ControlNetVSR: Video Super-Resolution using ControlNet

Naveen Reddy Dyava
Columbia University
nd2794@columbia.edu

Abstract

This report introduces a novel approach to address video super-resolution (VSR) by employing Diffusion Models (DM), called ControlNetVSR. Leveraging the Stable Diffusion X4 upscaler, a pre-trained DM designed for single image super-resolution, we adapt it to the VSR context using ControlNet. Through the integration of the ControlNet module, which harnesses spatially aligned texture data from neighboring low-resolution frames, we enable Temporal Conditioning. This mechanism guides the synthesis process, ensuring the production of temporally consistent images. To make ControlNet suitable for our specific requirements, we modify the original ControlNet architecture. Our method only enhances the reconstruction quality but also significantly improves the perceptual fidelity of the resultant frames compared to conventional independent image super-resolution method employing Stable Diffusion X4 upscaler. The code is available at https://github.com/nrdyava/ControlNetVSR.

1. Introduction & Related Work

Video super-resolution (VSR) involves the reconstruction of a high-resolution (HR) video from a given low-resolution (LR) video sequence [9]. The advancement of deep learning research has facilitated the utilization of deep learning techniques for VSR [2], [3], [7], resulting in substantial advancements in the field. However, many existing deep learning-based approaches exhibit a trade-off between reconstruction quality and perceptual quality. This trade-off, known as the perception-distortion trade-off, suggests that enhancing reconstruction quality often comes at the expense of perceptual quality. Methods emphasizing high reconstruction quality typically yield high Peak Signal-to-Noise Ratio (PSNR) values (where higher values indicate better performance) but may lack photorealism and consequently exhibit high Learned Perceptual Image Patch Similarity (LPIPS [16]) values (where lower values are preferable).

The recent advancements of Diffusion Models (DMs) [11] in producing high-fidelity images have sparked interest in utilizing DMs for single-image super-resolution (SISR). These models have demonstrated the capability of DMs to generate detailed and photorealistic textures. However, directly applying SISR techniques to video super-resolution (VSR) isn't a preferred choice as it can result in temporally inconsistent frames.

In this project, we propose and implement a method for video super-resolution (VSR) employing Stable Diffusion [11] X4 upscaler and the ControlNet. Inspired by the efficacy of ControlNet [15] in incorporating spatially conditional controls into image synthesis, we adapt it to leverage spatially aligned texture information from neighboring low-resolution frames. This utilization guides the generation process, aiming to produce high-fidelity and temporally consistent video frames.

During each stage of the inference process, we employ motion estimation and motion compensation techniques to extract texture details from neighboring frames. This extracted information is fed into the ControlNet. Additionally, we spatially condition the synthesis process by concatenating the latent space embedding with a low-resolution input image. However, the original architecture of ControlNet isn't directly applicable to this VSR scenario due to dimensionality concerns. To address this, we modify the architecture of the input convolutional module, ensuring its compatibility and effectiveness for VSR applications. We train and test our model on REDS [10] dataset.

In summary, our main contributions are: We present ControlNetVSR, which utilizes Stable Diffusion X4 upscaler and the ControlNet. We propose a method to extract texture information from adjacent frames and use it to condition the frame synthesis. We modify the intial convolutional module of the ControlNet to make it suitable for VSR.

2. Methodology

2.1. A brief background on diffusion

Diffusion models [4] convert any complex data distribution into Gaussian Distribution and the recover data from it.

Figure 1. Image shows 4 consecutive frames (centre crops) for a video in the REDS4 Dataset. First row shows (128 x 128) low resolution image. Second row shows ground truth high resolution images (512 x 512). The third rows shows the high resolution images (512 x 512) synthesized by our Model.

They achieve it using Markov Chains. In the forward process random Gaussian Noise is added to the input data at each time step until the last time step. In the reverse process, a U-Net model predicts the amount of noise added from the starting to the current time step. Using the predicted noise, an estimated original image can be recovered using weighted subtraction of noise from the image at a given time step. A more efficient class of diffusion models are latent diffusion models [11]. Images are first converted into low dimensional latent space using the encoder of a pre-trained Variational Auto-encoder and then the same procedure as explained before is used to generate the latent vector of the original sample. After the sampling is done, the generated latent vector is passed through the decoder of a Variational Auto-encoder to transform the latent vector to image space.

2.2. A brief background on Stable Diffusion X4 upscaler

Stable Diffusion [11] X4 upscaler is a pre-trained Single Image Super-Resolution (SISR) model to upscale images by 4 times. It contains a variational auto-encoder which shrinks the dimension of any image vector by 4 times and converts it to latent space. The U-Net model in this case is different from other diffusion models. It takes 7 channels as input and outputs 4 channels as output. The first 4 channels of the input correspond to the latent vector at the given time and the remaining 3 channels correspond to the low-resolution image. By concatenating the low-resolution image with the latent vector, it can be used to condition the generation process. To upscale a 128 X 128 image to 512 X 512 image, a latent space of size 128 X 128 is used. During training a 512 X 512 image is passed through the encoder and the U-Net [12] will try to predict that latent embed-

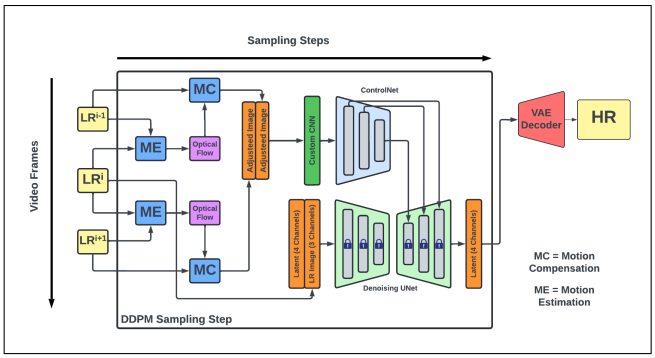

Figure 2. Architecture diagram of the model. Sampling steps from t=T to t=0 (left to right). Frame order from i = 1 to i = N (top to bottom). LR means a lower resolution image and HR means a Higher Resolution Image. Only ControlNet is trained and all other layers are frozen.

ding. During inference, the predicted latent will be passed through the decoder and a 512 x 512 image is generated.

2.3. Adding Temporal Texture Guidance using ControlNet

In the previous section we talked about adding spatial guidance to the diffusion process using the low resolution image. In order to generate temporally consistent frames, we need to add texture information from the adjacent frames. The hypothesis is that the adjacent frames will reveal information which could have been occluded in the current frame. To add temporal conditioning we make use of Control-Net [15]. Recent research in applications of ControlNet has shown tremendous potential in adding spatial controls to image generation. We condition the U-Net with texture guidance using ControlNet.

We get the texture information from the adjacent frames. We concatenate them and pass them into the ControlNet. The ControlNet architecture mentioned in the original ControlNet paper is not suitable for VSR task. Inorder to make it flexible for VSR, we remove the down sampling convolution block from the ControlNet and replace it with a custom CNN module that can take two 128 X 128 images and transform them into suitable dimension for the downstream ControlNet blocks.

2.4. Motion Estimation & Compensation

To obtain the texture information from the adjacent low-resolution frames of the current image. But due to the motion of objects in the images, we have to align them with the current frame in order to use them. First we estimate the motion between frames using optical flow from current frame to each frame. Then we warp these adjacent image using the optical flow values to align them with the current frame. These two adjusted adjacent images are stacked together and fed as input to the ControlNet. We use RAFT [14] Large Model for calculating Optical Flow.

2.5. Dataset & Metrics

We use the REDS dataset. REDS [10] is a realistic and dynamic scene dataset containing 300 video sequences. Each sequence has 100 frames at 1280 × 720 resolution. Following the work in [2], [3], we use sequences 000, 011, 015, and 020 for testing and all the others for training. We LPIPS, DISTS to evaluate the perceptual quality and PSNR and SSIM to measure reconstruction quality. We also use Warping Error (WE) [6] to evaluate of temporal consistency.

2.6. Training Process

The full architecture of the model is presented in the Figure 2. During training process we randomly sample a 128x128 image from the low-resolution images and corresponding 512x512 images from the high resolution images. We also sample adjacent frames from low-resolution images which are to the left and right of the current frame. We pass the high resolution image through the encoder of the auto-encoder and obtain a latent embedding of (128x128x4) dimension. Next we calculate the optical flow of the left and right frames with respect to the current frame. We use the optical flow to warp the left and right frames. These two frames are concatenated and passed into the ControlNet. At the same time we sample a random time step from 0 to maximum number of time steps (T = 1000). We calculate noise for this time step and add it to the latent embedding of the high resolution image. We concatenate the low-resolution image with the noised latent embedding and predict the noise added using a forward pass through the U-Net. The difference between calculated and estimated noise is used to calculate loss. We use MSE loss for this purpose. We only train the ControlNet [15] module and freeze every other weight in the model architecture. Due to lack of computational resources, we only perform 4 epochs of training. We use AdamW [5] optimizer with learning rate of 10^{-5}. We used a batch size of 2 on a Nvidia L4 instance and trained it for 2 days.

2.7. Inference

During inference we do only 50 sampling steps. We decided 50 to be optimal number of steps after running multiple experiments and doing cost-benefit analysis. Unlike training we don't have a high resolution image during inference. We start from the random noise and we keep predicting next latent vector. We concatenate the low resolution image with latent embedding and use it as a spatial conditioning. Also we use the texture information from the adjacent frames and pass it as an input to the ControlNet. After

50 steps, we pass the latent embedding through the decoder and we obtain a 512X512 image.

3. Experimental Results & Discussion

In the Table 1, we can see the metrics for different experiments. The SD-X4 means the the Stable Diffusion X4 Upscaler is run for each frame independently and the metrics are evaluated on the test set videos. The Ours experiment means the our ControlNetVSR model. The number in the bracket indicates that the number of inference steps.

Table 1. Metrics for Various Experiments and comparision with state of the art methods

Experiment	PSNR	SSIM	LPIPS	DISTS	WE
SD-X4	21.84	0.57	0.27	0.24	2.69
Ours (50)	23.18	0.65	0.19	0.16	**1.73**
Ours (100)	22.87	0.64	0.18	0.15	1.87
Ours (200)	22.67	0.63	0.18	0.15	1.94
BasicVSR++ [3]	32.38	0.907	0.131	0.068	-
RVRT [8]	**32.74**	0.911	0.128	0.067	-
StableVSR [13]	29.97	0.80	**0.097**	**0.045**	-

3.1. Baseline

We choose Stable Diffusion X4 upscaler (SD-X4) as a baseline. In this experiment, the SD-X4 model is run independently on each frames using Single Image Super-Resolution (SISR) mode. The reason for choosing this as baseline is because we stated that SISR will create temporally incoherent frames and we hypothesised that temporal texture guidance using ControlNet will enhance the VSR performance. In the subsequent sections, we comapre the performance of ControlNetVSR models to the baselines and see if our hypothesis is empirically valid. Note that all the evaluations are done on 512 X 512 centre crops of test images.

3.2. Temporal texture guidance using ControlNet improves VSR Performance

As we can see in the Table 1, the ControlNetVSR model (with 50 inference steps) performs better than the SD-X4 model. There is a marginal increase in the PSNR and SSIM metrics (the higher the better), but there is a big improvement in the LPIPS and DISTS metrics (the lower the better). We can also see improvement in Warping Error (WE). This suggests that use of temporal texture guidance using ControlNet signigicantly improves the perceptual quality of VSR. The same statements hold true for comparison of SD-X4 to ControlNetVSR with more number of inference steps. This result empirically proves our hypothesis.

3.3. Qualitative Analysis

In the Figure 1, the first row is low resolution frames of size 128X128 for consecutive frames in a video. In the second row we have ground truth high resolution frames (512X512). In the third row we have frames which are upscale using ControlNetVSR using 50 steps. Qualitatively we can see that the perceptual clarity is very good. Although some aspects like text and fine details like engravings on pillars don't get captured. But these problems can be resolved by running the model for more number of steps. However, we don't find much benefit in terms of improvement of perceptual quality metrics. So we limited ourself to 50 diffusion steps.

3.4. Perceptual Clarity Imporves with inference steps

We can see in the Table 1 the with the increase in number of inference steps, the LPIPS and DISTS value comes down. This suggests improvement in perceptual clarity. We can also observe that the values of PSNR and SSIM comes down with inference steps. This supports the argument about the trade-off [1] between perceptual clarity and the reconstruction clarity.

3.5. Drawbacks

Although this improves upon the SISR method, this method doesn't beat the State-of-the-art results. Also ControlNetVSR is not good at resolving human faces. Because of computational constraints, we couldn't more experiments to test its full potential. Even if this method works well and beats the state-of-the-art, this has very less practical relevance. Running 50 diffusion steps per frame is time consuming and also very less energy efficient. This experiment is only intended for academic purposes.

4. Conclusion

We introduced an approach to improve the quality of video super-resolution (VSR) by integrating diffusion models and ControlNet, resulting in ControlNetVSR. Our method involved adapting a pre-trained diffusion model originally designed for single-image super-resolution to the VSR context by incorporating Temporal Texture Conditioning through the ControlNetModule. Furthermore, we refined the ControlNet architecture to better suit the requirements of VSR. Through quantitative and qualitative analyses, we demonstrated that our model enhances perceptual quality in VSR. Additionally, we addressed the limitations of our model in the discussion section.

References

[1] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. *CoRR*, abs/1711.06077, 2017. 4

[2] 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. *CoRR*, abs/2012.02181, 2020. 1, 3

[3] Kelvin C. K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. *CoRR*, abs/2104.13371, 2021. 1, 3, 4

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3

[6] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. *CoRR*, abs/1808.00449, 2018. 3

[7] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention, 2022. 1

[8] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention, 2022. 4

[9] Hongying Liu, Zhubo Ruan, Peng Zhao, Fanhua Shang, Linlin Yang, and Yuanyuan Liu. Video super resolution based on deep learning: A comprehensive survey. *CoRR*, abs/2007.12928, 2020. 1

[10] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1996–2005, June 2019. 1, 3

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2

[13] Claudio Rota, Marco Buzzelli, and Joost van de Weijer. Enhancing perceptual quality in video super-resolution through temporally-consistent detail synthesis using diffusion models, 2023. 4

[14] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020. 3

[15] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1, 3

[16] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 1