VIDEO FRAME INTERPOLATION VIA RESIDUE REFINEMENT

Haopeng Li, Yuan Yuan, Qi Wang*

School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P.R. China

ABSTRACT

Video frame interpolation achieves temporal super-resolution by generating smooth transitions between frames. Although great success has been achieved by deep neural networks, the synthesized images stills suffer from poor visual appearance and unsatisfied artifacts. In this paper, we propose a novel network structure that leverages residue refinement and adaptive weight to synthesize in-between frames. The residue refinement technique is used for optical flow and image generation for higher accuracy and better visual appearance, while the adaptive weight map combines the forward and backward warped frames to reduce the artifacts. Moreover, all submodules in our method are implemented by U-Net with less depths, so the efficiency is guaranteed. Experiments on public datasets demonstrate the effectiveness and superiority of our method over the state-of-the-art approaches.

***Index Terms*—** Video frame interpolation, residue refinement, adaptive weight map, U-Net

1. INTRODUCTION

Video frame interpolation achieves video temporal super-resolution by generating smooth transitions between two consecutive frames [1, 2, 3, 4, 5]. It compensates the motion information and enriches changing details in videos so that the visual appearance is significantly improved. Because of its wide applications such as virtual view synthesis [6], frame rate up-conversion [7] and slow motion generation [8], video frame interpolation has been studied by researchers.

Learning-based methods dominate this field owning to the strong representation ability of deep neural networks [3, 8, 9, 10, 11]. The deep models aim to learn the mapping from two adjacent frames to the intermediate frame by iteratively adjusting the their parameters. The core of learning-based methods is to design appropriate network structure and reasonable loss function. The network is required to extract motion information and to synthesize images that contain more motion detains. The loss function measures the difference between the estimated frame and ground truth one for back-propagation.

*Corresponding author. This work was supported by the National Natural Science Foundation of China under Grant 61632018, 61825603, U1864204 and 61773316.

Despite the great success of deep convolutional neural networks in video frame interpolation, there are still a few limitations. 1) The generated frames suffer from artifacts such as blur and ghost effect. Those artifacts are caused by inaccurate motion estimation and less robust image fusion. 2) The deep neural networks have large models size and the computation is complex. Thus it is difficult to deploy the models to devices with less storage and computation ability.

To address those above limitations, we propose a deep neural network (RRIN) that exploits residue learning [12] and the adaptive weight map for accurate video frame interpolation. Residue learning is widely utilized in image synthesis tasks such as image super-resolution [13, 14], style transfer [14], face generation [15], etc. It shows great power of synthesizing photo-realistic images, and thus researchers are inspired to apply it to video frame interpolation [9, 10]. However, previous works use residue modules to extract features from given frames and warp the features to obtain in-between frames. In this work, residue learning is used for the refinement of estimated optical flow and warped frames. Such practice not only improves the accuracy of flow maps and visual appearance of images, but also reduces the difficulty of training by adding skip connections at flow or image level in networks. Besides, we introduce the adaptive weight map to combine the forward-warped frame and the backward-warped one. The learned map contains the information of lighting change and occlusion which widely exists in video and would cause unsatisfied artifacts. By leveraging the adaptive weight map, the warped frames are fused in a proper fashion so that the results are more visually appealing. Moreover, all submodules in RRIN are implemented by generalized U-Net [16] with less depths, which reduces the model size and computation complexity considerably. The contributions of this paper are summarized as follows:

- We present an effective and efficient deep convolutional network for video frame interpolation with compact structure. The proposed method outperforms the state-of-the-art approaches on Vimeo90K with less parameters and inference time.
- We propose to use residue learning for the refinement of optical flow and warped frames instead of feature extraction. Flow-level residue learning increases the ac-

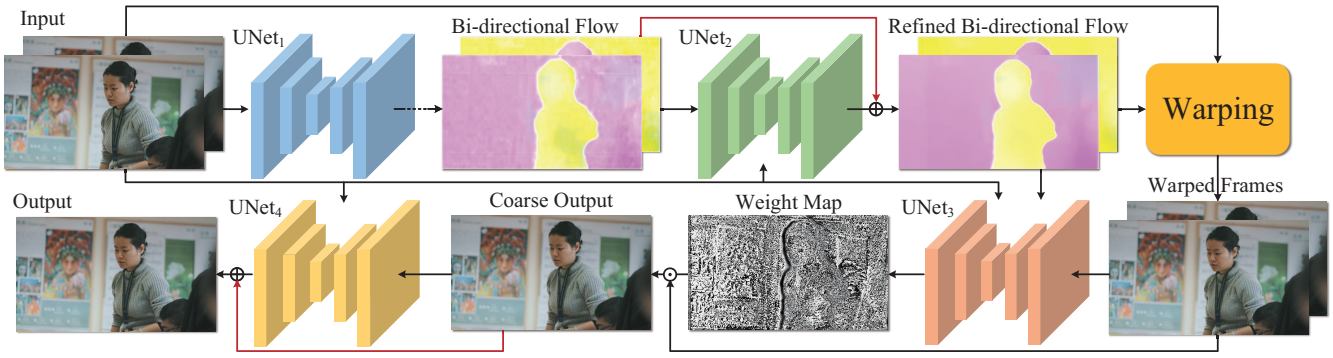

Fig. 1. Overview of the proposed RRIN. Given two consecutive frames in a video, RRIN first estimates the bi-directional optical flow. Then the arbitrary-time flow is refined by residue learning. We use the refined flow to warp the input frames forward and backward. So two approximations of the in-between frame are obtained. We linearly combined the two warped frames and the weight map is learned by a U-Net. After the coarse estimation is computed, RRIN uses residue learning to have it refined and outputs the final results.

curacy of optical flow while image-level residue learning improves the visual appearance of the results.

- We design the adaptive weight map to combine the forward and backward warped frames. By involving the information of lighting change and occlusion in the weight map, the warped frames are fused in an adaptive fashion, which alleviates the unsatisfied artifacts.

2. THE PROPOSED METHOD

2.1. Problem Definition

Given two consecutive frames $\boldsymbol{I}^0, \boldsymbol{I}^1 \in \mathbb{R}^{H\times W\times 3}$ (H, W are the height and width respectively), the aim of video frame interpolation is to generate a smooth transition $\hat{\boldsymbol{I}}^t$ ($t \in (0,1)$) that compensates the motion details in the video. If the ground truth in-between frame is denoted as $\boldsymbol{I}^t$, the task it to find a mapping G that takes $\boldsymbol{I}^0, \boldsymbol{I}^1$ and the time factor t as inputs and output the estimated $\hat{\boldsymbol{I}}^t$, that is to say,

$$G(\boldsymbol{I}^0, \boldsymbol{I}^1, t) = \hat{\boldsymbol{I}}^t \rightarrow \boldsymbol{I}^t. \tag{1}$$

In this paper, we propose to use deep neural network to achieve video frame interpolation. The ultimate goal is to train a network $G_{\boldsymbol{\theta}}$ that is parameterized by $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is learned by minimizing a certain loss function $\mathcal{L}$, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(G_{\boldsymbol{\theta}}(\boldsymbol{I}_i^0, \boldsymbol{I}_i^1, t), \boldsymbol{I}_i^t), \tag{2}$$

where $\{(\boldsymbol{I}_i^0, \boldsymbol{I}_i^1), \boldsymbol{I}_i^t\}_{i=1}^N$ is the set of training samples. In this work, we design a deep neural network specifically for video frame interpolation, which is described next.

2.2. Video Frame Interpolation via Residue Refinement

In this section, we elaborate the proposed video frame interpolation method (RRIN) which utilizes residue learning to estimate accurate in-between frames given two adjacent frames in a video. RRIN is a flow-based method and follows the typical procedure: motion estimation, frame warping [17] and post-processing. The overview of RRIN is shown in Fig. 1.

Given two consecutive frames $\boldsymbol{I}^0, \boldsymbol{I}^1$, RRIN first computes the bi-directional optical flow using a U-Net, i.e.,

$$\boldsymbol{F}_{0\rightarrow 1}, \boldsymbol{F}_{1\rightarrow 0} = \mathrm{UNet}_1(\boldsymbol{I}^0, \boldsymbol{I}^1). \tag{3}$$

Following [8], we estimate arbitrary-time optical flow by bi-directional interpolation, i.e.,

$$\hat{\boldsymbol{F}}_{t\rightarrow 0} = -(1-t)t\boldsymbol{F}_{0\rightarrow 1} + t^2\boldsymbol{F}_{1\rightarrow 0}, \tag{4}$$

$$\hat{\boldsymbol{F}}_{t\rightarrow 1} = (1-t)^2\boldsymbol{F}_{0\rightarrow 1} - t(1-t)\boldsymbol{F}_{1\rightarrow 0}. \tag{5}$$

However, this estimation works poorly around motion boundaries because the flow is not locally smooth in those regions [8]. To improve the accuracy of the estimated arbitrary-time optical flow, we propose to have them refined by residue learning. Specifically, $\boldsymbol{I}^0, \boldsymbol{I}^1, \hat{\boldsymbol{F}}_{t\rightarrow 0}, \hat{\boldsymbol{F}}_{t\rightarrow 1}$ are firstly concatenated along channel dimension, then they are sent to a U-Net to learn the residues of the arbitrary-time optical flow, i.e.,

$$\tilde{\boldsymbol{F}}_{t\rightarrow 0}, \tilde{\boldsymbol{F}}_{t\rightarrow 0} = \mathrm{UNet}_2(\boldsymbol{I}^0, \boldsymbol{I}^1, \hat{\boldsymbol{F}}_{t\rightarrow 0}, \hat{\boldsymbol{F}}_{t\rightarrow 1}). \tag{6}$$

Once the residues are obtained, the refined bi-directional arbitrary-time optical flow is computed as follows,

$$\boldsymbol{F}_{t\rightarrow 0} = \hat{\boldsymbol{F}}_{t\rightarrow 0} + \tilde{\boldsymbol{F}}_{t\rightarrow 0}, \tag{7}$$

$$\boldsymbol{F}_{t\rightarrow 1} = \hat{\boldsymbol{F}}_{t\rightarrow 1} + \tilde{\boldsymbol{F}}_{t\rightarrow 1}. \tag{8}$$

After the refined bi-directional optical flow is obtained, we warp [17] the input frames and generate two estimated

intermediate frames, i.e.,

$$\hat{\boldsymbol{I}}^{0\to t} = \mathrm{Warp}(\boldsymbol{I}^0, \boldsymbol{F}_{t\to 0}), \tag{9}$$

$$\hat{\boldsymbol{I}}^{1\to t} = \mathrm{Warp}(\boldsymbol{I}^1, \boldsymbol{F}_{t\to 1}). \tag{10}$$

In this paper, we propose to linearly combine the above estimated intermediate frames to obtain the coarse output. Specifically,

$$\hat{\boldsymbol{I}}_c^t = \frac{1}{1+\boldsymbol{\alpha}(t)} \odot \hat{\boldsymbol{I}}^{0\to t} + \frac{1}{1+\boldsymbol{\beta}(t)} \odot \hat{\boldsymbol{I}}^{1\to t}, \tag{11}$$

where $\boldsymbol{\alpha}(t) \odot \boldsymbol{\beta}(t) = \mathbf{1} \in \mathbb{R}^{H\times W}$ to satisfy that the sum of coefficients equals to 1, and $\odot$ is the element-wise multiplication operation. We only elaborate the design of $\boldsymbol{\alpha}(t)$ because $\boldsymbol{\beta}(t)$ has the same properties. Trivially, the reliability of $\hat{\boldsymbol{I}}^{0\to t}$ is positive correlated to $1-t$ and negative correlated to t, so we define $\boldsymbol{\alpha}(t)$ as follows,

$$\boldsymbol{\alpha}(t) = \frac{t}{1-t}\boldsymbol{M}_t, \tag{12}$$

where $\boldsymbol{M}_t \in \mathbb{R}^{H\times W}$ is the learned pixel-wise weight map for $\hat{\boldsymbol{I}}^{0\to t}$. In this case,

$$\boldsymbol{\beta}(t) = \frac{1-t}{t}\frac{1}{\boldsymbol{M}_t}. \tag{13}$$

The weight map is essential because it contains the information about lighting change and occlusion, which would have great impact on the interpolation performance. So we propose to leverage a U-Net to learn it in an end-to-end fashion. Instead of outputting $\boldsymbol{M}_t$ directly, the U-Net outputs two feature maps, and we use the pixel-wise ratio of them (after Sigmoid activation) as the final weight map. That is to say,

$$\boldsymbol{M}_0, \boldsymbol{M}_1 = \mathrm{UNet}_3(\boldsymbol{I}^0, \boldsymbol{I}^1, \hat{\boldsymbol{I}}^{0\to t}, \hat{\boldsymbol{I}}^{1\to t}, \boldsymbol{F}_{t\to 0}, \boldsymbol{F}_{t\to 1}), \tag{14}$$

$$\boldsymbol{M}_t = \frac{\sigma(\boldsymbol{M}_0)}{\sigma(\boldsymbol{M}_1)}. \tag{15}$$

The purpose for such practice is to improve the representation ability and the numerical stability during training and testing.

Although $\hat{\boldsymbol{I}}_c^t$ is a fine estimation of $\boldsymbol{I}^t$, it has the limitation of losing image details such as texture and clear edges. To overcome this shortcoming, we propose to have $\hat{\boldsymbol{I}}_c^t$ refined by residue learning. Another U-Net is utilized to learn the residue of the final output, i.e.,

$$\tilde{\boldsymbol{I}}^t = \mathrm{UNet}_4(\boldsymbol{I}^0, \boldsymbol{I}^1, \hat{\boldsymbol{I}}_c^t), \tag{16}$$

$$\hat{\boldsymbol{I}}^t = \hat{\boldsymbol{I}}_c^t + \tilde{\boldsymbol{I}}^t. \tag{17}$$

RRIN has four U-Nets, but they have different depths in consideration of different difficulties and model complexity. The details of each U-Net are shown in Table 1.

Sub-module	In Channel	Out Channel	Depth	#Parameters
UNet_1	6	4	5	10.99
UNet_2	10	4	4	2.73
UNet_3	16	2	4	2.74
UNet_4	9	3	4	2.73

Table 1. The details of each U-Net in RRIN, including the channel number of input, the channel number of output, depth and the number of parameters (given in million).

3. EXPERIMENTS

3.1. Experiment Setups

3.1.1. Training Set

The Vimeo90K dataset [18] is utilized as the training set. The Vimeo90K training set contains 51,312 triplets, each of which has 3 consecutive frames in a video. The resolution of all frames is 256×448. The first and third frames in the triplets are used as the inputs, and the second frame are considered as the ground truth in-between frame at $t = 0.5$. Besides, we use vertical/horizontal flipping and temporal reversion of the triplets to augment the training samples.

3.1.2. Loss Function

Pixel-wise loss functions such as MSE loss and l_1 loss are widely used to train image synthesis models. In this work, Charbonnier penalty function [10] is utilized as the loss function for its global smoothness and robustness to outliers. Specifically, the loss is calculated as

$$\mathcal{L}(\hat{\boldsymbol{I}}^t, \boldsymbol{I}^t) = \frac{1}{|P|}\sum_{\boldsymbol{p}\in P}\sqrt{(\hat{\boldsymbol{I}}^t(\boldsymbol{p}) - \boldsymbol{I}^t(\boldsymbol{p}))^2 + \varepsilon^2}, \tag{18}$$

where $P = [1, H] \times [1, W] \times [1, 3]$, and ε is set to 1×10^{-6} empirically during training.

3.1.3. Training Process

Mini-batch Adam algorithm [19] is adopted to train our model. The batch size is 4 for stable convergence and $\boldsymbol{\beta} = (0.9, 0.999)$. The initial learning rate is set to 1×10^{-4} and is decreased to 2×10^{-5} after 60 epochs, then remains constant for another 40 epochs. The whole process is implemented by PyTorch on NVIDIA GeForce 1080Ti.

3.1.4. Evaluation Metrics and Testing Sets

The proposed method is evaluated by two widely-used metrics: signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [20]. Note that we use *compare_ssim* in skimage library to match the implementation of SSIM in [20].

Method	Vimeo90K		UCF101		#Parameters	Runtime
	PSNR	SSIM	PSNR	SSIM	(millions)	(seconds)
DVF [1]	31.54	0.9320	34.12	0.9414	1.60	0.47
SepConv-L_1 [3]	33.80	0.9554	34.79	0.9476	21.60	0.20
SepConv-L_f [3]	33.45	0.9511	34.69	0.9452	21.60	0.20
Super Slomo [8]	33.44	0.9499	34.75	0.9474	39.61	0.09
MEMC-Net [9]	34.29	0.9622	34.96	0.9497	70.31	0.12
DAIN [10]	34.72	0.9641	34.99	0.9498	24.02	0.13
RRIN (Ours)	35.22	0.9643	34.93	0.9496	19.19	0.08

Table 2. Quantitative comparisons on Vimeo90K and UCF101. The number of model parameters (millions) and runtime (seconds per 480×640 image) are also shown.

Model	Vimeo90K		UCF101	
	PSNR	SSIM	PSNR	SSIM
w/o-RR	34.84	0.9625	34.58	0.9488
w/o-WM	34.90	0.9615	34.83	0.9486
RRIN	35.22	0.9643	34.93	0.9496

Table 3. Effectiveness of different components of our model.

We use two testing sets to qualitatively compare the proposed method with previous ones: Vimeo90K [18] and UCF101[21, 8]. The Vimeo90K testing set contains 3,782 triplets with the resolution of 256×448. The UCF101 dataset contains 397 triplets with resolution of 256×256.

3.2. Main Results

We compare RRIN with several previous video frame interpolation methods including DVF [1], SepConv [3], Super SloMo [8], MEMC-Net [9], DAIN [10]. DVF and Super SloMo are typical flow-based methods which model motions as pixel displacements. SepConv is a kernel-based method that considers motions as local convolution. MEMC-Net integrates optical flow and interpolation kernels to adaptively warp frames. DAIN uses depth information to detect occlusion and fuses optical flow, interpolation kernels and contextual features into a compact model, which is the state-of-the-art method of video frame interpolation.

As shown in Table 2, RRIN achieves the best PSNR and SSIM on Vimeo90K dataset. Note that there is a significant improvement in PSNR (0.5dB) compared to the state-of-the-art method. As for the results on UCF101 dataset, our method is comparable to MEMC-Net and DAIN (with slightly drop of performance). But MEMC-Net has much more parameters than RRIN does and we use 73% fewer parameters. Moreover, different from DAIN, RRIN does not adopt any pre-trained network for depth or flow estimation. In addition, RRIN still outperforms other flow-based and kernel-based methods significantly on UCF101. Besides, our method costs less time than others, which also indicates that RRIN is more suitable for lightweight devices. Fig. 2 shows interpolated results in UCF101 dataset by different methods.

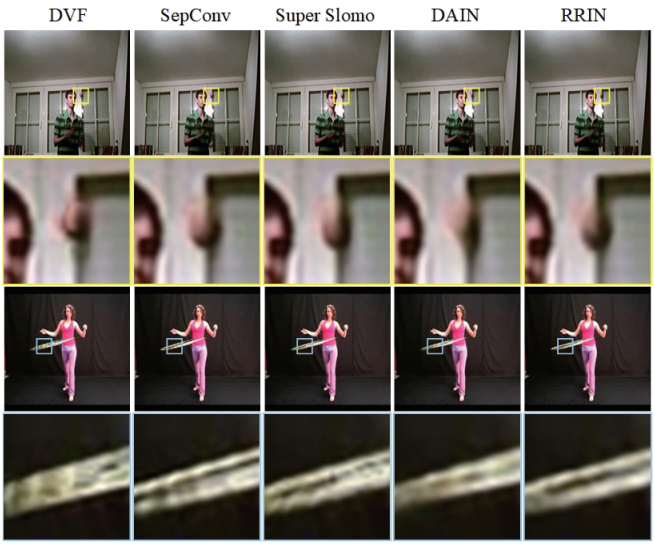

Fig. 2. Visual comparisons on the UCF101 dataset. RRIN is capable of restoring the contents (the ball and hoop).

3.3. Ablation Study

In this part we conduct ablation study to demonstrate the effectiveness of residue refinement and adaptive weight map. Two more models are constructed in this experiment: w/o-RR and w/o-WM. w/o-RR is a simplification of RRIN that omits the residue connections, i.e., the red lines in Fig 2. w/o-WM. is another simplification of RRIN that abandons the adaptive weight map and directly uses the average of warped frames as the coarse output. The results are shown in Table 3.

As shown in Table 3, RRIN outperforms the two simplification models, which indicates the influence of residue refinement and adaptive weight map. But two modules have different impact on PSNR and SSIM. Residue refinement shows the power in improving PSNR while the adaptive weight map is capable of increasing SSIM.

4. CONCLUSION

In this paper, we propose RRIN that uses residue refinement and adaptive weight map for video frame interpolation. The residue learning for the refinement of optical flow increases the accuracy while the refinement of course outputs improves the visual appearance. The learned weight map combines the forward and backward warped frames in consideration of lighting change and occlusion and thus reduces the artifacts. Besides, all sub-modules are implemented by U-Net with less depths, which guarantees the efficiency of RRIN. Experiments on two datasets demonstrate the effectiveness and superiority compared with the state-of-the-art approaches.

5. REFERENCES

[1] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala, "Video frame synthesis using deep voxel flow," in *IEEE International Conference on Computer Vision*, 2017, pp. 4473–4481.

[2] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung, "Phase-based frame interpolation for video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1410–1418.

[3] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.

[4] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations*, 2016.

[5] Tanaphol Thaipanich, Ping-Hao Wu, and C.-C. Jay Kuo, "Low complexity algorithm for robust video frame rate up-conversion (FRUC) technique," *IEEE Trans. Consumer Electronics*, vol. 55, no. 1, pp. 220–228, 2009.

[6] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5515–5524.

[7] Wenbo Bao, Xiaoyun Zhang, Li Chen, Lianghui Ding, and Zhiyong Gao, "High-order model and dynamic filtering for frame rate up-conversion," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3813–3826, 2018.

[8] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008.

[9] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang, "Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *CoRR*, vol. abs/1810.08768, 2018.

[10] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712.

[11] Haopeng Li, Yuan Yuan, and Qi Wang, "Fi-net: A lightweight video frame interpolation network using feature-level flow," *IEEE Access*, vol. 7, pp. 118287–118296, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[15] Guli Zhang Ziheng Zhang Kenny Mitchell Anpei Chen, Zhang Chen and Jingyi Yu, "Photo-realistic facial details synthesis from single image," *arXiv preprint arXiv:1903.10873*, 2019.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[18] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman, "Video enhancement with task-oriented flow," *CoRR*, vol. abs/1711.09078, 2017.

[19] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[20] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[21] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.