Revitalizing Legacy Video Content: Deinterlacing with Bidirectional Information Propagation

Zhaowei Gao[*1, 2]
zhagao@student.ethz.ch
Mingyang Song[*1, 2]
misong@student.ethz.ch
Christopher Schroers[2]
christopher.schroers@disneyresearch.com
Yang Zhang[2]
yang.zhang@disneyresearch.com

[1] ETH Zürich
Zürich, Switzerland
[2] Disney Research|Studios
Zurich, Switzerland

Abstract

Due to old CRT display technology and limited transmission bandwidth, early film and TV broadcasts commonly used interlaced scanning. This meant each field contained only half of the information. Since modern displays require full frames, this has spurred research into deinterlacing, i.e. restoring the missing information in legacy video content. In this paper, we present a deep-learning-based method for deinterlacing animated and live-action content. Our proposed method supports bidirectional spatio-temporal information propagation across multiple scales to leverage information in both space and time. More specifically, we design a Flow-guided Refinement Block (FRB) which performs feature refinement including alignment, fusion, and rectification. Additionally, our method can process multiple fields simultaneously, reducing per-frame processing time, and potentially enabling real-time processing. Our experimental results demonstrate that our proposed method achieves superior performance compared to existing methods.

1 Introduction

Interlaced video was developed in the early days of television to balance visual quality and technical constraints within limited bandwidth and refresh rates. It captured odd and even fields in alternating frames, combining them into interlaced frames for displaying on screens. While interlacing was once a useful technique, modern displays require progressive video, making interlaced formats obsolete. However, in the past, when interlacing videos, the original frames usually were not preserved. Consequently, deinterlacing has become crucial for the restoration of old interlaced content.

Deinterlacing involves estimating the content of absent lines within each field of an interlaced video signal, aiming to generate the complete frame information while ensuring visual

[*] These authors contributed equally to this work

quality and minimizing artifacts. A large variety of deinterlacing algorithms exists: Conventional Deinterlacing methods [8, 10, 24] can be categorized into intra-field interpolation, inter-field interpolation, and motion-based. Intra-field interpolation reconstructs the missing field by averaging pixel values available from the current field. Inter-field interpolation replicates information from neighboring fields to approximate the missing field. The outcome of such methods is generally unsatisfactory due to the simplicity of replicating and averaging pixel values. Despite involving motion detection and alignment, conventional motion-based methods are still insufficient in capturing accurate inter-frame correspondences. Fortunately, deinterlacing is perfectly suited for fully supervised training since the degradation process induced through interlacing is well-defined. This allowed to harness the expressive power of neural networks and to significantly surpass the previously available handcrafted reconstruction strategies across diverse input data [17, 31, 33, 34].

Sharing a similar goal of restoring missing information from observations, video super resolution[4, 5, 28], video frame interpolation[20], as well as image and video restoration [6, 14, 27] can offer valuable insights for video deinterlacing, especially when it comes to devising strategies for temporal propagation, alignment and fusion.

In order to make the most effective use of both spatial and temporal information in interlaced videos, we propose a Flow-guided Refinement Block (FRB). Opposed to [5], we introduce an additional fusion mechanism after the deformable convolutions. While [6] employs recurrent propagation, we leverage bidirectional parallel propagation [14] on each scale level. The main contributions of our work are:

- We propose a deep learning framework for deinterlacing that incorporates a mechanism for the propagation of temporal information in both image and latent space, as well as feature refinement. Our framework effectively tackles the restoration of interlacing artifacts, including combing and aliasing.
- Our model is lightweight and capable of simultaneously outputting six deinterlaced video frames which makes it a promising candidate for real-time applications.
- Our extensive experimental results demonstrate that our proposed method can remove complex interlacing artifacts and achieve state-of-the-art performance.

2 Related Work

2.1 Conventional Deinterlacing

Video deinterlacing in computer vision presents classic challenges, with methods falling into three categories: intra-field, inter-field, and motion-based. Intra-field techniques reconstruct frames independently but suffer from lower quality due to simplistic averaging [9]. Efforts to improve edge processing include bilateral filtering [26] locality and similarity adaption [24], and moving least square methods [25]. Inter-field methods aim to enhance quality by integrating temporal information but often yield unsatisfactory outcomes [10, 13]. Motion-based methods require accurate compensation, posing challenges with significant motion, resulting in visible artifacts [12, 18].

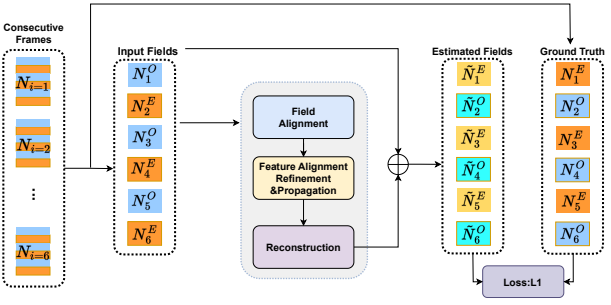

Figure 1: Overview of data processing during training.

2.2 Deep Learning-based Deinterlacing

Several deep learning-based deinterlacing networks have emerged in the past years, such as DICNN by Zhu et al. [34], emphasizing real-time processing, Liu et al. used deformable convolution and attention-residual blocks [17] and Zhao et al. used a two-stage ResNet Structure to deal with complex interlacing artifacts [33]. However, these approaches mainly focus on intra-frame deinterlacing and overlook temporal information. Bernasconi et al. [2] proposed a multi-field deinterlacing method, while VDNet [31] introduced an RNN-based framework but struggled with feature-domain alignment and handling artifacts with large motion. Meanwhile, video deinterlacing can be viewed as a type of video upscaling task, akin to vertical upscaling by a factor of 2. Existing methods rely on optical flow estimation [3], but alternatives like TDAN [23], EDVR [27], VFIT [20], BasicVSR++ [5], TMNet [29] and VRT [15] eliminate the need for motion estimation, utilizing deformable convolution for alignment and improving spatio-temporal upscaling techniques.

3 Method

3.1 Data processing pipeline

Our data processing pipeline is depicted in Fig. 1. We sample the odd or even field alternatively from 6 consecutive frames as the input to our model. The model predicts the rest of the corresponding even and odd fields and calculates the objective error during the training process. Specifically, the order of the input fields follows the role where the first field (N_1^O) from the odd-field of the first frame, then the second field (N_2^E) from the even-field from the second frame, and then alternates between odd and even for the subsequent input fields. The output is an estimation of the missing half-frame information, where the output order for the fields is even-field ($\tilde{N}_1^E$) for the first frame, odd-field ($\tilde{N}_2^O$) for the second frame, and then alternates between odd- and even-field for the subsequent frames.

3.2 The proposed method

As mentioned in Sec. 2, video processing tasks often benefit from the utilization of temporal information, however, it is also challenging. The difficulty lies in the need to aggregate information between multiple correlated frames in a video sequence that contains complex

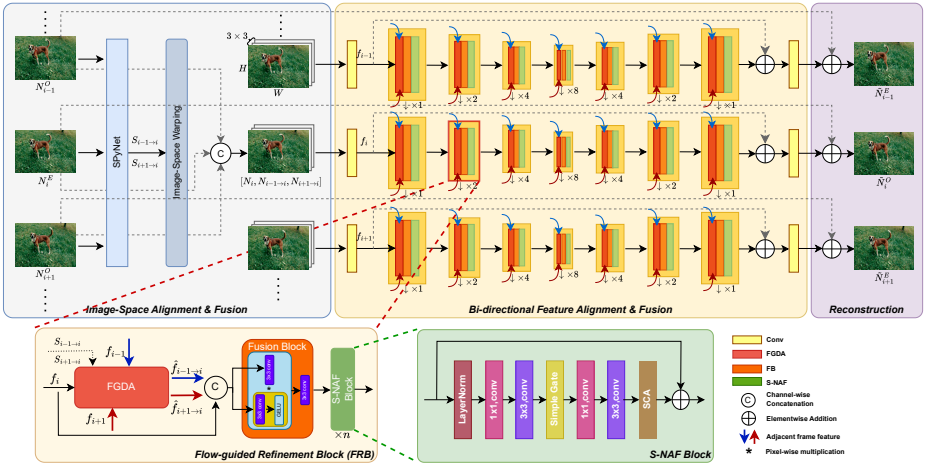

Figure 2: Overview of our deinterlacing network. We introduce forward-backward propagation to refine features bidirectionally. Specifically, within each propagation block, we introduce a Flow-guided Refinement Block (FRB). In the FRB, the FGDA block was designed to enhance offset diversity for the deformable convolution. It is followed by a Fusion block and S-NAF block to further refine the aligned features.

moving objects. Therefore, alignment and propagation of temporal sequence information become crucial.

The proposed overall architecture is shown in Fig 2. The alignment in our proposed method can be categorized into image space alignment and feature space alignment [15]. Feature alignment leverages a UNet-like structure and aligns at different scales. Building on the concept of BasicVSR++ [5], we propose a Flow-guided Refinement Block (FRB). It integrates Flow-guided Deformable Alignment [5] (FGDA) and a Fusion Block (FB) in conjunction with SimpleNAF [6, 21] (S-NAF) blocks. This helps to overcome instability during the training of Deformable Convolution Network (DCN), which can suffer from overflow issues. For information propagation, the commonly used unidirectional propagation transmits information from the first frame to the next in the video sequence. However, the information received by different frames is unbalanced. Specifically, the first frame receives no information from the video sequence except itself, whereas the last frame receives information from all the previous frames. Therefore, the later frames receive more information than earlier frames, which may result in sub-optimal outcomes and produce temporal artifacts, such as quality fluctuation over time. To address this, we developed a bidirectional information propagation scheme.

As shown in Fig 2, given an input of six consecutive fields, SPyNet [19] is first applied to estimate optical flow, S_i^k, between each pair of neighboring fields, followed by a forward and backward alignment of adjacent fields in the image domain, $N_i^{forward}$ and $N_i^{backward}$. Then the warped fields are concatenated with the input fields along the channel dimension. After that, a 3D convolutional layer is applied to extract features (g_i) from each field and warped field. In the Feature Alignment, Refinement, and Propagation (FARP) component, f_i^j from each Flow-guide Refinement Block (FRB) is then propagated under our bidirectional propagation scheme across corresponding scales, where alignment is performed by our FGDA module

and feature refinement is conducted by the FB and S-NAF modules. After propagation, the aggregated features are used to reconstruct the output image through convolutional layers.

In the following sections, a detailed description of the three components of our model will be presented respectively, including Field Alignment, Feature Alignment, Refinement & Propagation (FARP), and Reconstruction.

3.2.1 Field Alignment

We first perform alignment in the image domain. Alignment is achieved by utilizing a pre-trained SPyNet [19] to compute optical flow followed by forward and backward warping. It is worth noting that we apply spatial alignment at four different scales with corresponding optical flow. After warping and upsampling to the original scale, the original image fields N_i, and four pairs of $N^{forward}$ and $N^{backward}$ are concatenated along the channel dimension. Moreover, the four different scales of optical flows have been further utilized as inputs to the subsequent FRBs at various scales accordingly in the FARP component.

3.2.2 Feature Alignment, Refinement & Propagation (FARP)

We develop a bidirectional UNet-like scheme to facilitate refinement through propagation where the intermediate features are initially propagated independently both forward and backward in time and then down- and up-sampled and finally formed the aggregation process. Through this refinement process, the receptive field can be expanded and the information from different frames can be 'revisited' and employed for feature enhancement.

Specifically, after the field alignment, a 3D convolutional layer is applied to extract image features from the input. The features are then propagated under our bidirectional UNet-like propagation scheme in latent space, where alignment and refinement are performed in the feature domain under four various scales by our Flow-guided Refinement Block (FRB), as shown in Fig. 2.

In the following subsections, we provide a detailed explanation of the forward feature propagation in our proposed FRB module. The process for backward propagation is similarly defined.

Flow-guided Refinement Block (FRB) As shown in Fig. 2, let N_i be the input image, g_i be the feature extracted from the convolutional layer. f_i^j be the feature computed at the i-th timestep in the j-th propagation block. To compute the forward and backward feature of f_i^j, we first align f_{i+1}^{j-1} and f_{i-1}^{j-1} using the flow-guided deformable alignment (FGDA) module, respectively.

$$\hat{f}^j_{i-forward} = \text{FGDA}\left(f_i^{j-1}, f_{i+1}^{j-1}, S^k_{i\to i+1}\right), \text{ and } \hat{f}^j_{i-backward} = \text{FGDA}\left(f_i^{j-1}, f_{i-1}^{j-1}, S^k_{i\to i-1}\right), \tag{1}$$

where $S^k_{i\to i+1}$, $S^k_{i+1\to i}$ denote the optical flows at k-th scales from i-th field to the $(i+1)$-th and $(i-1)$-th field, respectively. And $f_i^0 = g_i$. The features from the current scale and from corresponding scales of adjacent fields are then concatenated and aggregated by an FB and then passed through multiple S-NAF blocks for further refinement. The S-NAF block was proposed in [21] and can make model architecture simpler and leaner. This operation can be formulated as below:

$$f_i^j = \text{S-NAF}\left(\text{FB}\left(\mathbb{C}\left(f_i^{j-1}, \hat{f}^j_{i-forward}, \hat{f}^j_{i-backward}\right)\right)\right) \tag{2}$$

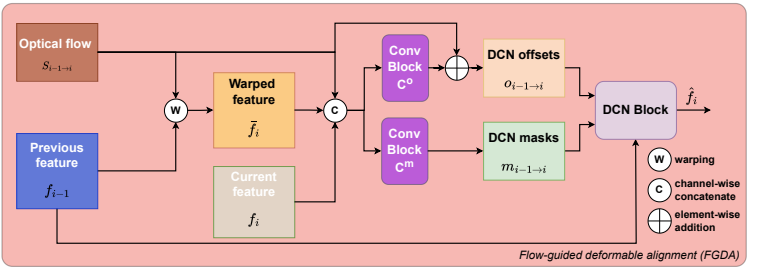

Figure 3: Illustration of the Flow-guided deformable alignment (FGDA) module.

where $\mathbb{C}$ denotes concatenation along channel dimension.

Flow-Guided Deformable Alignment (FGDA) We explain FGDA, a core component of our model which is added onto the concept of BasicVSR++ [5], for self-containing. Whereas the deformable alignment has achieved better performance over flow-based alignment, thanks to the offset diversity inherently introduced in deformable convolution (DCN)[7], the instability in vanilla DCN could lead to offset overflow, thus reducing final performance. Given the strong relation between the deformable alignment and flow-based alignment, optical flow is utilized to further guide deformable alignment, in order to fully utilize offset diversity and address the instability issue. The FGDA module has been illustrated in Fig. 3, we omit the superscript j and k in the notation, and only forward propagation has been demonstrated for simplicity.

Specifically, in Fig. 3, the current feature f_i at timestep i, the feature f_{i-1} computed from timestep $i-1$, and the optical flow $S_{i-1\to i}$ to the current field are the inputs. Firstly, f_{i-1} is forward warped with $S_{i-1\to i}$:

$$\bar{f}_i = \mathcal{W}(f_{i-1}, S_{i-1\to i}) \tag{3}$$

where $\mathcal{W}$ represents the spatial warping operation. The aligned features $\bar{f}_i$ are subsequently employed to calculate the DCN offsets $o_{i-1\to i}$ and modulation masks $m_{i-1\to i}$. Rather than directly computing the DCN offsets, the residue with respect to the optical flow is computed by Conv^O:

$$o_{i-1\to i} = S_{i-1\to i} + \text{Conv}^O\left(\mathbb{C}(f_i, \bar{f}_i, S_{i-1\to i})\right) \tag{4}$$

$$m_{i-1\to i} = \sigma\left(\text{Conv}^M\left(\mathbb{C}(f_i, \bar{f}_i, S_{i-1\to i})\right)\right) \tag{5}$$

where $\text{Conv}^{O,M}$ represents a stack of convolutional layers for o and m prediction respectively. σ denotes the sigmoid activation function. Subsequently, output feature $\hat{f}_i$ can be obtained by a DCN block with the input of feature f_{i-1}, offset $o_{i-1\to i}$ and mask $m_{i-1\to i}$.

$$\hat{f}_i = \text{DCN}(f_{i-1}, o_{i-1\to i}, m_{i-1\to i}) \tag{6}$$

The aforementioned formulations can be used for the forward propagation of a single field feature. The same process can be independently applied for backward propagation.

3.2.3 Reconstruction

Following feature refinement in FARP (Fig. 2), a 3D convolution reconstructs the predicted image's color information from the latent space. Skip connections (both latent and image

Method	Parameters (Million)	Runtime (ms)	VimeoTest		Vid4		SPMC		UDM10	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Liu-S[17]	0.52	169.33	40.45	0.9804	31.24	0.9524	36.73	0.9740	42.12	0.9872
VFIT-S[20]	0.51	46.85	40.79	0.9824	31.30	0.9541	40.89	0.9882	41.06	0.9836
DICNN-S[34]	0.54	20.35	41.42	0.9831	31.77	0.9559	40.58	0.9881	41.58	0.9844
VDNet-S†[31]	0.51	-	42.68	0.9848	32.26	0.9568	43.17	0.9907	42.48	0.9865
Ours-S	0.50	18.45	44.40	0.9906	34.20	0.9703	46.35	0.9959	44.49	0.9914
Liu-L[17]	9.12	1593.99	40.70	0.9810	30.61	0.9498	36.99	0.9749	42.27	0.9875
VFIT-L[20]	8.87	87.13	43.75	0.9891	34.07	0.9696	45.27	0.9945	43.51	0.9898
TMNet†[29]	12.44	-	45.70	0.9910	34.53	0.9698	47.26	0.9958	44.59	0.9912
VDNet-L†[31]	9.23	-	46.45	0.9922	34.83	0.9703	47.84	0.9965	45.52	0.9928
Ours-L	8.88	26.34	46.50	0.9935	35.46	0.9749	48.19	0.9972	46.20	0.9940

Table 1: Quantitative comparison (PSNR/SSIM). Red and blue colors represent the best and second-best performance, respectively. The runtime is calculated based on an image size of 256×256. †: numbers are taken from [31].

space) aid in residual learning, facilitating complex feature extraction and mitigating gradient vanishing, ultimately enhancing deinterlaced image quality.

4 Experiments

4.1 Training and Testing Datasets

We utilize datasets consisting of natural video sequences and synthesize the interlaced frames for both training and evaluation with the method mentioned in Sec. 3.1. We trained our models with the Vimeo-90K [30] training set that contains 64,612 sequences and tested our models on the remaining 7,824 testing sequences. To assess the generalization capability of our model across diverse data distribution, we utilized Vid4[16], SPMC[22], and UDM10[32] for additional testing without retraining or fine-tuning our models. Further details on training and evaluation settings can be found in supplementary materials.

4.2 Comparisons to existing methods

We compared our proposed method with existing deinterlacing and video frame interpolation methods. For the deinterlacing methods, we compared ours with VDNet[31], Liu[17] and DICNN[34]. For the video spatio-temporal upscaling methods, we choose the SOTA method TMNet[29] and VFIT[20] as the benchmark. We re-implemented the models Liu[17], DICNN[34] and trained VFIT[20], DICNN[34] and Liu[17] at two distinct parameter levels, 9 million (large) and 0.5 million (small), on Vimeo-90k train dataset[30].

As shown in Table 1, our large model, *Ours-L*, achieves state-of-the-art performance on all datasets and is the most efficient in terms of runtime and parameters. Moreover, our small model, *Ours-S*, also performed the best among all of the small models with the least amount of parameters used. Relative to DICNN's original parameter count of 0.07M, increasing the model's parameter count to 0.5M led to improved performance. However, when the parameter count was further increased to 9M, due to the simplicity of the network architecture, it may have resulted in a loss of robustness leading to unsatisfying results. Therefore, we have excluded it from the comparison.

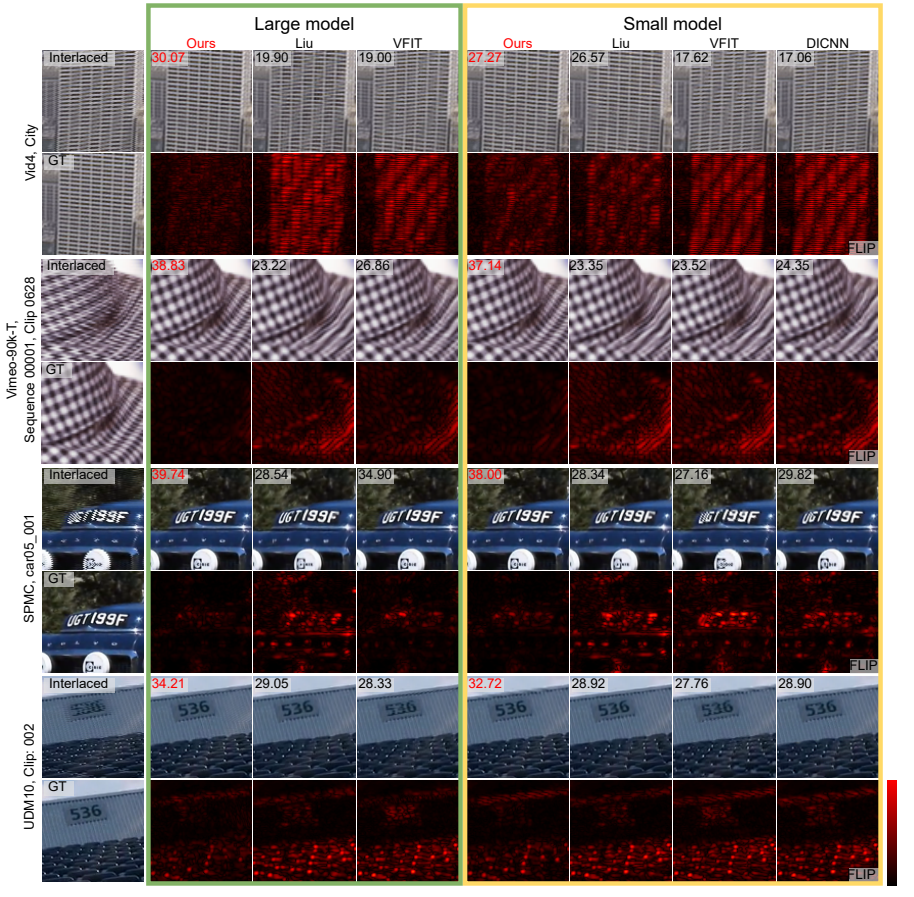

Figure 4: Visual comparisons of ours with existing deinterlacing methods. The first column shows the interlaced image and ground truth. The columns marked by green and yellow rectangles represent the results and FLIP[1] error maps from the large and small models, respectively. High intensity in FLIP maps indicates larger errors in the image. The PSNR values written in the top-left corner are computed for each crop.

4.3 Qualitative Results

In Fig. 4, we present qualitative comparisons between our approach and alternative methods. To intuitively demonstrate the discrepancy between the models' prediction and the ground truth, we visualize the pixel level FLIP [1] error maps where the brighter regions indicate more visible differences by human perception. While other approaches also have succeeded in eliminating interlaced artifacts, they often fail to handle areas with intricate textures and details. Notably, our approach consistently produces sharper results across various datasets and reduces combing and aliasing artifacts when generating deinterlaced frames compared with existing methods.

In order to demonstrate the model's performance on new types of content beyond the natural live-action content dataset, as shown in Fig. 5, our method can be generalized to animation content and consistently achieves superior performance without producing aliasing

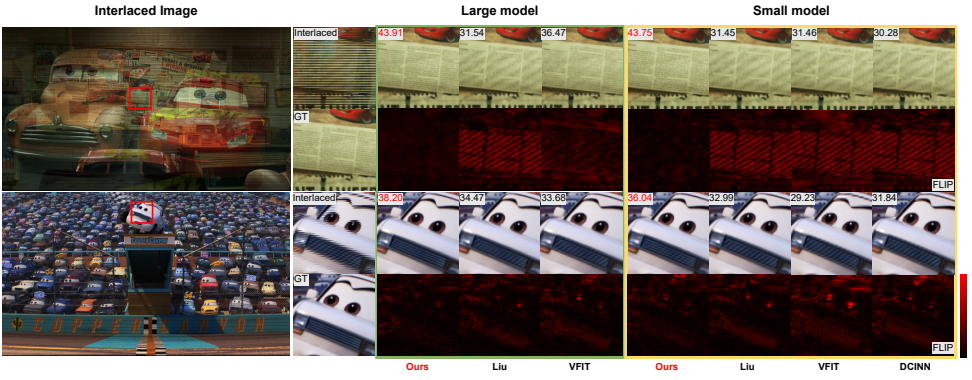

Figure 5: Visual comparisons showcase the deinterlacing results for animation content. Our method correctly restores the detail of the poster on the wall and the "nose" (intake grille) of the animated character. The PSNR values in the top-left corner are computed for each cropped region. High intensity in FLIP maps indicates larger errors in the image.

artifacts, even without fine-tuning on the animated content.

5 Ablation study

We devised several ablation studies to reason on our design and assessed the significance of each component within our network.

Impact of Image-level alignment. In our proposed method, the fields that enter the network undergo image-level alignment before proceeding to latent-level alignment, propagation, and aggregation. To attest to the necessity of temporal alignment in color space, we removed the Image-level alignment, which resulted in a slight decline across all test sets, named *w/o Image Alignment* in Table 2.

Impact of Bidirectional propagation. To motivate our bidirectional propagation approach to enlarge the temporal receptive field, we conducted a variant of our model utilizing only unidirectional propagation, labeled as *Unidirectional Propagation* in Table 2.

Impact of FGDA module. The effectiveness of feature alignment in the temporal domain has been thoroughly analyzed in [5]. To ensure the completeness of our work, we removed all the FGDA modules so that the receptive field is constrained within individual fields, and the quantitative results are shown in *w/o FGDA* in Table 2. Furthermore, as illustrated in Fig. 6, the significance of FGDA and bidirectional propagation scheme becomes more pronounced in regions that contain fine details and intricate textures.

Impact of S-NAF Block in FRB. To motivate our choice of S-NAF as basic blocks in the network, we substitute them with the conventional Conv-ReLU residual blocks, as shown in *Conv-ReLU Block* in Table 2. Our model with S-NAF offers a lighter architecture and improved performance.

	Parameters(M)	VimeoTest	Vid4	SPMC	UDM10
w/o Image Alignment	0.50	44.09	34.05	45.83	43.99
Unidirectional Propagation	0.50	43.35	33.37	44.13	44.50
w/o FGDA	0.53	41.24	32.76	41.66	42.71
Conv-ReLU Block	0.57	43.07	33.30	43.94	43.76
Our complete model	0.50	44.40	34.20	46.35	44.49

Table 2: Ablation study of the components. In each dataset, we evaluate in terms of PSNR. We conducted an ablation study on a small model (0.5M) across different datasets. To eliminate the influence of the reduced parameter count due to the absence of a specific component, we readjusted the network parameters to ensure they were all at the same parameter level, in order to ensure a fair validation of the effectiveness of each individual component.

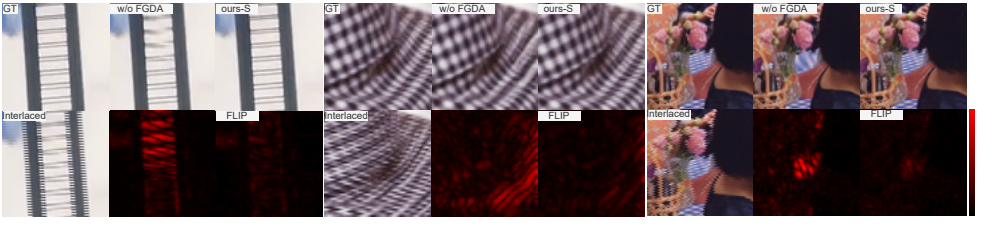

Figure 6: Visual results showcase the impact of the FGDA module in the ablation study. With the aid of the FGDA module, complex details are restored and aliasing artifacts are significantly alleviated.

6 Conclusion and future work

In this paper, we introduce a novel deep learning-based video deinterlacing framework. To the best of our knowledge, our model is the first deep learning-based deinterlacing framework that takes into account both image and feature space bidirectional alignment in conjunction with feature refinement. To address the interlacing artifacts, we first employed a pre-trained SPyNet to obtain the forward and backward optical flows at four different scales. These flows have been used for field alignment in the image space and also later in latent space. For more accurate feature information propagation, we proposed a feature refinement Block (FRB), performing bidirectional propagation and refinement across different scales to expand the receptive field while effectively enhancing the utilization of temporal information. In the reconstruction process, we employed a residual mechanism both in the latent space and image space, facilitating a more effective reconstruction of the deinterlaced image. Notably, our model was designed to be capable of concurrently processing six fields of interlaced images, which reduces the processing time significantly. Through our extensive experiments, we demonstrate that our proposed method achieves state-of-the-art results while also providing the potential for real-time deinterlacing applications.

Although the existing synthetic dataset has been effectively used in this work and may not fully represent the complexities of real-world legacy interlaced videos. Future work will focus on incorporating a broader range of real-world data to enhance the model's robustness and performance across diverse video content.

References

[1] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. ꟻLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3 (2):15:1–15:23, 2020. doi: 10.1145/3406183.

[2] Michael Bernasconi, Abdelaziz Djelouah, Sally Hattori, and Christopher Schroers. Deep deinterlacing. In *SMPTE Annual Technical Conf. Exhibition*, 2020.

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4778–4787, 2017.

[4] Kelvin C. K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond, 2021.

[5] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981, 2022.

[6] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022.

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[8] G. De Haan and E.B. Bellers. Deinterlacing-an overview. *Proceedings of the IEEE*, 86 (9):1839–1857, 1998. doi: 10.1109/5.705528.

[9] T DOYLE. Interlaced to sequential conversion for edtv applications. *2nd international workshop signal processing of HDTV*, 1998.

[10] Vinit Jakhetiya, Oscar C. Au, Sunil Jaiswal, Luheng Jia, and Hong Zhang. Fast and efficient intra-frame deinterlacing using observation model based bilateral filter. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5819–5823, 2014. doi: 10.1109/ICASSP.2014.6854719.

[11] Gwanggil Jeon, Jongmin You, and Jechang Jeong. Weighted fuzzy reasoning scheme for interlaced to progressive conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(6):842–855, 2009. doi: 10.1109/TCSVT.2009.2017309.

[12] O. Kwon, Kwanghoon Sohn, and Chulhee Lee. Deinterlacing using directional interpolation and motion compensation. *IEEE Transactions on Consumer Electronics*, 49 (1):198–203, 2003. doi: 10.1109/TCE.2003.1205477.

[13] Kwon Lee and Chulhee Lee. High quality spatially registered vertical temporal filtering for deinterlacing. *IEEE Transactions on Consumer Electronics*, 59(1):182–190, 2013. doi: 10.1109/TCE.2013.6490258.

[14] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

[15] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *IEEE Transactions on Image Processing*, 2024.

[16] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *CVPR 2011*, pages 209–216, 2011. doi: 10.1109/CVPR.2011.5995614.

[17] Yuqing Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Spatial-temporal correlation learning for real-time video deinterlacing. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[18] H Mahvash Mohammadi, Y Savaria, and JMP Langlois. Enhanced motion compensated deinterlacing algorithm. *IET Image Processing*, 6(8):1041–1048, 2012.

[19] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017.

[20] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *CVPR*, 2022.

[21] Mingyang Song, Yang Zhang, Tunç O Aydın, Elham Amin Mansour, and Christopher Schroers. A generative model for digital camera noise synthesis. *arXiv preprint arXiv:2303.09199*, 2023.

[22] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3360–3369, 2020.

[24] Jin Wang, Gwanggil Jeon, and Jechang Jeong. Efficient adaptive deinterlacing algorithm with awareness of closeness and similarity. *Optical Engineering*, 51(1):017003–017003, 2012.

[25] Jin Wang, Gwanggil Jeon, and Jechang Jeong. Moving least-squares method for interlaced to progressive scanning format conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11):1865–1872, 2013. doi: 10.1109/TCSVT.2013.2248286.

[26] Jin Wang, Zhensen Wu, and Jiaji Wu. Efficient adaptive deinterlacing algorithm using bilateral filter. *MATEC Web of Conferences*, 61:02021, 01 2016. doi: 10.1051/matecconf/20166102021.

[27] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[28] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution, 2020.

[29] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for controllable space-time video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6388–6397, 2021.

[30] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127: 1106–1125, 2019.

[31] Yin-Chen Yeh, Jilyan Dy, Tai-Ming Huang, Yung-Yao Chen, and Kai-Lung Hua. Vd-net: video deinterlacing network based on coarse adaptive module and deformable recurrent residual network. *Neural Computing and Applications*, 34(15):12861–12874, 2022.

[32] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3106–3115, 2019.

[33] Yang Zhao, Wei Jia, and Ronggang Wang. Rethinking deinterlacing for early interlaced videos, 2021.

[34] Haichao Zhu, Xueting Liu, Xiangyu Mao, and Tien-Tsin Wong. Real-time deep video deinterlacing, 2017.