https://doi.org/10.2352/EI.2022.34.14.COIMG-247
© 2022, Society for Imaging Science and Technology

OldVSR: A model for the video super-resolution and restoration of old real-world TV series

Tony Nokap Park and Taeyoung Na ; SK Telecom; Seoul, Korea

Abstract

With the recent advance in video super-resolution (VSR) techniques, there have been many requests for super-resolve real-world old analog TV series into high-definition digital content. As excellent classical TV series may receive little to no attention due to their poor video quality, restoring them would open new business opportunities for reusing old TV contents. A problem with restoring real-world old TV series is in the complex artifacts introduced by the old interlaced scanning and compression artifacts during the digitization of old analog videos. Though recent DNN-based VSR models perform nicely on clean videos, due to the artificial nature of interlacing and compression artifacts, they fail to restore old videos into a high-definition counterpart free from noticeable artifacts. In this work, we propose OldVSR for restoring old real-world TV series with artifacts of artificial nature. The proposed model implements a bidirectional recurrent structure with first and second-order propagation where each recurrent layer implements two main functions, i.e., Feature alignment (FA) and Pyramid feature aggregation (PFA). The outputs of the forward and backward layers are merged and upsampled to produce a High-Definition (HD) frame of the input standard-definition (SD) frame. We demonstrate through experiments that our proposed OldVSR can effectively remove artifacts of artificial nature from old videos and successfully restores old TV series.

1. Introduction

Video is one of the most closely integrated multimedia in our daily life. The consumers' expectations for better Quality-of-Experience (QoE) nowadays have been higher than ever before. Restoring poor-quality standard-definition (SD) videos has become an inevitable task with the popularity of high-definition (HD) displays because poor-quality videos could immediately disappoint a large volume of consumers regardless of their excellent story, eventually resulting in a substantial revenue loss for content providers.

Video super-resolution (VSR) is a challenging problem in computer vision that aims at recovering the high-resolution (HR) video using the information from the low-resolution (LR) counterparts. With the success of deep learning methods, VSR algorithms based on deep learning with diverse architectures have been studied extensively.

Because DNN-based VSR models show promising results, content providers often request the restoration of old analog TV series from SD into HD ones. As many excellent classical contents receive little to no attention due to their poor video quality, restoring them would benefit both the content producers and the content viewers in that content producers will enjoy the low cost of reproducing broadcast-ready content while content viewers could enjoy past videos full of old memories.

For DNN-based VSR models, a dataset composed of SD and HD frame pairs is necessary for training. But for real-world videos, it is not easy to find datasets with matching SD and HD pairs. Therefore, a degradation model generates the dataset for training existing DNN-based VSR models. For example, given a sharp HR image, the degradation model generates an LR counterpart to make a pair of training data. Many public datasets construct their LR frames using two representative degradation models: bicubic[1] or traditional[2, 3]. In the bicubic degradation model, simple bicubic interpolation generates the LR image. In the traditional model, the LR image is generated by applying a sequence of Gaussian kernel induced blur, bicubic downsampling, and simple noise model formally defined by:

$$x^{LR} = D_s \circ K(x^{HR}) + N_\theta. \tag{1}$$

That is, the HR image x^{HR} is convolved using a blur kernel K to get a blurry image, followed by a downsampling D_s with scale factor s and an addition of white Gaussian noise N with standard deviation θ to get the LR image x^{LR}. An example of publicly available dataset is the Realistic and dynamic scenes (REDS)[4]. REDS, one of the most used dataset for the training of VSR models, offers both the bicubic and the traditional degradation model generated training dataset with a downsampling scale factor of 4.

Public datasets have been an essential part of VSR development. They offer relatively sharp images without noticeable artifacts. But, the over-simplifying mathematical assumption of degradation models restricts the generation of LR images closer to real-world images. For example, in these degradation models, the noise is usually assumed to be AWGN which rarely matches the noise distribution of real images. Indeed, the noise could also stem from camera sensor noise and JPEG compression noise which are usually signal-dependent and nonuniform [6]. In addition, unlike the public datasets, real old videos may contain several types of complex artifacts.

Interlacing artifact is an example of a very common anomaly observed in many old TV series due to the early interlaced scanning protocols like PAL and NTSC they were produced. The interlaced scan is a display signal type in which one-half of the horizontal row pixels (odd-field) are refreshed in one cycle and the other half (even-field) in the next. The problem of this method is that the two fields cannot be aligned exactly, especially when there exist large movements of objects. This creates artifacts of artificial nature in the form of comb-teeth shape. Many deinterlace algorithms have been developed but severe artifacts may remain even after it has been applied.

Compression is another major source of artifacts. Especially block-based video coding schemes create various spatial artifacts due to block partitioned processing and quantization. These results as blurring, blocking, ringing, basis pattern effect, and color

Figure 1. *An illustration of visual artifacts found in old TV series. In this example, the deinterlacing and compression artifacts manifests as a stair pattern in the man's back shoulder. On the letter, we observe a comb-teeth pattern.*

Figure 2. ***Qualitative example.*** *The resotration result of existing DNN-based VSR models (EDVR [20] and BasicVSR [13]). These models fail to remove artifacts in old video frames.*

bleeding. Aside from the interlacing and compression artifacts, old videos may also include video acquisition artifacts, video post-processing artifacts, and many more.

All these artifacts, blur, and noise could be blended in random order and manifest in a diverse way to worsen the perceptual quality of the old video frames. Figure 1 shows an example where the deinterlacing and deblocking artifact are blended. In the case of the letter, a human perceives the discontinuous white and blank pattern as a hole that needs filling to make it into a proper letter. In the case of the stair pattern on the man's shoulder, a human would draw a line to repair it.

Contrary to a human, existing VSR models fails to restore the artifacts mentioned above. Because existing DNN-based VSR models have been trained mainly on clean datasets, their reconstruction ability is somewhat limited to work on clean images only. For instance, given a real-world frame, regardless of whether the blur or artifact has been modeled accurately or not, a few amounts of noise mismatch will be sufficient to cause a performance drop to VSR models. Indeed existing DNN-based VSR will unlikely extrapolate degradations they never learned. Instead, these models will reconstruct the unwanted pattern and make it more focused and salient. Figure 2 shows the reconstructed images from existing DNN-based VSR models.

The goal in developing OldVSR is to build a plausible model which can deal with unexpected artifacts in real-world old TV series to produce an HD video with enough quality to broadcast. In this work, we present the process of how we handle the anomalies caused by the blend of deinterlacing and deblocking artifacts present in Figure 1.

2. Model Architecture

OldVSR implements bidirectional recurrence structure for temporal aggregation of frames where each recurrent layer contains two main functions: Feature alignment (FA) and Pyramid feature aggregation (PFA). Given a set of LR frames, the forward and backward recurrent layers generate aligned and aggregated feature maps that are further merged and upsampled to produce a HD frames. In the rest of this section we present the details of our model's architecture. The structure of OldVSR is depicted in Figure 3.

2.1. Recurrence structure

The recurrent framework is popular for many video processing tasks including super-resolution[7, 8, 9, 10, 11, 12, 13]. The recurrent framework could either be unidirectional [8], bidirectional[13] , or omnidirectional[14]. Recently, Chan et al. [13] showed the effectiveness of bidirectional propagation over unidirectional counterpart in aggregating features sequentially. Moreover, [14] proposes an omniscient network that exploits the present state feature for performance improvement.

In this work, we adopt a bidirectional recurrence framework. In addition to the first-order recurrence, our model relaxes the first-order Markov property to use the second-order recurrences too. In general bidirectional settings, the frames are first propagated in the backward direction and then into the forward direction independently. Therefore, the backward and forward layers received additional information only from future and previous frames respectively. Restricting an interconnection from backward to forward recurrent layers, the forward layer can receive information from both the future and past frames, leading to more abundant information and better quality outputs. Given a set of LR frames $\{...,x_{t-2},x_{t-1},x_t,x_{t+1},x_{t+2},...\}$ and features propagated from forward layers $\{f^F_{t-2},f^F_{t-1}\}$ and backward layers $\{f^B_{t+1},f^B_{t+2}\}$ at respective times, we have

$$f^B_t = R_B(x_t,x_{t+1},x_{t+2},f^B_{t+1},f^B_{t+2}), \tag{2}$$

$$f^F_t = R_F(x_t,x_{t+1},x_{t+2},f^F_{t-1},f^F_{t-2},f^B_t), \tag{3}$$

where R_B and R_F denote the backward and forward recurrent layers, respectively.

2.2. Alignment

VSR models involve the handling of multiple consecutive video frames. Due to the motion of objects, the features in the neighbor frames are, in most cases, spatially misaligned. The alignment module plays the key role of explicitly aligning the

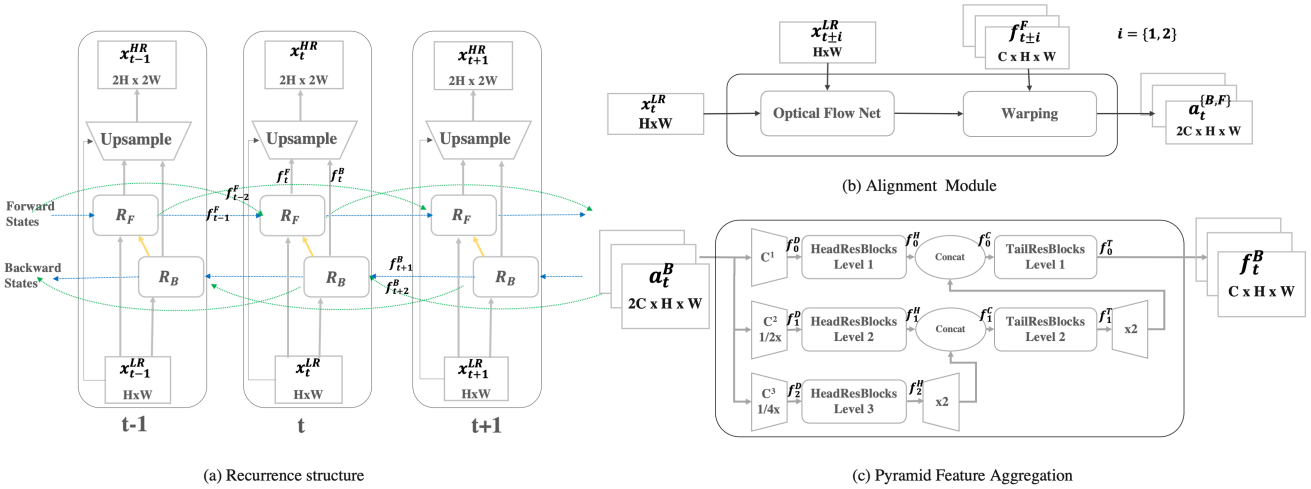

Figure 3. Model architecture of OldVSR *(a) The recurrence structure. (b) the alignment layer. (c) the pyramid feature aggregation layer*

features in the neighboring frames for subsequent feature aggregation.

There have emerged many different methods for alignment. Most of these methods fall into two main types: the one that uses optical-flow and the other that uses deformable convolution [25, 26] for alignment. In optical-flow-based methods[16, 12, 15, 17, 18, 19, 13], the inter-frame motion information is extracted first, and then the warping operation is performed between frames according to the inter-frame motion information to make one frame align with another one. In deformable alignment methods[20, 21, 23], an additional convolution filter computes the misalignment offsets from the concatenated input and neighbour feature maps.The deformable-convolution kernel adopts the computed misalignment offset and computes the final output from given input feature map. Compared to the previous method, the offset in the deformable alignment method replaces the role of optical flow. In practice, the training of deformable alignment often incurs instability which results in performance deterioration. But, when trained well it improves performance over the optical-based alignment.

In this work, we adopt a optical-flow-based alignment. Optical flow can be computed using either the traditional methods(e.g., Lucas&Kanade[28] and Druleas [29]) or deep learning based methods such as FlowNet[27], FlowNet 2.0[30], SPyNet[31], and PWC-Net[32]. SPyNet and PWC-Net are good choices as they implement a lightweight architecture that allows faster inference and shorter training times. We choose PWC-Net for accuracy gain that results from better construction of the feature pyramid and the cost volume. With PWC-Net we compute the optical flow o_t for backward and forward layers B and F given by:

$$o_t^{\{B,F\}} = \{O(x_t, x_{t\pm i}) | i = 1,2\} \tag{4}$$

Next, given the optical flow o_t, our model warps the feature map of the neighbor frame and aligns it to the target. The output of the warping is aligned feature map a_t given by:

$$a_t^{\{B,F\}} = \{W(f_{t\pm i}^{\{B,F\}}, o_t^{\{B,F\}}) | i = 1,2\} \tag{5}$$

We perform the warping on the feature map instead of the image for performance increase as demonstrated in [13]. The aligned feature map is then passed to the pyramid reconstruction layer for feature aggregation as we discuss next.

2.3. Feature aggregation

In VSR, a deep chain of residual blocks without batch normalization is the commonly used component for feature aggregation. This plain residual block processes the features at the same scale as the input (e.g., height x width). Some artifacts like the interlacing are more pronounced when there is a large displacement of objects. Simple downsampling could diminish the displacement thus alleviating the effect of interlacing artifacts. Downsampling is equivalent to amplification of the field-of-view (FOV) for convolution filter, therefore, gives a better chance to handle large displacements. But there is a tradeoff as relying only on downsampled frame results in a loss of details

The pyramid structure is a popular choice that implements the classical coarse-to-fine concept. The coarse-to-fine concept proceeds by processing the artifacts using a multi-scale image pyramid, starting from the coarsest level (the largest FOV) to the finest (the smallest FOV). By gradually refining artifacts through the pyramid levels, this approach can aggregate the features at multiple scales and handle large displacements better, therefore, improve the visual quality of reconstructed frames.

The Pyramid Feature Aggregation (PFA) module is the aggregator used in our model (Figure.3(c)). PFA performs feature aggregation at 3 multiple levels. First, given the aligned feature map as input, PFA convolves the input using a convolution with a stride 2^l, $l = \{0,1,2\}$ to compute downsampled feature map f_l^D. Next, PFA applies the head residual blocks to downsampled feature maps to compute the feature map f_l^H. Then, PFA concatenates current level feature map with the 2x upsampled tail feature map from next level to output a concatenated feature map f_l^C:

$$f_l^C = \left[U_{2x}(f_{l+1}^T), f_l^H\right], l = 0,1., \tag{6}$$

where, PFA applies the tail residual blocks to f_l^C, $l = 0,1$ to com-

pute the feature map f_l^T except the last layer where $f_2^T = f_2^H$. Through the multi-scale backward feature aggregation starting from level 2 to level 0, PFA outputs the feature map $f_t^{\{B,H\}} = f_0^T$. Given the intermediate features $f_t^{\{B,H\}}$, an upsampling module U_{2x} composed of multiple convolutions and Pixel-Shuffle [22] is used to generate the output HR frame:

$$x_t^{HD} = U_{2x}\left(f_t^B, f_t^H\right). \tag{7}$$

3. Experiments

In this section, we present our experiment. We first describe the process of collection of our dataset. Next, we explain the settings for training OldVSR. Finally, we present the comparison results of OldVSR to the existing VSR models.

3.1. Dataset

Based on the year of production, the old TV series categorize into two types: Those produced before and after the year 2000. Before 2000, TV series were filmed only in an analog format using SD filming equipment. Between 2000 to 2006, during the transition period from analog to digital standard, TV series were produced jointly in SD and HD formats. From TV series having both the SD and HD copies, we extracted 300 short clips of 60 frame pairs. Because the aspect ratio between SD and HD copy differs, we first align SD to HR copy and then remove extra pixels at the boundary to make an aligned sample pair.

TV series produced before 2000 have more titles and contain more diverse artifacts than those produced after 2000. Especially, interlacing artifacts are more severe and varied in the older series, and this is probably an additional effect caused by the limitations of the past compression technology while storing analog videos into digital format. Unfortunately, there is no high-quality frame paired with the corresponding low-quality frame. Therefore, to strengthen data variety, we synthesized interlaced videos from existing progressive videos. We selected 50 videos from varied genres of movies and TV Series. Select videos are progressive-scanned with Full-HD (1920x1080) format. We sampled 14 short clips of 60 consecutive frames from each video and obtained 700 clips forming a dataset of 1000 video clips in total.

We add artifacts to each clip as follows. First, we downsample Full-HD frames by a scale factor of 2 using bicubic interpolation to make the SD counterpart of the HD frames. Next, we produce interlaced frames by interleaving the upper-field from the first frame with the lower-field from the second frames for a given pair of frames(e.g.,$[t,t+1],[t+1,t+2]$), generating a frame with unchanged height at same frame rate. Then, we apply compression to the interlaced video using the H.264 codec with a constant rate factor (CRF) set at random from 28 to 38 to control the image quality at medium encoding speed. This blends interlace and compression artifacts. Finally, we deinterlace frames using "yet-another-interlace filter" (yadif) by keeping the frame rate intact. The synthetic dataset comprises downsample only, [downsample, interlace, compression] and [downsample, interlace, compression, deinterlace] clips with the mixing ratio of 0.5,0.25,0.25, respectively. We randomly select 20 percent of clips while maintaining mixing ratio and retaining these as a validation set. We use the remaining samples for training. For testing, we manually select real-world frames from old TV series with only SD copy available.

Table 1: Quantitative Results

Model	PSNR	SSIM
TDAN [23]	42.659	0.9812
EDVR-M [20]	39.931	0.972
EDVR-M woTSA [20]	42.103	0.9792
BasicVSR [13]	42.398	0.9802
OldVSR	**42.925**	**0.9821**

3.2. Settings

We train and test five models(TDAN[23], EDVR-M[20], EDVR-M woTSA[20], BasicVSR[13], and Ours) using the dataset mentioned in the previous subsection with 2x downsampling. For the existing DNN-based VSR models, we train each model using the same network and settings described on each paper except the upsampling layer modified to output a 2x upsampled frame. For OldVSR, we adopt Adam optimizer [34] with parameters $\left[\beta_1 = 0.9, \beta_2 = 0.999\right]$ and Cosine Annealing scheme [24]. The initial learning rate of the main network is set to $1.5\text{x}10^4$. We use pre-trained PWC-Net [32] as our flow network. The learning rate of the flow network is set to $2.5\text{x}10^5$. The total number of iterations is 600K, and the weights of the PWC-Net are fixed during the first 150K iterations. The batch size is 4 and the patch size of input LR frames is 64×64. We use Charbonnier loss [33] defined by $L = \sqrt{\|x_t^{HD} - x_t^{HD}\|^2 + \varepsilon^2}$ with $\varepsilon = 1 \times 10^{-12}$. It is known that Charbonnier loss better handles outliers and improves the performance over the conventional ℓ_2-loss [5]. We use 30, 20, and 10 residual blocks for the PFA's layers 1, 2, and 3, respectively. Layer1 consists of 12 head and 18 tail residual blocks, layer2 consists of 8 head and 12 tail residual blocks, and layer3 consists of 10 head residual blocks. We set all feature channels to 64.

3.3. Results

We conduct comparison experiments on four models mentioned above. We summarize the quantitative results in Table 1 and provide the qualitative comparison in Figure 4. As shown in Table 1, OldVSR achieves the best performance on the dataset presented in section 3.1. In particular, OldVSR outperforms EDVR-M [20], a light-weight version of large capacity sliding-window method EDVR, by up to 0.5dB in PSNR. When compared with the BasicVSR [13], OldVSR achieves improvements of 0.52dB. TDAN [23] qunatitatively achieves comparable PSNR and SSIM to our method, but we observed that TDAN fails to restore a faithful frames qualitatively. We present three qualitative comparisons in Figure 4. OldVSR successfully restores the broken lines produced due to the mixed interlace and compression artifacts often found on old TV series. In particular, OldVSR is the only method that smoothly connects the staircase-like line on the man's shoulder line (Figure 4 (a)), the line pattern in the car bumper and the edge of the bonnet (Figure 4. (b)), and the round edges of the hole on the right side of man's face and man's shoulder line(Figure 4(c)). TDAN restores the letter in Figure 4. (a) as OldVSR does but performs poorly on others.

Figure 4. ***Qualitative comparisons*** *(a) OldVSR restores correctly men's shoulder line. OldVSR and TDAN restores the letter (b) OldVSR restores correctly the pattern of the bumper and the edge of the bonnet. (c) OldVSR restores restores correctly the men's shoulder line and the edges of the round hole.*

4. Conclusion

In this work, we present OldVSR, a model for restoring old TV series that contains artifacts caused by the blend of interlacing and compression artifacts. We proposed an efficient pyramid structure for feature aggregation that solves artificially generated distortion. We also supplemented the lack of data for training through synthetic dataset generation. We showed qualitatively that contrary to existing DNN-based VSR models, OldVSR reconstructs smoothly missing hole and staircase pattern artifacts, often found in old TV series.

We are extending into OldVSR new architectures to enhance the quality of old TV series. The reason we deal with the blend interlace-deblocking artifact in the first place is the occurring frequency. But, there still exist other artifacts with less frequency of occurrence we will address in the future. In addition, we plan to extend the degradation model. Although methods for extending the degradation kernel or adding noise diversity are studied, their application is still unrealistic. Creating a more realistic degradation model based on the real-world old TV-series data we currently have is another task to be addressed in the future.

References

[1] Radu Timofte, Eirikur Agustsson, Luc Van Gool, MingHsuan Yang, Lei Zhang, Ntire 2017 challenge on single image super-resolution: Methods and results, CVPR Workshops, pg. 114–125, 2017.

[2] Assaf Shocher, Nadav Cohen, Michal Irani. zero-shot super-resolution using deep internal learning, ICCV, pg. 3118–3126, 2018.

[3] Ce Liu, Deqing Sun, On bayesian adaptive video super resolution, TPAMI, 2014.

[4] Seungjun Nah, Radu Timofte, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Kyoung Mu Lee, NTIRE 2019 Challenge on Video Deblurring: Methods and Results, CVPR Workshops, 2019.

[5] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, MingHsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution, CVPR, pg. 5835–5843, 2017.

[6] Tobias Plotz, Stefan Roth, Benchmarking denoising algorithms with real photographs, CVPR, pg. 1586–1595, 2017.

[7] Dario Fuoli, Shuhang Gu, Radu Timofte, Efficient video super-resolution through recurrent latent space propagation, ICCVW, 2019.

[8] Yan Huang, Wei Wang, Liang Wang, Video super- resolution via bidirectional recurrent convolutional net- works, TPAMI, 2018.

[9] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, Qi Tian, Video super-resolution with recurrent structure-detail network, ECCV, 2020.

[10] Jun Guo, Hongyang Chao, Building an end-to-end spatial-temporal convolutional network for video super-resolution, AAAI, pg. 4053–4060,2017.

[11] Xiaobin Zhu, Zhuangzi Li, Xiao-Yu Zhang,Changsheng Li, Yaqi Liu, Ziyu Xue, Residual invertible spatio-temporal network for video super-resolution, AAAI, pg. 5981–5988, 2019.

[12] Mehdi S M Sajjadi, Raviteja Vemulapalli, Matthew Brown, Frame-recurrent video super-resolution, CVPR, 2018.

[13] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, Chen Change Loy, BasicVSR: The search for essential com- ponents in video super-resolution and beyond, CVPR, 2021.

[14] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, Tao Lu, Xin Tian, Jiayi Ma, Omniscient Video Super-Resolution, ICCV, pg. 4429-4438, 2021.

[15] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, William T Freeman. Video enhancement with task-oriented flow, IJCV, 2019.

[16] Jose Caballero, Christian Ledig, Aitken Andrew, Acosta Alejandro, Johannes Totz, Zehan Wang, Wenzhe Shi, Realtime video super-resolution with spatio-temporal networks and motion compensation, CVPR, 2017.

[17] Muhammad Haris, Greg Shakhnarovich, Norimichi Ukita, Recurrent back-projection network for video super- resolution, CVPR, 2019.

[18] Jingwei Xin, Nannan Wang, Jie Li, Xinbo Gao, Zhifeng Li, Video face super-resolution with motion-adaptive feedback cell. AAAI, pg.468–475, 2020.

[19] Muhammad Haris, Greg Shakhnarovich, Norimichi Ukita, Space-time-aware multi-resolution video enhancement, CVPR, pg. 2859–2868, 2020.

[20] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, Chen Change Loy, EDVR: Video restoration with enhanced deformable convolutional networks, CVPR Workshop, 2019.

[21] Hua Wang, Dewei Su, Chuangchuang Liu, Longcun Jin, Xianfang Sun, Xinyi Peng, Deformable non-local network for video super-resolution. IEEE Access, 2019.

[22] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, CVPR, pg. 1874–1883, 2016.

[23] Yapeng Tian, Yulun Zhang, Yun Fu, Chenliang Xu, TDAN: Temporally deformable alignment network for video super-resolution, CVPR, 2020.

[24] Ilya Loshchilov, Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. arXiv:1608.03983, 2016.

[25] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei, Deformable convolutional networks, ICCV, pg. 764–773, 2017.

[26] Xizhou Zhu, Han Hu, Stephen Lin, Jifeng Dai,Deformable ConvNets v2: More Deformable, Better Results, Deformable ConvNets V2: More deformable, better results, CVPR, pg. 9300–9308, 2019.

[27] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, FlowNet: Learning optical flow with convolutional networks, ICCV, pg. 2758–2766, 2015.

[28] Bruce D. Lucas, Takeo Kanade, An iterative image registration technique with an application to stereo vision, IJCAI, pg. 674–679, 1981.

[29] Marius Drulea, Sergiu Nedevschi, Total variation regularization of local-global optical flow, ITSC, pg. 318–323, 2011

[30] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, Thomas Brox, FlowNet 2.0: Evolution of optical flow estimation with deep networks, CVPR, 2017, pg. 1647–1655, 2017.

[31] Anurag Ranjan, Michael J. Black, Optical flow estimation using a spatial pyramid network, CVPR, pg. 2720–2729, 2017

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, Jan Kautz, PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume, CVPR, pg. 8934–8943, 2018.

[33] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging, ICIP, 1994.

[34] Diederik Kingma, Jimmy Ba. Adam: A method for stochastic optimization, ICLR, 2015.

Author Biography

Tony Nokap Park received his Ph.D. in computer engineering from Seoul National University in 2008. From 2008 to 2015, he has worked in the Corporate Research Center at Samsung SDI where his work has focused on automated industrial inspection of defects and automated prediction of battery lifecycle and performance. In 2016, he joined the T3K at SK Telecom where he worked on modeling natural language processing models for chatbots and then video processing models for super-resolution, denoising, and inpainting.

Taeyoung Na received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2014. He was with LG Electronics, Osan, Korea, as a Research Engineer, where he developed audio and speakers from 2004 to 2006. In 2014, he joined Samsung Electronics, Suwon, Korea, as a Senior Engineer, where he developed various video coding methods for smart devices. Since 2017, he has been with SK Telecom, where he has led SUPERNOVA Development Team. His current research interests include video qualtiy enhancement with deep learning methods.