Video Super-Resolution Based on Multiple Networks Merging

Video Super-Resolution Based on Multiple Networks Merging

Weiye Shao[a], Zhihai Xu[*], Huajun Feng[b], Qi Li[c]

School of Zhejiang University, Hangzhou 310027, China

*Corresponding author e-mail: xuzh@zju.edu.cn, [a]21630052@zju.edu.cn, [b]fenghj@zju.edu.cn, [c]liqi@zju.edu.cn,

Abstract. Video or image super-resolution technology is designed to recovery a high-resolution image from a low-resolution video or image. In recent years, deep neural networks have developed rapidly and have been applied in many digtal image processing tasks. In this paper, we choose an optical flow network to effectively exploit temporal relation within multiple consecutive video frames. In addition, we propose a weight distribution network which gives weight images of different high-resolution images obtained by various super-resolution network methods. This architecture combines advantages of different methods and provides a more accurate high-resolution image. We build a dataset with high-definition video, and use this dataset to train and test our networks. We compare our algorithm with other super-resolution methods and show that it performs a state-of-the-art results.

1. Introduction

Video or image super-resolution (SR) algorithm has been an important topic in digital image processing domain for long time. SR is proposed to recover a high-resolution image from a low-resolution image. In many realms, this technology has been widely used, such as security area [1] [2], satellite imaging [3] [4], Ultra High Definition Televisions (HDTV), and mobile photography.

Recovering high-resolution images from low-resolution images is an ill-posed problem because of the loss of high-frequency information, i.e. there are many possibilities for recovering high-resolution images, so redundant information is needed to determine a good high-resolution image. The single-frame SR technique recovers from only one image and utilizes the spatial correlation within the image. Different from single-frame SR, video SR or multi-frame image SR obtains high-resolution images through consecutive multiple low-resolution images, which not only utilizes spatial correlation information within one image, but also utilizes time correlation information between different frames. These years, research on super-resolution have been greatly developed, however, existing algorithms still do not solve this problem very well whether it is single-frame SR or video SR.

Before the popularity of deep learning, single-frame SR mostly uses learning dictionary learning approaches, kernel regression approaches [7] and some other approaches. The video SR algorithm mainly give the relationship function between a series of low-resolution images and high-resolution images, then optimize this function, such as maximizing high-resolution images probability by Bayesian maximal posteriori estimate method, to get the most probable value [7]-[9]. Dictionary learning is also developed from single image SR to video SR [10].

In recent years, inspired by plenty of successes with deep learning, single-frame SR extracts the features of the image through neural network to utilize the spatial correlation of the image, and then recovers the high-resolution images by these features as well as spatial redundancy information [11]-[17]. The video SR deep learning method is mainly developed by the single-frame SR method. Because the video SR need to use the temporal information between frames, estimating the motion information between different frames and compensating inter-frame motion are very important. A series of video SR papers have proposed a plenty of approaches in motion estimation compensation or in combining multiple images [18]–[24].

1.1. Related work

Before the popularity of deep learning, dictionary learning approaches is dominant for single image SR, such as ANR [5], A+ [6]. Dictionary learning methods generate a high-resolution dictionary and a low-resolution dictionary, then define low-resolution images as a sparse linear combination of dictionary atoms in low-resolution dictionary coupled to the high-resolution dictionary.

Dong et al. first introduce deep learning into the SR area and propose SRCNN [11], which is a three-layer convolutional network structure. SRCNN need to upscale at first and the first layer extracts and represents the input image features, and the second layer performs nonlinear mapping, and final layer reconstructs high resolution. image. Base on this method, Dong et al. propose FSRCNN [12], which mainly uses smaller convolution kernels and more layers, and finally enlarges the size through the deconvolution layer, so it can directly input the original low-resolution image into the network, which reduces the amount of computation. A sparse coding network (SCN) is proposed by Wang et al. according to the sparse representation method for low-resolution images SR [17]. Kim et al. propose VDSR [13], assuming that the high-resolution image and the low-resolution image are similar in low-frequency information, so the residual network structure is used, which makes the networks much deeper. A low-resolution image was up sampled to the target size by interpolation and input to VDSR, and then add this image to the residuals which learned by the network to get the output of the network. Kim et al. propose another algorithm, DRCN [14], which introduce the recurrent neural network structure for the first time. An efficient sub-pixel convolution network called ESPCN [15] is proposed by Shi et al. This method extracts features directly on the low-resolution image size. After three convolutions, a series of feature maps are obtained, and the feature maps are arranged to obtain high-resolution images. Ledig et al. apply the anti-generation network (GAN) to SR and propose SRGAN [16]. This method consists of a generation network and a discriminant network. The generative network part is composed of residual network. It is mainly used to generate high-resolution images. The discriminative model uses VGG and is responsible for judging whether the generated high-resolution images are qualified. At the same time, the loss function is changed, and the mean square error (MSE) is no longer used as the loss function.

Liao et al. first use deep learning on video SR. The network, termed EDSR, obtain different SR maps by changing different motion compensation parameters, and then used neural networks to combine these SR maps to recover high-resolution maps. [18]. Huang et al. use bidirectional cyclic neural networks to

Exploit timing information [22]. Kappeler et al. improve SRCNN and propose VSRnet, which can be applied to video SR. This method calculates the motion field by the traditional optical flow method, then these images are aligned according to motion estimation and inputted to the neural network to obtain a SR image. [19]. VESPCN is proposed by Caballero et al. [20], which combines ESPCN, the motion detection and compensation module to form an end-to-end network. Yang et al. proposed a video SR based on a space-time residual network as a structure [21]. Tao et al. use the motion detection method in VESPCN to estimate motion, and then map all the input frames into the high-resolution grids according to the motion detection result, and then used the LSTM structure to selectively memorize the information of the multi-frame picture [23]. Liu et al. design a multi-branch neural network structure, and input different number low-resolution images for each branch, and use one branch as a time modulation branch for weight estimation of other branches, finally merged these images and gain final result [24].

For video SR, motion detection and compensation between different frames is very important. The mainly method is the optical flow method. With the development of deep learning, an optical flow network was proposed [25]. In the last few years, several new optical flow networks are proposed, and the accuracy has made considerable improvements [26]-[27].

1.2. Motivation and contributions

Although many approaches have been proposed to improve the accuracy of high-resolution image, the result is still much room for improvement. Because current SR methods can perform well in some images and perform bad in other images, we consider to merge this SR network methods. In this paper, we use state-of-the-art optical flow networks to preprocess input frames, which can effectively estimate motion between consecutive frames and utilize temporal correlation information. Also, we proposed a multiple-approaches merging network architecture and a weight distribution network which gives weight images of different high-resolution drafts obtained by various super-resolution methods. Finally, we build a dataset with high-definition video, and use this dataset to train and test our networks and compare our algorithm with other super-resolution methods.

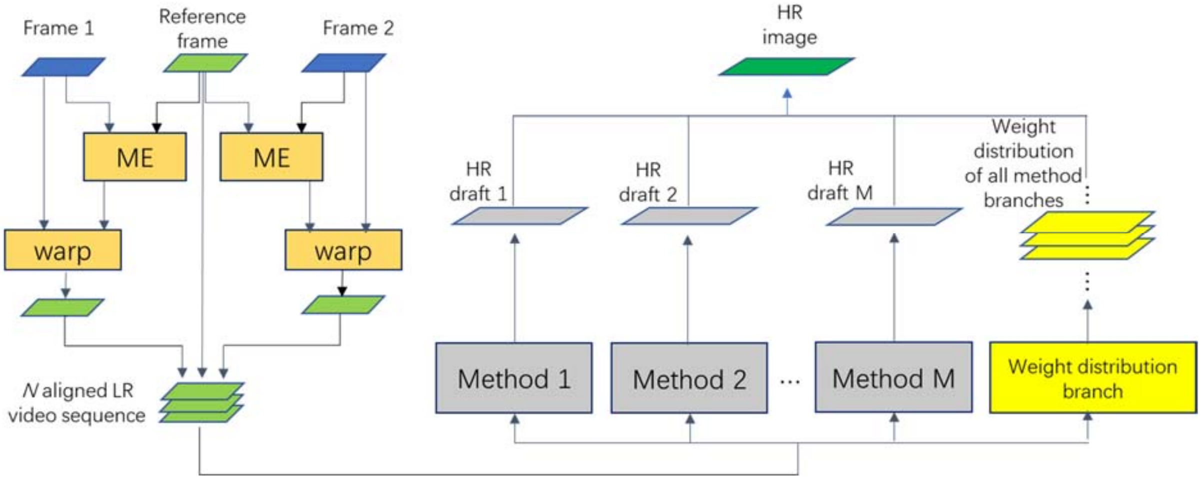


Figure 1. Proposed design for video SR. In order to simplify the figure, the number of input frame is only 3 here. It can input more frames into the network. Every method branch produce one HR draft, and weight distribution branch produces M weight images. Final HR image is the combination of these drafts.

2. Methods

In this section, we will introduce the network structure proposed in this paper, and give explanations and details of the network architecture.

2.1. Overview

The main purpose of this paper is to improve the accuracy of video super-resolution. The number of consecutive images is *N*, and choose one of the frames to use as a reference frame. The optical flow network takes each other *N-1* images and the reference frame as input and produce a motion field. Then, warp the image to produce the warped image according to motion field. After all images are aligned, these images are inputted into *M* branches. For each branch, a different network structure is used. We use three single-frame super-resolution network frameworks, that are SRCNN, FSRCNN and ESPCN, and we make some modifications based on these networks. Finally, *M* super-resolution drafts are obtained.

The aligned *N* images are also inputted into the weight distribution network branch, and this network produce *M* weight images. These weight images corresponding *M* SR drafts, and give weight coefficient of each pixel, i.e. each HR draft has a corresponding weight image and the weight image determines the ratio of each draft in final HR image. Then the weight is combined with all the super-resolution drafts to obtain the final result. The overview of the network is shown in Figure 1.

2.2. Motion estimation and compensation

We use the FlowNet-C [25] network as the motion estimation part to produce motion filed on multi-frame pictures. Compared with other networks for motion detection, such as FLowNet-S [25], FlowNet2.0 [26], it balance quality and computation cost, i.e., the accuracy of FlowNet-S is acceptable in our approach, and the amount of calculation is not too large. After obtaining a motion field map between two frames from the optical flow network, then warp image to the reference one.

2.3. Network architecture

N consecutive aligned images are inputted to each branch structure. The number of channels per image is *c*, and the final up-sampling ratio is *u*. So, the input layer of each branch network is $N \times c$ channels. For branch one, a network 5-layer convolutional network similar to SRCNN is used. The *N* aligned frames need to up-sampling to the target size by bicubic interpolation before inputting. The first convolutional layer convolution has 64 kernels in the size of 5×5 with stride 1. This is followed by 3 layers which all are convolutional layers with 32 convolution kernels in the size of 3×3 and stride 1. The fifth convolutional layer has *c* convolution kernels in the size of 3×3 with stride 1. Each convolutional layer is followed by a ReLU layer which is activation function.

For branch two, a network similar to FSRCNN is used, which has 7 convolutional layers and 1 deconvolution layer. The first convolutional layer convolution has 56 kernels in the size of 5×5 with stride 1. The second convolutional layer convolution has 12 kernels in the size of 1×1 with stride 1. This is followed by 4 convolutional layers with 12 convolution kernels in the size of 3×3 and stride 1. The seventh convolutional layer has 56 kernels in the size of 1×1 with stride 1. Each convolutional layer is followed by a ReLU layer. The last layer is a deconvolution layer, which uses *c* kernels in the size of 9 × 9 and stride *u*.

For branch three, a network similar to ESPCN is used, which has 9 convolutional layers and 1 sub-pixel convolutional layer. The first convolutional layer convolution has 64 kernels in the size of 5×5 with stride 1 This is followed by 8 convolutional layers with 32 convolution kernels in the size of 3×3 and stride 1. Each convolutional layer is followed by a TanH layer which is activation function. The last layer is a sub-pixel convolutional layer which has $u^2 \times c$ kernels in the size of 3×3 with stride 1 to produce $u^2 \times c$ channels feature images and rearranges these images to *c* channels high resolution images.

Although we only use three different networks as branch, it can add more SR models as new branch.

Weight distribution network consists of five convolutional layers and one SoftMax layer, and the input is also *N* aligned images. The *N* aligned frames need to up-sampling to the target size by bicubic interpolation before inputting. The first convolutional layer convolution kernel has 32 kernels in the size of 5×5 with stride 1. The second to fourth convolution kernels have 16 kernels in the size of 3×3 with stride 1. After the first four convolutional layers, they are all follow by a layer of ReLU. The fifth convolutional layer convolution kernel has *M* kernels in the size of 3×3 with stride 1. The last layer is SoftMax layer, which is used to make the sum of each pixel's weights among all weight images is one.

The final high-resolution image is formed by each HR draft pixel-wise multiple its corresponding weight images and summed up. It is defined as

$$\boldsymbol{H} = \sum_{i=1}^{M} \boldsymbol{D}_i * \boldsymbol{W}_i \tag{1}$$

Where H is output high-resolution image, $\boldsymbol{D_i}$ is the i-th HR draft image, $\boldsymbol{W_i}$ is the i-th weight image. * means pixel-wise multiple here.

2.4. Loss function

We first train every network branch respectively, and use MSE loss as loss function. It is defined as

$$\text{Loss} = \|\boldsymbol{D}_i(\boldsymbol{l};\boldsymbol{\theta_{di}}) - \boldsymbol{G}\|_2^2 \tag{2}$$

Where i is the i-th branch, $\boldsymbol{D}_i(\boldsymbol{l};\boldsymbol{\theta_{di}})$ is HR draft, $\boldsymbol{G}$ is ground truth image, $\boldsymbol{l}$ are all the associated aligned LR frames, $\boldsymbol{\theta_d}$ are parameters of network branch.

Then we train weight distribution network, and also use MSE loss as loss function. It is defined as

$$\text{Loss} = \left\|\sum_{i=1}^{M} \boldsymbol{D}_i(\boldsymbol{l};\boldsymbol{\theta_d}) * \boldsymbol{W}_i\,(\boldsymbol{l};\boldsymbol{\theta_\omega}) - \boldsymbol{G}\right\|_2^2 \tag{3}$$

Where $\boldsymbol{D_i}$ is the i-th HR draft image, $\boldsymbol{W_i}$ is the i-th weight image, $\boldsymbol{\theta_\omega}$ are parameters of weight distribution network, $\boldsymbol{G}$ is ground truth image.

3. Experiments and Results

In this section, we show our experiments and results.

3.1. Datasets

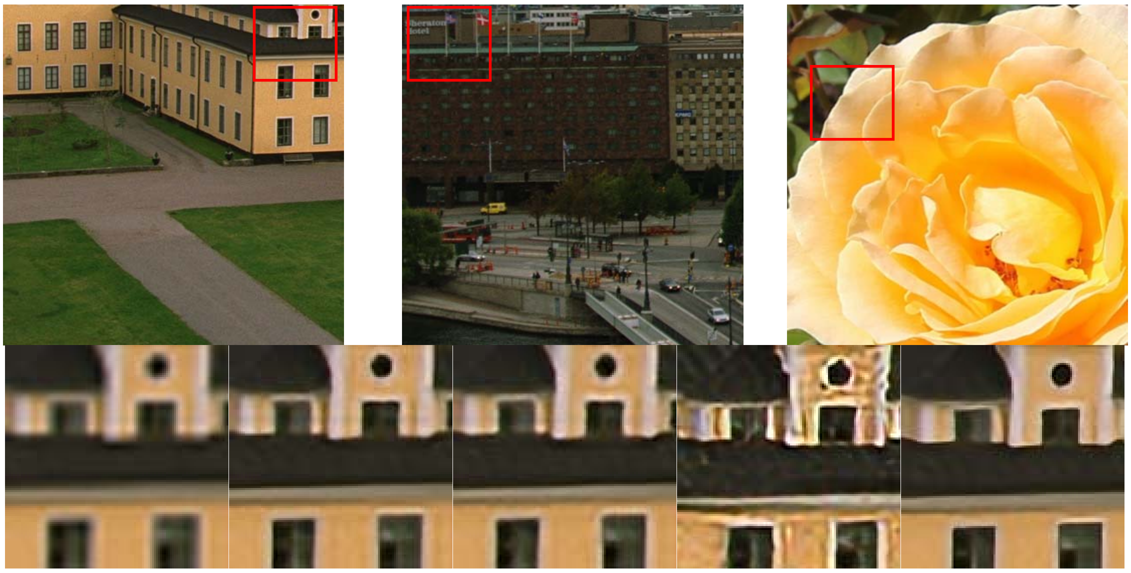
We download 50 1080p HD videos from CDVL database. Most of them contain various situation. We randomly collect 50 sequences from each HD videos and randomly choose 95% of these sequences as training data, and the rest sequences are for validation and testing. The corresponding LR sequences are down-sampled by interpolation from HR sequences.

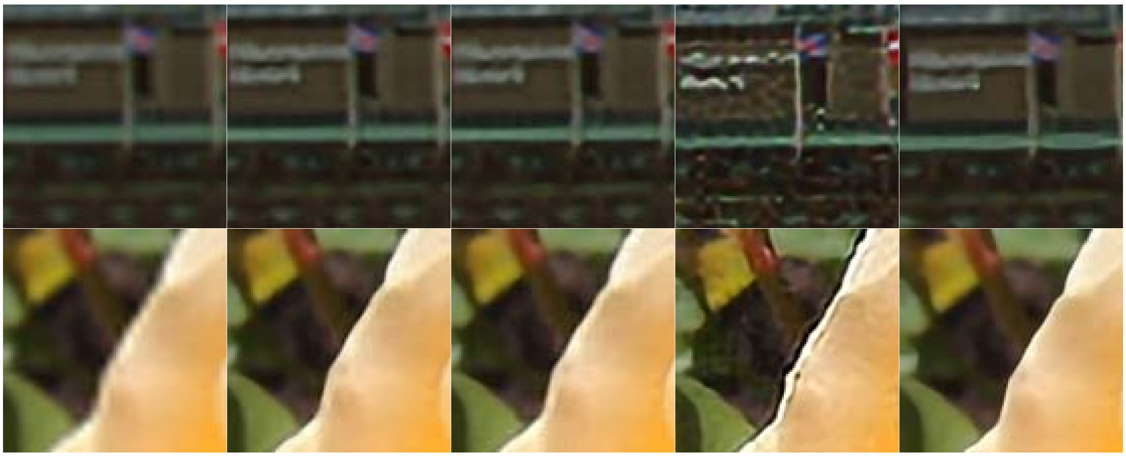
3.2. Implementation details

For model training, we first convert each frame to YCbCr color space and only consider the luminance channel. Hence the number of channels per image c is 1. We input 3 frames to the network, so N is 3. We use Stochastic Gradient Descent solver with learning rate of 0.0001, and momentum is 0.9, weight decay is 0.0001. We stop training after 1million iterations. First, we train every branch network use loss function (2). Then we fix the parameters of every branch network and use loss function (3) to train weight distribution network.

3.3. Image super-resolution results

We evaluated our method in test data with an upscale factor of 4, and show the results in Figure 2 and Table 1. There are the result from other SR methods: SRCNN [11], FSRCNN [12], and SPMC [23] in Table 1, and Figure 2. We compare different methods by two indexes which are PSNR and SSIM. As show in Table 1 and Figure 2, our results have better indexes than other several methods.





(a) Bicubic (b) SRCNN (c) FSRCNN (d) SPMC (e) Ours

Figure 2. The three images on the top are ground truth. From top to bottom is building, city, flower. All images use five methods to super-resolution.

Table 1. Comparison with other SR methods (PSNR/SSIM) .

	Bicubic	SRCNN[11]	FSRCNN[12]	SPMC[23]	Ours
Building	28.68/0.886	30.33/0.905	30.43/0.906	28.13/0.834	**30.98/0.9129**
City	29.89/0.800	30.95/0.826	30.98/0.828	28.72/0.762	**31.07/0.832**
Flower	32.22/0.949	34.50/0.964	34.92/0.967	30.88/0.931	**36.65/0.970**

4. Conclusion

In this paper, we have proposed a new network architecture for video SR. Our method combines multiple deep learning approaches of SR and we use a weight distribution network to produce weights of each approaches. This architecture combines advantages of different methods and provides a more accurate high-resolution image. Results show that our approach outperform the current state of the art in video SR.

References

[1] W. Zou and P. C. Yuen. Very Low Resolution Face Recognition in Parallel Environment. IEEE Transactions on Image Processing, 21: 327–340, 2012.

[2] Suresh, K. V., Kumar, G. M., & Rajagopalan, A. N. (2007). Superresolution of license plates in real traffic videos. IEEE Transactions on Intelligent Transportation Systems, 8 (2), 321-331. Su An, Qiao Xue-guang, Jia Zhen-an, et al. Temperature and pressure responsive characteristics of polymer packaged fiber Bragg grating with large dynamic range [J]. Chinese Journal Of Lasers, 2005, 32 (2): 224-227.

[3] H. Demirel and G. Anbarjafari. Discrete wavelet transformbased satellite image resolution enhancement. IEEE Transactions on Geoscience and Remote Sensing, 49 (6): 1997–2004, 2011.

[4] M. W. Thornton, P. M. Atkinson, and D. a. Holland. Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using SR pixel-swapping. International Journal of Remote Sensing, 27 (3): 473–491, 2006.

[5] Timofte, R., De Smet, V., Van Gool, L.: Anchored Neighborhood Regression for Fast Example-Based Super Resolution. ICCV (2013) 1920{1927Morey W W, Meltz G, Glenn W H. Fiber Bragg grating sensors [C]. SPIE, 1989, 2507: 98-107.

[6] Timofte, R., De Smet, V., & Van Gool, L. (2014, November). A+: Adjusted anchored neighborhood regression for fast SR. In Asian Conference on Computer Vision (pp. 111-126). Springer, Cham.

[7] Liu, C., & Sun, D. (2011, June). A Bayesian approach to adaptive video super resolution. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 209-216). IEEE.

[8] Villena, S., Vega, M., Molina, R., & Katsaggelos, A. K. (2009, September). Bayesian SR image reconstruction using an l1 prior. In Proceedings of 6th international symposium on image and signal processing and analysis (pp. 152-157).

[9] Villena, S., Vega, M., Babacan, S. D., Molina, R., & Katsaggelos, A. K. (2013). Bayesian combination of sparse and non-sparse priors in image super resolution. Digital Signal Processing, 23(2), 530-541.

[10] Dai, Q., Yoo, S., Kappeler, A., & Katsaggelos, A. K. (2015, September). Dictionary-based multiple frame video SR. In Image Processing (ICIP), 2015 IEEE International Conference on (pp. 83-87). IEEE.

[11] Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image SR using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2), 295-307.

[12] Dong, C., Loy, C. C., & Tang, X. (2016, October). Accelerating the SR convolutional neural network. In European Conference on Computer Vision (pp. 391-407). Springer, Cham.

[13] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Accurate image SR using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1646-1654).

[14] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Deeply-recursive convolutional network for image SR. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1637-1645).

[15] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., ... & Wang, Z. (2016). Real-time single image and video SR using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1874-1883).

[16] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017, July). Photo-Realistic Single Image SR Using a Generative Adversarial Network. In CVPR (Vol. 2, No. 3, p. 4).

[17] Wang, Z., Liu, D., Yang, J., Han, W., & Huang, T. (2015). Deep networks for image SR with sparse prior. In Proceedings of the IEEE International Conference on Computer Vision (pp. 370-378).

[18] Liao, R., Tao, X., Li, R., Ma, Z., & Jia, J. (2015). Video SR via deep draft-ensemble learning. In Proceedings of the IEEE International Conference on Computer Vision (pp. 531-539).

[19] Kappeler, A., Yoo, S., Dai, Q., & Katsaggelos, A. K. (2016). Video SR with convolutional neural networks. IEEE Transactions on Computational Imaging, 2 (2), 109-122.

[20] Caballero, J., Ledig, C., Aitken, A. P., Acosta, A., Totz, J., Wang, Z., & Shi, W. (2017, July). Real-Time Video SR with Spatio-Temporal Networks and Motion Compensation. In CVPR (Vol. 1, No. 2, p. 7).

[21] Yang, W., Feng, J., Xie, G., Liu, J., Guo, Z., & Yan, S. (2018). Video SR based on spatial-temporal recurrent residual networks. Computer Vision and Image Understanding, 168, 79-92.

[22] Huang, Y., Wang, W., & Wang, L. (2015). Bidirectional recurrent convolutional networks for multi-frame SR. In Advances in Neural Information Processing Systems (pp. 235-243).

[23] Tao, X., Gao, H., Liao, R., Wang, J., & Jia, J. (2017, October). Detail-revealing deep video SR. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy (pp. 22-29).

[24] Liu, D., Wang, Z., Fan, Y., Liu, X., Wang, Z., Chang, S., ... & Huang, T. S. (2018). Learning Temporal Dynamics for Video SR: A Deep Learning Approach. IEEE Transactions on Image Processing, 27(7), 3432-3445.

[25] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE

International Conference on Computer Vision (pp. 2758-2766).

[26] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017, July). Flownet 2.0: Evolution of optical flow estimation with deep networks. In IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 2, p. 6).

[27] Sun, D., Yang, X., Liu, M. Y., & Kautz, J. (2018). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8934-8943).