Journal of Artificial Intelligence and Autonomous Intelligence Received 10-04-2024; Accepted 16-07-2024; Published 17-07-2024

RBPGAN: Recurrent Back-Projection GAN for Video Super Resolution

Israa Fahmy israafahmy@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Marwah Sulaiman marwahisham@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Zahraa Shehabeldin zahraaagamal@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Mohammed Barakat mohamedyasser36@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Mohammed El-Naggart Mohamed_elnaggar@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Dareen Hussein dareenhussein@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Moustafa Youssef moustafa.youssef@aucegypt.edu
Department of Computer Science and Engineering
The American University in Cairo, Cairo
Egypt

Hesham M. Eraqi heraqi@amazon.com
Amazon, Last Mile Geospatial Science, Seattle,WA
USA

Corresponding Author: Israa Fahmy, et al.

Abstract

Citation: RBPGAN: Recurrent Back-Projection GAN for Video Super Resolution. Journal of Artificial Intelligence and Autonomous Intelligence. Research. 2024;1(1):3. Pages: 32-49

Video Super Resolution (VSR) has emerged as a crucial task in the field of Computer Vision due to its diverse applications. In this paper, we propose the Recurrent Back-Projection Generative Adversarial Network (RBPGAN) for VSR, aiming to generate temporally coherent videos while preserving spatial details. RBPGAN integrates two state-of-the-art models to leverage their strengths without compromising the accuracy of the output video. The generator in our model is inspired by the RBPN system, while the discriminator draws from TecoGAN. Additionally, we employ a Ping-Pong loss to enhance temporal consistency over time. Our approach results in a model that surpasses previous works in producing temporally consistent details, as demonstrated through both qualitative and quantitative evaluations across different datasets.

Keywords: Video Super Resolution (VSR), Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), Temporal Coherence, Recurrent Projection.

1. INTRODUCTION

Video Super Resolution (VSR) is the process of generating High Resolution (HR) videos from Low Resolution (LR) videos. Videos are one of the most common types of media shared in our daily lives. From entertainment purposes like movies to security purposes like surveillance camera footage, videos have become increasingly important. Consequently, VSR has also gained significance. The need to modernize old videos or enhance security camera footage to identify faces has become critical in recent years. VSR aims to enhance the quality of videos to meet these needs.

Similar to VSR, but older, is Image Super Resolution (ISR), which involves generating a single high-resolution image from a single low-resolution image. Since a video is essentially a sequence of frames (images), VSR can be seen as ISR applied to each frame in the video. While this analogy is useful because many ISR techniques can be slightly modified for VSR, there are major differences between VSR and ISR. The main difference is the temporal dimension in videos, which does not exist in images. The relationship between a frame in a video and other frames in the sequence makes VSR more complex than ISR.

In this research, various VSR methods will be explored. These methods are mainly clustered into two categories: methods with alignment and methods without alignment. We will compare the different methods across various datasets and discuss the results. Among the methods we studied, we chose two models as the base models for our research. We further explore these base models and conduct experiments with them.

This paper aims to minimize the trade-off between temporal coherence and the quality of VSR. To achieve this, we propose a Generative Adversarial Network (GAN) that combines components from each of the base models to achieve the best of both worlds. Our methodology, experimentation, and results are presented in the following sections. Finally, we conclude the paper and propose future recommendations for further research.

2. RELATED WORK

Based on our review of the literature, Deep Learning-based methods targeting the Video Super Resolution problem can be divided into two main categories: methods with alignment and methods without alignment. Alignment means that the input LR video frames should be aligned before being fed into the model. Within the methods with alignment, existing models can be divided into two sub-categories: methods with Motion Estimation and Motion Compensation (MEMC), and methods with Deformable Convolution (DC). Within the methods without alignment, existing models can be divided into four sub-categories: 2D convolution, 3D convolution, RNNs, and Non-Local based methods. In this section, the state-of-the-art methods belonging to each category will be discussed.

2.1 Methods with Alignment

2.1.1 Motion Estimation and Motion Compensation (MEMC)

First, the Temporally Coherent Generative Adversarial Network (TecoGAN) [1], The network proposes a temporal adversarial learning method for a recurrent training approach that can solve problems like Video Super Resolution, maintaining the temporal coherence and consistency of the video without losing any spatial details, and without resulting in any artifacts or features that arbitrarily appear and disappear over time. The TecoGAN model is tested on different datasets, including the widely used Vid4, and it is compared to the state-of-the-arts ENet, FRVSR, DUF, RBPN, and EDVR. TecoGAN has significantly less trainable weights than RBPN and EDVR. It scores PSNR of 25.57, and its processing time per frame is 41.92 ms. TecoGAN is able to generate improved and realistic details in both down-sampled and captured images.However, one limitation of the model is that it can lead to temporally coherent yet sub-optimal details in certain cases such as under-resolved faces and text.
Second, the recurrent back-projection network (RBPN) [2]. This architecture mainly consists of one feature extraction module, a projection module, and a reconstruction module. The recurrent encoder-decoder module integrates spatial and temporal context from continuous videos. This architecture represents the estimated inter-frame motion with respect to the target rather than explicitly aligning frames. This method is inspired by back-projection for MISR, which iteratively calculates residual images as reconstruction error between a target image and a set of its corresponding images. These residual blocks get projected back to the target image to improve its resolution. This solution integrated SISR and MISR in a unified VSR framework as SISR iteratively extracted various feature maps representing the details of a target frame while the MISR were used to get a set of feature maps from other frames. This approach reported extensive experiments to evaluate the VSR and used the different datasets with different specs to conduct detailed evaluation of strength and weaknesses for example it used the Vid4, and SPMCS which lack significant motions. It proposes an evaluation protocol for video SR which allows to differentiate performance of VSR based on the magnitude of motion in the input videos. It proposes a new video super-resolution benchmark allowing

evaluation at a large scale and considering videos in different motion regimes.

2.1.2 Deformable Convolution methods (DC)

The Enhanced Deformable Video Restoration (EDVR) [3], model was the winning solution of all four tracks of the NTIRE19 competition. In addition, outperformed the second-best solution. Also, this solution performed better when compared to some of the state-of-the-art solutions. EDVR is a framework that performs different video super-resolution and restoration tasks. The architecture of EDVR is composed of two main modules known Pyramid, Cascading, and Deformable convolutions (PCD) and Temporal and Spatial Attention (TSA). EDVR was trained on the REDS dataset, which contains 240 training videos and 60 videos divided equally for validation and testing. Each video in the REDS dataset is a 100 consecutive frame short clip.

2.2 Methods Without Alignment

2.2.1 2D convolution

Generative adversarial networks and perceptual losses for video super-resolution [4]. The model uses a GAN to generate high-resolution videos. The generator and the discriminator in the GAN consist both of many convolutional layers and blocks. The generator first generates a high-resolution frame, and the discriminator decides whether the output from the generator is a generated frame or a ground-truth (GT) image. If the discriminator decides it is a generated frame, then the generator uses the output of the discriminator to generate a better, closer to GT, high-resolution frame. The process is then repeated multiple times until the discriminator accepts the output of the generator as a GT image.

2.2.2 3D convolution

The dynamic filter network can generate filters that take specific inputs and generate corresponding features. The dynamic upsampling filters (DUF) [5], use a dynamic filter network to achieve VSR. The structure of the dynamic upsampling filter and the spatio-temporal information learned from the 3D convolution led to a comprehensive knowledge of the relations between the frames. DUF performs filtering and upsampling operations and uses a network to enhance the high-frequency details of the super-resolution result.

2.2.3 RCNNS

RCNNS is a very powerful network [6], developed a stochastic temporal convolutional network (STCN) by incorporating a hierarchy of stochastic latent variables into TCNs, allowing them to learn representations over a wide range of timescales. The network is divided into three modules: spatial, temporal, and reconstruction. The spatial module is in charge of extracting features from a series of LR frames. Temporal module is a bidirectional multi-scale convoluted version Motion estimation of LSTM that is used to extract temporal correlation between frames. The latent random variables in STCN are organized in accordance with the temporal hierarchy of the TCN blocks, effectively spreading them across several time frames. As a result, they generated a new auto-regressive model that combines the computational advantages of convolutional architectures with the expressiveness of hierarchical stochastic latent spaces. The model in STCN is meant to encode and convey information across its hierarchy.

2.2.4 Non-Local methods

There is a progressive fusion network for vSR that is meant to make greater use of spatio-temporal information that has shown to be more efficient and effective than existing direct fusion, slow fusion, and 3D convolution techniques through a technique known as Progressive Fusion Video Super-Resolution Networks in Exploiting Non-Local Spatio-Temporal Correlations (PFNL). This is presented in Progressive Fusion Video Super-Resolution Network via Exploiting Non-Local Spatio-Temporal Correlations [7]. That enhanced the non-local operation in this progressive fusion framework to circumvent the MEMC methods used in prior VSR techniques.This was done by adding a succession of progressive fusion residual blocks (PFRBs). The suggested PFRB is designed to make greater use of spatiotemporal information from many frames. Furthermore, the PFRB's multi-channel architecture allows it to perform effectively even with little parameters by employing a type of parameter sharing technique. That created and enhanced the non-local residual block (NLRB) to directly capture long-range spatiotemporal correlations. So, this can be summarized into three major components: a non-local resblock, progressive fusion residual blocks (PFRB), and an upsampling block. The non-local residual blocks are used to extract spatio-temporal characteristics, and PFRB is proposed to fuse them. Finally, the output of a sub-pixel convolutional layer is added to the input frame, which is then upsampled using bicubic interpolation to produce the final super-resolution results.

3. Our Model and Contribution

This paper proposes a Generative Adversarial Network that combines the generator of RBPN to achieve high accuracy and the discriminator of TecoGAN to improve temporal coherence, while reducing model size.

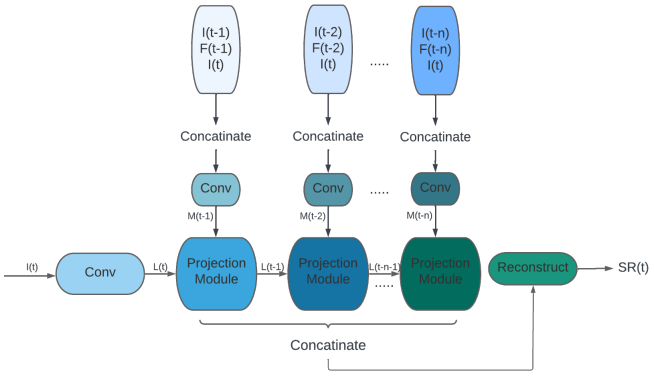

Figure 1: RBPN Architecture

3.1 RBPN

The Recurrent Back Projection Network calculates residual images as the reconstruction error between the target image and a set of six neighboring frames, exploiting temporal relationships between adjacent frames [2]. The network mainly consists of three modules: a feature extraction module, a projection module, and a reconstruction module, as shown in FIGURE 1.

The feature extraction module performs two operations: it extracts features directly from the target frame and from the concatenation of the neighboring frames, and calculates the optical flow from the neighboring frames to the target frame. The projection module consists of an encoder and a decoder. The encoder is composed of multiple image super-resolution (MISR), single image super-resolution, and residual blocks. The decoder consists of a strided convolution and a residual block. The decoder takes the output of the previous encoder to produce the LR features, which are then fed to the encoder of the next projection module. The reconstruction module takes the output of the encoder from each projection module, concatenates them, and produces the final SR results.

RBPN is specifically chosen as the generator for the proposed network because it contains modules that jointly use features across layers, known as back-projection [2]. It offers superior results by combining the benefits of the original MISR back-projection approach with Deep Back-Projection Networks (DBPNs), which perform SISR by estimating the SR frame using the LR frame through learning-based models. Combining these two techniques results in superior accuracy produced by the RBPN network [8, 9].

3.2 TecoGAN

In this network, the generator, denoted by G, generates high-resolution (HR) frames from low-resolution (LR) input frames. It takes the LR frames and the previously estimated HR frames as inputs and feeds them into the motion estimation module to obtain the optical flow. The optical flow is then used to warp the previous HR frames, which are fed into

convolutional modules to generate the HR frame [1]. The discriminator, denoted by D, is spatio-temporal based and its main role is to compare the generated HR frames with the ground truth.

This network also proposes a new loss function named ”Ping-Pong,” which focuses on the long-term temporal flow of the generated frames to make the results more natural without artifacts. Additionally, it has a relatively low number of parameters for a GAN network, approximately 3 million parameters, resulting in an inference time of around 42 ms [1]. The discriminator guides the generator to learn the correlation between the LR input and the HR targets. It penalizes the generator if the generated frames contain less spatial detail or unrealistic artifacts compared to the target HR and the original LR frames. The architecture of the discriminator is shown in FIGURE 2.

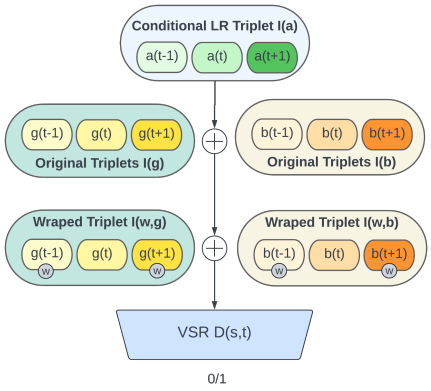

Figure 2: Discriminator Architecture

There is an issue that appears when super-resolving at large upscaling factors, which is usually seen with CNNs [10]. Therefore, the proposed network chose TecoGAN as a discriminator to mitigate the issue of a lack of finer texture details. The discriminator is trained to differentiate between super-resolved images and original photo-realistic images.

RBPN is specifically chosen as the generator for the proposed network because it contains modules that jointly use features across layers, known as back-projection [2]. It offers superior results by combining the benefits of the original MISR back-projection approach with Deep Back-Projection Networks (DBPNs), which perform SISR by estimating the SR frame using the LR frame through learning-based models.

Our proposed architecture, called RBPGAN, combines the strengths of RBPN and TecoGAN as the generator and discriminator, respectively. The main goal is to recover precise photo-realistic textures and motion-based scenes from heavily down-sampled videos, thereby improving temporal coherence while reducing model size. The architecture of the proposed network is shown in FIGURE 1.

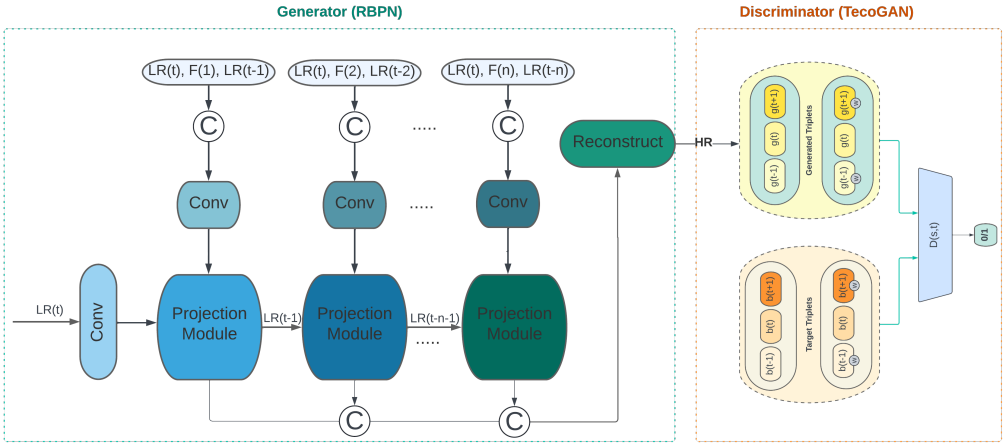

Figure 3: RBPGAN Architecture

4. DATASETS AND METRICS

4.1 Datasets

We conducted our training experiments using the Vimeo-90k dataset and the training dataset created by TecoGAN publishers, referred to as VimeoTecoGAN. For testing, we utilized the Vid4 and ToS3 datasets, with detailed descriptions available in TABLE 1. To generate the low-resolution (LR) frames from high-resolution (HR) input frames during training, we applied 4x down-sampling using bicubic interpolation, also known as the Gaussian Blur method. This approach enabled self-supervised learning by automatically creating input-output pairs without human intervention.

For testing, we obtained comparable assessment results on the Tears of Steel datasets (ToS3 scenes: room, bridge, and face) alongside the Vid4 dataset. To ensure consistency across all methods, we followed the procedures outlined in previous works [5, 11]. Specifically, we excluded spatial borders within 8 pixels of the image sides, adjusted the borders to ensure the LR input image was divisible by 8, and disregarded the first and last few frames for spatial and temporal metrics (first two and last two frames for spatial metrics, and first three and last two frames for temporal metrics) to accommodate inference requirements.

Additionally, we experimented with our own LR video sessions focusing on bodily motions. When comparing these sessions to other datasets and metric breakdowns, we found that our measures effectively captured human time perception.

4.2 Evaluation Metrics

While the visual results offer a first indication of the quality of our technique, quantitative assessments are critical for automated evaluations over greater numbers of samples. Because

Table 1: Details of the datasets used in the experiments.

Dataset	Resolution	#clips	Frames/clip	#Frames
Vimeo90K	448x256	13,100	7	91,701
VimeoTecogan	Varies	265	120	31,800
Vid4	Varies	4	Varies	684
ToS3	1280x534	3	233, 166, 150	549

ground-truth data is available, we will concentrate on the VSR assignment in this section. We give metrics evaluations of several models in relation to existing geographical metrics. We also justify and suggest two new temporal metrics for measuring temporal coherence. The usual criterion for evaluating the quality of super-resolution results mainly includes Peak signal-to-noise ratio (PSNR) and Structural index similarity (SSIM). PSNR is the ratio of an image's maximum achievable power to the power of corrupting noise that affects the quality of its representation. To calculate the PSNR of a picture, it must be compared to an ideal clean image with the highest potential power. Higher outcomes are preferable. A single SR frame's PSNR can be calculated as

$$PSNR = 10l(\frac{MAX^2}{MSE}) \tag{1}$$

where MAX is the color value's maximum range, which is commonly 255 and MSE is the mean squared error. Generally, a greater PSNR value indicates higher quality. While SSIM measures the similarity of structure between two corresponding frames using an uncompressed or distortion-free image as a baseline. A higher SSIM value indicates higher quality. PSNR may be more sensitive to Gaussian noise, whereas SSIM may be more sensitive to compression errors. Their values, however, are incapable of reflecting video quality for human vision. That implies, even if a video has a very high PSNR value, it may still be unpleasant for humans. As a result, deep feature map-based measures like LPIPS [12], can capture more semantic similarities. The distance between picture patches is calculated using LPIPS (Learned perceptual image patch similarity). Higher implies more distinct. Lower values indicate a closer match.LPIPS indicates the perceptual and semantic similarity to the ground truth. In other words, lower LPIPS means a more natural video. Additionally, ToF is used to calculate the pixel-wise difference of movements inferred from successive frames.

$$tOF = ||OF(b_{t-1}, b_t) - OF(g_{t-1}, g_t)|| \tag{2}$$

5. LOSS FUNCTIONS

The loss functions used while training our model are as follows:

1. GAN Loss (min-max loss):
   We use the Vanilla GAN loss, which is the simplest form of the GAN loss, for the adversarial

training. The generator tries to minimize it while the discriminator tries to maximize it.

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))] \quad \text{Eqn.1}$$

Here, $D(x)$ is the discriminator's estimate of the probability that real data instance x is real, and $D(G(z))$ is the discriminator's estimate of the probability that a fake instance is real. E is the expected value over all data instances.

2. Pixel loss:
   Minimizes the pixel-wise squared differences between Ground Truth and generated frames.

$$\|g_t - b_t\|_2 \quad \text{Eqn.2}$$

3. Ping Pong Loss:
   Proposed by TecoGAN model, effectively avoids the temporal accumulation of artifacts, and targets generating natural videos that are consistent over time. PP loss uses a sequence of frames with the forward order as well as its reverse. Using an input number of frames of length n, we can form a symmetric sequence $a_1, ...a_{n-1}, a_n, a_{n-1}, ...a_1$ such that when feeding it to the generator, the forward results should be identical to the backward result [1].

$$\sum_{t=1}^{n-1} \|g_t - g_t'\|_2 \quad \text{Eqn.3}$$

Here, the forward results are represented with g_t and the backward results with g_t'

4. Feature/perceptual Loss:
   Encourages the generator to produce features similar to the ground truth ones by increasing the cosine similarity of their feature maps. It ensures more perceptually realistic and natural generated videos. Our discriminator features contain both spatial and temporal information and hence are especially well suited for the perceptual loss.

$$1.0 - \frac{\Phi(I^g_{s,t}) * \Phi(I^b_{s,t})}{\left\|\Phi(I^g_{s,t})\right\| * \left\|\Phi(I^b_{s,t})\right\|} \quad \text{Eqn.4}$$

Where $I^g = \{g_{t-1}, g_t, g_{t+1}\}, I^b = \{b_{t-1}, b_t, b_{t+1}\}$

5. Warping Loss:
   Used while training the motion estimation network (F) that produces the optical flow between consecutive frames.

$$\sum \|a_t - W(a_{t-1}, \mathrm{F}(a_{t-1}, a_t))\|_2 \quad \text{Eqn.5}$$

Where $W()$ is the warping function, $F()$ is the flow estimator, and a_t is the LR frame in position t.

6. EXPERIMENTS

During the training process, GANs' generative and discriminative models interact with each other to achieve greater perceptual quality than other standard models. As a result, GANs are widely used in the field of Super Resolution. To handle large-scale and unknown degradation difficulties in VSR tasks, we rely on the remarkable ability of GAN models' deep feature learning. We also refer to the TecoGAN method's design and introduce the spatio-temporal adversarial structure to aid the discriminator's understanding and learning of the distribution of spatio-temporal information, avoiding the instability impact in the temporal domain that standard GANs suffer from. Additionally, we introduce a more accurate generator module based on the RBPN model into the TecoGAN design to ensure quality and improve temporal coherence.

In all our experiments, we focus on the 4× Super Resolution factor as it provides satisfactory results and requires a reasonable amount of training. We used a crop size of 32x32 and Gaussian downsampling. All experiments were conducted using the following specifications to enable the dense nature of the training phase: 64GB of DDR4 RAM, 2.80GHz Intel Core i9-10900F CPU, NVIDIA GeForce RTX 3090 (1 x 24 GB) GPU, and Ubuntu 20.04.3 LTS operating system.

We will now present and explain the experiments we conducted in sequence and later discuss their results comparatively.

First, we started by training and testing our two base models (TecoGAN and RBPN) to ensure their correctness and reliability before integrating them to produce our model. Then, we integrated them as discussed in Section 2. Subsequently, we performed experiments on our model with different parameters, loss functions, etc., until we reached the best outcome. The final model is then compared with other state-of-the-art models in terms of PSNR, SSIM, LPIPS, and ToF metrics.

1. **Experiment 1: Reduced RBPN Model Size**
   As discussed, RBPN is the base model we are using for our model's generator. We started by training and testing it. The model size was very large, and we encountered memory-related issues, so we reduced its size by decreasing the number of neighbor frames passed to the projection modules. This resulted in a decreased size and resolved our problems. The training of this experiment took around 1 hour per epoch, and we trained it for 150 epochs using the VimeoTecoGAN dataset and other parameters as in the original published model.
2. **Experiment 2: Lightweight TecoGAN**
   We trained and tested the TecoGAN model, which demonstrated adequate results with fewer parameters compared to other state-of-the-art models. We used one GPU for the first training

experiment of TecoGAN implemented using the TensorFlow framework. The results were encouraging, but the training took more than 170 hours to complete. Therefore, we restructured the network to be more lightweight and less computationally dense. The new implementation provides a model with a smaller size and better performance than the official TecoGAN implementation, as shown in TABLE 2. We also trained it using a more computationally powerful machine (with a 24GB GPU), and the results show that the reduced model has less training time than the official implementation by a factor of 6.7x. This implementation was done using the PyTorch framework to make it compatible with RBPN in the integration phase.

Table 2: Comparison between Official TecoGAN implementation and reduced model performance.

Methods	PSNR	SSIM	tOF (x10)	tLP (x100)	LPIPS (x10)	Processing time (ms/frame)
Official TecoGAN	25.57	-	1.897	0.668	1.623	41.92
Lightweight TecoGAN	26.030	0.802	0.199	0.510	0.156	41.92

3. **Experiment 3: Model Integration**
   After ensuring the correctness, reliability, and readiness of the two base models for the integration phase, we began integrating RBPN as the generator with the spatio-temporal discriminator from TecoGAN to create our GAN model and prepare it for some experiments. The integration was challenging due to many differences in functions' interfaces, dependencies used, training datasets, and the lack of generalization to fit any other dataset, as well as the coding style. We experimented in two ways: replacing the existing generator of TecoGAN with RBPN in the TecoGAN environment, and adding the spatio-temporal discriminator of TecoGAN to RBPN, transforming a feed-forward model to a generative model. After solving all issues, we produced our model: RBPGAN - Recurrent Back Projection Generative Adversarial Network.

We will now discuss the experiments done on RBPGAN (our model) to monitor the model's potential and test our hypothesis.

1. **Experiment 4.1**
   We used all the loss functions mentioned in the previous section (Ping Pong loss, Pixel loss, Feature loss, Warping loss, and GAN loss). We used 2 neighbor frames per frame, but due to the use of ping pong loss, this number is doubled to create the backward and forward paths. Therefore, the generator used 4 neighbor frames per frame. We trained both the generator and discriminator simultaneously from the beginning, using the VimeoTecoGAN dataset. Training took around 3.5 days using the specifications mentioned.
2. **Experiment 4.2**
   We used the same loss functions as in Experiment 4.1, except the Ping Pong loss, to observe its effect on the results. We used 3 neighbor frames per frame, started the training of the generator and discriminator together, and used the same dataset and other parameters as in Experiment 4.1. The training took around 3 days.
3. **Experiment 4.3**
   This experiment is the same as Experiment 4.2, except that we first trained the generator solely

for some epochs and then started the training of the GAN using this pre-trained part. The training took around 3 days using the same dataset, number of neighbors, and other parameters.

4. **Experiment 5**
   We trained the RBPN model with the same number of neighbors, crop size, dataset, and other unifiable parameters as we did for our model in the previous experiments to ensure a fair comparison between it and our model.

7. RESULTS

In this section, we present the results and metrics evaluation of our conducted experiments. The performance of our model is assessed using the Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), temporal Optical Flow (tOF), and Structural Similarity Index Measure (SSIM) metrics across different datasets.

TABLE 3 provides a comparative analysis of the experiments conducted on the **Vid4** dataset. The results indicate that Experiment 4.2 achieved the best performance with a PSNR of 25.74, an LPIPS of 1.44, a tOF of 2.35, and an SSIM of 0.762, outperforming the other experiments in all metrics.

Similarly, TABLE 4 shows the results for the **ToS3** dataset. In this case, Experiment 4.1 achieved the highest PSNR of 32.89, while Experiment 4.2 demonstrated superior performance in terms of LPIPS, tOF, and SSIM metrics, with values of 0.69, 1.64, and 0.880, respectively.

Table 3: Comparative analysis between all conducted experiments on our model for the **Vid4** dataset.

Metric Name	Experiment 4.1	Experiment 4.2	Experiment 4.3
PSNR	25.58	**25.74**	25.56
LPIPS	1.47	**1.44**	1.45
tOF	2.46	**2.35**	2.40
SSIM	0.756	**0.762**	0.751

Table 4: Comparative analysis between all conducted experiments on our model for the **ToS3** dataset.

Metric Name	Experiment 4.1	Experiment 4.2	Experiment 4.3
PSNR	**32.89**	32.85	32.78
LPIPS	0.78	**0.69**	0.75
tOF	**1.60**	1.64	1.62
SSIM	0.872	**0.880**	0.869

Overall, Experiment 4.2 yields the best results collectively, and therefore we will use it for comparison with state-of-the-art models, as shown in TABLE 5 and TABLE 6. FIGURE 4

and FIGURE 5 present examples from the Vid4 dataset illustrating the performance of our model.

Figure 4: Walk image. Top: Low Resolution (LR), Bottom: RBPGAN output.

Figure 5: Calendar image. Top: Low Resolution (LR), Bottom: RBPGAN output.

Table 5: Comparison between Experiment 4.2 (Ours) and state-of-the-art methods on the **Vid4** dataset.

Metric	Experiment 4.2 (Ours)	TecoGAN	RBPN (3 neighbors)	BIC	ENet	DuF
PSNR	25.74	25.57	26.71	23.66	22.31	**27.38**
LPIPS	**1.44**	1.62	2.00	5.04	2.46	2.61
tOF	2.35	1.90	2.19	5.58	4.01	**1.59**
SSIM	**0.756**	0.770	0.801	NA	NA	0.815

Table 6: Comparison between Experiment 4.2 (Ours) and state-of-the-art methods on the **ToS3** dataset.

Metric	Experiment 4.2 (Ours)	TecoGAN	RBPN (3 neighbors)	BIC	ENet	DuF
PSNR	32.85	32.65	34.32	29.58	27.82	**34.60**
LPIPS	**0.69**	1.09	1.10	4.17	2.40	1.41
tOF	1.64	1.34	1.54	4.11	2.85	**1.11**
SSIM	**0.880**	0.892	0.915	NA	NA	NA

8. DISCUSSION

Our hypothesis aimed to merge the highly realistic output of RBPN with the temporally coherent output of TecoGAN to achieve a smooth, high-quality output for our model. In the results section, we presented the metrics of our model's output, demonstrating higher quality compared to TecoGAN and greater temporal cohesion compared to RBPN. Although we hoped our model would generate HR videos with qualities equivalent to RBPN, the quality of the generated videos was not less than that of RBPN and was higher than that of TecoGAN. This outcome is attributed to the GAN training incorporated into our model and the significant need for fine-tuning. Due to time and hardware limitations, we trained all models for the same number of epochs, despite some models using different datasets and GANs typically requiring more time to achieve convergence. Specifically, we trained our GAN model for only 51 epochs, although it might have achieved higher stability with more epochs.

When assessing our model using the LPIPS metric, which measures temporal cohesion, we found that our model surpasses both base models, TecoGAN and RBPN. This improvement is due to the discriminator aiding the generator in learning further temporal cohesion. Further enhancements for other metrics could have been achieved with more time for fine-tuning our GAN model. We could have explored different combinations of loss functions and varied learning rates to reach the optimal training conditions.

9. LIMITATIONS AND FURTHER WORK

Our model required more powerful computational resources; however, we were limited to a 2-GPU machine with 64GB of memory. This limitation extended the duration of our experiments, as each took longer than usual on the available hardware. Moreover, while our technique produces extremely realistic results for a wide variety of natural scenes, in some cases—such as under-resolved faces and text in VSR, or tasks with dramatically differing motion between two domains—our method can provide temporally coherent but suboptimal details. It would be interesting to combine our technique with motion translation from contemporaneous work in these instances [13]. Therefore, we recommend using different downsampling methods to introduce more generalization to the model and training the model on more augmented datasets that focus on faces and text.

Although RBPGAN combines temporal coherence and high video accuracy, several ideas could be explored to improve it. Visual images often prioritize the foreground, which frequently includes subjects such as individuals, over the background. To enhance perceptual quality, we could separate the foreground and background and have RBPGAN perform ”adaptive VSR” by applying different rules for each. For example, we might use a larger number of frames to extract features from the foreground compared to the background. Additionally, there is ongoing research on accelerator techniques to speed up network training and inference time, potentially leading to real-time VSR transitions. The most promising techniques we found include Convolutional Computation Acceleration, Efficient Upsampling, and Batch Normalization Fusion. We anticipate that these methods will provide a useful basis for a wide range of generative models for real-time HR video generation.

Our model shows great promise by combining the strengths of RBPN and TecoGAN, yet further fine-tuning and extended training could potentially yield even better performance across all metrics.

10. CONCLUSION

We successfully validated our hypothesis and achieved the highest results in terms of temporal coherence. While current adversarial training produces generative models across various fields, temporal correlations in the generated data have received far less attention. Our approach focuses on enhancing learning objectives and presents a temporally self-supervised method to address this gap. For sequential generation tasks such as video super-resolution and unpaired video translation, natural temporal shifts are crucial.

The reduced Recurrent Back-Projection Network in our model extracts information from each context frame, then combines and processes this information within a refinement framework based on the back-projection concept in multi-frame super resolution. The inter-frame motion is estimated concerning the target, which aids in producing more temporally coherent videos.

Our approach not only demonstrates significant improvements in temporal coherence but also sets the stage for further advancements in video generation tasks. By refining our methods

and exploring additional computational resources, we can continue to push the boundaries of video super-resolution and related fields.

11. ACKNOWLEDGMENTS

We wish to acknowledge the help provided by the technical and support staff in the Computer Science and Engineering department of the American University in Cairo (AUC). We would also like to express our deepest appreciation to our supervisors Prof. Dr. Cherif Salama, Prof. Dr. Hesham Eraqi, and Prof. Dr. Moustafa Youssef, who guided us through the project.

References

[1] Chu M, Xie Y, Leal-Taixé L, Thuerey N. Temporally Coherent Gans for Video Super-Resolution (tecogan). 2018. ArXiv preprint: https://arxiv.org/pdf/1811.09393

[2] Haris M, Shakhnarovich G, Ukita N. Recurrent Back-Projection Network for Video Super-Resolution. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019:3897-3906.

[3] Wang X, Chan KC, Yu K, Dong C, Change Loy C. EDVR: Video Restoration With Enhanced Deformable Convolutional Networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2019.

[4] Lucas A, Lopez-Tapia S, Molina R, Katsaggelos AK. Generative Adversarial Networks and Perceptual Losses for Video Super-Resolution. IEEE Trans Image Process. 2019;28:3312-3327.

[5] Jo Y, Oh SW, Kang J, Kim SJ. Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation. In Proceedings of the IEEE conference on Comput Vis Pattern Recognit. 2018:3224-3232.

[6] Dieng AB, Kim Y, Rush AM, Blei DM. Avoiding Latent Variable Collapse With Generative Skip Models. 2018. ArXiv preprint: https://arxiv.org/pdf/1807.04863

[7] Yi P, Wang Z, Jiang K, Jiang J, Ma J. Progressive Fusion Video Super-Resolution Network via Exploiting Non-local Spatio-Temporal Correlations. In Proceedings of the IEEE/CVF Int Conf Comput Vis. 2019:3106-3115.

[8] Irani M, Peleg S. Improving Resolution by Image Registration. Cvgip: Graphical Models and Image Processing. 1991;53:231-239.

[9] Irani M, Peleg S. Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency. J Vis Commun Image Represent. 1993;4:324-335.

[10] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, 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. 2017:4681-4690.

[11] Sajjadi MS, Bachem O, Lucic M, Bousquet O, Gelly S. Assessing Generative Models via Precision and Recall. Adv Neural Inf Process Syst. 2018;31.

[12] Zhang H, Goodfellow I, Metaxas D, Odena A. Self-Attention Generative Adversarial Networks. In International conference on machine learning. PMLR. 2019;7354-7363).

[13] Aberman K, Weng Y, Lischinski D, Cohen-Or D, Chen B. Unpaired Motion Style Transfer From Video to Animation. ACM Trans. Graph. 2020;39:64.