applied sciences

MDPI

Article

Efficient Video Frame Interpolation Using Generative Adversarial Networks

Quang Nhat Tran and Shih-Hsuan Yang *

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; quangtn@dlu.edu.vn
* Correspondence: shyang@ntut.edu.tw

Received: 27 July 2020; Accepted: 6 September 2020; Published: 8 September 2020

Featured Application: The potential applications of frame interpolation include frame rate up-conversion, video compression, video streaming, and related video sequence processing.

Abstract: Frame interpolation, which generates an intermediate frame given adjacent ones, finds various applications such as frame rate up-conversion, video compression, and video streaming. Instead of using complex network models and additional data involved in the state-of-the-art frame interpolation methods, this paper proposes an approach based on an end-to-end generative adversarial network. A combined loss function is employed, which jointly considers the adversarial loss (difference between data models), reconstruction loss, and motion blur degradation. The objective image quality metric values reach a PSNR of 29.22 dB and SSIM of 0.835 on the UCF101 dataset, similar to those of the state-of-the-art approach. The good visual quality is notably achieved by approximately one-fifth computational time, which entails possible real-time frame rate up-conversion. The interpolated output can be further improved by a GAN based refinement network that better maintains motion and color by image-to-image translation.

Keywords: deep learning; generative adversarial network (GAN); frame interpolation

1. Introduction

Frame interpolation is a technique that generates intermediate frames between the adjacent ones in a video sequence. In the dyadic case, given two frames x_{n-1} and x_{n+1}, the frame interpolation process predicts the frame x_n and consequently doubles the frame rate of the output sequence. Frame interpolation techniques based on deep learning has recently been considered [1–3]. The popular network models include the convolution neural network (CNN) and generative adversarial network (GAN), typically under the supervised learning framework.

A GAN comprises two adversarial parts, a generator and a discriminator. The generator manages to generate a plausible output while the discriminator manages to distinguish the real data from that fake output. GANs have shown promising results in various image processing applications including frame prediction [4–6]. Most of the existing GAN based frame prediction methods use complicated models in which each of them takes a different responsibility such as generation or rectification [7,8]. Those methods also require auxiliary data and long training time.

This study aims for an efficient GAN framework that produces high-quality interpolated frames without intermediate data and complex network architecture. Motivated by the Deep Convolutional Generative Adversarial Network (DCGAN) [4], one of the most popular GAN frameworks for random image generation, we propose using a semi-supervised frame interpolation framework (Figure 1). The generator encodes a pair of frames to produce the generated output, and the discriminator applies several down-sampling operations on an input frame for classification. Furthermore, the generated output can undergo a refinement network that adopts the image-to-image translation to improve motion and color of the generated output. Besides, a loss function that jointly combines the adversarial loss and the reconstruction loss for pixel/structure-based similarity and motion blur degradation is also proposed.

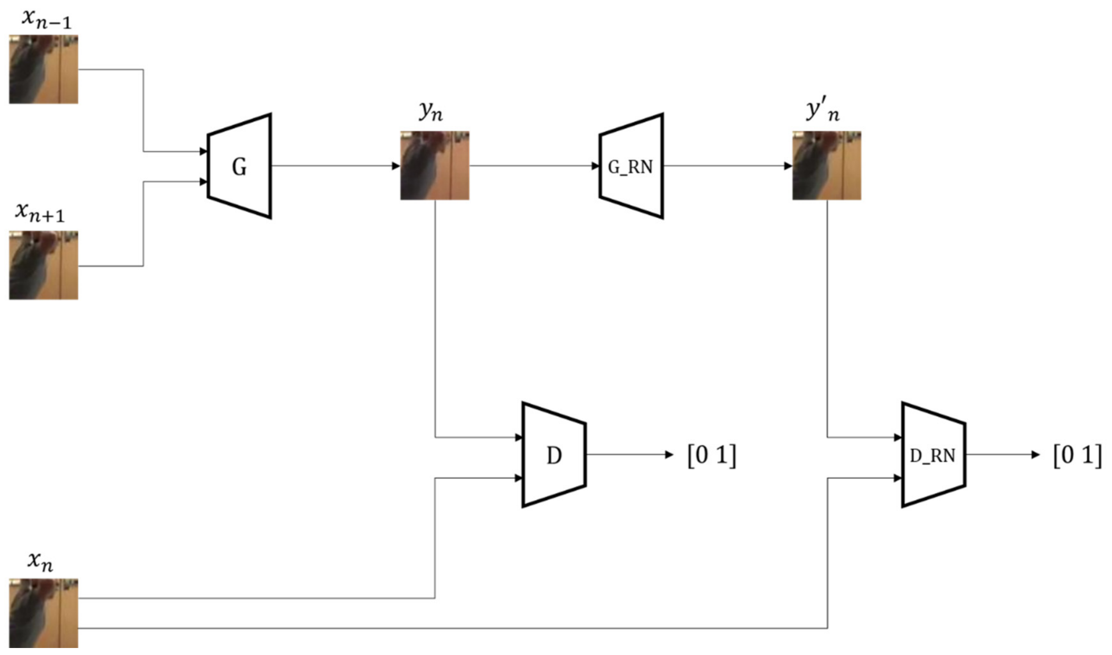


Figure 1. The proposed framework, in which the framework derives the interpolated frame y_n from two given ones. Two given input frames x_{n-1} and x_{n+1} are fed into the generator network G for an interpolated frame. The interpolated frame y_n can go through the second generator with the refinement network, denoted by G_RN, for further improvement. D and D_RN represent the discriminators of the original network and refinement network, respectively.

The rest of the paper is organized as follows. In Section 2, we review the related work on GANs, particularly in model design, image generation, and video frame generation. Section 3 describes the proposed approach. Section 4 demonstrates the experiments which include qualitative and quantitative evaluation of the proposed method and presents analyses. Finally, we conclude the paper in Section 5.

2. Related Works

2.1. *Generative Adversarial Networks*

A generative adversarial network (GAN) [9] is a machine learning framework for data generation, which attempts to generate new data that follow similar distribution as those in the training sets. A GAN includes two adversarial networks, the generator network and the discriminator network. The generator strives to generate a plausible output to fool the discriminator while the discriminator tries to classify generated outputs correctly, keeping from being fooled. Let x denote a source vector from the true data source X, $G(x)$ denote the generator's output, and $D(x)$ be the probability of x coming from the true data source rather than the generated output. Goodfellow et al. [9] formulated the adversarial loss $V(G, D)$ of GANs as,

$$V(G,D) = \mathrm{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathrm{E}_{x \sim p_g}[\log(1 - D(x)] \tag{1}$$

The training of GANs is a minimax problem that finds D and G such that

$$\max_{D} \min_{G} V(G, D) \tag{2}$$

That is, the generator G tries to produce samples that minimize $\log(1 - D(x))$, and simultaneously the discriminator D aims to maximize the probability of assigning the correct label (to both true samples and generated outputs of G). Following that strategy iteratively, both networks enhance their performance in the training process and produce more and more desirable output. Since 2014, the GAN has been used to solve various problems in image and video applications including image deblurring [10], realistic visual manipulation [11] and video frame generation [12–14].

Various improvements and extensions have been made on GANs. One notable direction is the introduction of nonrandom input models such as conditional GAN [15], StyleGAN [16], and SRGAN [17]. In contrast to the models with random inputs [4,18,19], nonrandom input models use labeled data or real images instead of noise as in the input, which allows GANs to generate outputs of specific features.

2.2. Image Generation and Video Frame Generation

GANs have shown their efficiency in numerous image applications. In the beginning, many researchers used GANs to generate images from scratch or simple labeled data. Later, cross-domain GANs, which transform data from a domain to another, such as from sketch to an image [11], from description text to image [20,21], and from photographs to painting [22], were developed. Other applications of GANs in image processing include super-resolution [4], image inpainting [23], image rendering [24], de-raining [25], and medical image processing [26]. Isola et al. [6] proposed a conditional GAN structure for image processing based on the image-to-image translation.

Video frame generation is a technique that produces new frames based on the existing image sequence with temporal semantics. There are two types of frame generation, frame extrapolation and frame interpolation as shown in Table 1.

In this study, we focus on frame interpolation, where an intermediate frame is predicted from adjacent frames. Traditional interpolation techniques are realized by estimating the relative motion of pixels, blocks, or meshes. One classic pixel-based approach is the optical flow, however, which requires heavy computation and performs poorly for videos containing abrupt changes in brightness or motion. Consequently, gradient-based [27] and phase-based [28] approaches were proposed. Deep learning based approaches, instead, directly synthesize the intermediate frame using neural networks and generally provide more satisfactory results than conventional methods.

The networks considered in video frame generation include the convolutional neural networks (CNNs) [32,33], recurrent neural networks (RNNs), long short term memory networks (LSTM), and GANs. Niklaus et al. [1] designed a fully-convolutional CNN and estimated pairs of 1D kernels for all pixels to produce the interpolated frame. Bao et al. [34] proposed a video frame interpolation method based on the bi-directional optical flow and depth maps estimation. Meyer et al. [35] proposed a phase-based deep learning method that predicts the interpolated frame level by level with a phase-based loss function. For GAN-based approaches, Kwon and Park [14] proposed using the retrospective cycle GAN framework to predict a future frame by four previous frames. Santurkar et al. [7] used a DCGAN based approach for frame interpolation to enhance video compression performance. In their work, the authors used a feed-forward network to predict a latent vector as the input for the generator and produced interpolated frames based on the linear interpolation in latent space. Wen et al. [8] concatenated two GANs to generate the intermediate frames between keyframes, in which the first generator learns motions, and the second one adds more details into the first network's output. In that work, the authors also enhanced the input data with new frames that were obtained by pixel-wise linear interpolation. The method, application, dataset, and result of deep learning-based frame interpolation are summarized and listed in Table 1. Although the deep learning based approaches generally achieve good performance, most of them use complex networks and auxiliary data and therefore a long time for training and processing is required. Moreover, many GAN based approaches still have difficulties in characterize irregular motion and image details.

Table 1. Deep learning-based approaches for frame interpolation.

Author	Method	Application	Dataset	Result
Santurkar et al. [7]	Process data in the z-domain with a CNN based network and a GAN. The framework also transmits every N-th frame for further improvement.	Image/video compression.	For Mcode (video compression framework): the KTH actions dataset.	The videos reach PSNR/SSIM of 22.85 dB/0.611 while skipping the alternate frame in compression.
Wen et al. [8]	Concatenate two GANs for generation with the additional input data obtained by the linear interpolation operation.	Generate N frame(s) between given ones (N ≥ 1).	The KTH actions dataset; the UCF101 dataset; he Google Robotic Push dataset.	Generate seven frames between two given ones from the UCF101 dataset with the PSNR in the range of 28 dB to 30 dB.
Koren et al. [29]	Utilize CNN and GAN for generation and refinement, respectively.	Generate a frame in-between two given ones.	Xiph.org Video Test Media: 'Bus', 'Football', 'News', and 'Stefan'.	Correct in structure and most of the details, but not close to the ground truth, particularly motion.
Amersfoort et al. [30]	Propose a multi-scale GAN that includes a multi-scale residual estimation module and a CNN based refinement network.	Frame interpolation.	Collection of 60 fps videos from Youtube-8m.	Achieve visual quality comparable to SepConv [1] at x47 faster runtime.
Meyer et al. [31]	Use a phase-based deep learning method that consists of a neural network decoder.	Frame interpolation for challenging scenarios, coping with large motion (larger than existing methods).	DAVIS video dataset.	Produce preferable output in challenging cases containing motion blurs and brightness changes. May produce more disturbing artifacts.

3. The Proposed Method

3.1. Network Architecture

This research is targeted at an effectual GAN-based frame interpolation framework without creating auxiliary data. As is illustrated in Figure 1, the proposed frame interpolation network G takes two input frames x_{n-1} (previous frame) and x_{n+1} (future frame) to derive y_n as a good estimate of the current frame x_n. Each frame is a two-dimensional image with three color channels (red, green, blue). The proposed discriminator D reports a classification value in the range of 0.0 and 1.0 for y_n, which indicates the confidence of y_n for being a real frame. Motivated by the DCGAN [4], our networks are fully convolutional, composed of 2D convolutional layers, 2D transposed convolutional layers, batch normalization layers, and Leaky Rectification Linear Unit (LeakyReLU) layers.

The complete frame interpolation process is separated into two phases, analysis and synthesis, as shown in Figure 2. The analysis phase extracts a feature map from the given input frames, while the synthesis phase produces an output frame from the feature map. No pooling function is involved in our approach. Instead, we use strides in convolutional layers to control the size of the feature map. The discriminator uses down-sampling processes to generate the classification result. The essential parameters of the proposed generator network and discriminator network are in Tables 2 and 3, respectively. Specific values of the coefficients involved in the network are obtained through the training process.

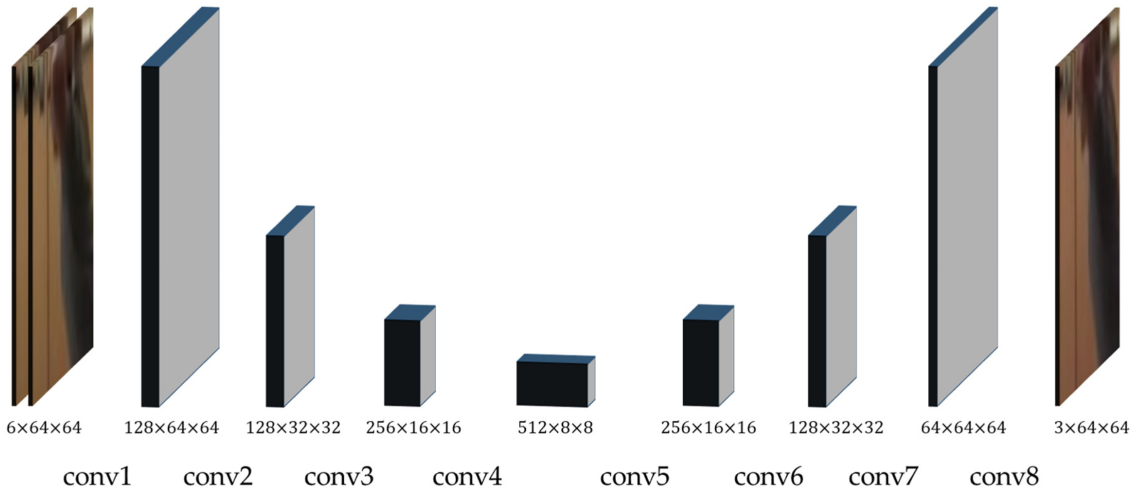


Figure 2. The model of the proposed generator network. The network takes two $3 \times 64 \times 64$ matrices and uses the previous and the future frames as the input. The details of the parameters are listed in Table 2.

Table 2. Proposed generator network's structure.

Layers	Details	Output Size
1	Conv2d(3, 128, stride = 1) BatchNorm2d LeakyReLU	$128 \times 64 \times 64$
2	Conv2d(4, 128, stride = 2) BatchNorm2d LeakyReLU	$128 \times 32 \times 32$
3	Conv2d(4, 256, stride = 2) BatchNorm2d LeakyReLU	$256 \times 16 \times 16$
4	Conv2d(4, 512, stride = 2) BatchNorm2d LeakyReLU	$512 \times 8 \times 8$
5	TransposedConv2d(4, 256, stride = 2) BatchNorm2d LeakyReLU	$256 \times 16 \times 16$
6	TransposedConv2d (4, 128, stride = 2) BatchNorm2d LeakyReLU	$128 \times 32 \times 32$
7	TransposedConv2d (4, 64, stride = 2) BatchNorm2d LeakyReLU	$64 \times 64 \times 64$
8	TransposedConv2d (3, 3, stride = 1) tanh	$3 \times 64 \times 64$

Table 3. Proposed discriminator network's structure.

Layers	Details	Output Size
1	Conv2d(4, 64, stride = 2) LeakyReLU	$64 \times 32 \times 32$
2	Conv2d(4, 128, stride = 2) BatchNorm2d LeakyReLU	$128 \times 16 \times 16$
3	Conv2d(4, 256, stride = 2) BatchNorm2d LeakyReLU	$256 \times 8 \times 8$
4	Conv2d(4, 512, stride = 2) BatchNorm2d LeakyReLU	$512 \times 4 \times 4$
5	Conv2d(4, 1, stride = 1) Sigmoid	$1 \times 512 \times 512$

To further improve the interpolated frame, particularly the motion and color, the proposed framework applies a refinement network proposed in [6]. This network is the image-to-image translation network that includes a generator network *G_RN* and a discriminator network *D_RN*. The generator *G_RN* translates an input frame similar to the ground truth, particularly in motion area.

3.2. Loss Functions

The training process of GAN-based networks relies on the loss function that measures the discrepancy between the generated model (or data) and the true model (or data). In this paper, the similarity of a generated frame is judged by a combination of two types of losses, which include the adversarial loss and the reconstruction loss. The overall loss function is expressed as follows,

$$L\left(X,\ \hat{X}\right) = \lambda_1 \times L_{\mathrm{adv}}\left(X,\ \hat{X}\right) + \lambda_2 \times L_2\left(X,\ \hat{X}\right) + \lambda_3 \times L_{\mathrm{MS\_SSIM}}\left(X,\ \hat{X}\right) + \lambda_4 \times L_{\mathrm{GDL}}\left(X,\ \hat{X}\right). \quad (3)$$

where X and $\hat{X}$ denote the original frame (ground truth) and the reconstructed frame, respectively, L_{adv} indicates the adversarial loss, i.e., the difference between the probability distributions and the other three terms L_2, $L_{\text{MS\_SSIM}}$ and L_{GDL} refer to the reconstruction data losses, as explained below. The coefficients λ_1, λ_2, λ_3, and λ_4 are non-negative weights for each of the loss terms.

We have explained the general adversarial loss L_{adv} in Section 2.1. Specifically in our work, the Binary Cross Entropy (BCE) loss function (Equation (4)), which measures the binary cross-entropy between the classification result $D(\boldsymbol{x})$ and the target value $\boldsymbol{y}$, is adopted.

$$L_{\text{adv}}(D(\boldsymbol{x}),\, \boldsymbol{y}) = \ L_{\text{BCE}}(d,\, \boldsymbol{y}) = -\frac{1}{N}\sum_{n=1}^{N}(y_n \log(d_n) + (1 - y_n)\log(1 - d_n)) \tag{4}$$

The target value y_n represents the expected label data between 0 (false) and 1 (true) and d is the discrimination value for the frame $\boldsymbol{x}$.

This study falls into the category of nonrandom input approaches. The errors (reconstruction data losses) between the generated output and ground truth can be calculated for the training set. The first term is the L_2 loss (mean square error, MSE)

$$L_2(\boldsymbol{x},\, \boldsymbol{y}) = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2, \tag{5}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the generated output and the ground truth, respectively, and N is the number of pixels within a frame. The L_2 loss function performs the pixel-by-pixel comparison and helps to sharpen the output image. The second term is multi-scale SSIM (MS_SSIM), suggested by Zhao et al. [36] for image restoration, as defined in Equation (6).

$$L_{\text{MS\_SSIM}}(\boldsymbol{x},\, \boldsymbol{y}) = 1 - \text{MS\_SSIM}(\boldsymbol{x},\, \boldsymbol{y}) \tag{6}$$

MS_SSIM well preserves the structural similarity in different scales and facilitates smoother output.

Third, we apply the gradient difference loss (GDL) function [12] for combating the motion blur. The GDL loss, which considers the relative difference of neighboring pixels between the generated output $\boldsymbol{x}$ and ground truth $\boldsymbol{y}$, is calculated as follows

$$L_{GDL}(\boldsymbol{x},\, \boldsymbol{y}) = \sum_{i,j}\left\| x_{i,j} - x_{i-1,j} \right| - \left| y_{i,j} - y_{i-1,j} \right\|^{\alpha} + \left\| x_{i,j} - x_{i,j-1} \right| - \left| y_{i,j} - y_{i,j-1} \right\|^{\alpha}, \tag{7}$$

where α is an integer greater or equal to 1 and we set α to two in our implementation.

4. Experimental Results and Analyses

4.1. Dataset

The UCF101 dataset [37] is a large diversified video dataset, which includes 13,320 realistic videos from 101 action categories with various moving objects in static and dynamic environments. The videos in the UCF101 dataset have a fixed frame rate of 25 FPS (frames per second) and a resolution of 320 × 240 pixels. Soomro et al. [37] divided these videos into five types, Body-Motion Only, Human-Human Interaction, Human-Object Interaction, Playing Musical Instrument, and Sports.

In our experiment, we use a part of Body-Motion Only videos for both training and testing. The training dataset includes 377 videos, while the testing dataset comprises 59 videos. These videos come from the following action categories: BabyCrawling, BlowingCandles, BodyWeightSquats, HandstandWalking, JumpingJack, PullUps, RopeClimbing, Surfing, Swing, TrampolineJumping, and WallPushups. The selected videos cover a wide range of scenes with diverse motion and texture characteristics.

We also perform the outside test by using the pre-trained model on other datasets, such as other types in the UCF101 dataset, the CUHK dataset [38], and the high-speed Sintel (Sintel) dataset [39]. The CUHK dataset contains 150 FPS videos of resolution of 640 × 360 pixels, while the Sintel dataset includes 1080 FPS videos of resolution of 2048 × 872 pixels.

4.2. Implementation and Training

We group every three consecutive frames as a patch in sliding windows as shown in Figure 3. The generator network takes the first and third frames from the patch to derive the output (interpolated second frame), while the discriminator network compares the generated output and genuine second frame. The proposed framework takes 32 patches for every training interval and uses the mini-batch gradient descent algorithm to train the network. The input to the network is cropped into blocks of 64 pixels by 64 pixels. It was observed that this size was large enough to capture motions of videos and suitable for training. Cropping positions are randomly chosen to produce diversity.

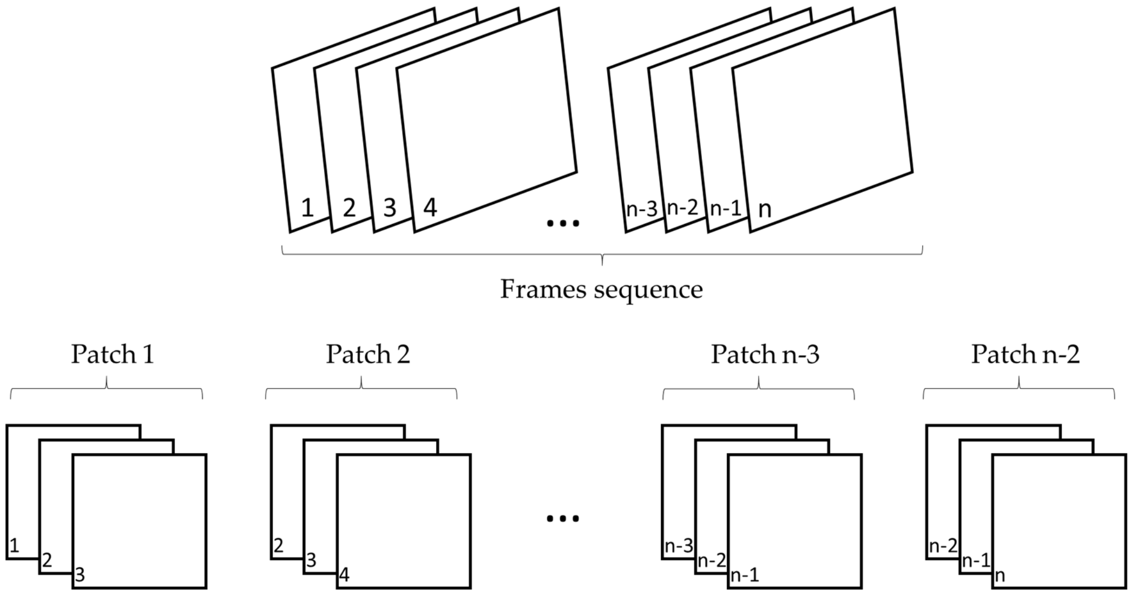


Figure 3. The method for grouping frames into correlated patches. Here, the neighbors share some same frames.

We use PyTorch for implementing the proposed method. To train the GAN, we use the Adam optimizer [40] with a decay of the first-order momentum of the gradient of 0.5, and a decay of the second-order momentum of the gradient of 0.999 [4]. A learning rate of 0.0002 is chosen for training. A summary of parameters and the corresponding values used in implementation is listed in Table 4. Besides, the selected values of λ_1, λ_2, λ_3, λ_4 in the loss function (3) are 0.05, 1.0, 6.0, 1.0. The proposed network is trained on a single NVIDIA GeForce GTX 960 GPU for 400 epochs.

For the refinement network, we adopt the implementation of the pix2pix model of Isola et al. [6]. The source code is shared on GitHub under the BSD license [41]. Our implementation that includes all networks is also published on GitHub [42]. The activity diagrams of our implementations are provided in the Appendix A for reference.

Table 4. Configuration parameters.

Parameters	Values
Number of epochs	400
Batch size	32
Size of input image	64
Input's channels	3
Learning rate	0.0002
Adam: β1	0.5
Adam: β2	0.999

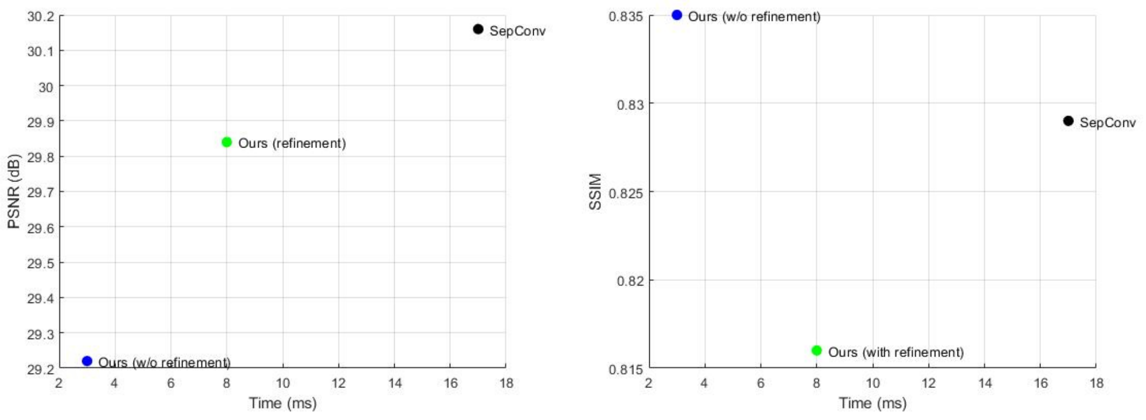
4.3. Objective Evaluation

In our evaluation, two widely used metrics for video quality, peak signal to noise ratio (PSNR) and structural similarity index (SSIM) [43], are adopted for objective evaluation. The objective performance of the proposed method is shown in Table 5. Higher PSNR but lower SSIM is achieved with refinement. The proposed method is first compared with the MCode framework, a frame interpolation approach proposed by Santurkar et al. [7]. Tested on the KTH (hand-waving) dataset [44], the MCode gives an average PSNR of 22.85 dB and SSIM of 0.609 for the interpolated frames. The proposed method, on the other hand, provides a much higher average PSNR of 33.18 dB and SSIM of 0.919. The *p* value of the associated statistical hypothesis test equal to 0.01 substantiates the significantly better performance of the proposed framework.

Table 5. Objective evaluation on various datasets for the proposed method.

	Proposed Method (without Refinement)		**Proposed Method (with Refinement)**	
Dataset	**PSNR**	**SSIM**	**PSNR**	**SSIM**
UCF101:Body Motion Only	28.97	0.827	29.52	0.809
UCF101:Human-Human Interaction	27.50	0.789	27.60	0.762
UCF101:Human-Object Interaction	31.37	0.867	31.72	0.832
UCF101:Playing Musical Instrument	32.54	0.881	32.48	0.852
UCF101:Sport	29.10	0.821	29.42	0.797
CUHK	32.63	0.935	32.20	0.910
Sintel	32.64	0.914	33.11	0.912

The proposed method is further compared with [1] (SepConv, the state-of-the-art-solution), and the performance is shown in Table 6. All the approaches have comparable quality metrics, while the proposed approaches have more stable performance (less standard deviation in PSNR and SSIM), and the proposed method (without refinement) has a slightly higher SSIM. This framework also requires less time (80 h) for training, wherein contrast SepConv requires 216 h. The computation complexity of the proposed method (network) is 0.78 GFLOPs (without refinement) and 1.08 GFLOPs (with refinement) for processing 64-by-64-pixel block(s). Additionally, without applying the refinement step, our pre-trained framework takes approximately 60 ms to process a frame that has a resolution of 320-by-240 from the UCF101 dataset, which is much less than that of SepConv. As a consequence, more than 16 frames can be generated in a second, and the frame rate up-conversion for 320-by-240 videos from 15 fps to 30 fps can be achieved. Figure 4 illustrates a good performance achieved by the proposed method in terms of the quality (PSNR/SSIM) and complexity (time) relationship.



(**a**) (**b**)

Figure 4. The average metric values, (**a**) PSNR and (**b**) SSIM, with respecting the processing time of various approaches, which include SepConv (black), the proposed approach without the refinement (blue), and the proposed approach with the refinement (green), to derive a 64-by-64-pixel output.

Table 6. Comparisons of the proposed method and SepConv [44] on UCF101 [1,2]. The best value of each column is in bold.

Approach	Training Time (Hours)	Processing Time (ms)	PSNR				SSIM			
			Min	Max	Avg	Stdev	Min	Max	Avg	Stdev
SepConv [1]	216	340	12.53	43.44	**30.16**	5.90	0.161	**0.998**	0.829	0.136
Ours (w/o refinement)	**80**	**60**	12.60	39.83	29.22	**4.97**	**0.170**	0.984	**0.835**	0.126
Ours (with refinement)	167	160	**12.61**	**44.67**	29.84	5.58	0.169	0.972	0.816	**0.120**

[1] Categories action from the UCF101_Body Motion Only dataset. [2] The videos have a resolution of 320-by-240-pixel.

4.4. Subjective Results

Examples of the generated frames are shown in Figures 5–7. It can be observed that the proposed method maintains correct motion and most textural details in the scene. Besides, the inclusion of the refinement network derives the motion areas more favorably. For a better point of view on the similarity between the ground truth and the generated frames, the histograms of pixels are graphed in Figure 8, which illustrate their similarity. The correlation coefficients between the histograms confirm the accuracy of the proposed method and the advantage of incorporating the refinement process. In Figure 9, we compare the generated outputs of the proposed method with SepConv [1]. It can be seen that the proposed approach keeps more structural details of the objects in the scene. Besides, the refinement process improves the estimation of color and motion (Figure 7).

We do an ablation study on the loss functions. Using a combination of the adversarial loss, L1 loss, and GDL yields a PSNR of 27.97 dB and SSIM of 0.816. Using a combination of the adversarial loss, L2 loss, and GDL yields a PSNR of 27.84 dB and SSIM of 0.746. The proposed combination of four loss functions, including the MS-SSIM loss, yields PSNR of 28.62 dB and SSIM of 0.810. Figure 10 shows the generated frames. The proposed method creates frames that keep more details.

In summarization, our method interpolates a frame from two given inputs within a short time. Moreover, the output is correct in motion, details, and color. Without the refinement, our framework produces preferable frames in human vision. The refinement network improves the output frame in the similarity to the ground truth, although it may reduce the frame's quality in some cases. Hence, each approach is suitable for different kinds of applications. The proposed method may not be able to capture irregular motion correctly. More training sequences or a conditional GAN model will be tested in the future.

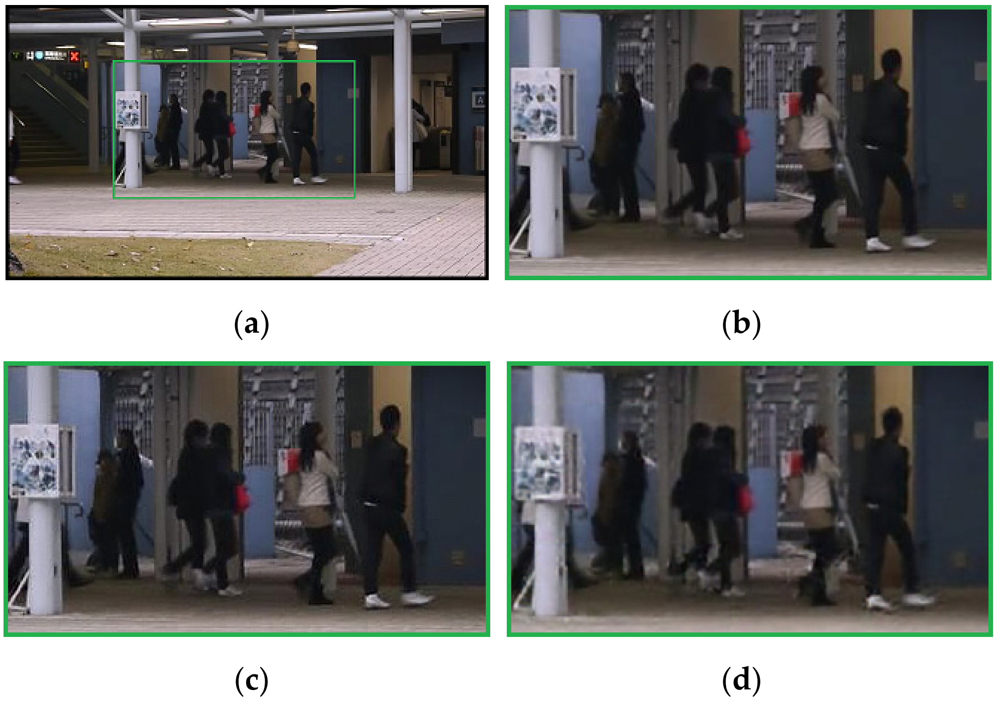


Figure 5. Experimental results on the CUHK dataset. The ground truth is shown in the first row and the generated outputs are shown in the second row: (**a**) The full ground truth, (**b**) the ground truth, (**c**) the output of the proposed method without refinement (PSNR = 32.87, SSIM = 0.935), (**d**) the output of the proposed method with refinement (PSNR = 31.39, SSIM = 0.914).

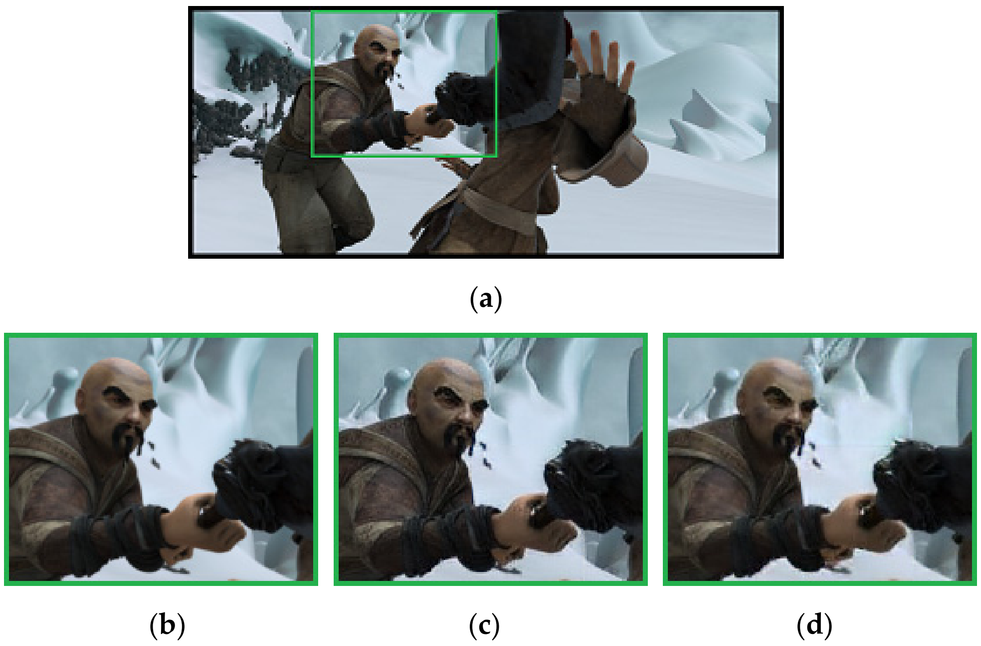
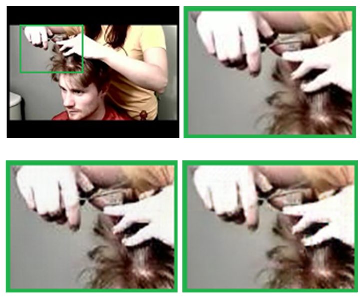
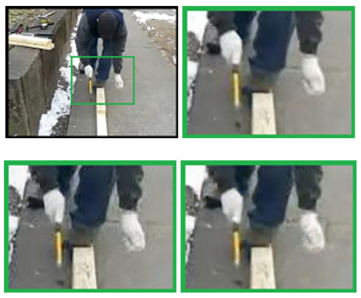
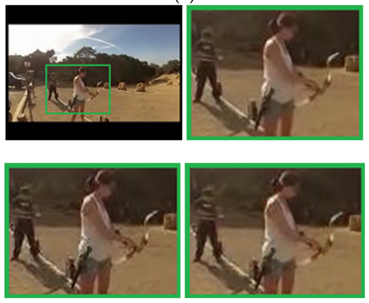


Figure 6. Experimental results on the Sintel datasets. The figures are demonstrated in two rows: (**a**) The full ground truth, (**b**) the ground truth, (**c**) the output of the proposed method without refinement (PSNR = 27.96, SSIM = 0.860), (**d**) the output of the proposed method with refinement (PSNR = 29.08, SSIM = 0.871).


(**a**)


(**b**)


(**c**)

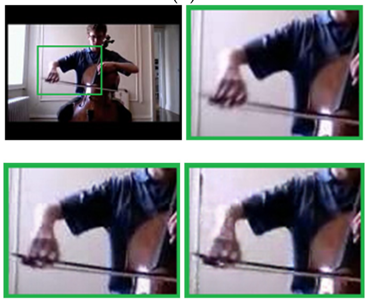
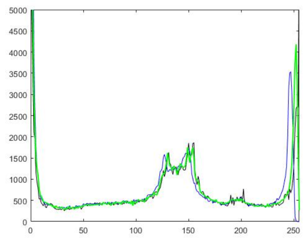
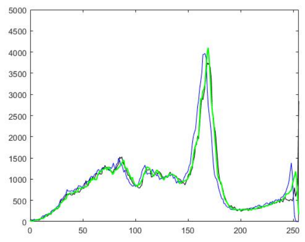
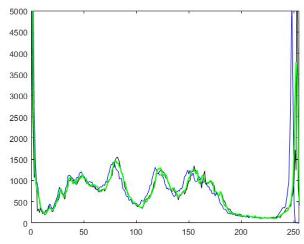

(**d**)

Figure 7. Experimental results on cross-domain action categories from the UCF101 dataset. For each group, the ground truth is shown in the first row, and the generated output of the proposed method is shown in the second row, in which the left one is the output without refinement, and the right one is the refinement output. Quality metrics for the generated output: (**a**) w/o refinement: PSNR = 30.56, SSIM = 0.834; with refinement: PSNR = 30.16, SSIM = 0.782; (**b**) w/o refinement: PSNR = 27.85, SSIM = 0.856; with refinement: PSNR = 28.51, SSIM = 0.852; (**c**) w/o refinement: PSNR = 30.68, SSIM = 0.849; with refinement: PSNR = 32.78, SSIM = 0.839; (**d**) w/o refinement: PSNR = 34.03, SSIM = 0.896; with refinement: PSNR = 34.39, SSIM = 0.845.


(**a**)


(**b**)


(**c**)

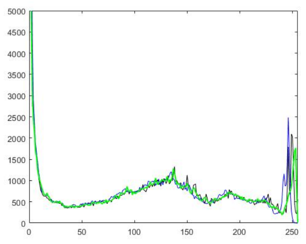

(**d**)

Figure 8. Histogram comparison between the ground truth (black line), the output without refinement (blue line), and the output with refinement (green line) on cross-domain action categories from the UCF101 dataset. Each diagram corresponds to a same-label-group in Figure 7. The correlation coefficient between the histograms of two pairs of frames, which are the ground truth-the output without refinement and the ground truth-the output with refinement, are (**a**) 0.6855 and 0.8886, (**b**) 0.8731 and 0.9609, (**c**) 0.5298 and 0.8933 and (**d**) 0.7996 and 0.9767.

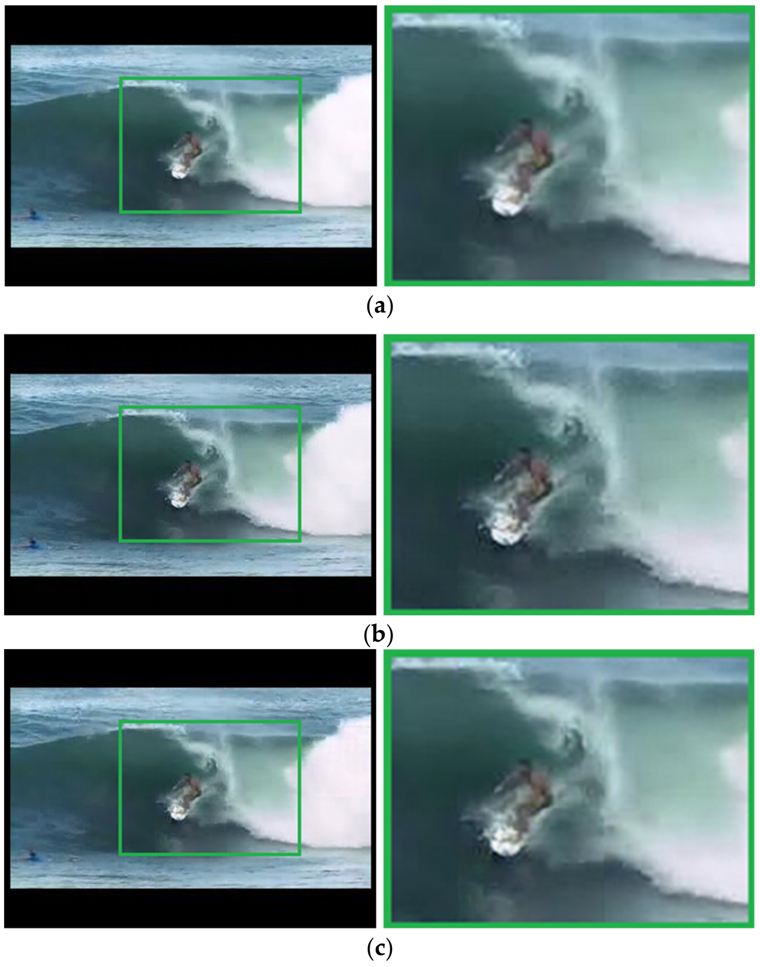


Figure 9. Comparison of SepConv and the proposed method on the generated output for the UCF101 dataset. The proposed framework keeps more structural details. Quality metrics for the generated output: (**a**) SepConv, PSNR = 30.35, SSIM = 0.796; (**b**) the proposed method (w/o refinement), PSNR = 29.66, SSIM = 0.839; (**c**) the proposed method (with refinement), PSNR = 30.71, SSIM = 0.807.

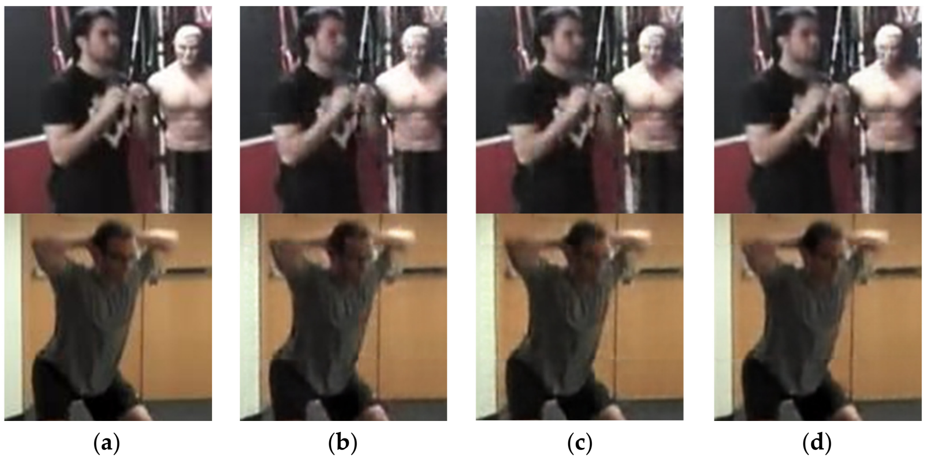


Figure 10. Comparison of various combination of loss functions on the proposed frame interpolation network and the UCF101 dataset. The proposed loss function output is preferable output in human vision. The figures are demonstrated in four columns. From left, they are (**a**) the ground truth, (**b**) the combination of the adversarial loss, L1 loss, and GDL, (**c**) the combination of the adversarial loss, L2 loss, and GDL, and (**d**) the proposed combination of loss function.

5. Conclusions

We have presented a lightweight GAN for frame interpolation. The proposed method neither involves complex networks and nor requires auxiliary data. A suitable loss function is designed, and the ablation study on the loss functions indicates that our choice can increase PSNR and keep more details for the generated frames. The experimental results substantiate a favorable visual quality given by the proposed method as compared to the state-of-the-art approaches, both objectively and subjectively, with a more stable performance. Among the investigated methods, the proposed method without refinement achieves a high average SSIM value of 0.835 with the shortest training time and encoding time. The pre-trained framework in our experiment takes approximately 60 ms to derive a frame from the UCF101 dataset videos with our experiment environment. The PSNR versus time diagram substantiates the advantage of the proposed method. We also investigate the use of a GAN-based refinement process, which is found to improve the estimation of color and motion.

Author Contributions: Conceptualization, Q.N.T. and S.-H.Y.; Methodology, Q.N.T.; Software, Q.N.T.; Validation, Q.N.T., and S.-H.Y.; Formal Analysis, Q.N.T.; Investigation, Q.N.T. and S.-H.Y.; Resources, S.-H.Y.; Data Curation, Q.N.T.; Writing—Original Draft Preparation, Q.N.T.; Writing—Review & Editing, S.-H.Y.; Visualization, Q.N.T.; Supervision, S.-H.Y.; Project Administration, S.-H.Y.; Funding Acquisition, S.-H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported in part by the Ministry of Science and Technology of Taiwan (R.O.C.) under Grant MOST 108-2221-E-027-065–MY2.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Activity Diagram

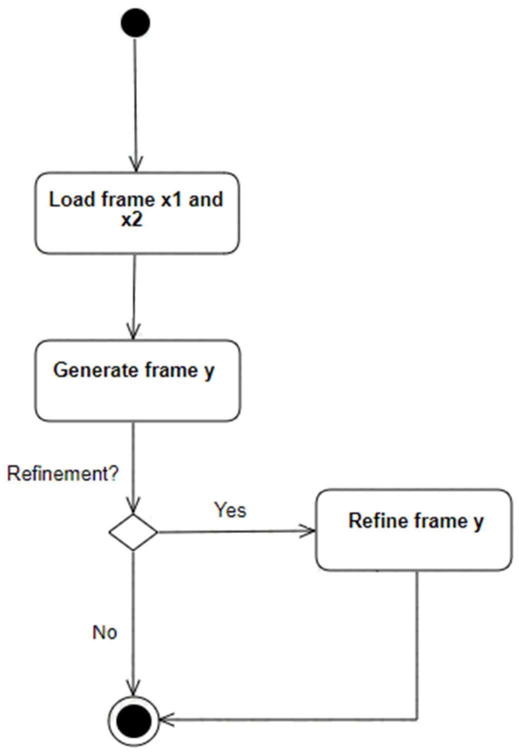


Figure A1. The activity diagram of the process of interpolation. The frame interpolation network produces an output frame from two given input ones. Then, the optional refinement step involves for further improvement.

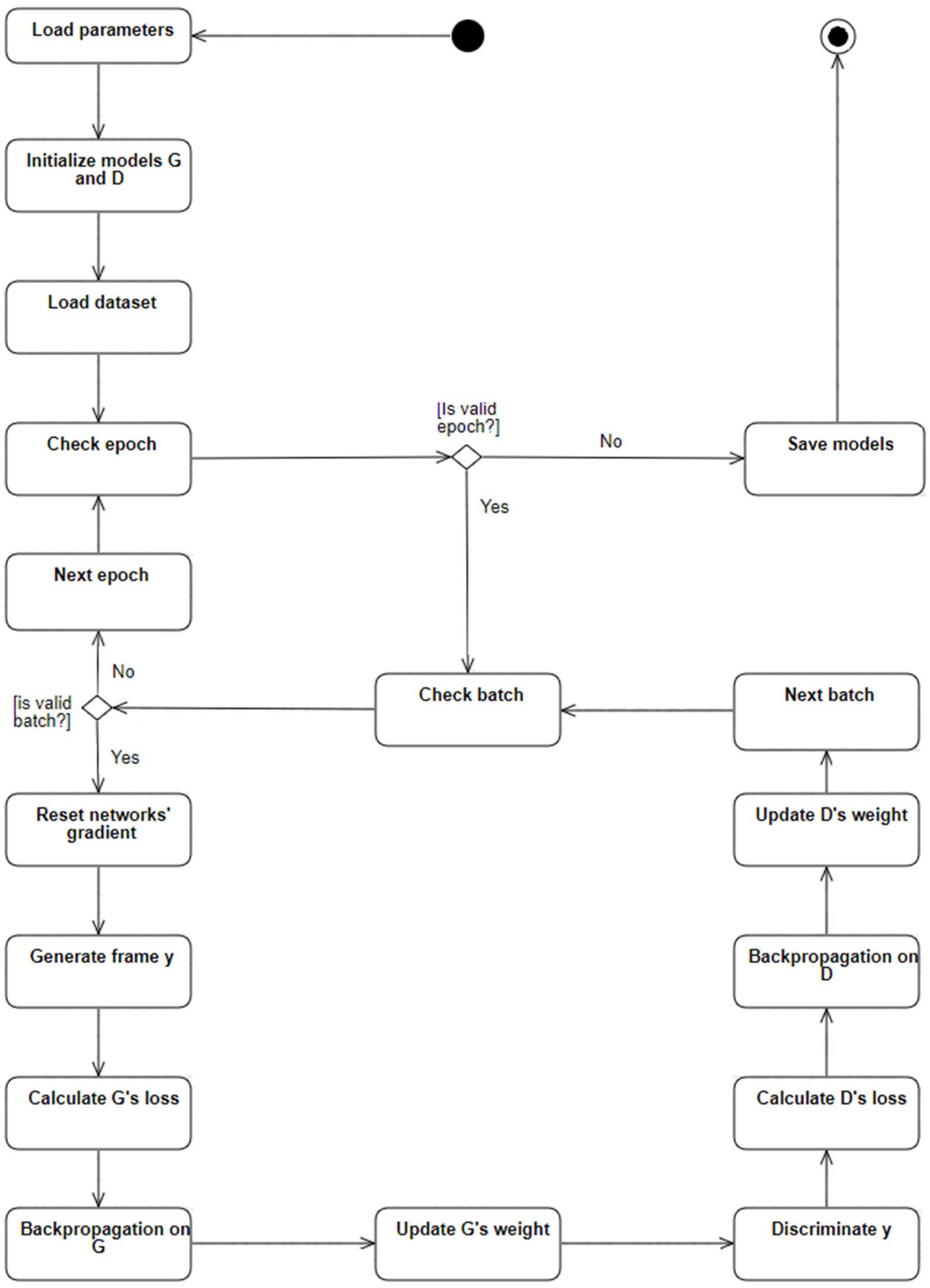


Figure A2. The activity diagram of the training process of the proposed frame interpolation network.

References

1. Niklaus, S.; Mai, L.; Liu, F. Video Frame Interpolation via Adaptive Separable Convolution. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
2. Liu, Z.; Yeh, R.A.; Tang, X.; Liu, Y.; Agarwala, A. Video Frame Synthesis Using Deep Voxel Flow. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
3. Niklaus, S.; Liu, F. Context-Aware Synthesis for Video Frame Interpolation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

4. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434v2.
5. Sung, T.L.; Lee, H.J. Image-to-Image Translation Using Identical-Pair Adversarial Networks. *Appl. Sci.* **2019**, *9*, 2668. [CrossRef]
6. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
7. Santurkar, S.; Budden, D.; Shavit, N. Generative Compression. In Proceedings of the Picture Coding Symposium, San Francisco, CA, USA, 24–27 June 2018.
8. Wen, S.; Liu, W.; Yang, Y.; Huang, T.; Zeng, Z. Generating Realistic Videos From Keyframes With Concatenated GANs. *Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2337–2348. [CrossRef]
9. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; NIPS: Montreal, QC, Canada, 2014.
10. Hong, M.; Choe, Y. Wasserstein Generative Adversarial Network Based De-Blurring Using Perceptual Similarity. *Appl. Sci.* **2019**, *9*, 2358. [CrossRef]
11. Zhu, J.Y.; Krahenbuhl, P.; Shechtman, E.; Efros, A.A. Generative Visual Manipulation on the Natural Image Manifold. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; Volume 9099, pp. 597–613.
12. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
13. Lin, J.; Liu, D.; Li, H.; Wu, F. Generative adversarial network-based frame extrapolation for video coding. In Proceedings of the International Conference on Visual Communications and Image Processing, Taichung, Taiwan, 9–12 December 2018.
14. Kwon, Y.H.; Park, M.G. Predicting Future Frames using Retrospective Cycle GAN. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
15. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784v1.
16. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
17. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
18. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
19. Berthelot, D.; Schumm, T.; Metz, L. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv* **2017**, arXiv:1703.10717.
20. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Huang, X.; Wang, X.; Metaxas, D. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–19 October 2017.
21. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1947–1962. [CrossRef] [PubMed]
22. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
23. Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context Encoders: Feature Learning by Inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
24. Cui, Y.; Wang, W. Colorless Video Rendering System via Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Artificial Intelligence and Computer Applications, Dalian, China, 29–31 March 2019.

25. Xiang, P.; Wang, L.; Wu, F.; Cheng, J.; Zhou, M. Single-Image De-Raining With Feature-Supervised Generative Adversarial Network. *IEEE Signal Process. Lett.* **2019**, *26*, 650–654. [CrossRef]
26. Dirvanauskas, D.; Maskeliūnas, R.; Raudonis, V.; Damaševičius, R.; Scherer, R. HEMIGEN: Human Embryo Image Generator Based on Generative Adversarial Networks. *Sensors* **2019**, *19*, 3578. [CrossRef] [PubMed]
27. Mahajan, D.; Huang, F.; Matusik, W.; Ramamoorthi, R. Moving gradients: A path-based method for plausible image interpolation. *ACM Trans. Graph.* **2009**, *28*. [CrossRef]
28. Fleet, D.J.; Jepson, A.D. Computation of component image velocity from local phase information. *Int. J. Comput. Vis.* **1990**, *5*, 77–104. [CrossRef]
29. Koren, M.; Menda, K.; Sharma, A. *Frame Interpolation Using Generative Adversarial Networks*; Stanford University: Stanford, CA, USA, 2017.
30. Amersfoort, J.V.; Shi, W.; Acosta, A.; Massa, F.; Totz, J.; Wang, Z.; Caballero, J. Frame Interpolation with Multi-Scale Deep Loss Functions and Generative Adversarial Networks. *arXiv* **2019**, arXiv:1711.06045.
31. Meyer, S.; Wang, O.; Zimmer, H.; Grosse, M.; Sorkine-Hornung, A. Phase-based frame interpolation for video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
32. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazırbas, C.; Golkov, V. FlowNet: Learning Optical Flow with Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015.
33. Reda, F.A.; Sun, D.; Dundar, A.; Shoeybi, M.; Liu, G.; Shih, K.J.; Tao, A.; Kautz, J.; Catanzaro, B. Unsupervised Video Interpolation Using Cycle Consistency. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), 27 October–2 November 2019.
34. Bao, W.; Lai, W.S.; Ma, C.; Zhang, X.; Gao, Z.; Yang, M.H. Depth-Aware Video Frame Interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
35. Meyer, S.; Djelouah, A.; McWilliams, B.; Sorkine-Hornung, A.; Gross, M.; Schroers, C. PhaseNet for Video Frame Interpolation. *arXiv* **2018**, arXiv:1804.00884.
36. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss Functions for Image Restoration with Neural Networks. *IEEE Trans. Comput. Imaging* **2017**, *3*, 47–57. [CrossRef]
37. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Action Classes from Videos in the Wild. *arXiv* **2012**, arXiv:1212.0402.
38. Lu, C.; Shi, J.; Jia, J. Abnormal Event Detection at 150 FPS in Matlab. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013.
39. Janai, J.; Güney, F.; Wulff, J.; Black, M.J.; Geiger, A. Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
40. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
41. Pix2pix. Available online: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix (accessed on 15 July 2020).
42. Frame_interpolation_GAN. Available online: https://github.com/tnquang1416/frame_interpolation_GAN (accessed on 26 August 2020).
43. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*. [CrossRef] [PubMed]
44. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004.