Content-Adaptive Model Update of Convolutional Neural Networks for Super-Resolution

기세환, 김문철
한국과학기술원 전기 및 전자 공학과
shki@kaist.ac.kr, mkim@ee.kaist.ac.kr

Content-Adaptive Model Update of Convolutional Neural Networks for Super-Resolution

Sehwan Ki, Munchurl Kim
Korea Advanced Institute of Science and Technology Dep. Of Electronic Engineering

요 약

Content-adaptive training and transmission of the model parameters of neural networks can boost up the SR performance with higher restoration fidelity. In this case, efficient transmission of neural network parameters are essentially needed. Thus, we propose a novel method of compressing the network model parameters based on the training of network model parameters in the sense that the residues of filter parameters and content loss are jointly minimized. So, the residues of filter parameters are only transmitted to receiver sides for different temporal portions of video under consideration. This is advantage for image restoration applications with receivers (user terminals) of low complexity. In this case, the user terminals are assumed to have a limited computation and storage resource.

1. Introduction

Convolutional neural networks (CNN) for super-resolution (SR) have shown very promising performance with high fidelity of restoration [2-4]. There have been many studies on CNN-based SR which is still a very hot topic in computer vision. Also, it is possible to deliver encoded video at low spatial resolutions efficiently and to reconstruct them with high restoration fidelity at higher resolution using such CNN-based SR methods at the receiver sides. Moreover, content-adaptive training and transmission of the model parameters of neural networks can boost up the SR performance with higher restoration fidelity. In this case, efficient transmission of neural network parameters is essentially needed. This contribution proposes a novel method of compressing the network model parameters based on the training of network model parameters in the sense that the residues of filter parameters and content loss are jointly minimized. So, the residues of filter parameters are only transmitted to receiver sides for different temporal portions of video under consideration. This is advantage for image restoration applications with receivers (user terminals) of low complexity. In this case, the user terminals are assumed to have a limited computation and storage resource.

2. Method

In neural networks for content processing in [1], it was identified that transmission of updated neural network (NN) models is required for different temporal portions of the video. In order to avoid the overhead of NN model parameters for each temporal portion, it may be possible to transmit the parameter residues between the two NN models: one NN model may trained with overfitting to a temporal portion of a whole video; and the other one may be trained

with overfitting to the whole video. Also, the NN model trained for the entire video is transmitted once before video transmission, while the NN model trained for the temporal portion is transmitted before transmitting the temporal portion of the video. This is very effective for NN-based image restoration applications. In this proposal, we present a framework of transmitting the parameter residues between two NN models for super-resolution allocations where one NN model is trained with overfitting for a temporal portion and the other one is for the entire sequence.

2.1 Proposed Replicated Light-weight residual dense network for super resolution (RLRD-SR Net) with light-weight residual dense blocks (LwRDBs)

We propose a replicated light-weight residual dense network for super-resolution (RLRD-SR Net) which is composed of light-weight residual dense blocks (LwRDBs). Each LwRDB consists of three convolutional layers, the first two of which are implemented with depth-wise separable convolutions to reducing the number of training parameters and the last is a 1×1 convolution. The RLRD-SR Net has a cascade of N duplicated LwRDBs with shared parameters. The cascade of N duplicated LwRDBs with shared parameters in the RLRD-SR Net allows to enhance SR performance without increasing the number of NN model parameters. Furthermore, since the residual weights are transmitted, the weight compression efficiency is greatly enhanced by minimizing the residual weights during training.

2.2 Training of RLRD-SR Net

- Training dataset

Generally, test data is not used for deep learning training for generalization. But, since the proposed method is intended to train the RLRD-SR Net in a content-adaptive manner, the RLRD-SR Net is trained with overfitting to a specific temporal portion of a video or the entire video. Then the trained NN model parameters are transmitted. We use a training sequence of 320 frames which constitute 10 different scenes, each of which has 32 frames. The training sequence has 1920×1080@24fps as ground truth, and its low resolution version is generated by 2× downsampling and is encoded and decoded using HM16.17 reference software.

- Loss functions

The total loss function to train the RLRD-SR Net is composed of an SR loss and a weight residual (WR) cost. The SR loss is an L1 loss between ground truth frame (y) and generated SR frame ($\hat{x}$) as follow:

$$L_{SR} = L1(\hat{x}, y) \quad (1)$$

The WR cost is used to minimize the weight residues between the pre-trained reference weights (W_{ref}) for the entire sequence and the training weights (W_{train}) for a temporal portion during training as:

$$L_{WRM} = L1(W_{ref}, W_{train}) \quad (2)$$

The NN model with the pre-trained reference weights for the entire sequence is denoted as AS-model (trained model for all scenes). When training the AS-model, the total loss is one the SR loss (L_{SR}) for training. The NN model trained for one temporal portion of the entire video, it is denoted as OS-model (trained model for one scene), and the total loss consists of the SR loss and the WR cost (C_{WR}), which is given by

$$L_{total} = L_{SR} + \lambda C_{WR} \quad (3)$$

where λ is a hyper parameter and is often empirically determined. For small λ values, SR performances are increased, but the weight residual cost is increased, thus the weight compression performance is degraded, and vice versa for large λ values. In our experiments, we use λ = 0.003.

- Training conditions

First, the AS-model is trained by using the entire training video frames (320 frames). We do not use any augmentation method (only random crop is used) because we want that the network is over-fitted to the training video data. The input patch size is 60×60 and batch size is 2. The network is trained up to 64,000 iterations and learning rate is 0.001. Secondly, the OS-model is trained by using the each scene's frames (32 frames) and the trained AS-model parameters were used as reference weights. The OS-model and AS-model have the same architecture of the RLRD-SR Net, but they are trained with different training data. In training the OS-model, the input patch size is 400×400 and the initial parameters of the OS-models are set to be pre-trained AS-model parameters. The other training conditions of the OS-model are the same as those of the AS-model.

- Compression of NN model parameters

The AS-model and OS-model parameters are compressed according to the method [5]. The weight compression in [5] is taken place with binary masking for zero weights and K-means clustering for non-zero weights. Since the OS-models are trained so as to minimize the weight residues between the AS-model and the OS-model, the weight residues are finally compressed and transmitted.

3. Experiments

In order to compare the SR performance of the proposed method, we use 3 SR models which are bicubic interpolation, the AS-model, and the OS-model corresponding to the scene for up-scaling the scene of the video. Table 1 shows the PSNR and bpp values for 10 scene of the video when λ is set to 0.003 and the number of centroids of K-means clustering is set to 1,000 for the AS-model and to 300 for the OS-model. The bpp of the bicubic interpolation (bpp_{bic}) is obtained by dividing the bit size (bit_{video}) of all frames of the each scene encoded at 1200kbps using HM16.17 by the height (H), width (W) and the number of frames(n) of the up-

scaled video.

$$bpp_{bic} = \frac{bit_{video}}{W \times H \times n} \quad (4)$$

For the bpp computation of AS-model (bpp_{AS}), we have to additionally consider the bits of the trained parameters (bit_{AS}) for the initial transmission.

Table 1. The PSNR results and bpp for 3 SR models

	Bicubic interp.		AS-model		OS-model	
	PSNR	bpp	PSNR	bpp	PSNR	bpp
Scene 1	32.97	0.0266	34.06	0.0267	34.31	0.0276
Scene 2	42.02	0.0234	42.63	0.0236	42.85	0.0244
Scene 3	35.95	0.0167	37.14	0.0168	37.45	0.0176
Scene 4	34.19	0.0260	35.19	0.0262	35.60	0.0270
Scene 5	40.27	0.0220	41.67	0.0221	41.93	0.0229
Scene 6	32.12	0.0334	35.04	0.0336	35.75	0.0346
Scene 7	23.66	0.0381	24.49	0.0383	24.62	0.0393
Scene 8	31.69	0.0120	32.25	0.0121	32.41	0.0130
Scene 9	30.13	0.0199	30.69	0.0201	30.76	0.0209
Scene 10	27.75	0.0179	27.92	0.0180	28.02	0.0189

The bit size of the AS-model is divided by the number of scenes (S) to obtain the size assigned to one scene as

$$bpp_{AS} = \frac{bit_{video} + bit_{AS}/S}{W \times H \times n} \quad (5)$$

For the bpp computation of OS-model (bpp_{OS}), we need to consider the bits of trained residual parameters ($bit_{OS\text{-}residual}$) for transmission as

$$bpp_{OS} = \frac{bit_{video} + bit_{AS}/S + bit_{OS-residual}}{W \times H \times n} \quad (6)$$

It is worthwhile to see the PSNR-bpp performance for the three SR models. Fig. 1 shows the PSNR-bpp performance for bicubic interpolation, AS-model and OS-model. As can be seen in Fig. 1, the OS-model outperforms the bicubic interpolation with about 3dB higher PSNR, and is superior to the AS-model with about 0.7 dB higher PNSR for "Scene 6" of the video.

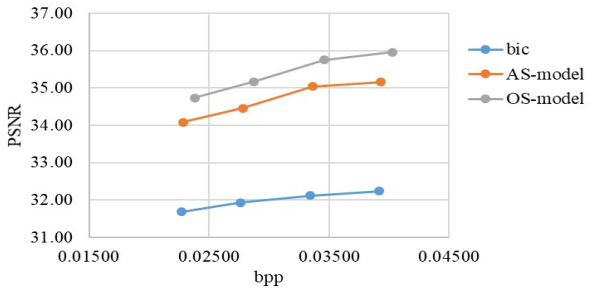

Figure 1. The PSNR results and bpp for 3 SR models for various target bitrates for "Scene 6"

4. Conclusions

We propose a novel method of compressing the network model parameters based on the training of network model parameters in the sense that the residues of filter parameters and content loss are jointly minimized. So, the residues of filter parameters are only transmitted to receiver sides for different temporal portions of video under consideration. This is advantage for image restoration applications with receivers (user terminals) of low complexity. In this case, the user terminals are assumed to have a limited computation and storage resource.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00419, Intelligent High Realistic Visual Processing for Smart Broadcasting Media).

References

[1] W. Bailer, et al, "Use cases and requirements for compressed representation of neural networks," ISO/IEC JTC1/SC29/WG11 N17924, Oct. 2018.

[2] J. Kim, et al., "Accurate image super-resolution using very deep convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2016.

[3] Zhang et al, "Image super-resolution using very deep residual channel attention networks." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[4] Zhang, et al. "Residual dense network for image super-resolution." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018.

[5] Caglar, et al, "Response to the Call for Proposals on Neural Network Compression: Training Highly Compressible Neural Networks", ISO/IEC JTC1/SC29/WG11 MPEG2019/m47379, March. 2019.