Learning-based Video Coding for CLIC 2021

David Alexandre, Wen-Hsiao Peng, Hsueh-Ming Hang
Dept. of Electronics Engineering
National Yang Ming Chiao Tung University, Taiwan
davidalexandre.eed05g@nctu.edu.tw, wpeng@cs.nctu.edu.tw, hmhang@nctu.edu.tw

Abstract

We proposed a learning-based video compression system and submitted it to the video track of Challenge on Learned Image Compression (CLIC) 2021. Our system consists of a neural-network based motion compensation unit and a neural-network based residual signal compressor. The motion estimation part includes PWC-Net with local attention residual blocks. The residual coding part includes an autoencoder with hyperprior and a carefully designed Refine-Net. Our design suggests a way of using the learning-based video compression technique to meet the challenge. Our team name is commlab005.

1. Introduction

In 2021, Challenge on Learned Image Compression launched a video track challenge. Given short clips of 60 frames at 720p resolution, each team needs to submit a decoder and compressed file with the maximum size not exceeding 1,309,062,500 bytes when calculated using the formula: model size + data size/0.019. To participate in this challenge, we designed a learning-based video compression framework, which has neural network-based components for motion estimation (ME-Net), motion coding, residual coding, and refinement. In this fact sheet, we briefly describe our design in section 2 and present the experimental results in section 3.

2. Proposed Method

2.1. Framework Design

As shown in Fig.1, our system consists of 4 major components: ME-Net (for motion estimation), Motion Coding (for encoding the motion information), Residual Coding (for encoding the motion-compensated differences), and Refine-Net (for improving the decoded image quality). The conventional Group of Pictures (GOP) frame structure is adopted but we have only one I-frame (intra) and a number of P-frames (inter) in one GOP. We use the VVC intra coding scheme [4] to encode the intra frame. Our CNN-based modules are applied only to the P-frames. First, the ME-net (based on PWC-Net with enhancement) uses the reference frame x_{t-1} and the target frame x_t to produce the motion vector field v_t. Then, we calculate the residual frame r_t which is the difference between the motion compensated frame $\bar{x}_t$ and the target frame. We encode the motion field and the residual frame using an autoencoder-based compressor with hyperprior proposed by Balle, et al. [1] to produce compressed files. Receiving a compressed file, the decompressor recovers $\hat{v}_t$ and $\hat{r}_t$ at the decoder side. In the cases where the motion vectors fail to produce good quality motion compensated frame, we skip the motion coding part and perform only the residual coding on the frame difference without motion compensation; in other words, the decoded reference frame is used as the replacement for the motion compensated frame in reconstruction. To achieve multi-rate compression, we implemented the quantization scaling factor in the compressors (for both motion vector v_t and residual frame r_t), which was proposed by Lu, et al. [2].

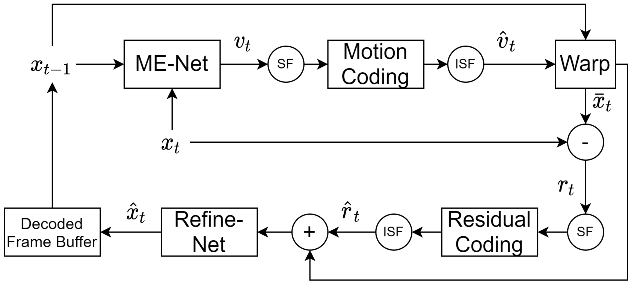

Figure 1: Our learning-based framework for video compression. We implemented learning-based components for P-frame coding, and it consists of ME-Net, Motion Coding, Residual Coding, and Refine-Net.

2.2. Motion Estimation-Net

We adopted PWC-Net for motion estimation but we added a Refine-Net to improve motion field quality and to reduce the bit rate in coding. The Refine-Net includes the local attention residual block (LARB) proposed by Vaswani, et al. [3].

Given the inputs of reference x_{t-1} and target frame x_t, PWC-Net produces a motion vector field $\bar{v}_t$ and put it into the motion enhancement network (MV Refine-Net) to produce the refined motion vector field v_t. In the training phase for ME-Net, we calculate the MSE loss between the motion compensated frame and the target frame. To speed up PWC-Net in computing MVs, the input frames in PWC-Net are downsampled to 1/4 of its original scale. To get back the full-size vector field, we perform an upsampling process. For the MV Refine-Net, we use two LARB blocks. Fig. 2. shows our ME-Net design which includes PWC-Net and MV Refine-Net.

2.3. Residual and Motion Compressor

As said earlier, we adopted the compressor design from Balle, et al [1]. The inputs to the residual compressor and motion compressor are normalized to (-1, 1). The image format is YUV. The loss function used for the compressor training is MSE between the target and the reconstructed of residual and motion vector , $L_{MSEr} = MSE(r_t, \hat{r_t})$ and $L_{MSEv} = MSE(v_t, \hat{v_t})$.

2.4. Refine-Net

We incorporate a residual Refine-Net together with the residual decompressor to reconstruct the target frame. It is made of residual blocks and it uses the multi-scale information derived from the motion compensated frame and the reconstructed residual frame. During the scaling process, we use convolution with stride 2 to do downsampling and then do bilinear upsampling later. The compressor is first trained without the Refine-Net. Then, both Refine-Net and compressor are trained together to produce the final model. The loss function used in the final training step is the MS-SSIM between the target frame and the reconstructed frame, $L_{MSSSIM} = 1 - MSSSIM(x_t, \hat{x}_t)$. The network structure is shown in figure 3.

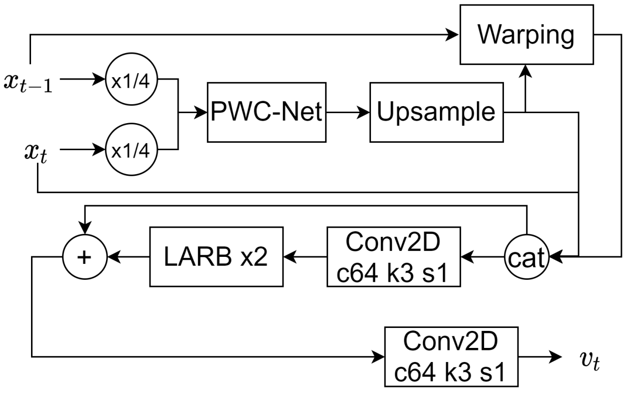

Figure 2: Our ME-Net designed to produce better motion vector representation for the motion-compensated frame by incorporating MV Refine-Net after PWC-Net.

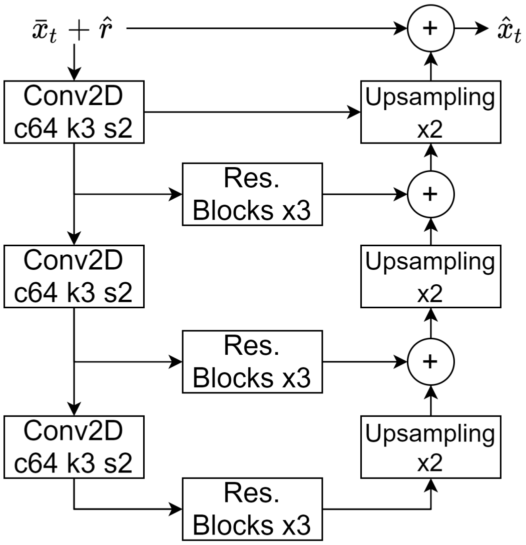

Figure 3: The Refine-Net has a multi-scale structure using the residual blocks. c64, k3, s2 means 64 channel and 3x3 convolution with stride 2.

3. Experiments

We trained the components using the UVG dataset and validate it using the CLIC 2021 validation dataset. In this Challenge, we empirically adjusted the bit rate for each frame clip to meet the total rate requirement. The GOP is set to 30 with a scaling factor between 0.35 to 0.45. Our decoder submission achieves an MS-SSIM value of 0.9502 on the CLIC validation set. Our team name is commlab005.

4. Acknowledgment

This work is partially supported by the Ministry of Science and Technology, Taiwan under Grant MOST 109-2634-F-009-020 through Pervasive AI Research (PAIR) Labs, National Yang Ming Chiao Tung University, Taiwan. Also, we would like to thank National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[2] Yadong Lu, Yinhao Zhu, Yang Yang, Amir Said, and Taco S Cohen. Progressive neural image compression with nested quantization and latent ordering. *arXiv preprint arXiv:2102.02913*, 2021.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[4] Fraunhofer Versatile Video Decoder (VVdeC). https://github.com/fraunhoferhhi/vvdec.