Taylor & Francis
Taylor & Francis Group

Journal of Intelligent Transportation Systems
Technology, Planning, and Operations

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: https://www.tandfonline.com/loi/gits20

A deep learning approach for quality enhancement of surveillance video

Dandan Ding, Junchao Tong & Lingyi Kong

To cite this article: Dandan Ding, Junchao Tong & Lingyi Kong (2019): A deep learning approach for quality enhancement of surveillance video, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2019.1670659

To link to this article: https://doi.org/10.1080/15472450.2019.1670659

JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS
https://doi.org/10.1080/15472450.2019.1670659

A deep learning approach for quality enhancement of surveillance video

Dandan Ding, Junchao Tong, and Lingyi Kong

School of Information Science & Engineering, Hangzhou Normal University, Hangzhou, China

ABSTRACT

The growing number of surveillance cameras imposes great demand on high efficiency video coding. Although modern video coding standards have significantly improved the video coding efficiency, they are designed for general video rather than surveillance video. The special characteristics of surveillance video leave a large space for further performance improvement. In this paper, we leverage a deep learning approach to enhance the quality of compressed surveillance video. More specifically, we formulate the problem of frame enhancement as a regression problem and design a Residual Squeeze-and-Excitation Network (RSE-Net), to address it. RSE-Net extensively exploits the non-linear mapping between the reconstructed frame and the ground truth, with only a small number of parameters. Moreover, By improving You Only Look Once (YOLO) network, we successfully detect the grouped vehicles within a frame. A novel model training scheme is then developed through learning from the grouped vehicles. With the proposed scheme, we train a global model for both foreground and background of surveillance video. Experimental results show that our method achieves average 0.40 dB, 0.22 dB and 0.24 dB PSNR gains over H.265/HEVC anchor in AI, LDP and RA configurations, and produces visually pleasing results when applied to compressed surveillance video.

ARTICLE HISTORY
Received 31 October 2018
Revised 16 September 2019
Accepted 18 September 2019

KEYWORDS
Convolutional Neural Network; image coding; intelligent vehicles; multimedia; traffic surveillance

1. Introduction

Nowadays, there are thousands of surveillance cameras deployed at intersections and along roads for urban traffic. The network constituted by these cameras, so called the dense video surveillance system, produces a huge amount of video data every second. Moreover, such videos are generally stored for several days or even months in case of reviewing special events. Although video coding techniques have been applied to compress these videos, very large memories are still required for storage.

To address the issues above, one way is to further improve the coding efficiency of surveillance video. Relative to previous coding tools, we may explore more efficient tools at the encoder side to achieve higher compression ratio. In this manner, the coding standard would be modified, which is unfriendly to the decoder. Another way is to compress the original video into lower bitrate with higher Quantization Parameters (QP), which will accordingly result in low-quality reconstructed frames. In such scenarios, we may enhance both the subjective quality and objective quality of the decoded frames, which is equivalent to improving the coding efficiency at the encoder side.

The compression of surveillance video has attracted lots of studies. Some work focuses on developing fast algorithms to reduce the complexity of encoders. Such algorithms may introduce certain coding loss. For example, Pushkar and Bharadwaj (2014) propose a skip decision and reference frame selection method for H.264/AVC encoder (ITU-T, 2003) to process surveillance video. Chen, Liu, Li, and Wu (2017) conduct a fast detection of background coding units to skip some prediction modes of H.265/HEVC (ITU-T, 2013). Consequently, the encoding time is reduced by 77.0% with 1.0% performance loss. These methods are all based on the modern video coding standards, such as H.264/AVC and H.265/HEVC, which are designed for general videos. The surveillance videos, however, are quite different from natural videos. Generally, surveillance videos are divided into background and foreground, where the background remains almost unchanged and the foreground is mainly vehicles.

Some work utilizes the background to provide better references for the inter coding of surveillance video. For instance, Zhang, Tian, Huang, Dong, and Gao (2014) regard the G-picture as the long-term reference frame to improve the background prediction in

CONTACT Dandan Ding DandanDing@hznu.edu.cn School of Information Science & Engineering, Hangzhou Normal University, Hangzhou, China.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/gits.

H.265/HEVC. Yang, Chen, Liu, Chen, and Li (2017) propose to detect the dynamic texture content by the histogram of motion direction algorithm. The texture content is then removed at the encoder and restored at the decoder directly. There are also many studies on improving the coding efficiency of foreground. Ma, Liu, Peng, and Wu (2017) observe that there are always similar vehicles passing through one static camera. Hence, they build a vehicle library and put high-quality copies of the similar vehicles into the library for additional references of inter prediction. As the vehicles may disappear from one camera and appear in another, Xiao et al. (2016) establish a 3D vehicle model library for coding.

It can be seen that previous work usually optimizes the encoding procedure of either the background or the foreground. However, the traffic flow is dynamically changing in a day, e.g., there is usually heavy traffic congestion at rush hour but seldom traffic at midnight. A general algorithm, which takes both the background and foreground into consideration, is then expected. Besides, the parameters and algorithms of prior work are generally set and tuned for different scenarios manually. It is promising to design an adaptive method which is capable of manipulating various scenarios.

In this paper, we develop a learning-based approach to enhance the quality of surveillance video. We recognize that the content of surveillance video is not as abundant as natural videos. Especially in the case of static cameras, the foreground is mostly vehicles and the background remains unchanged. We propose to employ Convolutional Neural Network (CNN) to learn such characteristics automatically. The proposed method is directly applied at the decoder side for the quality enhancement of compressed frames, without any modification to the encoding algorithms.

The highlights of this paper are summarized as follows.

1. We formulate the frame enhancement problem as a regression problem that targets to minimize the difference between the reconstructed frame and the original frame. Afterwards, we develop a novel CNN, namely RSE-Net, to address this problem. RSE-Net explores not only the correlations among neighboring pixels but also the relationship across channels to boost performance.
2. We propose a robust model training scheme for the enhancement of surveillance video. First, we improve the network You Only Look Once (YOLO) to detect the grouped vehicles in foreground. The training set is then generated from the vehicles detected. Instead of training separate models for the background and foreground, we train a global model for holistic performance improvement.
3. We integrate the proposed method into H.265/HEVC decoder for out-loop filtering, to demonstrate its effectiveness. We perform extensive experiments to evaluate the performance of the proposed network and the training strategy, compared to state-of-the-art methods. The computational complexity is also discussed.

The reminder of this paper is organized as follows. Section 2 presents the related work of CNN-based loop filters. Section 3 describes the structure of our proposed RSE-Net as well as the model training strategy. Section 4 gives the experimental results and analysis. Section 5 concludes this paper.

2. Related work

There have been many studies on the CNN-based video enhancement. Most of the solutions are originated from the super-resolution problem. For example, by adding an extra layer to SRCNN (Dong, Deng, Loy, & Tang, 2015), Dong, Loy, He, and Tang (2016) present a 4-layer CNN model, namely ARCNN, to reduce the artifacts of JPEG images. ARCNN inputs the compressed image and outputs the refined one. Experimental results show that more than 1.0 dB PSNR gains can be obtained over JPEG.

For video compression, a very deep CNN network, namely VDSR (Kim, Lee, & Lee, 2016), is employed for loop filtering in video coding. VDSR includes 20 layers and each layer employs a 3×3 kernel size. Results show that VDSR achieves around 4.25% BD-rate gain whereas the number of its parameters is as large as 664, 704 (Kang, Kim, & Lee, 2017). Dai, Liu, and Wu (2017) propose a Variable-filter-size Residue learning CNN (VRCNN) to replace the in-loop filter of H.265/HEVC. VRCNN is simply structured by 4 layers and adopts variable filter size. Results show that VRCNN as a replacement of in-loop filter saves average 4.6% BD-rate under All Intra (AI) configuration with 54, 512 parameters. Park and Kim (2016) directly retrain SRCNN to replace the module SAO of H.265/HEVC. It is shown that average 1.6% and 2.8% BD-rate saving is obtained under Low Delay P (LDP) and Random Access (RA) configuration, respectively. Kang et al. (2017) propose a Multi-modal/Multi-scale

Table 1. Performance comparison: using the model trained for natural videos to enhance the quality of surveillance videos (PSNR/dB).

Test surveillance sequence	HM16.9	VRCNN (Dai et al., 2017)	VDSR (Kim et al., 2016)
1	33.66	33.70	33.63
2	32.90	32.87	32.80
3	31.24	31.30	31.31
4	31.49	31.53	31.51
Average	32.32	32.35	32.31

CNN network (MMS-net). An input image is down-sampled first and restored through the lower scale network. Then the output image is fed into a higher scale network concatenated with the original input image. Although up to 8.5% BD-rate saving is achieved, the number of parameters in MMS-net is 42 times of VRCNN. Jia, Wang, Zhang, Wang, and Ma (2017) employ a Spatial-Temporal Residue Network (STResNet) as a new filter after the in-loop filter of H.265/HEVC. Both current block and co-located block in reference frame are taken into consideration by the network. Results show that average 1.3% BD-rate reduction is obtained in RA configuration. A Residual Highway CNN (RHCNN) is proposed by Zhang et al. (2018) following the in-loop filter of H.265/HEVC. Different models are trained for different frame types. The average bit-rate saving is 5.70%, 5.68% and 4.35% for I, P and B frame, respectively.

The above work is focused on the encoder side. There are also many methods proposed at the decoder side for compressed video enhancement (He et al., 2018; Kuanar, Conly, & Rao, 2018; Song et al., 2018; Wang, Chen, & Chao, 2017; Yang, Chen, et al., 2017; Yang, Xu, Wang, & Li, 2018). For example, Wang et al. (2017) develop a 10-layer network of 3×3 kernel size, namely Deep CNN-based Auto Decoder (DCAD). It equals to boost the coding efficiency averagely 5.0%, 6.4%, 5.3% and 5.5% BD-rate for AI, LDP, LDB and RA configurations, respectively. Yang, Xu, and Wang (2017) adopt a scalable CNN architecture for different frame type, termed DSCNN, to conduct enhancement at the decoder side. Results show that DSCNN achieves better performance than DCAD. Furthermore, Yang et al. (2018) design a multiple frame enhancement on basis of DSCNN and attain higher performance.

It is noted that all the work above is designed and trained for processing natural video. Deep networks with extensive parameters are employed to learn the diverse features of natural video. The surveillance video, however, is quite different from the natural video. Therefore, it is unsuitable to apply the same network models to surveillance video.

To verify the point further, we employ two typical networks, namely VRCNN (Dai et al., 2017) and VDSR (Kim et al., 2016), to enhance the quality of compressed surveillance frames at the decoder side, using their models trained for natural video. The test surveillance frames are first encoded by the reference software of H.265/HEVC, referred to as HM16.9, with QP value 37. At the decoder side, the reconstructed frame is decoded and the PSNR against original frame is calculated. Then, VRCNN and VDSR are utilized to further enhance the quality of the reconstructed frame. The results are shown in Table 1. Obviously that although VRCNN and VDSR are effective for natural video, they fail to process surveillance video.

The above results provide evidence that it is beneficial to develop a specific approach for the surveillance video enhancement. In this paper, we propose a novel CNN structure and develop a robust model training strategy. With the scheme proposed, we train a global model instead of separate models for the background and foreground and achieve significant gains. The proposed scheme can be directly applied at the decoder side without any modification to the encoding algorithms.

3. Proposed network structure and training strategy

3.1. Proposed network structure

The structure of the proposed network is illustrated in Figure 1. It is comprised of two Squeeze-and-Excitation blocks (SE blocks) and a group of REsidual blocks (RE blocks), referred to as Residual Squeeze-and-Excitation Network (RSE-Net). The input frame is first processed by a convolutional layer with kernel size 3×3 for shallow feature extracting, i.e.,

$$S = F_{sfe}(W_0, X), \tag{1}$$

where $S = [s_1, s_2, ..., s_C]$ represents the function of shallow feature extraction, X is the input data and W_0 is the corresponding weight. These shallow features then go through one SE block, several stacking RE blocks and the other SE block, consecutively. At last, a 3×3 convolutional layer is employed to generate residuals. The final output is produced by summing the residuals and the input data.

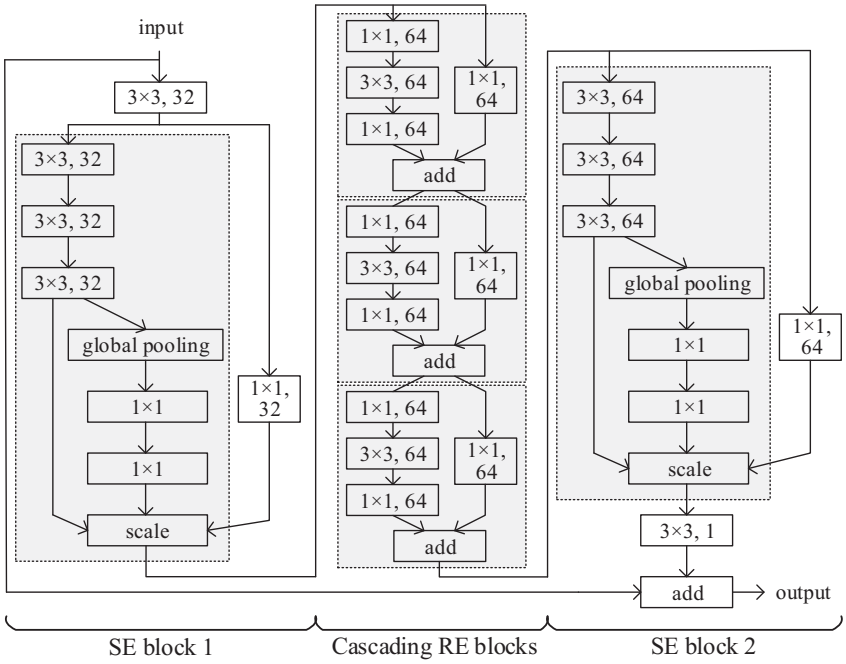

Figure 1. The structure of our proposed network. The dashed boxes indicate the SE block and the RE block, respectively. The parameters in each convolutional layer imply the "size of kernel, number of channels". Additionally, there is a ReLU activation after each convolutional layer, which are omitted here.

It is a challenging task to develop new CNN structures, typically involving the selection of many hyper parameters and layer configurations. Here we introduce the concept of SE block in Squeeze-and-Excitation Network (SENet) (Hu, Shen, & Sun, 2018) at the beginning and the end of RSE-Net to re-calibrate channel features adaptively. Within each SE block, after three consecutive convolutions, a global pooling operation is conducted to compress the features of channels. An $M \times N$ channel is finally squeezed to be represented by a number, which is the average of $M \times N$ values in the channel. Thus, C values corresponding to C channels are obtained in the squeezed stage. Formally, suppose the channel is represented by $O = [o_1, o_2, ..., o_C]$, a statistic $q \in \mathbb{R}^C$ is generated by shrinking O through spatial $M \times N$ dimensions, where the c^{th} element q_c is calculated by

$$q_c = F_{sq}(o_c) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} o_c(i,j). \tag{2}$$

Afterwards, the excitation process is invoked. It works similarly to the gate mechanism in Recurrent Neural Network (RNN). In the excitation stage, two 1×1 convolutional layers are introduced for non-linear mapping. Through learning the relationships across channels, the weight of each channel is obtained. The weight actually implies the degree of importance to each channel. At the end of SE block, each channel is scaled by its corresponding weight. In this manner, the important channels are emphasized and the unimportant ones are suppressed. To meet these criteria, a gating mechanism with Sigmoid activation σ is employed, i.e.,

$$v = F_{ex}(q, W) = \sigma(g(q, W)) = \sigma(W_2 \delta(W_1 q)), \tag{3}$$

where δ refers to the ReLU function, $W_1 \in \mathbb{R}^{C \times C}$ and $W_2 \in \mathbb{R}^{C \times C}$.

The output of the SE block is then obtained by rescaling the transformation output O with the activation v, i.e.,

$$h_c^{se1} = F_{scale}(o_c, v_c, b_c) = v_c \cdot o_c + b_c, \tag{4}$$

where $H^{SE1} = [h_1^{se1}, h_2^{se1}, ..., h_C^{se1}]$ and $F_{scale}(o_c, v_c)$ refers to channel-wise multiplication between the feature map $o_c \in \mathbb{R}^{H \times W}$ and the scalar v_c, respectively.

Considering that the networks may suffer from the performance degradation problem, we introduce an enhanced residual unit to add a bias in (4), described as

$$b_c = \delta(W_3 s_c), \tag{5}$$

where W_3 is the weight of identity branch.

It is noted that in the proposed RSE-Net, the first SE block is not exactly the same as the second one. Considering that the results of the first SE block will be strengthened by the second one, we only assign a small number of parameters to the first SE block. That is, we employ a light-weighted SE block with 32 channels, so called as SE block 1, at the beginning.

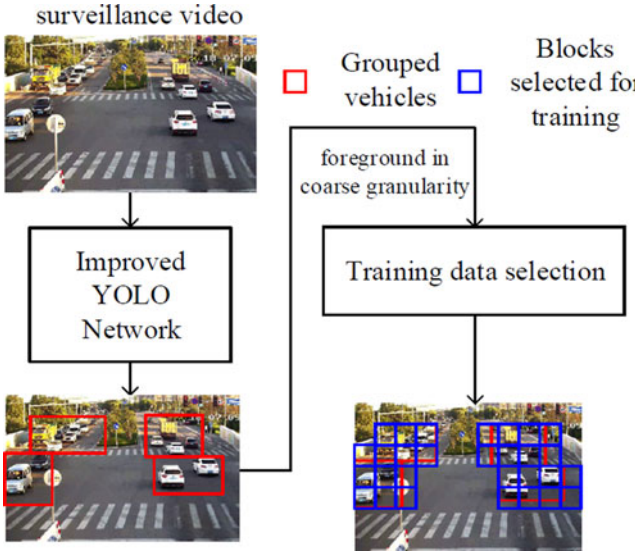

Figure 2. Framework of the proposed training scheme. YOLO is improved to detect the foreground in a coarse granularity. Then the training dataset is established based on the foreground detected.

However, as the network goes deeper, it is prone to distinguish the importance of each channel. Then, a heavy-weighted SE block with 64 channels, namely SE block 2, is applied. In this way, the number of parameters is controlled within a certain range.

Following the first SE block, N RE blocks are cascaded for further non-linear mapping. Each RE block consists of three convolutional layers. As shown in Figure 1, the corresponding kernel size is 1×1, 3×3 and 1×1, respectively. There is a local shortcut within each SE block for identity mapping. After N RE blocks, we can obtain

$$G = F_{RE,n}(F_{RE,n-1}(\cdots F_{RE,1}(H^{SE1})\cdots)), \tag{6}$$

where $F_{RE,n}$ implies the n^{th} RE block, $n \in [1, N]$, and $G = [g_1, g_2, ..., g_C]$.

After the stacking RE blocks, we invoke SE block 2 to re-calibrate the channels again and obtain the output

$$h_c^{se2} = F_{scale}(g_c, v_c, b_c) = g_c \cdot v_c + b_c. \tag{7}$$

Notice that the parameters v and b are different in SE block 1 and SE block 2. Finally, the global residuals $H^{SE2} = [h_1^{se2}, h_2^{se2}, ..., h_C^{se2}]$ are added to the global shortcut to generate the output $\tilde{Y} = X + H^{SE2}$.

3.2. Proposed model training scheme

Besides the network structure, it is also critical to design an effective model training strategy, so as to explore the best potential of the network. Taking into account the characteristics of surveillance video, we propose a robust model training scheme, which trains a global model for both the foreground and background of surveillance video.

Since the surveillance frame is decomposed into background and foreground, it is intuitive to train separate models for them. Ideally, within each picture, the foreground and background scenarios may employ their own models for enhancement. However, we find that it is time consuming to train such a model for the background. On one hand, the content of background remains unchanged in a sequence. Hence, we can only derive limited training data from one sequence. On the other hand, although the foreground is mostly vehicles, the background can vary from sequence to sequence. Consequently, it requires a fairly large number of videos to collect data for sufficient training.

To address the problems above, in this work, we propose to train a global model for both foreground and background processing. The framework of our proposed approach is illustrated in Figure 2. YOLO (Redmon & Farhadi, 2018), which is a state-of-the-art real-time object detection system, is introduced in our framework for vehicle detection. Rather than detect each vehicle accurately, the algorithm of YOLO is improved here to detect vehicles in a coarse granularity. It indicates that the neighboring vehicles are recognized and merged as a group of vehicles. All the content left in an image is regarded as background. In this manner, each $N \times N$ block is checked whether it is belonged to background or foreground. The determination principle is that as long as there exists foreground in the block, the block is selected into training dataset. As illustrated in Figure 2, the blocks in blue

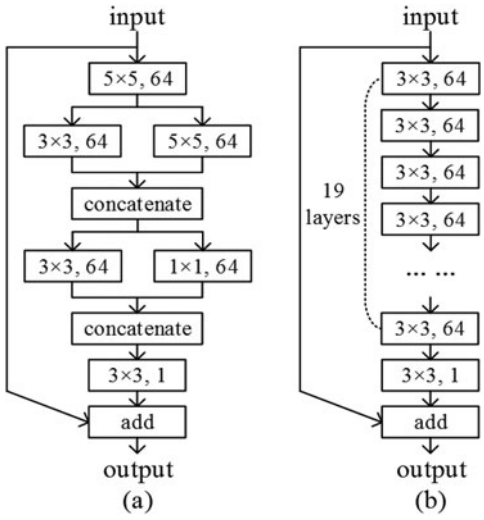

Figure 3. The structure of VRCNN (a) and VDSR (b).

squares are all selected as training data, although some blocks contain no vehicle.

It is worth noting that we target to train a global model to deal with both the foreground and background. Hence the training database should include content of both. In order to produce a training set with rich content, we employ YOLO to help remove the redundant background. Note that we do not directly employ the original YOLO to recognize each vehicle accurately. Instead, we modify it to identify a group of neighboring vehicles. The training set is then established on basis of the grouped vehicles.

As such, we achieve a balance between the vehicle content and the background information. The proposed scheme can successfully pick out most of the vehicles in an image, and furthermore add a certain amount of background to elevate the generalization capability of our CNN model.

4. Experimental results

In this section, we present the experimental results to validate the effectiveness of our approach. The experiments are conducted on the reconstructed frame at the decoder side for further quality improvement. We compare our approach with two typical CNNs used for video quality enhancement, i.e., a shallow network VRCNN (Dai et al., 2017) and a deep network VDSR (Kim et al., 2016), respectively. We also compare the computational complexity and subjective quality with the two networks. Structures of VRCNN and VDSR are illustrated in Figure 3. Note that three RE blocks are employed in our RSE-Net for all the experiments of section 4.2.

4.1. Training and settings

In our experiments, seven surveillance sequences in different scenarios are used for training. We pick one frame per 50 frames and a total of 200 frames are obtained for each sequence. Then 1400 frames are collected. Each frame is split into numerous 64×64 patches, with the stride of 64, considering both the training time and storage cost. Our test set contains four sequences with different scenarios. The snapshots of training and test sequences are illustrated in Figures 4 and 5, respectively. We also visualize part of our training set in Figure 6.

The results are tested under default AI, LDP and RA configurations of HM16.9. Two typical QP values, i.e. {37, 42}, are used to encode the video sequences. The first 50 frames of each sequence are employed for test. The PSNR and SSIM of luma component are evaluated to measure the performance.

The models are trained by minimizing the difference between the reconstructed frame and the original frame. Suppose the input picture is denoted as $X_i, i \in \{1, 2, ..., N\}$, and the ground truth is Y_i. The end-to-end mapping function $F(\cdot)$ with the parameters θ is learned by minimizing the loss function

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|F(X_i; \theta) - Y_i\|_2^2. \quad (8)$$

The mini-batch gradient descent is employed in the training, with the mini-batch size set to 64, momentum parameter to 0.9, and weight decay to 10^{-4}. For weight initialization, we use the same method as He, Zhang, Ren, and Sun (2016), which is shown to be suitable for networks utilizing ReLU activation. The initial learning rate is set to 0.003 and then decreased to half every 4 epochs. We also adopt the adjustable gradient clipping to boost the convergence rate while suppressing exploding gradients. All of our experiments are implemented on Caffe (Jia et al., 2014) platform. Training a RSE-Net with three residual blocks roughly takes 4 hours on the NVIDIA GeForce GTX 1080Ti. Our code and model are available on our Github website (https://github.com/IVC-Projects/svideo).

4.2. Performance comparison

Our approach is integrated into H.265/HEVC decoder to test the coding efficiency. As we know, H.265/HEVC has built the baseline in-loop filter including Deblocking and SAO to filter the raw reconstructed frames. Our proposed scheme is applied after them as an out-loop filter to further boost the quality of reconstructed frames.

Figure 4. The snapshots of our training sequences.

Figure 5. The snapshots of our test sequences which are numbered from 1 to 4 from left to right.

Figure 6. The visualization on part of our training set, left: foreground training set; middle: background training set; right: frame-level training set.

4.2.1. Compare with different network structures

We compare the overall performance of our RSE-Net with VRCNN and VDSR under the same test conditions. We can see from Table 2 that at QP value 37, average 0.40 dB, 0.22 dB and 0.24 dB PSNR gains are obtained over H.265/HEVC baseline in AI, LDP and RA configurations, respectively. When QP value is set to 42, the gains increase to 0.47 dB, 0.28 dB and 0.32 dB.

It also can be seen that {0.16, 0.15} dB, {-0.01, 0.05} dB and {0.07, 0.12} dB PSNR gains are derived by our network compared to VRCNN at QP values {37, 42} in AI, LDP and RA configurations. As VDSR contains 20 convolutional layers, its learning capability is higher than VRCNN. Using much less parameters, our RSE-Net performs better than VDSR. There are {0.04, 0.14} dB, {0.05, 0.04} dB and {0.02, 0.03} dB PSNR gains compared to VDSR under different configurations.

4.2.2. Model training scheme comparison

We show the results of our proposed training method in Table 3. The experiment is conducted in AI configuration at QP value 37. The first frame of each test

Table 2. Performance comparison of VRCNN, VDSR and RSE-Net in different configurations.

		$QP = 37$				$QP = 42$			
Cfg	Test sequence	HM16.9	VRCNN	VDSR	RSE-Net	HM16.9	VRCNN	VDSR	RSE-Net
AI	1	33.55	33.81	33.94	34.02	29.94	30.28	30.30	30.44
	2	32.70	32.85	32.98	33.05	29.00	29.26	29.27	29.41
	3	31.64	31.96	32.08	32.09	28.21	28.59	28.60	28.79
	4	31.80	32.03	32.13	32.13	28.45	28.74	28.75	28.85
	Avg. PSNR	32.42	32.66	32.78	32.82	28.90	29.22	29.23	29.37
	Avg. SSIM	0.9223	0.9251	0.9267	0.9268	0.8697	0.8761	0.8755	0.8773
LDP	1	31.51	31.78	31.72	31.79	28.21	28.49	28.50	28.56
	2	31.62	31.80	31.74	31.79	28.33	28.52	28.53	28.55
	3	30.01	30.27	30.20	30.25	27.09	27.33	27.38	27.42
	4	30.29	30.51	30.45	30.48	27.28	27.50	27.45	27.52
	Avg. PSNR	30.86	31.09	31.03	31.08	27.73	27.96	27.97	28.01
	Avg. SSIM	0.9107	0.9144	0.9135	0.9135	0.8575	0.8635	0.8643	0.8645
RA	1	31.41	31.61	31.65	31.67	28.71	28.93	29.00	29.04
	2	32.34	32.49	32.55	32.58	29.04	29.22	29.32	29.34
	3	30.19	30.37	30.42	30.45	27.58	27.80	27.89	27.94
	4	30.57	30.74	30.78	30.79	27.80	27.97	28.05	28.06
	Avg. PSNR	31.13	31.30	31.35	31.37	28.28	28.48	28.57	28.60
	Avg. SSIM	0.9188	0.9212	0.9220	0.9225	0.8705	0.8745	0.8769	0.8771

Table 3. Performance comparison among different model training methods (PSNR/dB).

Test sequence	HM16.9	Frame-level training	Separate-fore./back. training	Proposed training
1	33.66	34.01	34.04	34.12
2	32.90	33.04	33.04	33.18
3	31.24	31.51	31.53	31.59
4	31.49	31.72	31.75	31.80
Avg.	32.32	32.57	32.59	32.67

sequence is tested for comparison. We can see that our proposed scheme obtains around 0.35 dB PSNR over the anchor H.265/HEVC.

To validate the effectiveness of our training strategy, we retrain a global model using the whole frames, so called as the frame-level training. It can be seen from Table 3 that our model achieves average 0.10 dB PSNR over the frame-level model.

Besides, we also train different models for the foreground and background and then apply them to corresponding scenarios. Our model selection principle is that as long as there exist vehicles in a block, the foreground model is applied to filter this block. Otherwise, the background model is applied. Table 3 compares the results of using the separate foreground/background model against the proposed global model. We can observe that the proposed global model performs better.

4.2.3. Foreground performance comparison

Since the traffic incidents generally occur in foreground, we care more about the quality of foreground in practice. It is more significant to obtain a high-quality foreground. More specifically, we compare the performance of the foreground among different approaches. We can see from Table 4 that the proposed scheme can achieve {0.38, 0.36} dB, {0.51, 0.36} dB and {0.25, 0.30} dB gains over H.265/HEVC baseline at QP values {37, 42}. Our scheme also performs better than VRCNN and VDSR. Compared to VRCNN, {0.19, 0.09} dB, {0.39, 0.21} dB and {0.13, 0.14} dB PSNR gains are obtained. Although VDSR is deeper than our RSE-Net, RSE-Net performs better. Average {0.28, 0.12} dB, {0.44, 0.22} dB and {0.09, 0.18} dB PSNR increments are achieved at QP values {37, 42}.

4.3. Comparison on computational complexity

Apart from the performance improvement, computational complexity is also a critical evaluation criteria. We compare PSNR versus the number of parameters in our and the other two methods. For fair comparison, all of the networks are trained with the proposed training scheme. The first frame of each test sequence is tested at $QP = 37$ in AI configuration. The average PSNR value is demonstrated for comparison.

As observed in Figure 7, VRCNN has the least parameters and its PSNR is much lower than that of RSE-net and VDSR. VDSR performs better than VRCNN but its number of parameters is as large as 664, 704. We try different layouts of our RSE-Net by increasing the number of layers, i.e., the number of the cascading residual blocks is changed to 3, 6, 9 and 12, respectively. It can be seen that RSE_6, RSE_9 and RSE_12 surpasses VDSR with less parameters. As the number of layers increases from RSE_3 to RSE_9,

Table 4. Foreground performance comparison among different networks.

		QP = 37				*QP* = 42			
Case	Test sequence	HM16.9	VRCNN	VDSR	RSE-Net	HM16.9	VRCNN	VDSR	RSE-Net
AI	1	36.16	36.24	36.22	36.46	34.05	34.25	34.26	34.29
	2	36.67	36.78	36.52	37.03	34.44	34.72	34.68	34.81
	3	34.65	34.94	34.84	35.08	32.53	32.83	32.77	32.95
	4	34.56	34.84	34.85	35.00	32.42	32.71	32.68	32.84
	Avg. PSNR	35.51	35.70	35.61	35.89	33.36	33.63	33.60	33.72
	Avg. SSIM	0.9304	0.9348	0.9370	0.9379	0.8785	0.8879	0.8872	0.8902
LDP	1	34.50	34.61	34.59	34.98	32.93	33.12	33.08	33.28
	2	35.05	35.12	35.07	35.56	33.64	33.81	33.80	34.05
	3	33.73	33.89	33.83	34.23	32.19	32.32	32.34	32.56
	4	34.03	34.17	34.12	34.57	32.35	32.47	32.44	32.68
	Avg. PSNR	34.33	34.45	34.40	34.84	32.78	32.93	32.92	33.14
	Avg. SSIM	0.9043	0.9121	0.9092	0.9132	0.8503	0.8624	0.8646	0.8656
RA	1	34.20	34.30	34.34	34.42	33.01	33.14	33.08	33.26
	2	35.15	35.27	35.30	35.42	33.72	33.90	33.85	34.06
	3	33.79	33.91	33.96	34.03	32.39	32.55	32.53	32.70
	4	34.19	34.31	34.37	34.45	32.50	32.67	32.65	32.81
	Avg. PSNR	34.33	34.45	34.49	34.58	32.91	33.07	33.03	33.21
	Avg. SSIM	0.9063	0.9118	0.9129	0.9138	0.8607	0.8692	0.8720	0.8738

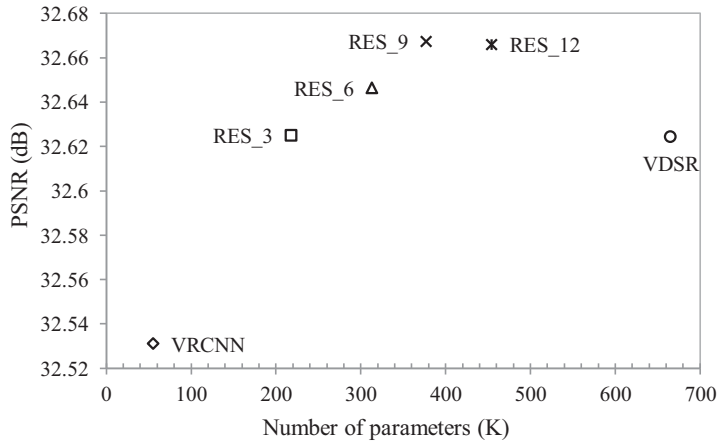

Figure 7. A visual illustration of the coding gain versus the number of parameters in different network structures.

the performance increases linearly. However, when the number of layers reaches 12, i.e., RSE_12, it turns out that the gain is decreased slightly. In this work, we choose RSE_3 for implementation. RSE_3 reduces the parameters to 217, 628, which is only 32.7% of the parameters in VDSR, while achieving comparable performance.

4.4. Comparison on visual quality

In addition to the objective quality, we also show some visual comparisons in Figure 8. The test pictures are all compressed with QP value 37 in LDP configuration, which is widely accepted in surveillance video compression. Due to the limitation of surveillance camera and the environment issue, artifacts are introduced in the compressed frames, as illustrated in the left column. The in-loop filter of H.265/HEVC only reduces certain compression artifacts. Our proposed out-loop filter, successfully removes the major artifacts and produces frames of high subjective quality. In Figure 8a, the left side of the car is seriously blurred. The proposed approach corrects the blur and smooths the mirror and door. In Figure 8b, the plate number is very fuzzy and the edge of the car is corrupted by noise. Although the original H.265/HEVC in-loop filters improve the visual quality, it still contains thorns and blocks. Our CNN-based approach clear them and the enhanced frames look pleasing. It is the similar case in Figure 8c and d.

5. Conclusion

In this paper, we develop a CNN-based approach to enhance the quality of compressed surveillance video. We design RSE-Net to extract the extensive characteristics of surveillance video. RSE-Net is structured by residual network and squeeze-and-excitation network,

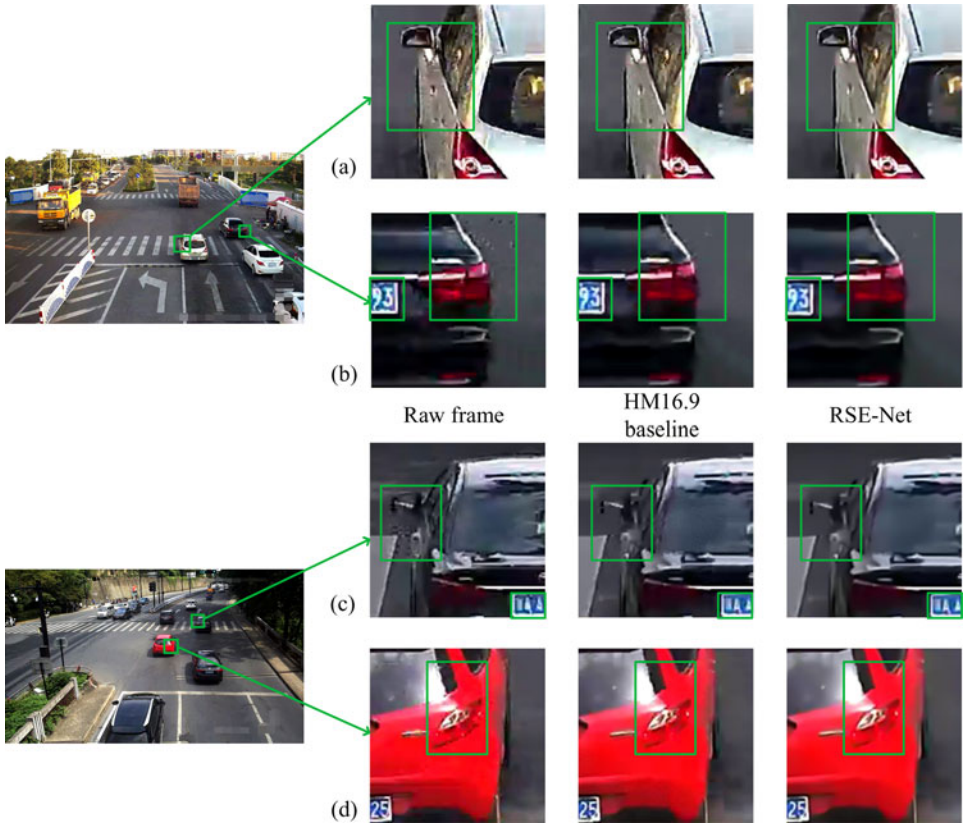

Figure 8. Visual comparisons between H.265/HEVC anchor and the proposed RES-Net.

taking into consideration both the channel-wise correlation among pixels and the relationships across channels. Meanwhile, considering the characteristics of surveillance video, a robust model training strategy is developed. Finally, we train a global model to handle both the foreground and background of surveillance video. Experimental results prove the effectiveness of our network and model training strategy. The coding efficiency of our approach outperforms two state-of-the-art approaches with only a small number of parameters. Besides, the subjective quality of compressed surveillance video is also significantly improved.

Currently, we utilize the spatial information within a frame for enhancement. Considering that surveillance videos are highly correlated in the temporal domain, it is potential to develop a spatio-temporal approach to achieve better performance, which will be our work in the future. Besides, it is also valuable to investigate how to reduce the computational complexity in the enhancement procedure.

Funding

The work was partially supported by the National Key R&D Program of China under Grant 2017YFB1002803.

References

Chen, F., Liu, D., Li, H., & Wu, F. (2017). *Fast encoding of surveillance videos based on HEVC*. 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, pp. 1–4.

Dai, Y., Liu, D., & Wu, F. (2017). *A convolutional neural network approach for post-processing in HEVC intra coding*. The 24th International Conference on MultiMedia Modeling (MMM 2017), Reykjavik, Iceland, pp. 28–39.

Dong, C., Deng, Y., Loy, C., & Tang, X. (2015). *Compression artifacts reduction by a deep convolutional network*. 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, pp. 576–584.

Dong, C., Loy, C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. 2016. doi:10.1109/TPAMI.2015.2439281

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Identity mappings in deep residual networks*. European Conference on Computer Vision, Amsterdam, The Netherlands, pp. 630–645.

He, X., Hu, Q., Han, X., Zhang, X., Zhang, C., & Lin, W. (2018). *Enhancing HEVC compressed videos with a partition-masked convolutional neural network*. The 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, pp. 216–220.

Hu, J., Shen, L., & Sun, G. (2018). *Squeeze-and-excitation networks*. Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 7132–7141.

ITU-T. (2003). Infrastructure of audiovisual services – Coding of moving video: Advanced video coding for generic audiovisual services. Audiovisual and Multimedia System. Standard No. H.264:2003.

ITU-T. (2013). Infrastructure of audiovisual services – Coding of moving video: High efficiency video coding. Audiovisual and Multimedia System, Standard No. H.265:2013.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). *Caffe: Convolutional architecture for fast feature embedding*. Proceedings of the 22nd ACM international conference on Multimedia, November 3–7, 2014, Orlando, FL, pp. 675–678.

Jia, C., Wang, S., Zhang, X., Wang, S., & Ma, S. (2017). *Spatial-temporal residue network based in-loop filter for video coding*. 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, pp. 1–4.

Kang, J., Kim, S., & Lee, K. (2017). Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec. 2017 IEEE International Conference on Image Processing (ICIP 2017), Beijing, China, pp. 26–30.

Kim, J., Lee, J., & Lee, K. (2016). *Accurate image super-resolution using very deep convolutional networks*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, pp. 1646–1654.

Kuanar, S., Conly, C., & Rao, K. (2018). *Deep learning based HEVC in-loop filtering for decoder quality enhancement*. Picture Coding Symposium (PCS), San Francisco, CA, pp. 164–168.

Ma, C., Liu, D., Peng, X., & Wu, F. (2017). *Surveillance video coding with vehicle library*. 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 270–274.

Park, W., & Kim, M. (2016). CNN-based in-loop filtering for coding efficiency improvement. IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP 2016), Bordeaux, France, pp. 1–5.

Pushkar, G., & Bharadwaj, A. (2014). Skip decision and reference frame selection for low-complexity H.264/AVC surveillance video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(7), 1156–1169, 2014.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. https://arxiv.org/abs/1804.02767.

Song, X., Yao, J., Zhou, L., Wang, L., Wu, X., Xie, D., & Pu, S. (2018). *A practical convolutional neural network as loop filter for intra frame*. The 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, pp. 1133–1137.

Wang, T., Chen, M., & Chao, H. (2017). A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC. 2017 Data Compression Conference (DCC 2017), Snowbird, UT, pp. 410–419.

Xiao, J., Hu, R., Liao, L., Chen, Y., Wang, Z., & Xiong, Z. (2016). Knowledge-based coding of objects for multi-source surveillance video data. *IEEE Transactions on Multimedia*, *18*(9), 1691–1706. 2016. doi:10.1109/TMM.2016.2581590

Yang, K., Chen, F., Liu, D., Chen, Z., & Li, W. (2017). *Surveillance video coding with dynamic textural background detection*. 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 2736–2740.

Yang, R., Xu, M., & Wang, Z. (2017). *Decoder-side HEVC quality enhancement with scalable convolutional neural network*. IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, pp. 817–822.

Yang, R., Xu, M., Wang, Z., & Li, T. (2018). *Multi-frame quality enhancement for compressed video*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT.

Zhang, Y., Shen, T., Ji, X., Zhang, Y., Xiong, R., & Dai, Q. (2018). Residual highway convolutional neural networks for in-loop filtering in HEVC. *IEEE Transactions on Image Processing*, *27*(8), 3827–3841. doi:10.1109/TIP.2018.2815841

Zhang, X., Tian, Y., Huang, T., Dong, S., & Gao, W. (2014). Optimizing the hierarchical prediction and coding in HEVC for surveillance and conference videos with background modeling. *IEEE Transactions on Image Processing*, *23*(10), 4511–4526. doi:10.1109/TIP.2014.2352036