.
- **Class C** and **Class D** dataset results are provided as supplementary material.
- Reconstructed frame distortion is measured using **PSNR** (Peak-signal-to-noise-ratio) and **MS-SSIM**(Multi scale structural similarity index).
- Four models are trained using four different $\boldsymbol{\lambda}$ $values$ ($\lambda = 256, 512, 1024, 2048$).
- **Adam optimizer** is used.
- The **mini-batch** size is set as **4** .
- The resolution of training images is **256 $\times$ 256.**

Experiments (Experimental Results)

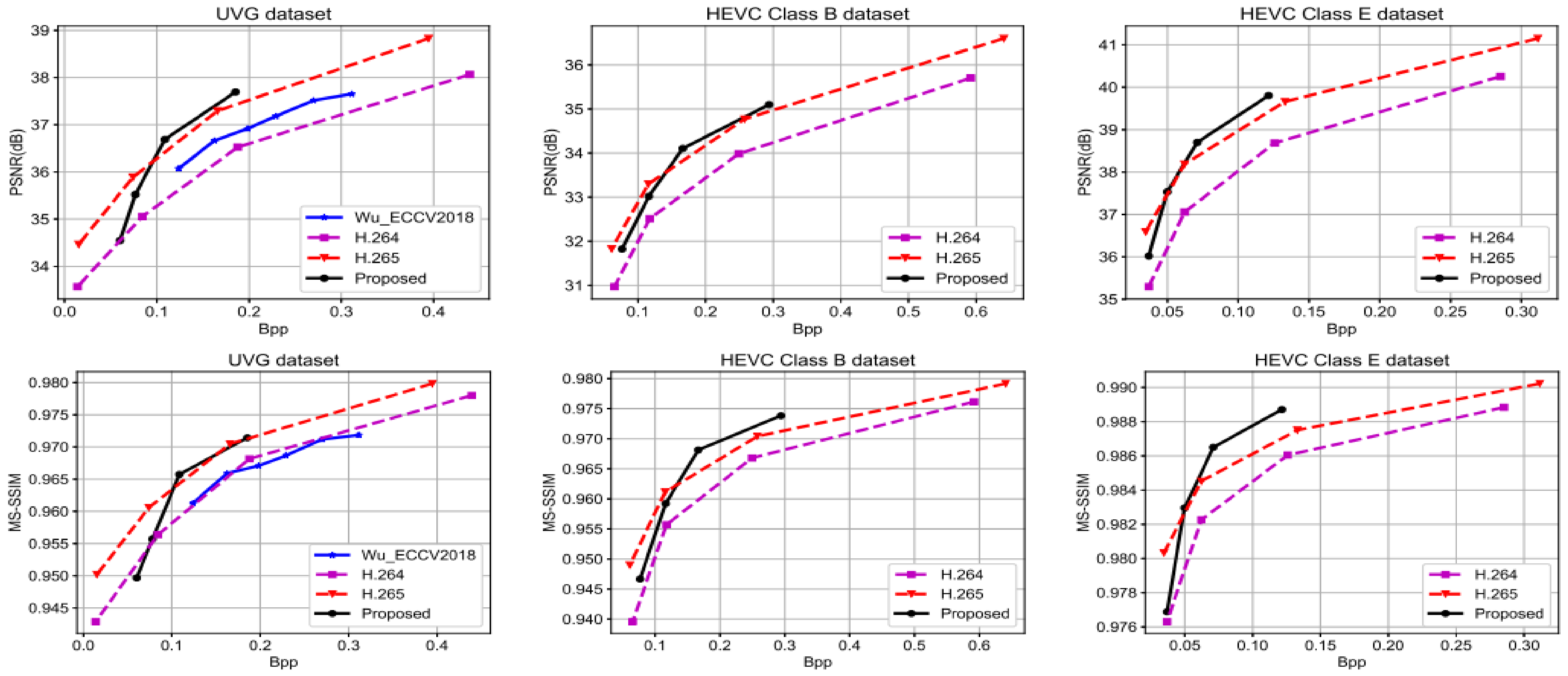


Figure 5: Comparison between our proposed method with a learning-based video codec technique, H.264 and H.265 .Our method outperforms H.264 when measured by both PSNR and MS-SSIM. Meanwhile, our method achieves similar or better compression performance when compared with H.265 in terms of MS-SSIM.

Experiments (Experimental Results)

- The compressed frames for H.264 and H.265 are generated using FFmpeg.
- The GOP sizes for the UVG dataset and HEVC dataset are 12 and 10, respectively.
- On UVG dataset the proposed method achieve 0.6 dB gain at same Bpp (Bit per pixel) level over Wu_ECCV2018 (learning based video compression system).

Experiments (Ablation Study and Model Analysis)

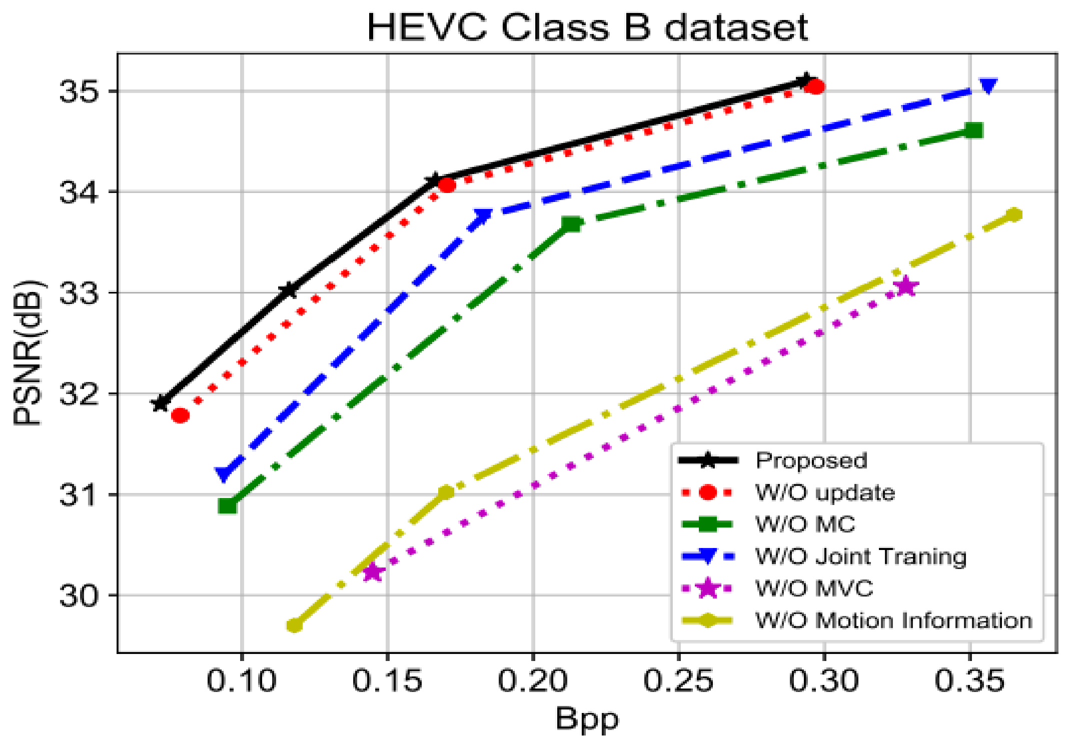


Figure 6: Ablation study. We report the compression performance in the following settings. 1. The strategy of buffering previous frame is not adopted(W/O update). 2. Motion compensation network is removed (W/O MC). 3. Motion estimation module is not jointly optimized (W/O Joint Training). 4. Motion compression network is removed (W/O MVC). 5. Without relying on motion information (W/O Motion Information).

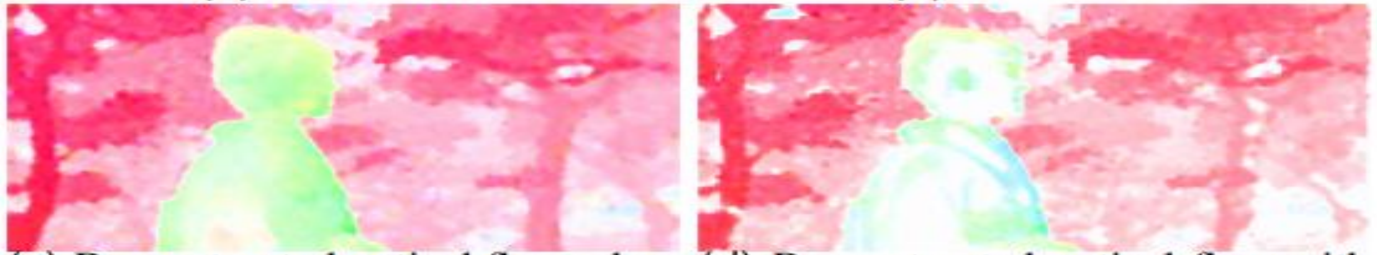
- The buffering strategy provide about 0.2dB gain at the same bpp level.
- The PSNR without the motion compensation network will drop by about 1.0 dB at the same bpp level .
- Without using any motion estimation approach leads to more than 2dB drop in PSNR.

Fix ME		W\O MVC		Ours	
Bpp	PSNR	Bpp	PSNR	Bpp	PSNR
0.044	27.33	0.20	24.32	0.029	28.17

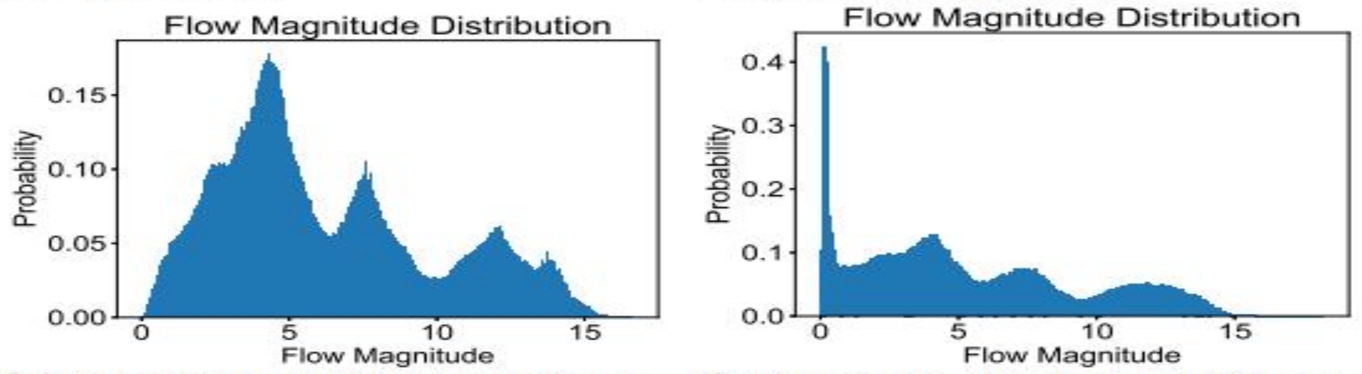
Table 1: The bit cost for encoding optical flow and the corresponding PSNR of warped frame from optical flow for different setting are provided .

Experiments (Ablation Study and Model Analysis)

(a) Frame No.5 (b) Frame No.6



(c) Reconstructed optical flow when fixing ME Net. (d) Reconstructed optical flow with the joint training strategy.

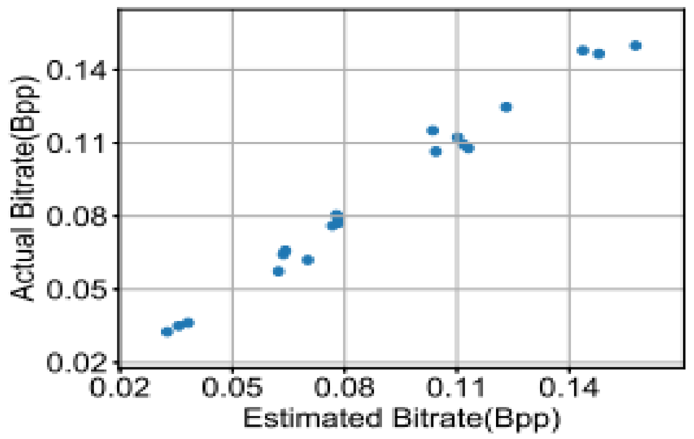


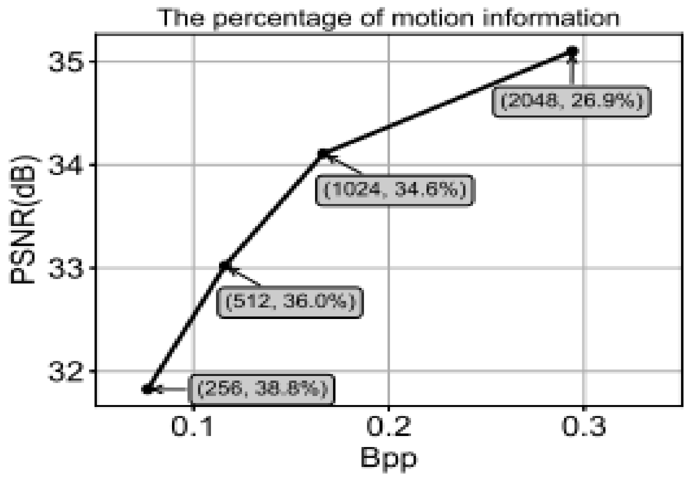
(e) Magnitude distribution of the optical flow map (c). (f) Magnitude distribution of the optical flow map (d).

Figure 7: Flow visualize and statistic analysis.

- Visual comparisons between frame 5 and frame 6 from the kimono sequence.
- It needs 0.045bpp for encoding the optical flow map in Fig. 7 (c) while it only needs 0.038bpp for encoding optical flow map in Fig. 7 (d).
- The reconstructed optical flow by using joint training strategy method contains more pixels with zero flow magnitude (e.g., in the area of human body) .

Experiments (Ablation Study and Model Analysis)





(a) Actual and estimated bit rate. (b) Motion information percentages.

Figure 8: Bit rate analysis.

- Compare the estimated bit rate and the actual bit rate by using arithmetic coding .
- Estimated bit rate is closed to the actual bit rate.
- *When* λ in function $\lambda * D + R$ becomes larger, the whole Bpp also becomes larger while the corresponding percentage of motion information drops.

Conclusion

- Proposed fully end-to-end deep learning framework for video compression.
- Inherits the advantages of both classic predictive coding scheme in the traditional video compression standards and the powerful non-linear representation ability from DNNs.
- Outperforms the widely used H.264 video compression standard and the recent learning-based video compression system.
- Provides a promising framework for applying deep neural network for video compression.

Thank You