# Video Enhancement Network Based on CNN and Transformer

YUAN Lang[1], HUI Chen[1], WU Yanfeng[1],
LIAO Ronghua[1], JIANG Feng[2], GAO Ying[3]
(1. Harbin Institute of Technology, Harbin 150001, China;
2. Sichuan University of Science & Engineering, Zigong 643002, China;
3. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202404011
https://kns.cnki.net/kcms/detail/34.1294.TN.20241120.1435.004.html, published online November 20, 2024
Manuscript received: 2023-07-29

**Abstract:** To enhance the video quality after encoding and decoding in video compression, a video quality enhancement framework is proposed based on local and non-local priors in this paper. Low-level features are first extracted through a single convolution layer and then processed by several conv-tran blocks (CTB) to extract high-level features, which are ultimately transformed into a residual image. The final reconstructed video frame is obtained by performing an element-wise addition of the residual image and the original lossy video frame. Experiments show that the proposed Conv-Tran Network (CTN) model effectively recovers the quality loss caused by Versatile Video Coding (VVC) and further improves VVC's performance.

**Keywords:** attention fusion mechanism; H.266/VVC; transformer; video coding; video quality enhancement

**Citation** (Format 1): YUAN L, HUI C, WU Y F, et al. Video enhancement network based on CNN and transformer [J]. ZTE Communications, 2024, 22(4): 78 - 88. DOI: 10.12142/ZTECOM.202404011
**Citation** (Format 2): L. Yuan, C. Hui, Y. F. Wu, et al., "Video enhancement network based on CNN and transformer," *ZTE Communications*, vol. 22, no. 4, pp. 78 - 88, Dec. 2024. doi: 10.12142/ZTECOM.202404011.

## 1 Introduction

According to the 2023 China Internet Annual Report released by QuestMobile[1], online video platforms continue to catch users' attention through a series of show content. In September 2023, the number of active users of Tencent Video had reached 416 million. Meanwhile, the emergence of 4K and 8K ultra-high definition (HD) videos and the rapid development of concepts such as virtual reality and meta-universe, have led to a fast rise in the total amount of video data flowing across the Internet. These developments have placed higher requirements for video technologies, such as high definition, small data transmission resources, and little storage space.

The current video compression technology[2-3] typically exploits data redundancy in the spatial and temporal domains for compression. Based on the final effect, the compression methods can be classified into two categories: lossless compression and lossy compression. The former optimizes the way data is stored without affecting the content, but its compression ratio is not high enough. In contrast, lossy compression takes the advantage of the human eye's insensitivity to color and sensitivity to brightness, and intentionally removes trivial information. Though the final decoded video is not identical to the original, it maintains the visual effect and can achieve an ideal compression ratio. However, when the compression ratio is limited, lossy compression introduces serious artifacts.

In recent years, deep learning technology has achieved great progress in various task fields. We can effectively obtain local information using convolutional neural networks (CNN). However, limited by the receptive field of the small kernel, CNN can only scan local areas step by step, which means global features are thus ignored. To alleviate the problem, we can utilize Transformer that uses a self-attention mechanism to construct long-range correlations and directly capture global textures. In visual tasks, to reduce computational complexity, Transformer splits the whole image into non-overlapping patches and then calculates the similarity between them. However, this method breaks the continuity of the adjacent block's edge, leading to inferior local modeling results. To combine the strengths of both models, we design a feature extractor based on both CNN and Transformer to obtain restoring features from different perspectives, and thus enhance the quality of video frames. To further improve the effectiveness of the network, we introduce the channel attention fusion mechanism, which selects the features extracted by CNN

**This work was supported by the Key R&D Program of China under Grant No. 2022YFC3301800, Sichuan Local Technological Development Program under Grant No. 24YRGZN0010, and ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-03-2019-12.**

and Transformer with an adaptive fusion ratio.

To effectively improve the quality of the decoded lossy video, we propose a video enhancement network based on convolution and Transformer, dubbed Conv-Tran Network (CTN). We use two feature extraction modules containing CNN and Transformer respectively as the backbone and introduce a multi-branch structure to ensure that the two modules do not interfere with each other while extracting recovery features. First, we employ a convolution layer to fuse the lossy video frame and its corresponding quantization-parameter (QP) value information, which raises the dimension and obtains low-level features. Second, these features are processed by several conv-tran blocks (CTB) and then transformed into a residual image after another convolution layer. The final reconstructed video frame is obtained by adding the residual image and the original lossy frame. Overall, the main contributions of CTN are as follows:

1) We propose a novel video enhancement network based on CNN and Transformer, which utilizes the local representations and global similarity to improve the reconstruction quality of videos.

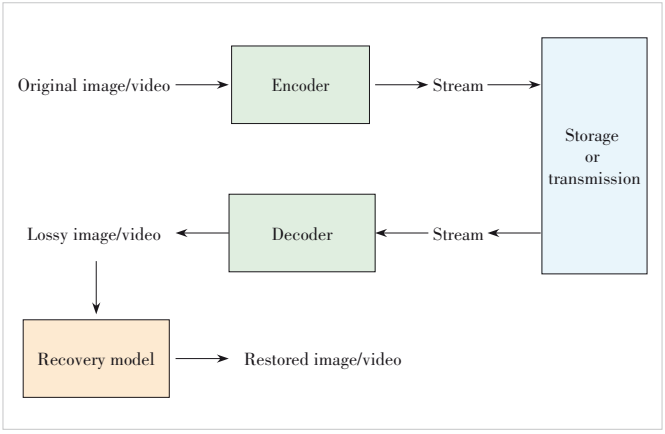
2) A channel attention fusion mechanism is presented, which can effectively exploit the local and non-local priors to enhance the correlation between inter-components.

3) Compared to the existing methods, CTN can extensively recover quality loss after Versatile Video Coding (VVC) decoding and further enhance the coding performance of VVC.

## 2 Related Work

The vision restoring tasks can be divided into image compression restoring and video compression restoring. The specific process is shown in Fig. 1.

First, the encoder compresses the original lossless images or videos to generate a data stream for reducing the bandwidth utilization and storage space. Then, the decoder decodes the data stream to restore the images or videos when needed. However, the encoding and decoding processes usually introduce compression noise into the data, requiring additional recovery algorithms to restore the lossy data. The final goal is to ensure that the images or videos recovered by the model are as similar as possible to the original ones.




▲Figure 1. Process of restoring quality

### 2.1 Recovery Task

Image coding has been developed earlier than video coding and is easier to study, so many works are based on the Joint Photographic Experts Group (JPEG) or JPEG2000 to test the quality enhancement results of the recovery algorithms. The earliest work using neural networks to enhance the performance of codec is Artifacts Reduction CNN (ARCNN)[4] proposed by DONG et al., which uses four layers of convolutional neural networks to recover JPEG compression artifacts through four steps: feature extraction, feature enhancement, map learning, and reconstruction, and finally improves the peak signal-to-noise ratio (PSNR) by more than 1 dB on five classic test pictures. WANG et al. used the prior knowledge of image splitting and discrete cosine transform in JPEG and proposed a dual-domain recovery method[5] to enhance images from the frequent and spatial domain, which obtains better enhancement results with less time complexity than ARCNN. Moreover, some works concentrate on network structures, such as an iterative approach proposed by ORORBIA et al., to enhance JPEG images using recurrent neural networks[6].

The first work for video coding recovery is Variable-filter-size Residue-Learning CNN (VRCNN)[7] proposed by DAI et al., which draws from ARCNN. They designed a four-layer post-processing convolutional neural network to replace the deblocking filter and sample adaptive offset for enhancing video quality, saving an average bit rate of about 4.6% on High Efficiency Video Coding (HEVC) in the luminance component. Different video frames that are compressed differently produce different compression noise. YANG et al.[8] designed and trained the Decoder-Side Scalable Convolutional Neural Network (DS-CNN). According to the types of encoded frames, they trained DS-CNN-I for I frames and DS-CNN-B for B frames. Experiments have shown that DS-CNN-I can improve the quality of 0.35 dB in HEVC codec for I frames, while DS-CNN-B achieves the same improvement of 0.35 dB for B frames.

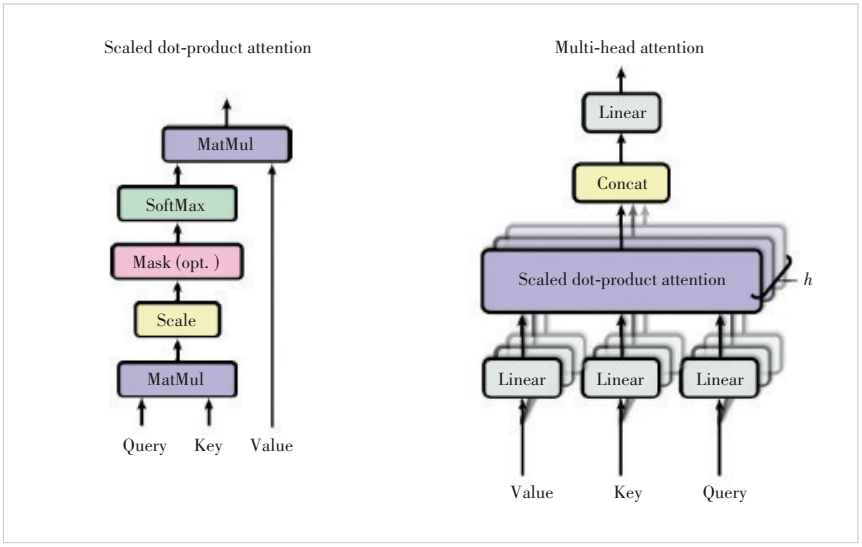
During the 26th meeting of the JVET Video Joint Expert Group held in April 2022, QI et al.[9] proposed that video quality enhancement should not just rely on the information extracted from the spatial domain but also utilize some essential information from the frequent domain to improve the decoded frame. They introduced a multi-scale frequent-domain recovery network, which starts from the network structure and saves the bit rate of 3.07% on the Y component.

### 2.2 Transformer

Classical neural network structures, such as multilayer perceptron (MLP), CNN, recurrent neural networks (RNN), and long short-term memory (LSTM), have been proven effective in

some specific tasks and domains. Transformer is first proposed in natural language processing[10], which utilizes a self-attention structure (shown in Fig. 2) to mine the global correlation between word vectors. This non-recursive global attention calculation method reduces training time and decreases the performance degradation caused by long-term dependencies.

The first vision work that introduces the Transformer structure is the Vision Transformer (ViT),[11] proposed by DOSOVITSKIY et al. To extract the global correlation of the entire image, the network divides the whole image into several blocks (i.e., patches) by analogy with the words in the field of natural language processing. Each block is projected and flattened to form a one-dimensional token, and these token sequences are exactly input into the network for encoding. Another task-oriented token is concatenated to extract the features for the recognition task, and positional encoding is used to preserve the absolute position information of the image blocks. Experiments show that the structure achieves better image recognition results than classical CNN structure models, proving the feasibility and superiority of the pure Transformer structure in the field of vision.




▲Figure 2. Self-attention mechanism

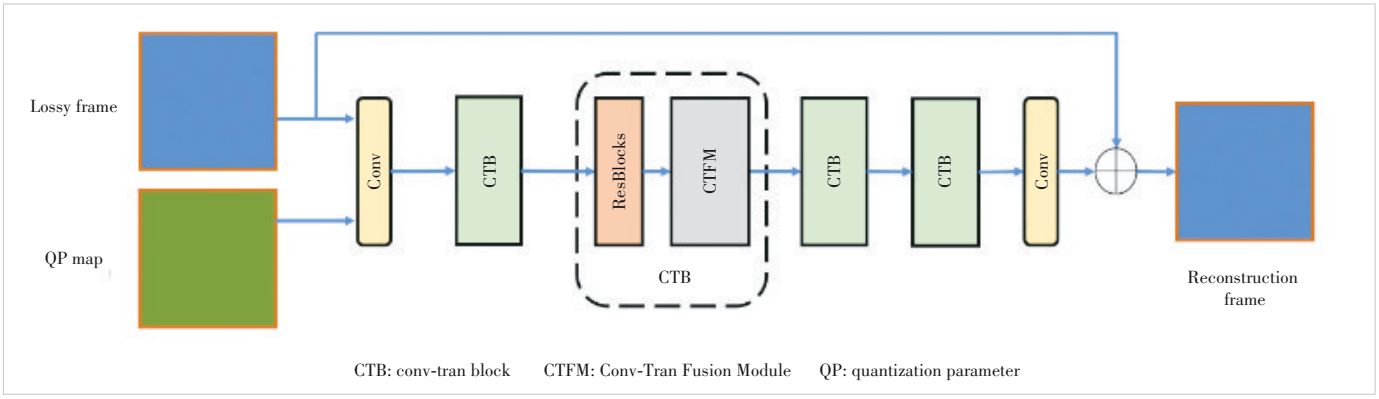
Swin Transformer[12] proposed by LIU et al. is a recent work inspired by the Transformer architecture in 2021. Its goal is to reduce the computational complexity of the original Transformer while maintaining or even improving its performance. To achieve this, instead of computing the correlation between all pairs of patches, Swin Transformer only calculates the correlation between patches within a certain window. This reduces the computational complexity from $O(n^2)$ to $O(wn)$, where $n$ is the number of patches and $w$ is the size of the window. To achieve global modeling capabilities, Swin Transformer shifts the windows in the former layer, allowing information to flow between different parts of the image. Swin Transformer can use this window shifting mechanism to achieve high performance with much less computation than the original Transformer.

CHEN et al.[13] proposed a Transformer-based pre-trained image processing model called Image Processing Transformer (IPT) for image restoration tasks. The model employs the classical Transformer structure as the backbone and is composed of a transformer encoder and decoder. To handle various recovery tasks, IPT uses multiple CNN-based task headers and task tails. The IPT model surpasses previous single-task image-processing models in multiple image-processing tasks. However, IPT also reveals some of the Transformer structure's characteristics: to make a pure Transformer model to accomplish visual tasks, a surprisingly large training set is required, and the model should be trained for a large number of epochs to learn the characteristics of visual image data.

LIANG et al.[14] proposed a new super-resolution network, SwinIR, which combines the self-attention and CNN structures to design the Residual Swin Transformer Block (RSTB). Experiments demonstrate that SwinIR achieves better super-resolution image results than the IPT model with fewer parameters. To explore the potential of the network structure, ZHANG et al.[15] also combined CNN and Transformer to build a recovery network. Unlike the serial structure, this work uses Swin Transformer and residual convolution modules in parallel to extract features and concatenate them and then uses a 1×1 convolutional layer to fuse the features adaptively. To further improve the performance, the work employs the classic U-Net framework, which fully utilizes the priority of different features extracted from the images by multi-scale methods. Finally, the structure achieves a competitive denoising result to SwinIR with lower computational complexity, which shows that the fusion network structure still has much potential to be further exploited. Inspired by this work, we propose a Conv-Tran Fusion Module (CTFM), combining the local and global modeling structures of CNN and Transformer to provide stronger feature extraction capabilities.

## 3 Proposed Method

The general structure of CTN is shown in Fig. 3. The input consists of two parts: the video frame with noise and the corresponding QP value. We focus on restoring the video frames that are generated by VVC in the random access coding mode. The compressed video sequences in this mode have different coding time series and orders, which means the video frames are assigned different QP values according to their time series layers. Meanwhile, different quantization parameters usually represent different degrees of distortion. Therefore, introduc-

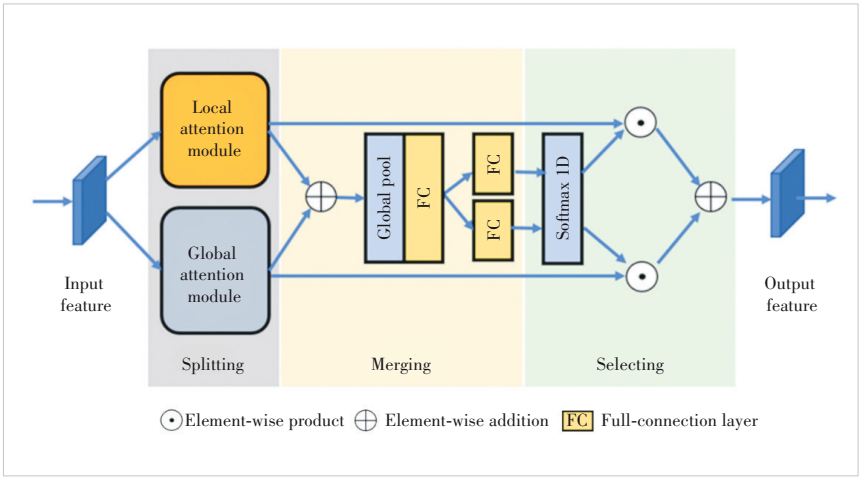



▲Figure 3. Proposed video restoring network structure

ing this auxiliary information facilitates guiding the network to learn and eliminate different degrees of compression noise.

### 3.1 Overall Pipeline

We first apply a convolution layer to fuse the lossy video frame and the corresponding QP value information. The designed structure raises the channel dimension and extracts the low-level features of the image. Then the features above go through several CTBs (Fig. 3) and are transformed into a residual image by another convolution layer. The final reconstructed video frame is obtained by adding the residual image and the original lossy frame. Moreover, we normalize the QP value by scaling it to 1/63 of the original value. Through the whole process, CTB deeply extracts features from the input. CTB itself includes several residual blocks ahead to preprocess the input image and obtain a representation suitable for the CTFM (Fig. 4) to recover the features. The structure of ResBlock includes a 3×3 convolution layer at the beginning and the end. For the non-linear part, we use a Parametric Rectified Linear Unit (PReLU) activation function in the middle, and an addition connection is then created to obtain a stronger generalization ability of the residual block.




▲Figure 4. Conv-tran fusion module

Specifically, the network input is a video frame consisting of three channels: Y, U, and V. After the local and global feature fusion of several CTB blocks, the correlation of luminance and chrominance components has been improved. Moreover, in the following formulas, our designed loss function in Eq. (8) weights the three components separately based on the human eye's sensitivity to different visual information. This will fully exploit the correlation between luminance and chrominance to achieve better performance.

### 3.2 Conv-Tran Fusion Module

The purpose of the proposed fusion sub-module CTFM is to integrate the characteristics of CNN and Transformer and thus provide stronger feature extraction capability for CTB blocks. The combination modes of CNN and Transformer can be divided into two types: serial combination, such as SwinIR, and parallel combination, such as Swin-Conv-Unet. We comprehensively design the fusion structure with reference to the latter's parallel combination and further improve their fusion mode by using the channel attention mechanism. Specifically, instead of dividing the input features during the combination proces, we first make the whole input go through the CNN and Transformer respectively to ensure the integrity of the input features. Secondly, we introduce a channel attention fusion structure inspired by SKNet[16]. The final output of the CTFM module is a weighted sum of the recovery information output by the two branches.

CTFM is structured into three parts: splitting, merging, and selecting. In the splitting step, the input feature $F_{in}$ is sent to a Local Attention Module (LAM) that is based on the convolution layer, and a Global Attention Module (GAM) that consists of a Transformer. The two modules analyze the features from different perspectives to obtain a local representation $F_{la}$ and a global representation $F_g$ respectively. In order to deeply com-

bine the modeling capabilities of LAM and GAM, in the merging step, we obtain the weights of the above two modules. The channel fusion mechanism of SKNet is actually adopted in this step, and we can get the two weights $\tilde{W}^c_{la}$ and $\tilde{W}^c_{ga}$ of the two channels. In the implementation, the merging module first performs an additional operation on the representations output by the two modules, as shown in Eq. (1). And then, $F_{gla}$ is processed by a global average pooling layer to get the statistical information on the channel. The information passes through a full connection layer (FCS), which transfers the information from the recovery domain to the weight domain. Finally, the channel weights $\tilde{W}^c_{la}$ and $\tilde{W}^c_{ga}$ of the two modules are extracted through another linear layer, as shown in Eq. (2).

$$F_{gla} = F_{la} + F_{ga}, \tag{1}$$

$$\tilde{W}^c_{la}, \tilde{W}^c_{ga} = \mathrm{FCS}\left(\mathrm{GlobalPool}\left(F_{gla}\right)\right). \tag{2}$$

In the selection part, first, we use SoftMax ("SM" in the equation) to normalize the two above channel weights $\tilde{W}^c_{la}$ and $\tilde{W}^c_{ga}$, to obtain the final fusion weights $W^c_{la}$ and $W^c_{ga}$, as shown in Eq. (3). Then the restoration features output by the LAM and GAM modules are weighted and summed with these weights to obtain a deep fusion result $F_{out}$, as shown in Eq. (4).

$$W^c_{la}, W^c_{ga} = \mathrm{SM}\left(\tilde{W}^c_{la}, \tilde{W}^c_{ga}\right), \tag{3}$$

$$F_{out} = W^c_{la} \times F_{la} + W^c_{ga} \times F_{ga}. \tag{4}$$

### 3.3 Module Details

The fusion method used by CTFM contains the local feature extraction module LAM and the global feature extraction module GAM, which are based on CNN and Transformer respectively. Their module details are shown in Fig. 5.

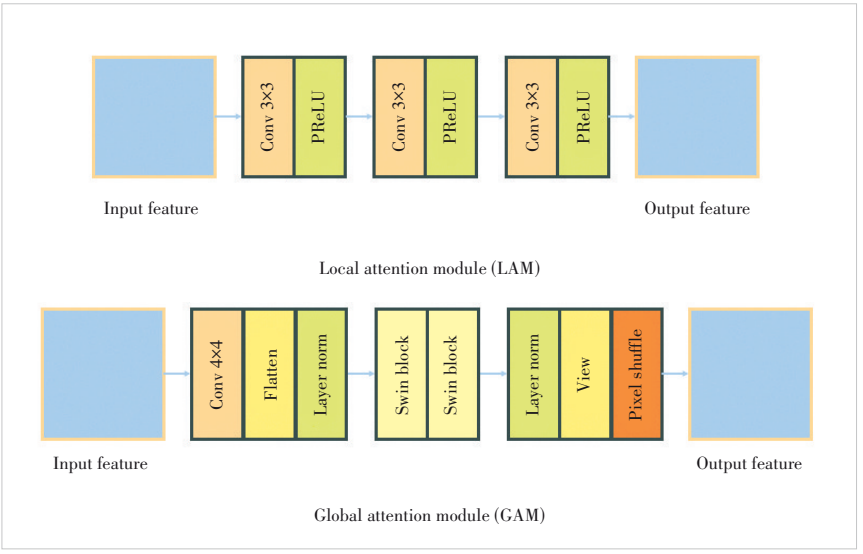
The local feature extraction module LAM is based on the CNN structure, using three cascaded 3×3 convolution layers, and after each convolution layer, the PReLU activation function is used for nonlinear mapping. LAM uses the limited receptive field of the convolution kernel to observe and process the local area features and obtains the feature representation from the local angle. The global feature extraction module GAM extracts features based on the structure of the Swin Transformer[12]. The input image is three-dimensional, but Transformer can only process two-dimensional information. Therefore, the two problems that need to be solved are how to convert three-dimensional features into two-dimensional features for Transformer to process (embedding), and how to convert the dimension back after Transformer outputs features. Our design scheme for the above two problems is as follows. For the GAM (W, H, C) shaped input, we first use the 4×4 convolution layer with the kernel size equal to the stride to complete the block-splitting operation. After this, the shape turns to (W/4, H/4, 16 × C), and then we use the transpose and flatten functions to obtain a two-dimensional shape of (16 × C, W/4 × H/4).

For the output of Transformer, on the contrary, the output of the features by Swin Block needs to be embedded reversely. First, we use the view function to transform the features from 2D to 3D (W/4, H/4, 16×C). Then we employ pixel-shuffle to up-sample the feature (W, H, C). It should be noticed that the window size of the Swin Transformer block is set to 8, that is, each window calculates the global similarity of 8×8 on 16C-dimensional vectors; the shift distance is set to 4, which is half of the window size, to ensure that the information can be fully exchanged between the shifted windows. The number of multi-headed self-attention blocks is set to 4, which ensures that the global feature representation can be extracted from multiple angles.

## 4 Experiments

### 4.1 Datasets

We use BVI-DVC[17] as the training set, and the standard test sequence defined in JVET Common Test Condition (CTC)[18] as the test set. BVI-DVC includes 800 video sequences in the YUV420 format, with various resolutions: 3 840×2 176, 1 920×1 088, 960×544, and 480×272. Each resolution category contains 200 video sequences and we choose 191 video sequences from each resolution of BVI-DVC under CTC. The standard test sequences defined by CTC (Table 1) cover differ-




▲Figure 5. Detailed structure of LAM and GAM

ent resolutions, frame rates, bit depths, and video sequences with features such as complex foreground, complex background, simple or strenuous movement, and complex textures.

Before the experiment, we obtain the video sequence with compressed noise for the training and test by encoding and decoding the original video using the VVC codec. We use the VTM11.0-NNVC coding framework[19] to compress and encode all video sequences and decode the stream output to obtain a video sequence with compressed noise. These lossy video sequences and the original lossless video sequences contribute to the dataset together. We employ the random access encoding pattern, which has the maximum compression ratio and can effectively test the recovery effect of our recovery model. To train and test the recovery effect of the model under different bit-rate conditions, each video sequence is encoded and decoded on the following five QP values: 22, 27, 32, 37, and 42.

The dataset details are as follows. To simplify the training set, only the first 16 frames are compressed into the training set for videos with a resolution of 3 840×2 176. While for other lower resolution sequences, the first 64 frames are compressed and included in the training set. Each video is produced to get five corresponding lossy videos according to the mentioned QP value requirements. The final training set contains a total of 198 640 images. It should be noted that the bit depth of the test set video sequence is different. Therefore, to normalize the dataset, all 8-bit test videos are processed to 10 bits in advance.

### 4.2 Training Details

All experiments in this paper are performed on Ubuntu 18.04.1 using NVIDIA GeForce RTX 3060 with a total video memory size of 24G. Pytorch 1.10.0 is used as the deep learning framework with the CUDA version 11.4. We update the network model parameters using the Adam optimizer with the optimization coefficients set to 0.9 and 0.999. The initial learning rate is set to 2e−4, and for every 1e5 epoch, the learning rate decreases by 0.5, with the final learning rate decreasing to 5e − 5. The video frames to be recovered are in the YUV420 format, where every four luminance components share a pair of chromaticity components. As neural networks cannot accept irregular input, we convert the YUV420 video frame into the YUV444 format using the nearest neighbor interpolation method before the input lossy video frame is restored. After we obtain the recovered video frame, the components of the restored frame are down-sampled back to the YUV420 format using 2×2 average pooling to compare with the lossy video frame.

▼Table 1. Standard testing sequences in CTC

| Class | Video Sequence | Frames | Resolution | FPS | Bit Depth |
|---|---|---|---|---|---|
| Class A1 | Tango2 | 294 | 3 840×2 160 | 60 | 10 |
| | FoodMarket4 | 300 | 3 840×2 160 | 60 | 10 |
| | Campfire | 300 | 3 840×2 160 | 30 | 10 |
| Class A2 | CatRobot | 300 | 3 840×2 160 | 60 | 10 |
| | DaylightRoad2 | 300 | 3 840×2 160 | 60 | 10 |
| | ParkRunning3 | 300 | 3 840×2 160 | 50 | 10 |
| Class B | MarketPlace | 600 | 1 920×1 080 | 60 | 10 |
| | RitualDance | 600 | 1 920×1 080 | 60 | 10 |
| | Cactus | 500 | 1 920×1 080 | 50 | 8 |
| | BasketballDrive | 500 | 1 920×1 080 | 50 | 8 |
| | BQTerrace | 500 | 1 920×1 080 | 60 | 8 |
| Class C | RaceHorses | 300 | 832×480 | 30 | 8 |
| | BQMall | 600 | 832×480 | 60 | 8 |
| | PartyScene | 500 | 832×480 | 50 | 8 |
| | BasketballDrill | 500 | 832×480 | 50 | 8 |
| Class D | RaceHorses | 300 | 410×240 | 30 | 8 |
| | BQSquare | 600 | 410×240 | 60 | 8 |
| | BlowingBubbles | 500 | 410×240 | 50 | 8 |
| | BasketballPass | 500 | 410×240 | 50 | 8 |

CTC: Common Test Condition FPS: frames per second

As to the parameters, for all the resblocks in the model, we set the kernel size, padding and stride to 3, 1, and 1. In Swin Transformer, the window size is 8 while the extraction depth is 6. Considering the input frame 256×256, the average decoding time for four classes from D to A is 52.99 s, 193.26 s, 1 188.66 s, and 4 232.59 s. Furthermore, we measure the computational complexity of the proposed model. Under the metrics of floating point operations (FLOPs), our model achieves a complexity of $1.673\,1\times10^{11}$ FLOPs.

During training, we randomly divide video frames into several blocks with the size of 256×256 from the dataset, randomly flip them, rotate them for data augmentation, and then put them into the network. The batch size is set to 6, and the model is trained for $2\times10^{6}$ iterations. To be noticed, Class B, C and D video frames are fully input into the network for the recovery test. However, the resolution of Class A is too high because it occupies too many video memory resources during the calculation. Therefore, the video frames of Class A are divided into image blocks of 960×960 with 32 overlapping pixels for testing, and the output is the average value of the corresponding position of each image block.

### 4.3 Evaluation Criteria

In video coding enhancement tasks, two commonly used evaluation methods are the video quality after decoding and the stream size during video encoding. To assess video quality, PSNR is the most commonly used metric for measuring the similarity between the original and the recovered signals. The equation is as follows:

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{\left(255 \ll \left(\mathrm{bitDepth} - 8\right)\right)^2}{\mathrm{MSE}}\right), \tag{5}$$

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j} - \widetilde{x_{i,j}}\right)^2, \tag{6}$$

where bitDepth represents the bit depth of the video frame, MSE is the mean square error, $x$ is the original signal, $\tilde{x}$ represents the signal to be evaluated, and $M$ and $N$ are the length and width of the video frame, respectively. The larger the PSNR is, the smaller the distortion of the signal. Other visual evaluation criteria, such as structural similarity (SSIM) and multi-scale structural similarity (MS-SSIM)[20], are designed based on the human eye's view of the image to extract structured information and thus are more sensitive to local structure transformations. PSNR is commonly used due to its convenience and consistency with image quality, while MS-SSIM is also used as an auxiliary evaluation index to make up for the deficiency of the PSNR. To obtain quantitative results, we calculate the outputs of all video frames according to the quantitative indicators, and the mean value of all video frames is the final quantitative result of the video. The bit rate is also important for measuring coding performance, as a lower bit rate indicates higher compression. To test the recovery effect of the enhancement model on the decoded videos under different QP, we select Bjontegaard Delta-Rate (BD Rate) as the evaluation metrics of coding performance. When the BD Rate is negative, a higher bit rate is saved under the same objective quality, and the codec has a better effect after using this recovery method. We calculate the BD Rate based on PSNR and SSIM to measure the enhancement effect of the mode on VVC.

We employ Charbonnier Loss as the loss function for training CTN. It has two advantages: firstly, it alleviates the problem that small gradients occur when L1 Loss approaches 0. Secondly, for values far from zero, Charbonnier Loss does not cause gradient explosions. Assuming the original lossless data is represented as $x$ and the data recovered by the model after restoration is represented as $\tilde{x}$, with $N$ samples used for each iteration during training, the expression of Charbonnier Loss is shown in the following equation.

$$L_{chb}(x,\tilde{x}) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x-\tilde{x})+\epsilon^2}. \tag{7}$$

Due to the human eye's different sensitivity to luminance and chrominance, we use different weights for YUV. Specifically, the weight ratio is Y:U:V = 10:1:1, and the final loss function used for model optimization is as follows.

$$L_{\text{total}} = 10\times L_{\text{chb}}\left(x_Y,\widetilde{x_Y}\right) + L_{\text{chb}}\left(x_U,\widetilde{x_U}\right) + L_{\text{chb}}\left(x_V,\widetilde{x_V}\right). \tag{8}$$

### 4.4 Results and Analysis

The CTN-Channel (CTN-C) model is tested on a GPU throughout the whole process. After training, the recovery effect is tested under QP = 32. The quantitative outcomes are presented in Table 2, which reveals that our proposed restoration model, CTN-C, can improve the average PSNR of the decoded lossy video on Y, U, and V components by 0.07 dB,

▼Table 2. PSNR under the random access mode

| Class | Video Sequence | VTM11.0-NNVC | | | | VTM11.0-NNVC with CTN-C | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Y-PSNR /dB | U-PSNR /dB | V-PSNR /dB | Decoding Time/s | Y-PSNR /dB | U-PSNR /dB | V-PSNR /dB | Decoding Time/s |
| A1 | Tango2 | 38.86 | 47.51 | 44.89 | 65.64 | 38.88 | 47.63 | 44.83 | 4 181.96 |
| | FoodMarket4 | 41.21 | 46.00 | 46.16 | 70.30 | 41.23 | 46.02 | 46.19 | 4 234.27 |
| | Campfire | 36.52 | 35.97 | 39.81 | 80.85 | 36.54 | 36.03 | 39.86 | 4 267.28 |
| A2 | CatRobot | 38.41 | 40.82 | 41.37 | 65.04 | 38.45 | 40.90 | 41.44 | 4 250.03 |
| | DaylightRoad2 | 36.47 | 44.15 | 41.85 | 69.64 | 36.51 | 44.29 | 41.91 | 4 278.40 |
| | ParkRunning3 | 36.50 | 32.89 | 34.59 | 106.77 | 36.53 | 33.00 | 34.63 | 4 285.42 |
| B | MarketPlace | 36.86 | 41.96 | 42.80 | 36.46 | 36.89 | 42.16 | 42.93 | 1 270.89 |
| | RitualDance | 38.73 | 44.30 | 44.25 | 35.87 | 38.80 | 44.44 | 44.45 | 1 269.88 |
| | Cactus | 35.60 | 38.87 | 41.22 | 27.34 | 35.61 | 38.92 | 41.30 | 1 059.97 |
| | BasketballDrive | 36.31 | 42.06 | 42.23 | 46.94 | 36.33 | 42.16 | 42.39 | 1 078.13 |
| | BQTerrace | 34.38 | 40.84 | 43.44 | 30.64 | 34.35 | 40.96 | 43.48 | 1 265.19 |
| C | BasketbalDrill | 35.79 | 39.87 | 40.10 | 6.83 | 35.85 | 40.05 | 40.32 | 204.68 |
| | BQMall | 35.82 | 41.07 | 42.03 | 10.80 | 35.91 | 41.34 | 42.26 | 247.90 |
| | PartyScene | 32.59 | 38.04 | 38.89 | 10.98 | 32.71 | 38.34 | 38.98 | 208.78 |
| | RaceHorses | 33.75 | 37.51 | 39.50 | 9.57 | 33.79 | 37.69 | 39.65 | 128.08 |
| D | BasketballPass | 34.18 | 39.82 | 38.13 | 3.20 | 34.33 | 40.12 | 38.42 | 58.09 |
| | BQSquare | 32.64 | 40.59 | 41.68 | 1.91 | 32.98 | 40.76 | 41.92 | 67.57 |
| | BlowingBubbles | 32.58 | 37.51 | 38.28 | 2.88 | 32.71 | 37.81 | 38.38 | 58.01 |
| | RaceHorses | 33.13 | 37.16 | 38.49 | 1.37 | 33.24 | 37.44 | 38.70 | 34.16 |
| Overall | | 35.81 | 40.37 | 41.04 | 35.95 | 35.88 | 40.53 | 41.16 | 1 707.83 |

CTN: Conv-Tran Network PSNR: peak signal-to-noise ratio

0.17 dB and 0.12 dB, respectively. For the video sequence that has the best effect, our obtained PSNR improvement is 0.34 dB, 0.30 dB and 0.29 dB. Additionally, the model enhances video quality in almost all video sequences, indicating its good generalization ability. Furthermore, the model takes both luminance and chrominance as inputs and effectively utilizes the correlation between different components in the image space to significantly enhance the visual effect of the chrominance components.

According to the JVET CTC, we also perform tests under five quantization factors QP = 22, 27, 32, 37 and 42. Compared to the decoded lossy video after VTM11.0-NNVC, the BD-rate evaluation results based on PSNR and MS-SSIM (MSIM in the table) are presented respectively. The evaluation data are shown in Table 3.

The recovery effect of CTN-C on VTM11.0-NNVC shows the following characteristics:

1) It achieves quality recovery for all test classes, with the most significant effect on Class D, which achieves a bit rate saving effect of 4.09% in the luminance component. This indicates that the proposed model performs better in recovering sequences with a small resolution.

2) CTN-C demonstrates more effective performance on the chrominance component than the luminance component. The main reason is that, during video compression, chrominance information is always down-sampled and much sparser because of the human visual system's greater sensitivity in luminance than chrominance. In the YUV420 format dataset used in our study, the luminance component maintains the same shape as the resolution, but the chrominance component has only half the width and height of the resolution. Meanwhile, the human eye is more sensitive to luminance than to chrominance[21 – 22]. Therefore, video coding and compression algorithms typically apply stronger compression to chrominance (U and V components). As a result, the chrominance becomes relatively sparser compared to the luminance in the video data and is easier to obtain better gain effects.

3) On average, the decoding time required by CTN-C increases by about 30 times, which is a weakness of the neural network compared to traditional image processing algorithms. Although GPU acceleration is used in the experiment, the decoding time still significantly increases.

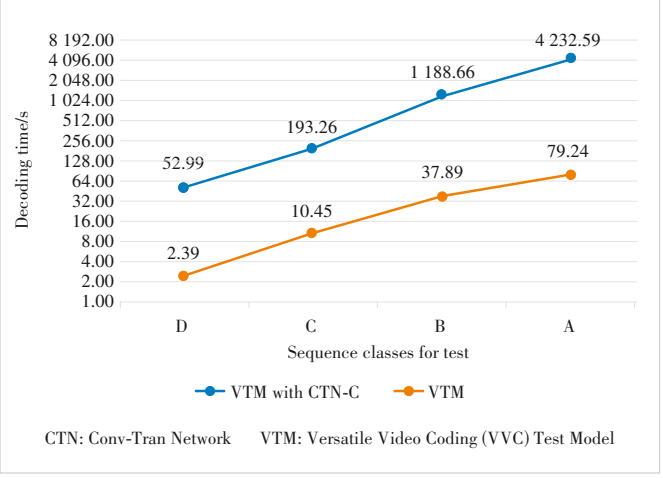
To further analyze the decoding time of the CTN-C at different resolutions, we summarize the average time for each test class in Fig. 6.

In Fig. 6, we can observe that the decoding time of CTN-C increases with video resolution since the neural network's calculation complexity on each pixel is fixed. As the resolution goes higher, the time consumption of CTN-C is proportional to it. In contrast, the time consumption of the VTM decoder is gradually lower than this proportional line, so the ratio of the two models' decoding time (CTN-C and VTM) increases with the test class resolution.

▼**Table 3. Improvement of CTN-C compared with VTM11.0-NNVC under the random access mode**

| Class | Y-PSNR | U-PSNR | V-PSNR | Y-MSIM | U-MSIM | V-MSIM | DecT |
|---|---|---|---|---|---|---|---|
| Class A1 | −0.80% | −2.87% | −0.12% | −1.02% | −5.32% | −1.28% | 5 415% |
| Class A2 | −1.54% | −6.75% | −2.47% | −1.45% | −5.30% | −1.35% | 5 268% |
| Class B | −0.27% | −6.22% | −4.84% | −1.21% | −5.91% | −4.31% | 3 137% |
| Class C | −1.91% | −8.13% | −5.53% | −1.40% | −6.47% | −3.67% | 1 849% |
| Class D | −4.09% | −8.58% | −6.39% | −1.99% | −6.53% | −3.91% | 2 220% |
| Overall | −1.70% | −6.68% | −4.19% | −1.42% | −5.97% | −3.15% | 3 087% |

MSIM: multi-scale structural similarity PSNR: peak signal-to-noise ratio




▲**Figure 6. Decoding time with respect to the test class**

However, in Table 3, it is paradoxical that the ratio of Class D with a smaller resolution is higher than that of Class C. This is because, to satisfy the minimum pixel partition condition of the Swin Transformer window, an additional mirror-filling operation on the borders of the video frame is required for Class D. As a result, CTN-C takes more time to recover the filled area. Consequently, the recovery time per pixel for the original video frame becomes longer, specifically 5.4e−4 unit time for D and 4.8e−4 for C, and the ratio of the two models' decoding time in Class D is longer than that in Class C.

Table 4 presents the details of the bit rate optimization of CTN-C for VTM11.0-NNVC on each video sequence. The average recovery effects on video frame quality at high QP (27 – 42) are −1.85%, −7.23%, and −4.50%, compared with the recovery effects of −1.62%, −6.14%, and −4.01% at low QP values (22 – 37). The increased QP value results in a higher compression ratio, leads to more information loss, and provides more space for the quality recovery model.

### 4.5 Ablation Study

We present ablation experiments to validate the effect of the recovery model's channel attention fusion mechanism proposed in the CTFM fusion module. Specifically, the channel attention fusion structure is removed from CTFM, and the outputs of LAM and GAM are directly added. Then we retrain

▼Table 4. BD rate (PSNR) of CTN-C on VTM11.0-NNVC under the random access mode

| Class | Video Sequence | Low QP (22 – 37) | | | High QP (27 – 42) | | |
|---|---|---|---|---|---|---|---|
| | | Y | U | V | Y | U | V |
| A1 | Tango2 | −0.78% | −6.16% | 2.14% | −1.04% | −7.10% | 2.35% |
| | FoodMarket4 | −0.62% | −0.21% | −0.73% | −0.87% | −0.56% | −0.82% |
| | Campfire | −0.48% | −1.16% | −1.46% | −1.07% | −2.00% | −2.27% |
| A2 | CatRobot | −1.75% | −6.47% | −3.11% | −1.98% | −7.69% | −3.81% |
| | DaylightRoad2 | −2.41% | −7.77% | −2.61% | −2.13% | −9.33% | −1.95% |
| | ParkRunning3 | −0.35% | −3.96% | −1.70% | −0.70% | −4.78% | −1.68% |
| B | MarketPlace | −1.05% | −9.68% | −6.17% | −1.23% | −10.11% | −7.17% |
| | RitualDance | −1.53% | −4.21% | −5.29% | −1.50% | −5.12% | −6.41% |
| | Cactus | −0.19% | −3.63% | −3.08% | −0.59% | −5.55% | −3.28% |
| | BasketballDrive | −0.54% | −4.98% | −5.63% | −0.90% | −4.91% | −5.62% |
| | BQTerrace | 3.18% | −6.99% | −3.04% | 1.66% | −5.73% | −2.75% |
| C | BasketbalDrill | −1.49% | −5.63% | −6.27% | −1.58% | −6.37% | -6.36% |
| | BQMall | −2.26% | −8.41% | −6.42% | −2.37% | −10.39% | −7.65% |
| | PartyScene | −3.22% | −9.02% | −2.67% | −2.88% | −11.47% | −3.38% |
| | RaceHorses | −0.57% | −6.47% | −5.76% | −1.05% | −8.04% | 6.85% |
| D | BasketballPass | −2.97% | −8.83% | −7.96% | −3.24% | −10.70% | −8.35% |
| | BQSquare | −8.20% | −6.02% | −7.88% | −8.13% | −6.53% | −8.39% |
| | BlowingBubbles | −3.14% | −8.62% | −2.49% | −3.11% | −10.78% | −3.70% |
| | RaceHorses | −2.39% | −8.40% | −6.13% | −2.42v | −10.26% | −7.32% |
| Overall | | −1.62% | −6.14% | −4.01% | −1.85% | −7.32% | −4.50% |

BD rate: Bjontegaard Delta-Rate CTN: Conv-Tran Network PSNR: peak signal-to-noise ratio QP: quantization-parameter

the model based on the experimental environment configuration. The modified model is named CTN-E. Table 5 shows the recovery outcomes obtained by CTN-E on VTM11.0-NNVC and we compare them with the gain effect of CTN-C. The results demonstrate that the channel attention fusion mechanism enhances the image recovery model's performance by saving 1.06%, 3.61% and 3.56% in the BD rate for the Y, U, and V components respectively. This indicates that the fusion mechanism effectively improves the model's image recovery capabilities.

To better demonstrate the superiority of the feature extraction module proposed in our study, we conduct extra ablation experiments to prove that both LAM and GAM are necessary parts of our model. Concretely, we remove LAM and the fusion mechanism from CTFM, remain only GAM, and name the modified model CTN-G. Then we also remove GAM and the fusion structure, leave LAM alone, and name it CTN-L. The two models are then trained under the same experimental environment and hyperparameters. Table 6 shows the output of three models on VTM.

It is observed that CTN-C performs the best in almost all classes, owing to its two-perspective feature extraction module and adaptive fusion mechanism. Moreover, due to Transformer employed by CTN-G, it can better explore global features and has superior modeling capabilities compared to traditional CNNs, leading to better performance on the test set compared with CTN-L.

### 4.6 Visualization

The video quality enhancement model is evaluated using the

▼Table 5. Gain effects of CTN-C compared with CTN-E

| Class | VTM with CTN-C | | | VTM with CTN-E | | |
|---|---|---|---|---|---|---|
| | Y-PSNR | U-PSNR | V-PSNR | Y- PSNR | U- PSNR | V- PSNR |
| Class A1 | −0.80% | −2.87% | −0.12% | −0.20% | −1.91% | −0.41% |
| Class A2 | −1.54% | −6.75% | −2.47% | −0.74% | −3.04% | −0.40% |
| Class B | −0.27% | −6.22% | −4.84% | −0.11% | −2.82% | −0.36% |
| Class C | −1.91% | −8.13% | −5.53% | −0.59% | −3.38% | −0.96% |
| Class D | −4.09% | −8.58% | −6.39% | −1.88% | −3.98% | −0.99% |
| **Overall** | −1.70% | −6.68% | −4.19% | −0.64% | −3.07% | −0.63% |

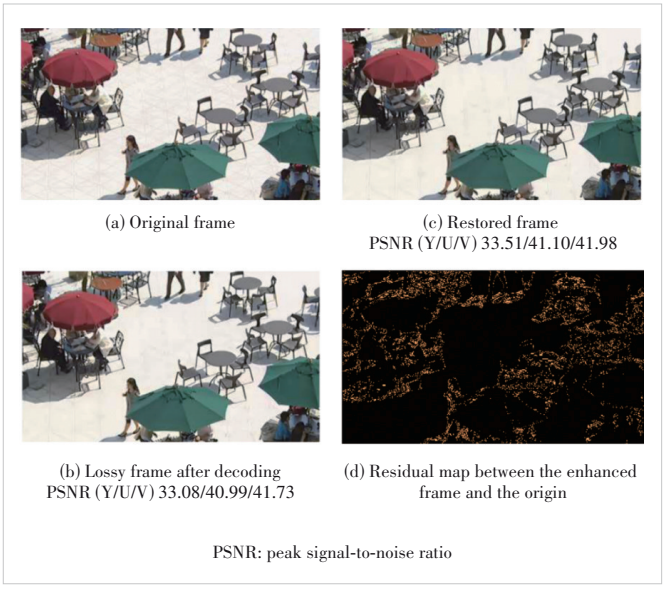
CTN: Conv-Tran Network PSNR: peak signal-to-noise ratio VTM: Versatile Video Coding (VVC) Test Model

▼Table 6. Comparison of CTN-G and CTN-L with CTN-C on VTM

| Class | VTM with CTN-C | | | VTM with CTN-G | | | VTM with CTN-L | | |
|---|---|---|---|---|---|---|---|---|---|
| | Y-PSNR | U-PSNR | V-PSNR | Y-PSNR | U-PSNR | V-PSNR | Y-PSNR | U-PSNR | V-PSNR |
| Class A1 | −0.80% | −2.87% | −0.12% | −0.42% | | −0.37% | −0.33% | −1.74% | −0.15% |
| Class A2 | −1.54% | −6.75% | −2.47% | −1.13% | −3.77% | −1.75% | −0.69% | −3.58% | −0.95% |
| Class B | −0.27% | −6.22% | −4.84% | −0.33% | −2.79% | −2.06% | −0.18% | −2.23% | −1.36% |
| Class C | −1.91% | −8.13% | −5.53% | −0.92% | −3.19% | −2.24% | −0.74% | −2.93% | −1.58% |
| Class D | −4.09% | −8.58% | −6.39% | −1.64% | −3.53% | −3.35% | −1.13% | −2.46% | −1.87% |
| Overall | −1.70% | −6.68% | −4.19% | −0.88% | −2.83% | −2.25% | −0.49% | −1.17% | −1.07% |

CTN: Conv-Tran Network PSNR: peak signal-to-noise ratio VTM: Versatile Video Coding (VVC) Test Model

fourth frame of the BQSquare in the CTC test video sequence Class D at QP value 32. The output recovery frame of CTN-C is visualized and analyzed, and the findings are presented in Fig. 7. To get a straightforward understanding of the lossy position after encoding and the effect of video enhancement, we generate a residual map as shown in Fig. 7d on the luminance component. In Fig. 7d, the enhanced lossy frame in Fig. 7c is compared with the original video frame in Fig. 7a to observe the position of the video frame after being enhanced. The different positions with brighter colors indicate a higher difference in the residual map. As shown in Fig. 7d, the loss of the luminance component is gathered in the texture complex areas like tables, sunshade edge areas, chairs, and human bodies, indicating that the proposed video quality-enhancement model CTN-C can effectively identify the damaged areas and repair them, but still cannot be identical to the origin.




▲Figure 7. Visualization of the enhanced frame and the original

## 5 Conclusions

In this paper, we propose the adaptive fusion restoration model CTN based on CNN and Transformer, which effectively removes compression noise introduced by video codec in the random access coding mode. We focus on designing the fusion module CTFM of CTN, where a multi-branch structure is used to extract restoration features from two perspectives using CNN and Swin Transformer. Then a channel attention mechanism is used to deeply fuse the two features. We also show the detailed structure of the local and global modules LAM and GAM. Experiments show that the proposed model, CTN, takes both brightness and chroma components as inputs, effectively utilizes the inter-component correlation in the spatial domain, and significantly restores the visual effect on the chroma component. Compared with existing methods, CTN can extensively recover quality loss after VVC coding and further enhance the coding efficiency of VVC.

**References**

[1] QuestMobile. QuestMobile's 2023 annual report on china internet core trends (condensed version) [R/OL]. (2023-12-19) [2024-03-20]. https://www.questmobile.com.cn/research/report/1737028262113153026
[2] WIEGAND T, SULLIVAN G J, BJONTEGAARD G, et al. Overview of the H.264/AVC video coding standard [J]. IEEE transactions on circuits and systems for video technology, 2003, 13(7): 560 – 576. DOI: 10.1109/TCSVT.2003.815165
[3] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. IEEE transactions on circuits and systems for video technology, 2012, 22(12): 1649 – 1668. DOI: 10.1109/TCSVT.2012.2221191
[4] DONG C, DENG Y B, LOY C C, et al. Compression artifacts reduction by a deep convolutional network [C]//Proc. IEEE International Conference on Computer Vision (ICCV). IEEE, 2015: 576 – 584. DOI: 10.1109/ICCV.2015.73
[5] WANG Z Y, LIU D, CHANG S Y, et al. D3: deep dual-domain based fast restoration of JPEG-compressed images [C]//Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 2764 – 2772. DOI: 10.1109/CVPR.2016.302
[6] ORORBIA A G, MALI A, WU J, et al. Learned neural iterative decoding for lossy image compression systems [C]//Proc. Data Compression Conference (DCC). IEEE, 2019: 3 – 12. DOI: 10.1109/DCC.2019.00008
[7] DAI Y Y, LIU D, WU F. A convolutional neural network approach for post-processing in HEVC intra coding [M]//Lecture notes in computer science. Cham: Springer International Publishing, 2016: 28 – 39. DOI: 10.1007/978-3-319-51811-4_3
[8] YANG R, XU M, WANG Z L. Decoder-side HEVC quality enhancement

with scalable convolutional neural network [C]//Proc. IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2017: 817 – 822. DOI: 10.1109/ICME.2017.8019299
[9] QI Z Y, JUNG C, LIU Y, et al. CNN-based post-processing filter for video compression with multi-scale feature representation [C]//Proc. IEEE International Conference on Visual Communications and Image Processing (VCIP). IEEE, 2022: 1 – 5. DOI: 10.1109/VCIP56404.2022.10008797
[10] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proc. 31st International Conference on Neural Information Processing Systems. NIPS, 2017: 6000 – 6010. DOI: 10.48550/arXiv.1706.03762
[11] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: transformers for image recognition at scale [EB/OL]. (2020-10-22)[2024-03-20]. https://arxiv.org/abs/2010.11929
[12] LIU Z, LIN Y T, CAO Y, et al. Swin transformer: hierarchical vision transformer using shifted windows [C]//Proc. IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2021: 9992 – 10002. DOI: 10.1109/ICCV48922.2021.00986
[13] CHEN H T, WANG Y H, GUO T Y, et al. Pre-trained image processing transformer [C]//Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 12294 – 12305. DOI: 10.1109/CVPR46437.2021.01212
[14] LIANG J Y, CAO J Z, SUN G L, et al. SwinIR: image restoration using swin transformer [C]//2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). IEEE, 2021. DOI: 10.1109/ICCVW54120.2021.00210
[15] ZHANG K, LI Y W, LIANG J Y, et al. Practical blind image denoising via swin-conv-UNet and data synthesis [J]. Machine Intelligence Research, 2023, 20: 822 – 836. DOI: 10.1007/s11633-023-1466-0
[16] LI X, WANG W H, HU X L, et al. Selective kernel networks [C]//Proce. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019: 510 – 519. DOI: 10.1109/CVPR.2019.00060
[17] MA D, ZHANG F, BULL D R. BVI-DVC: a training database for deep video compression [J]. IEEE transactions on multimedia, 2022, 24: 3847 – 3858. DOI: 10.1109/TMM.2021.3108943
[18] LIU S, SEGALL A, ALSHINA E, et al. Common test conditions and evaluation procedures for neural network-based video coding technology [Z]. JVET-W2016, 2021
[19] JVET. VTM-11.0-NNVC, VVC test model [EB/OL]. [2024-03-20]. https://vcgit. hhi. fraunhofer. de/jvet-ahg-nnvc/VVCSoftware_VTM/-/tree/VTM11.0_nnvc
[20] WANG Z, BOVIK A C, SHEIKH H R, et al. Image quality assessment: from error visibility to structural similarity [J]. IEEE transactions on image processing, 2004, 13(4): 600 – 612. DOI: 10.1109/tip.2003.819861
[21] CHEN L H, BAMPIS C G, LI Z, et al. Perceptual video quality prediction emphasizing chroma distortions [J]. IEEE transactions on image processing, 2021, 30: 1408 – 1422. DOI: 10.1109/TIP.2020.3043127
[22] HEINDEL A, PRESTELE B, GEHLERT A, et al. Enhancement layer coding for chroma sub-sampled screen content video [J]. IEEE transactions on circuits and systems for video technology, 2022, 32(2): 788 – 801. DOI: 10.1109/TCSVT.2021.3061944

## Biographies

**YUAN Lang** (1190201114@stu.hit.edu.cn) is now a senior student at Harbin Institute of Technology (HIT), China. He is working at the Research Center of Intelligent Interface and Human-Computer Interaction, HIT. His research interests include deep learning, image coding, and compressive sensing.

**HUI Chen** received his BS degree in software engineering from Yanshan University, China in 2017, MS degree in software engineering, and PhD in computer science and technology from Harbin Institute of Technology, China in 2020 and 2024, respectively. He has been a visiting scholar with the College of Computing and Data Science, Nanyang Technological University, Singapore since 2023. He is currently with the School of Future Technology, Nanjing University of Information Science and Technology, China. His research interests include image compression, quality assessment, and multimedia security.

**WU Yanfeng** is currently an undergraduate student at Harbin Institute of Technology (HIT), China. He has joined the Research Center of Intelligent Interface and Human-Computer Interaction, HIT. His research interests include machine learning and deep learning, with a current focus on video coding and compressive sensing.

**LIAO Ronghua** is pursuing a master's degree at the Department of Computing, Harbin Institute of Technology, China. His research interests include compressed sensing and video quality assessment.

**JIANG Feng** received his BS, MS and PhD degrees from the Harbin Institute of Technology (HIT), China in 2001, 2003, and 2008, respectively, all in computer science. He is currently a professor with the Department of Computer Science, Sichuan University of Science & Engineering, China and a visiting scholar with the School of Electrical Engineering, Princeton University, USA. His research interests include computer vision, image and video processing, and pattern recognition.

**GAO Ying** received her master's degree in mathematics from Hohai University, China in 2011. Since then, she has been engaged in the research on Internet of Things (IoT), video surveillance, video transmission, and video coding. She has applied for over 100 patents in these fields. She is currently a chief engineer of standard pre-research at ZTE Corporation and a member of the State Key Laboratory of Mobile Network and Multimedia Technology in China, where she mainly focuses on the research and standardization of video coding technology.