An Image Coder With CNN Optimizations

Yundong Zhang*[1], Jianhua Hu*[2], Ming Li*[2], Changsheng Xia*[2], Jinwen Zan[1]
Zhangming Huang[2], Weiquan Liao[2], Dekai Chen[2], Jing Nie[2]

[1]Vimicro AI Chip Technology Corporation
The National Key Laboratory of Digital Multimedia Chip Technology
Room607,6/F, Shining Tower, No.35 Xueyuan Road, Haidian District, Beijing 100191 China
[2]Guangdong Vimicro Microelectronics Corporation
Building 16,Hengqin Financial District, Hengqin New Area, Zhuhai City, Guangdong Province, P.R.C
hu.jianhua@zxelec.com

Abstract

Convolutional neural networks (CNNs) has achieved great success in image processing and computer vision, especially in high level vision applications, such as classification and image compression. In this paper, CNN based optimizations have been proposed to improve the performance of an open source image coder, and the coding gain mainly comes from three modules: firstly, a classification CNN is employed to generate a region of interest (ROI) map, highlighting the part of the image containing more visual information that might be more sensitive to coding loss than other part, and thus guiding the bit allocation; secondly, a remedy CNN is introduced on the reconstructioned YUV image, to learn and compensate for the coding loss; thirdly, adaptive loop filter(ALF alorithm is applied to carry out color space conversion, and to minimize the color information loss during conversion. The improvement of the proposed optimizations, both objectively and subjectively, has been demonstrated on the CLIC validation data set.

1. Introduction

In recent years, image compression attracts increasing interest in image processing and computer vision due to its potential applications in many vision systems. Many image compression methods have been developed to efficiently compress the image such as JPEG, WebP, H.265 and H266. But compressed images and videos often suffer from block and ringing artifacts for areas with rich texture and sharp edges, especially when the bit-rate is relatively low. Human vision naturally focuses on familiar objects, and is particularly sensitive to distortions of these objects as compared to distortions of background details [1], so in this paper, we have improved the subjective quality of region of interesting by using a higher bit rate to encode, and lowering the bit rate in background region to guarantee high coding efficiency. Also we proposed an approach to reduce artifacts after reconstructing image by applying deep CNNs in-loop filter. Moreover, due to the lossy of yuv-to-rgb translation, we proposed a adaptive loop filter filter in RGB color space.

*These authors share first-authorship

2. The Proposed Compression Methods

In the proposed approach, we develop our codec based on the VTM/H266 platform. Three algorithms have been proposed to improve the performance of H266 codec: CNN based in-loop filter (CNNIF) after reconstruction frame; CNN based regions of interesting control different quantization parameters(QP); ALF to enhance the coding performance.

Our image compression framework is shown in Fig 1. Each image is split into block-shaped regions, and coded using intra prediction and other coding modules. The residual signal of intra prediction is transformed by a linear spatial transformation. The transform coefficients are then scaled, quantized and coded with entropy coding. Moreover, we apply different QP values according to the ROI mapping which is generated using CNN network to improving the subjection quality. The raw input image is RGB fromat, we transfer it into yuv420 format, and then encode to bitstreams with H266 encoder. After reconstructing the yuv420 data by the h266 decoder and interpolating chroma u/v by cubic interpolation, we get yuv444 data, and then a CNN network is used to filter luma y and chroma u/v data to get better image quality. We transform yuv444 into the RGB data. Finally we apply a RGB color space based adaptive loop filter to filter the RGB data and pack it to PNG format.

2.1. CNN ROI Use Different QP(ROIDQP)

In a traditional classification cnn, e.g.VGG-16, there are two fully-connected (non-convolutional) layers as the final layers of the network. The final layer has one neuron for every class in the training data, and the final step in the in-

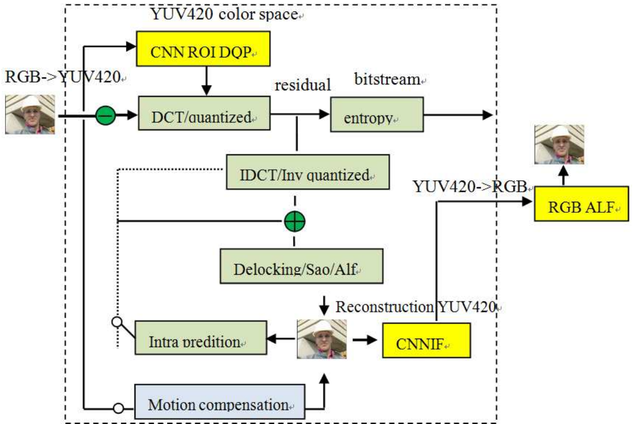

Figure 1. The architecture of our image codec

ference is to normalize the activations of the last layer to sum to one. The second to last layer, however, is fully connected to the last convolution layer, and a non-linearity is applied to its activations. Since the output of this layer is connected directly to the classification layer, each class will in essence learn a weight for each feature map from the final convolution layer. So we refer VGG-16 network architectures and revise a little in the final layers of the network. The detailed structure of train network model is listed in Table 1. After useing 13 convolution layers(channel: 64, 64, 128, 128, 256, 256, 256, 512, 512, 512, 512, 512, 512), the size of output convolution layers is [14][14][512], and then we apply 3 convolution layers more(channel: 32, 1024, 1024). So we can get last convolution layers with size of [14][14][1024], which can be output for generating ROI mapping, gap[1024] can be get by calculating the mean value of last convolution layers([14][14][1024] reduce_mean with axis[1,2]), also we will generated each of class's probability(here we use 256 classes) by multiply a weight coefficients matrix[1024][257] with gap[1024], so we can get each of classes output probability.

We signicantly reduce the number of classes by collapsing these sets of similar classes to a single, because these combined classes have similar structure and are within the same general category, also the map produced will be almost identical, so we trained our model with the Caltech-256 dataset [2], the training is carried out by optimizing the loss between classes probability and labels based on cross-entropy method.

After our model have been trained, we apply this model to generate ROI mapping. Because our model only support input image size of 224x224x3, so we scaled down the source image to size of 224x224x3, and then we can get 257 class probability and the last convolution layer [14][14][1024] with this model, class probability from max to min statistics show that only several class probability have big probability, after five top classes, most of image's

main path	image(224x224x3)
3x3x64, s = 1	
3x3x64, s = 1	
max-pool, s = 2	
3x3x128, s = 1	
3x3x128, s = 1	
max-pool, s = 2	
3x3x256, s = 1	
3x3x256, s = 1	
3x3x256, s = 1	
max-pool, s = 2	
3x3x512, s = 1	
3x3x512, s = 1	
3x3x512, s = 1	
max-pool, s = 2	
3x3x512, s = 1	
3x3x512, s = 1	
3x3x512, s = 1	[14][14][512]
3x3x32, s = 1	
3x3x1024, s = 1	
3x3x1024, s = 1	last convolution
reduce-mean[1, 2]	gap [1024]
gap x W[1024][257]	class-prob[257]

Table 1. Network architecture of ROI

probability value is decrease of 50 percent compared to the max probability value, so we pick the top five classes whcih use argsort function to reorder these class probabilitys and combine them via a sum weighted by their rank. For example, we find the max probability class is 67 class. Firstly, we extract the 67th class's weight coefficients W[1024] from all class's weight coefficients W[1024][257]. Secondly, we resize the last convolution layer [14][14][1024] to size of [224][224][1024](along with axis[1,2]) using bilinear interpolation which is the same size as the input image, and then we reshape [224][224][1024] to [50176][1024]. Thirdly, after multiplying [50176][1024] matrix by weight coefficients vector W[1024], we can get the ROI mask (cur_classmap) size of [224][244], and then normalize as :x = (x-min)/(max-min)

The ultimate roi map can be calculate as below(from low probability to high probability):

```
for(i = 0; i <5; i++) {
    if (i == 0)
        roi_map = cur_classmap
    else
        roi_map = (roi_map + cur_classmap)/2
}
```

After have get roi_map size of [224][244], then we scale up to size of source image by bilinear interpolation.

Our frame level QP value is 36 when encode the H266 bitstream, this is base QP value, and each of coding unit(CU) may have their own QP value, so we can set a QP offset (max_qp_offset) in H266 encode relative to frame QP. This is to say, when the ROI mapping value is high, we set a small QP value, otherwise a big QP value. Each of cu we can calculate a max QP offset (cur_cu_qp_offset) in the H265 encode is shown in Fig 2:

```
cu_delta = cu_average_value - global_average_value
max_diff = global_max_value - global_min_value
left_diff = global_average_value - global_min_value
right_diff = global_max_value - global_average_value
if abs(cu_delta) < k*0.5*max_diff
    cur_cu_qp_offset = 0
if (cu_average_value < global_average_value)
    cur_cu_qp_offset = cu_delta/left_diff
if (cu_average_value > global_average_value)
    cur_cu_qp_offset = cu_delta/right_diff
```

Where cu_average_value means the average ROI mapping value of current CU and global_average_value means the average ROI mapping value of all CU, k is response coefficient. what's more, global_max_value is max value of ROI mapping and global_min_value is the min value of ROI mapping.

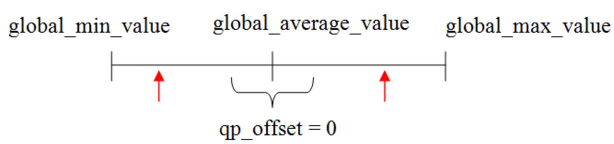

Figure 2. The calculation of CU QP offset

2.2. Reconstruction Frame Filtering With CNNs

In-loop filtering is an important technique in current mainstream codec for improving the quality of compressed image, so we design a novel CNN architecture and further boost the performance of in-loop filtering. The input of the network is the reconstruction frame in YUV color space, let's denote lumina of the H266 decoded image as Y. The proposed CNN model focuses on learning the residuals between the decoded Y and the ground truth lumina X of source image. Our goal is to fit a mapping function X≈F(Y)+Y that reverses image degradation due to H266 codec as much as possible. The whole architecture of our network is shown in Fig 3. We wish to learn the F by training a CNN, which conceptually consists of two operations: the feature extraction and image detail's reconstruction. The filter reconstruction frame model is a fully CNN network that consists of a set of convolution layers and nonlinear layers cascades. To extract both the local and the global image features, all outputs of the hidden layers are concatenated at the end of feature ex-traction as skip connections from different layer domains. After concatenating all of the features, a simple reconstruction net is used to reconstruct the image details. Input Y is fed into the network, residual is output from the second last layer, finally adding Y to form a F(Y)+Y function.

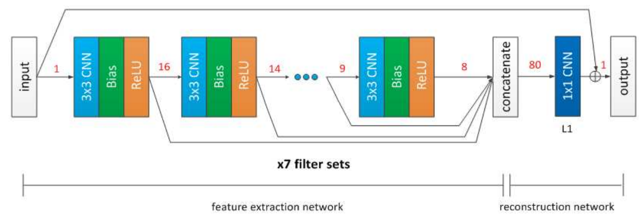

Figure 3. CNN filter structure

Although the reconstruction network is trained only on the luminance channel, there is great improvement in chroma uv reconstruction, which using the trained net of luminace. The model is not specifically designed to be an end-to-end solution. On the contrary, the proposed optimizes an end-to-end mapping. It is faster at speed because of less layers and channels.

The feature extraction part is responsible for extracting hidden features of the H266's reconstructed image. It consists of 7 consecutive 2d-convolution layers. We use Relu as activation function for each layer. We optimize the number of filters at each layer. The 7 filters with output feature num N are as follows: 16,14,12,11,10,9,8. All filter kernel size is k=3. The concatenation concatenates all layers' outputs, therefore the channels num in concatenation is 80.

The detail reconstruction part is responsible for outputing the residue of the image. Because of all of the hidden features are concatenated at the input layer of the reconstruction network, the dimension of input data is rather large. So we only use a 1x1 CNN filter as final mapping to generate output residual pixels data, not only reduces the dimensions of the previous layer for faster computation with less information loss, but also adds more nonlinearity to enhance the potential representation of the network [3]. The output residual is then added with the original input Y.

The training inputs are Y patches decoded by H266 decoder using qp36. We trained our model with BSD200 dataset[4]. Patch size is 64x64. Using MSE as the loss function favors a high PSNR.

3. ALF In RGB Color Space

Because there is a loss of YUV420 to YUV444 and YUV444 color to RGB color space, we appply ALF method to reduce loss during translation in RGB color space and R, G, B each have their own filter coefficients and master switch. The ALF was designed to minimize the error between the original frame and the reconstructed frame. It use different method in reconstructed image to divide the pixels into different classification. The pixels of the same category will share the same filter coefficients. ALF switch for each Block and its taps are decided based on the rate distortion cost. Finally, the filter parameters are coded into bitstream with entropy coding.

Ther are two pixel classification methods are employed, block-based (BA) and region-based (RA) [5]. Here we apply RA pixel classification method. In RA, the pixels are classified according to their locations with a predefined table. Therefore, for each pixel category, a special adaptive loop filter is estimated by minimizing the mean square error(MSE). In order to repeat the filter process at decoder, multiple groups of filter parameters are needed to be transmit, e.g. 16 categories in RA, which will consume lots of coding bits. Therefore, an adaptive merging process is employed to reduce the filter numbers needed to be transmit. Merging is conducted on pixel categories and their corresponding ALF coefficients based on the rate distortion optimization

Although the ALF is useful for the whole category of reconstructed pixels, it may degrade the quality of reconstructed pixels in some local areas, adaptive filter switching scheme is utilized based on different block sizes. It splits the whole image into 32x32 block size and makes a decision on each region whether filter is on or off.

In order to reduce the computational complexity, there is only one types of taps filter are defined, which are 9x7 illustrated in Fig.4. The filter taps are decided according to the rate distortion cost.

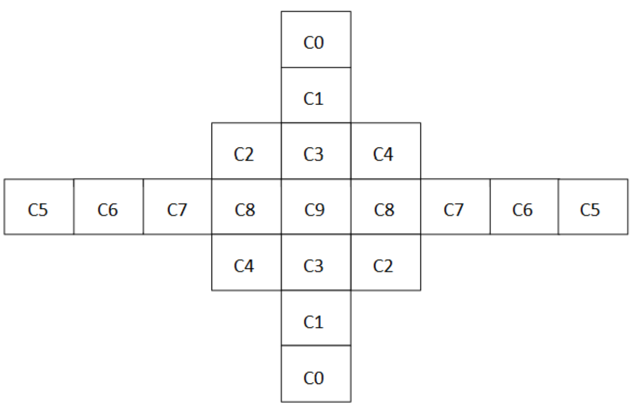

Figure 4. 9x7 filter shape

4. Result and Discussion

The encode bistream is compliance with H266 standard by ROIDQP. If CNNs in-loop filter play an important pole in improving the quality of reconstruction image, the ALF enable may be off, otherwise the ALF will be more effective in improving the the quality of reconstruction image. In CLIC, we use valid validation of the dataset(102 images), after submitting the decode test, compliance with the require BPP(0.15 bpp below), our test result is 32.04 db. The Rate-PSNR curve of our proposed method is shown in Fig.5. The performace improvements of ROIDQP, reconstruction frame filtering with CNNs and ALF is shown in Table 2:

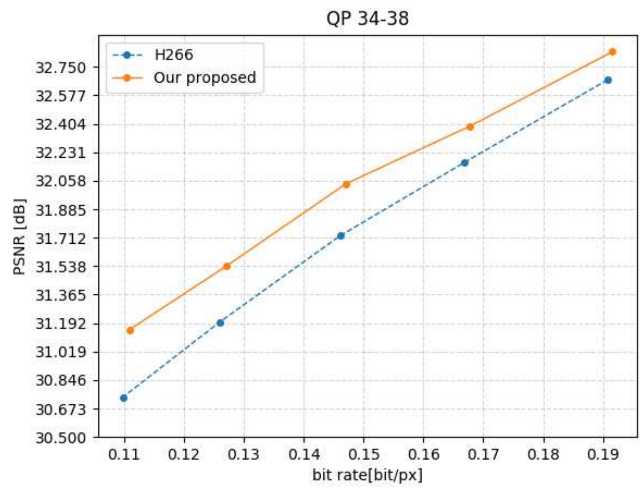

Figure 5. The Rate-PSNR curve of our proposed

	VTM4.0	ROIDQP	CNN filter	ALF
PSNR	31.735	31.806	31.973	32.040
MSSIM	0.958	0.959	0.959	0.959
BPP	0.1454	0.1482	0.1482	0.1491
Bs size	4578362	4665614	4665614	4695523

Table 2. Network architecture of ROI

In H266 encode cfg, we have opened paremater : AdaptiveQP, so each encode CU will use qp offset RDO according to the ROI mapping, we have checked the decode bitstream's QP is right(ROI have smaller QP). Although our proposed CNN ROI use different quantization parameter only have a little improvement of the psnr, there is ROI subjective image quality improvement when the RDO QP select smaller value. Our proposed CNN region of interesting detect of heat map and ROI improvement can be shown in Fig.6

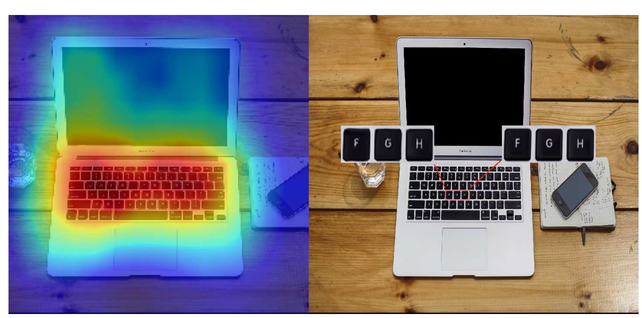

Figure 6. ROI heat map and ROI improvement

References

[1] M. Jiang, S. Huang, J. Duan, and Q. Zhao *Perceptual adaptation of objective video quality metrics*, in Proc. Ninth International Workshop on Video Processing and Quality Metrics (VPQM), 2015.

[2] G.Griffin, A.Holub, and P Perona, *Caltech-256 object category dataset*, 2007.

[3] Jin Yamanaka1, Shigesumi Kuwashima1 and Takio Kurita2. *Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network.*, 24th International Conference On Neural Information Processing (ICONIP 2017), 2017. 9

[4] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J. *Contour Detection and Hierarchical Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898–916 (2011)

[5] Xinfeng Zhang, Ruiqin Xiong, Siwei Ma, Wen Gao *Adaptive loop filter with temporal prediction*, Picture Coding Symposium, May 7-9, 2012.