EVSRNet：Efficient Video Super-Resolution with Neural Architecture Search

Shaoli Liu, Chengjian Zheng, Kaidi Lu, Si Gao, Ning Wang, Bofei Wang, Diankai Zhang, Xiaofeng Zhang, Tianyu Xu
State Key Laboratory of Mobile Network and Mobile Multimedia Technology, ZTE Corporation, Shenzhen, China
{liu.shaoli, zheng.chengjian, lu.kaidi, gao.si, wangning, wang.bofei, zhang.diankai, zhang.xiaofeng18, xu.tianyu1}@zte.com.cn

Abstract

With the development of convolutional neural networks (CNN), the super-resolution results of CNN-based method have far surpassed traditional method. In particular, the CNN-based single image super-resolution method has achieved excellent results. Video sequences contain more abundant information compare with image, but there are few video super-resolution methods that can be applied to mobile devices due to the requirement of heavy computation, which limits the application of video super-resolution. In this work, we propose the Efficient Video Super-Resolution Network (EVSRNet) with neural architecture search for real-time video super-resolution. Extensive experiments show that our method achieves a good balance between quality and efficiency. Finally, we achieve a competitive result of 7.36 where the PSNR is 27.85 dB and the inference time is 11.3 ms/f on the target snapdragon 865 SoC, resulting in a 2nd place in the Mobile AI（MAI）2021 real-time video super-resolution challenge. It is noteworthy that, our method is the fastest and significantly outperforms other competitors by large margins.

1. Introduction

In the past few years, CNN-based methods have achieved state-of-the-art results in various computer vision tasks. But these methods can only run on high-performance servers because of the massive parameters and high computational cost. At the same time, the intense demand for artificial intelligence applications in mobile devices has prompted academia and industry to study how to deploy CNN-based models on mobile devices. On the one hand, after continuous development, mobile systems on a chip (SoC) have achieved excellent hardware acceleration performance, which is comparable to desktop PCs. As a major mobile deep learning library, TensorFlow Lite (TFLite) [1] provides convenience for model deploying on mobile devices. On the other hand, the design of lightweight network with efficient operators(e.g. group convolution, pointwise convolution) can greatly improve the efficiency of the model. Model compression is another popular optimization approach to accelerating model, which includes network quantization, network pruning, low-rank approximation, etc. These technologies described above have greatly promoted the deployment of CNN-based methods on mobile devices.

The super-resolution plays an important role in many domains, such as medical imaging, astronomical images, face recognition in surveillance videos, etc.. Like other computer vision task, CNN-based method has dramatically boosted the performance of single image super-resolution by using carefully designed neural architectures. The successes encourage the community to further attempt CNN on the more challenging video super-resolution (VSR) problem. To study the temporal redundancy among neighboring frames, various VSR approaches have been proposed. But most VSR approaches are generally complex due to the use of network structures such as frame alignment, optical flow and 3-dimensional convolution. With the the rapid development of video business, both the research and industrial communities have attached much attention in real-time VSR, which aims at producing a high-resolution (HR) video from the corresponding low-resolution (LR) counterparts in real-time.

To promote the development of VSR technology, the MAI workshop 2021 holds a Real-Time Video Super-Resolution Challenge [2]. The aim of the competition is to super-resolve an input video to an output video in the spatial domain with a upsampling factor x4. Competitors are asked to balance the quality and the efficiency of the VSR model on the target mobile platform. The network efficiency is tested by using the professional AI Benchmark application [3,4]. It is an Android application designed to check the performance and the memory limitations associated with running AI and deep learning algorithms on mobile platforms, which uses the TFLite library as a backend for running all embedded deep learning models.

In this paper, We present Efficient Video Super Resolution Network (EVSRNet) for tackling the real-time VSR challenge. The experimental results show that the proposed EVSRNet is highly computational efficiency and achieves good results on both the validation and the test set of REDS [5].

2. Related work

SRCNN [6] is the first approach that uses CNN for single image super-resolution. SRCNN [6] further showed that traditional sparse-coding-based super-resolution methods can also be viewed as a deep convolutional network. It inspired researchers focus on deep learning based super-resolution. In order to reduce the memory and computational requirements, ESPCN [7] proposed a sub-pixel convolution layer to perform the feature extraction stages in the LR space. EDSR [8] used the residual-based learning mechanism of ResNet [9] without batch normalization layer due to the batch normalization layer can cause the loss of high frequency information. To make full use of the information from the original LR images and exploiting the inherent feature correlations in intermediate layers, SAN [10] introduced a non-locally enhanced residual group structure for further capture the long-distance spatial contextual information and a second-order channel attention module for better feature correlation learning.

Video super-resolution is developed from image super-resolution, existing approaches can be mainly divided into two categories. The first class of method is based on explicit motion compensation. Kappeler et at. [11] proposed to warp all neighboring frames to the reference frame based on the offline estimated optical flow. VESCPN [12] is the first end-to-end VSR method by jointly training optical flow estimation and spatial-temporal networks. MMCNN [13] cascaded an optical flow network and an image-reconstruction network to fully exploit spatio-temporal correlations between adjacent LR frames and reveal more realistic details. But these works are not suitable for real-time VSR since the computation of optical flow introduces heavy computational load. The second class of method is based on implicit motion compensation which explores advanced temporal modeling frameworks. Temporal modeling plays an important role in VSR. EDVR [14] proposed a video restoration framework including effective alignment module and fusion module with enhanced deformable convolutions. RRN [15] exploited previous frame and current frame as hidden state input, and incorporated identity mapping in hidden state to preserve the texture details through network layers. These VSR approaches achieve good reconstruction results and perform quickly on PC, But they still cannot be deploy on mobile devices due to the high computational cost. Ma et al. [16] speeded up the super-resolution network significantly through binarize the convolutional filters in residual block, but the accuracy of the network has dropped significantly.

Despite the great success of CNN-based methods, most of them are not suitable for mobile devices. In this work, we adopt parts of the ideas presented above to deal with the challenging real-time VSR task.

3. Approach

The real-time VSR challenge requires the input tensor of proposed model should accept 10 subsequent video frames and have a size of [1 x 180 x 320 x 30], where the first dimension is the batch size, the second and third dimensions are the height and width of the input frames from the REDS dataset[5], and the last dimension is the number of channels (3 color channels x 10 frames). The size of the output tensor should be [1 x 720 x 1280 x 30]. Due to this limitation, the proposed method will not be able to make full use of the inter-frame reference information. On the other hand, the final score of this challenge is shown in formula (1), which is calculated based on two metrics - the quality of the reconstructed results and the runtime of the model on the target snapdragon 865 SoC.

$$Score(PSNR,\ runtime) = \frac{2^{2*(PSNR-27.00)}}{runtime} \qquad (1)$$

Therefore, in addition to the quality of the restored image, we also need to pay attention to the efficiency of network operation. In order to solve this challenging problem, we have done extensive experiments. In this section, We introduce the establishment process of our proposed EVSRNet. We first analyze recently published super-resolution approaches and then describe our EVSRNet with neural architecture search.

3.1. Find super-resolution baseline

Due to time constraints, we initially wanted to design a network structure suitable for real-time VSR by analyzing the existing super-resolution approaches. On the one hand, we experiment with single-image super-resolution networks such as RFDN [17], IMDN [18], and EDSR [8], but the forward inference efficiency of these methods is too slow and the final score is not ideal. On the other hand, we experiment with excellent VSR methods. Due to the limitation of the challenge on the input format, we remove the backward reference input information of EDVR [14]. We fine-tune the adjusted EDVR [14] and the original RRN [15], test them on the target platform, the efficiency of these methods cannot meet our needs, RRN-5L is relatively optimal. By analyzing the basic units of RRN network, we decide to use network architecture search(NAS) to determine the optimal solution.

3.2. Design search space

As mentioned above, we choose the RRN [15] as our baseline. There are two hyper parameters in RRN, specificity, the number of channels and the number of residual modules. In order to reduce the search space of NAS, we use residual block as a basic building block to construct our model. As shown in Figure 1 and Table 1(cite

Table 1. Ablation on the number of residual block.

Number	2	3	4	5	6	7	8	9	10
PSNR(dB)	26.97	27.09	27.20	27.38	27.42	27.53	27.65	27.67	27.69

from reference [15]), two experiments are carried out to further reduce the search space.

As shown in Table 1, the PSNR growth slow after the number of residual block is greater than 8, so the best search space of residual block number is under 8.

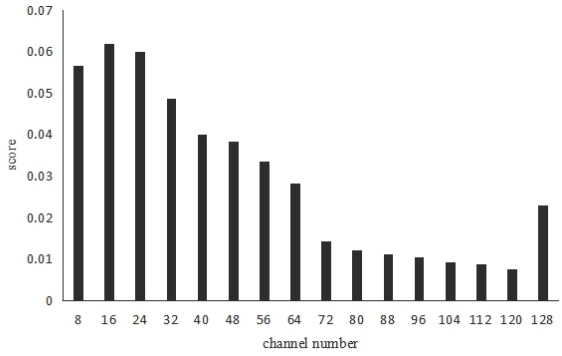


Figure 1. Ablation on the channel number of RRN.

In Figure 1, we set the number of residual modules as 5, the score is calculated according to formula (1). According to Figure 1, the best search space of channel number is under 32.

3.3. Network Architecture Search

In order to pursue a trade-off between the restoration capacity and the parameters of model, we use FGNAS [19] to search for an operation in each channel.

Table 2. The search space of operations.

Factor	Search Space
Convolution types	Normal
Convolution kernel sizes	1,3
Activation functions	ReLU
The number of channels	0,1,..,32
The number of residual block	0,1,...,8

Like most super-resolution network, RRN [15] can be divided into three sub-procedures: feature extraction, nonlinear mapping and restoration. Since most of the deep learning approaches concentrate on the feature extraction and nonlinear mapping, we design our search space on the two parts and fix the restoration part(composed of sub-pixel convolution layer [7] and resize operation). In specific, Table 2 illustrates the search space of operations.

The objective function of FGNAS [19] compose of two terms, one is the task-specific loss and the other is a regularizer penalizing such as parameters, FLOPs and latency. In our work, we use FLOPs of network as the regularizer penalizing, because it's easy to calculate. In specific, our objective function is shown in formula (2).

$$\min_{\theta,\psi} L(\theta,\psi)+\lambda\cdot R(\psi) \tag{2}$$

Where θ and Ψ are learnable parameters in the neural networks and the gating functions g(·), respectively, and λ is the hyper-parameter balancing the two terms. The detailed information about the two terms can refer to [19]. After NAS, we obtain four models by changing the λ value. These models are similar in network architecture. An overview of network architecture is shown in Figure 2. As shown in the figure, feature extraction includes multiple basic modules (BM) which are residual modules without batch normalization layer due to the batch normalization layer can cause the loss of high frequency information. The high-resolution residual map is obtained by adopting sub-pixel convolution layer [7] .

3.4. Model evaluation based on incomplete training

As mentioned above, we finally obtain four similar models. Due to time and computing resource constraints, only 400 epochs were trained for the above four models. The results are shown in Table 3.

Table 3. Training results of different models.

Model	n_c	n_b	PSNR(dB)	Runtime(ms/f)	Score
1	8	5	27.4203	17.3	0.1035
2	16	4	27.6647	25.6	0.0982
3	16	5	27.7138	27.1	0.0993
4	24	4	27.8082	31.8	0.0964

In Table 3, n_c represents the number of channels of the residual module, n_b represents the number of residual modules. The score is calculated according to formula (1). According to the final score, we choose model 1 with n_c=8, n_b=5 as our challenge network.

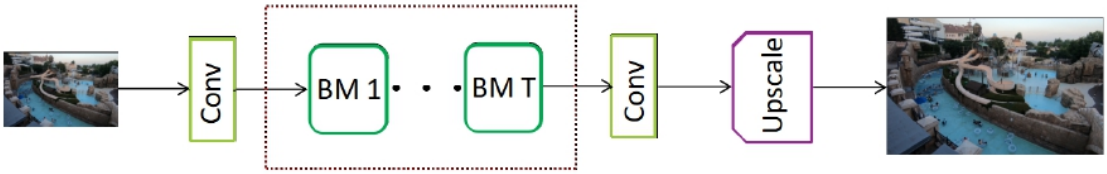


Figure 2. EVSRNet architecture overview.

3.5. Remove the resize operation

By means of analyzing the architecture of the network and conducting a series of experiments, we find that data-related operations are not conducive to GPU acceleration. Data switching between CPU and GPU seriously affects the efficiency of network forward inference. Therefore, we try to remove the resize operation and use the sub-pixel convolution layer [7] to directly output the restored image. We test the final model on the REDS val set. This change has minimal impact on PSNR, but can greatly improves the efficiency. On our mobile test device, the inference time has dropped from 17.3ms/f to 12.1ms/f.

4. Experiments

4.1. Dataset

The real-time VSR challenge uses the REDS dataset [5], which have a large diversity of contents and dynamic scenes. It is widely used in video super-resolution and video denoising tasks. REDS dataset consists of 300 video sequences containing 100 frames of 720 × 1280 resolution. To generate the LR data, the videos are bicubic downsampled by scale 4. In this challenge, the dataset is divided into 240 sequences for training, 30 sequences for validation, and 30 sequences for testing. We use REDS 120fps as an extra dataset, which is only differs from REDS in sampling frequency, to improve the image restoration result. The corresponding LR frames of REDS 120fps are generated by applying bicubic interpolation at scale 4.

4.2. Implementation Details

We adopt both REDS [5] and REDS 120fps [5] as the training set. In order to accelerate the training of EVSRNet, we randomly crop HR patches of size 512×512 from the HR images and LR patches of size 128×128 from LR images. The learning rate is initially set to 2e-4 and later down-scaled by a factor of 0.5 till 400 epoch. The training step completes after 1000 epochs. Our models are supervised by pixel-wise L1 loss function with Adam [20] optimizer by setting $\beta 1 = 0.9$, $\beta 2 = 0.999$ and weight decay of 5e-4. We pretrain the model on the REDS and then fine-tune on the REDS 120fps with batch size 4. Model implementation, training and exporting are conducted using Python 3.7.7 and Pytorch 1.5.0.

4.3. Comparisons with bicubic method

Figure 3 depicts the $\times 4$ VSR results of the REDS val dataset. As shown on the right side of Figure 3. The first row is the result of bicubic, the second row is the result of ours, The third row is the ground truth. It is obvious that our method produces sharper edges and finer details than bicubic method. At the same time, our VSR results is more close to the ground truth.

4.4. Results of the real-time VSR challenge

In this challenge, REDS test-set is used to evaluate the quality of the reconstructed results, and the runtime of the model on the actual snapdragon 865 SoC is tested by AI benchmark. The top three results are shown in Table 4, we achieves a competitive result of 7.36 on test-challenge set where the PSNR is 27.85 dB and the inference time is 11.3 ms/f, resulting in a 2nd place in the Mobile AI 2021 real-time VSR challenge[21]. It is noteworthy that, our method is the fastest.

Table 4. MAI challenge results. The best results are highlighted.

Team	PSNR(dB)	SSIM	Runtime(ms/f)	Score
1	**28.33**	**0.81**	19.9	**8.13**
2(ours)	27.85	0.8	**11.3**	7.36
3	27.99	0.8	18.0	5.61

4.5. Runtime of our method on several other mobile SoCs

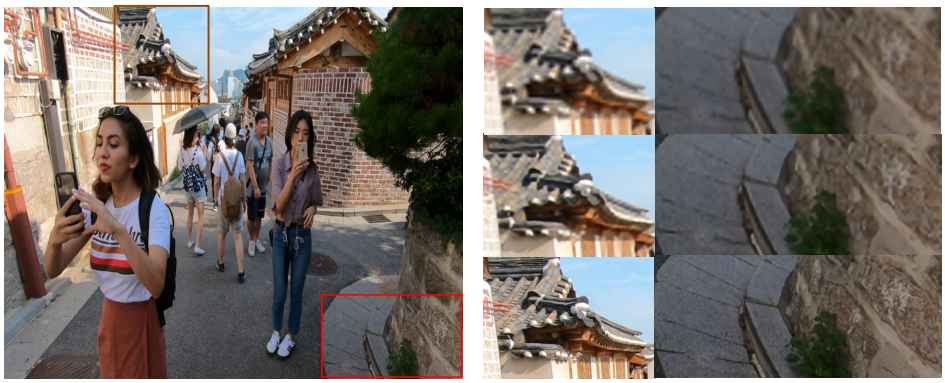
In order to evaluate the performance of our proposed method on different hardware platforms, we use AI Benchmark app (FP16 mode) [22] to test the runtime and report CPU(single thread), GPU and NPU or NNAPI results on several mobile devices. As shown in Table 5, in the GPU(FP16 + TFLite GPU delegate)mode, our model can run in real-time on most common mobile SoCs .

Table 5. Runtime on different mobile SoCs (ms/f).

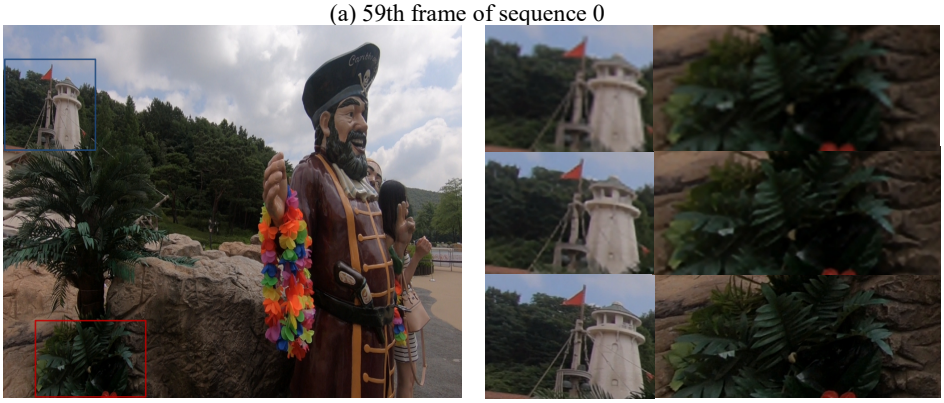
Smartphone	SoC	GPU	CPU	NNAPI
ZTE Nubia Red Magic 5G	Snapdragon 865	12.1	37.3	73.2
Samsung Galaxy S10	Snapdragon 855	14.8	44.0	74.0
Redmi K30 5G	Snapdragon 765	24.0	53.8	61.9
Huawei P40 PRO	Kirin 990	18.9	43.2	65.7
Huawei P30	Kirin 980	22.5	47.0	34.8
Huawei P10	Kirin 960	37.2	118.4	243.7

5. Conclusion

In this paper, We proposed a real-time video super-resolution network EVSRNet, which mainly composed of five residual blocks and sub-pixel convolution layer [7]. We show that our method is the fastest and can be easily ported to mobile devices. In future, we will further study the characteristics of video super-resolution and add some high-efficiency units to achieve a better balance between quality and efficiency.



(a) 59th frame of sequence 0



(b) 59th frame of sequence 6

Figure 3. Qualitative comparison on the REDS val datasets. Zoom in for better visualization.

References

[1] TensorFlow Lite. https://www.tensorflow.org/lite.

[2] https://competitions.codalab.org/competitions/28112.

[3] A. Ignatov et al., AI Benchmark: All About Deep Learning on Smartphones in 2019, 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019.

[4] Ignatov A , Timofte R , Szczepaniak P , et al. AI Benchmark: Running Deep Neural Networks on Android Smartphones[J]. European Conference on Computer Vision, 2018.

[5] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenges on video deblurring and superresolution: Dataset and study. In CVPRW, June 2019.

[6] C. Dong, C. Loy, K. He, and X. Tang, " Learning a deep convolutional network for image super-resolution," in ECCV, 2014.

[7] W. Shi, J. Caballero, F. Husz´ar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR 2016.

[8] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In CVPRW, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR 2016.

[10] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in CVPR, 2019.

[11] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. IEEE Transactions on Computational Imaging, 2(2):109–122, 2016.

[12] Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Hongyang Chao, and Tao Mei. Temporal deformable convolutional encoder-decoder networks for video captioning. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[13] Z. Wang, P. Yi, K. Jiang, J. Jiang, Z. Han, T. Lu, and J. Ma, " Multi-memory convolutional neural network for video super-resolution," IEEE Trans. Image Process., vol. 28, no. 5, pp. 2530–2544, May 2019.

[14] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In CVPR Workshops, 2019.

[15] Takashi Isobe,Fang Zhu,Xu Jia,ShengjinWang. Revisiting Temporal Modeling for Video Super-resolution. In: British Machine Vision Conference (BMVC) 2020.

[16] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In CVPR Workshops, pages 0–0, 2019.

[17] Liu J , Tang J , Wu G . Residual Feature Distillation Network for Lightweight Image Super-Resolution[J]. 2020.

[18] Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network.. ACM 2019
[19] Kim H , Hong S , Han B , et al. Fine-Grained Neural Architecture Search. 2019.
[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.
[21] Romero, Andres, Ignatov, et al.Real-Time Video Super-Resolution on Smartphones with Deep Learning, Mobile AI 2021 Challenge: Report. In CVPR Workshops, pages 0–0, 2021.
[22] https://ai-benchmark.com/download.