Video Super-Resolution using Recurrent Back-Projection Network

Mohit Kumar(19825)
Kaushik Kukadiya(19762)
MTech Artificial Intelligence

Indian Institute of Science Bangalore

December 9, 2022

Table of Contents

Introduction

- The goal of super-resolution (SR) is to enhance a low resolution (LR) image to a higher resolution (HR) image by filling in missing fine-grained details in the LR image.
- SISR super-resolves LR_t by utilizing spatial information inherent in the frame, independently of other frames in the video sequence. However, this technique fails to exploit the temporal details inherent in a video sequence resulting in temporal incoherence.
- MISR utilizes the missing details available from the neighboring frames $LR_{t-n}, \ldots, LR_t, \ldots, LR_{t+n}$ and fuses them for super-resolving LR_t.
- Convolutional neural networks (CNNs) have outperformed traditional approaches in terms of widely-accepted image reconstruction metrics such as peak signal to noise ratio (PSNR) and structural similarity (SSIM).

- However, CNNs lose finer texture details when superresolving at large upscaling factors
- A crucial aspect of an effective VSR system is the ability to handle motion sequences, which are often integral components of videos.
- Our method is inspired by recurrent back-projection networks (RBPNs).
- Back-projection iteratively calculates residual images as reconstruction error between a target image and a set of neighboring images. The residuals are then back-projected to the target image for improving super-resolution accuracy.
- The multiple residuals enable representation of subtle and significant differences between the target frame and its adjacent frames, thus exploiting temporal relationships between adjacent frames

- We build a model that leverages RBPN, which is based on the idea of integrating SISR and MISR in a unified VSR framework using back-projection which enables it to extract details from neighboring frames.
- Pixel-wise loss functions like L1 loss, used in RBPN struggle to handle the uncertainty inherent in recovering lost high-frequency details such as complex textures that commonly exist in many videos.
- We use a four-fold (MSE, perceptual, MAE, and TV) loss function. MSE loss focuses on optimizing perceptual similarity instead of similarity in pixel space. Also, we use a denoising loss function called TV loss.

Losses

$$MSE_t = \frac{1}{WH} \sum_{x=0}^{W} \sum_{y=0}^{H} \left((HR_t)_{x,y} - G_{\theta_G} (LR_t)_{x,y} \right)^2$$

where, $G_{\theta_G}(LR_t)$ is the estimated frame SR_t. W and H represent the width and height of the frames respectively.

$$MAE_t = \frac{1}{WH} \sum_{x=0}^{W} \sum_{y=0}^{H} \left| \left((HR_t)_{x,y} - G_{\theta_G} (LR_t)_{x,y} \right) \right|$$

where, $G_{\theta_G}(LR_t)$ is the estimated frame SR_t. W and H represent the width and height of the frames respectively.

$$\text{PerceptualLoss}_t = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\begin{array}{l} VGG_{i,j} (HR_t)_{x,y} - \\ VGG_{i,j} (G_{\theta_G} (LR_t))_{x,y} \end{array} \right)^2$$

where, $VGG_{i,j}$ denotes the feature map obtained by the j^{th} convolution (after activation) before the i^{th} maxpooling layer in the VGG-19 network. $W_{i,j}$ and $H_{i,j}$ are the dimensions of the respective feature maps in the VGG-19 network.

TV loss is defined as follows:

$$\frac{1}{WH}\sum_{i=0}^{W}\sum_{j=0}^{H}\sqrt{\begin{array}{l}\left(G_{\theta_G}\left(LR_t\right)_{i,j+1,k}-G_{\theta_G}\left(LR_t\right)_{i,j,k}\right)^2+\\ \left(G_{\theta_G}\left(LR_t\right)_{i+1,j,k}-G_{\theta_G}\left(LR_t\right)_{i,j,k}\right)^2\end{array}}$$

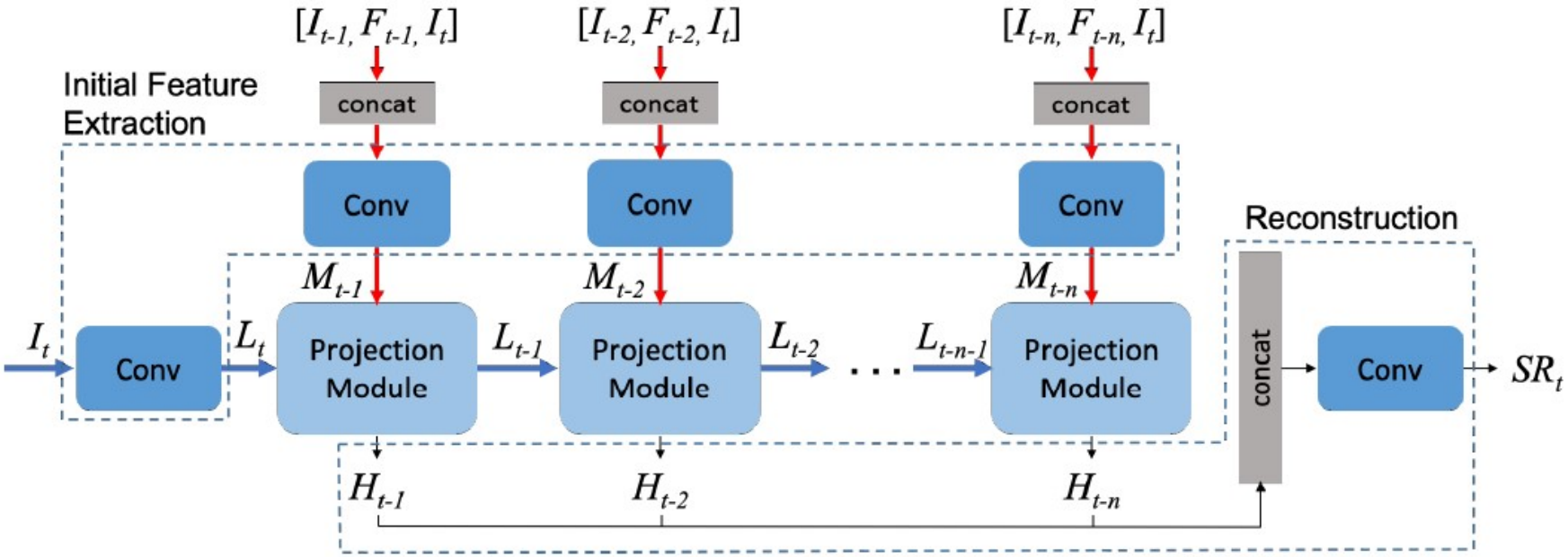
We define our overall loss objective for each frame as the weighted sum of the MSE, adversarial, perceptual, and TV loss components. The total loss of an input sample is the average loss of all frames.

$$\begin{aligned}\text{Loss}\left(SR_t\right) = &\alpha \times \text{MSE}\left(SR_t, HR_t\right)\\ &+\beta \times \text{PerceptualLoss}\left(SR_t, HR_t\right)\\ &+\gamma \times \text{MAE}\left(SR_t, HR_t\right)\\ &+\delta \times \text{TV Loss}\left(SR_t, HR_t\right)\end{aligned}$$

where,

$\alpha, \beta, \gamma, \delta$ are weights set as $0.5, 6 \times 10^{-3}, 0.5$ and 2×10^{-8} respectively.

Model Architecture



RBPN has two approaches that extract missing details from different sources: SISR and Multi Image SR (MISR)

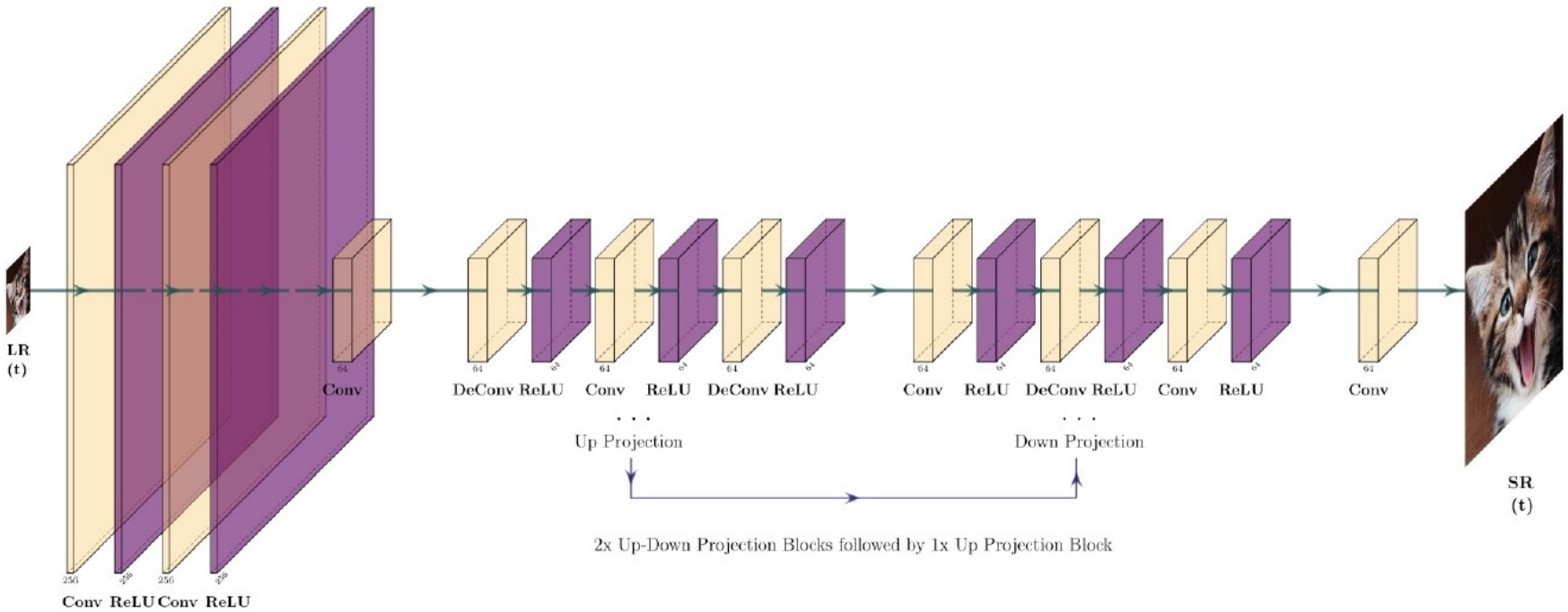


Figure: DBPN architecture for SISR, where we perform up-down-up sampling using 8×8 kernels with a stride of 4 and padding of 2. Similar to the ResNet architecture above, the DBPN network also uses Parametric ReLUs as its activation functions.

Enlarges LR frame independently of other frames

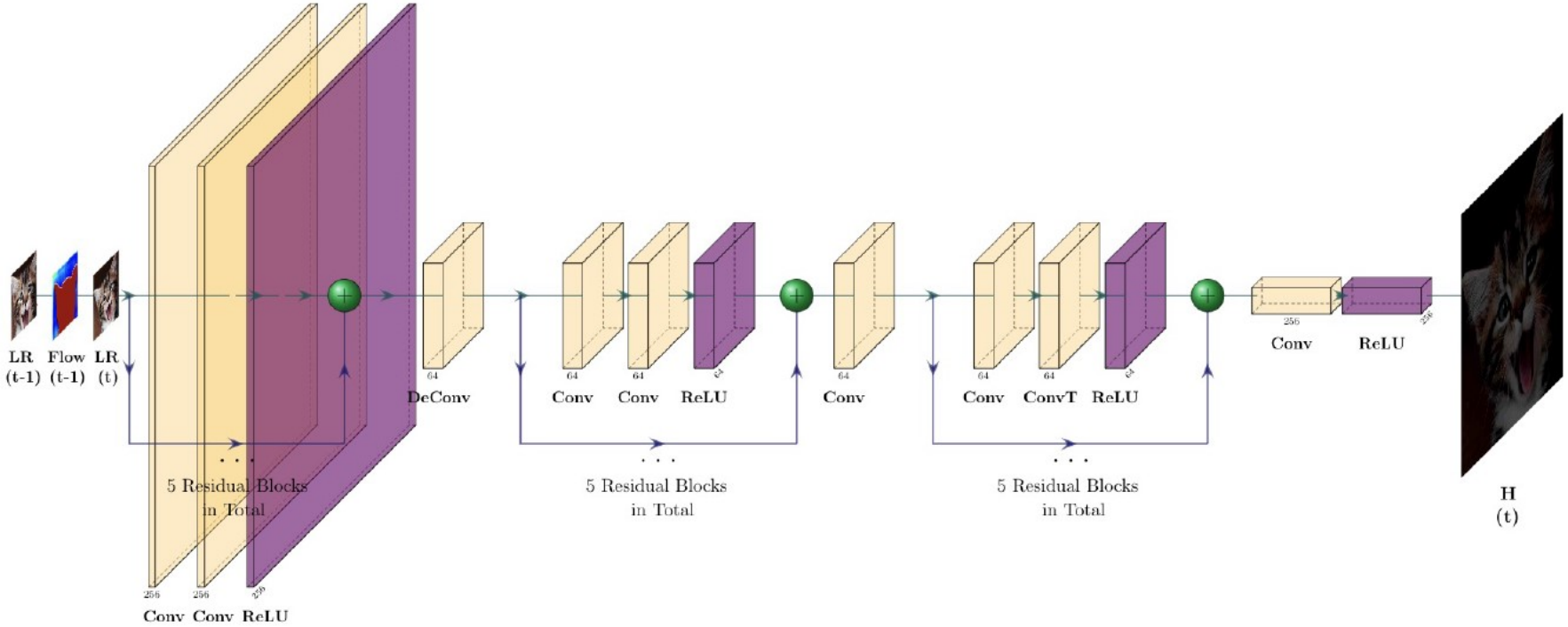
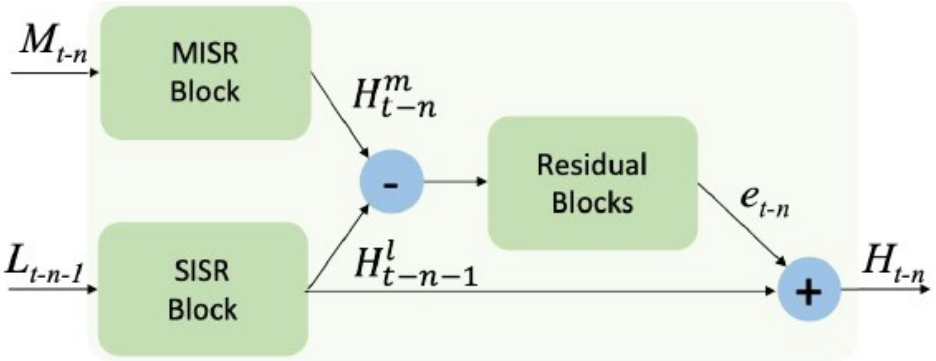
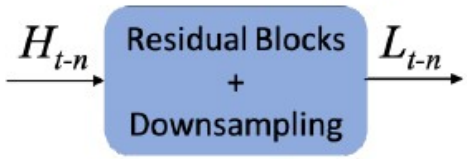


Figure: ResNet architecture for MISR that is composed of three tiles of five blocks where each block consists of two convolutional layers with 3×3 kernels, a stride of 1 and padding of 1. The network uses Parametric ReLUs [18] for its activations.

Computes residual features from a pair of input-to-neighbor frames and flow maps



(a) Encoder (the back-projection)



(b) Decoder

Encoder: $\mathrm{H}_{t-n} = \mathrm{Net}_E\left(L_{t-n-1}, M_{t-n}; \theta_E\right)$

Decoder: $\mathrm{L}_{t-n} = \mathrm{Net}_D\left(H_{t-n}; \theta_D\right)$

The encoder module Net_E:

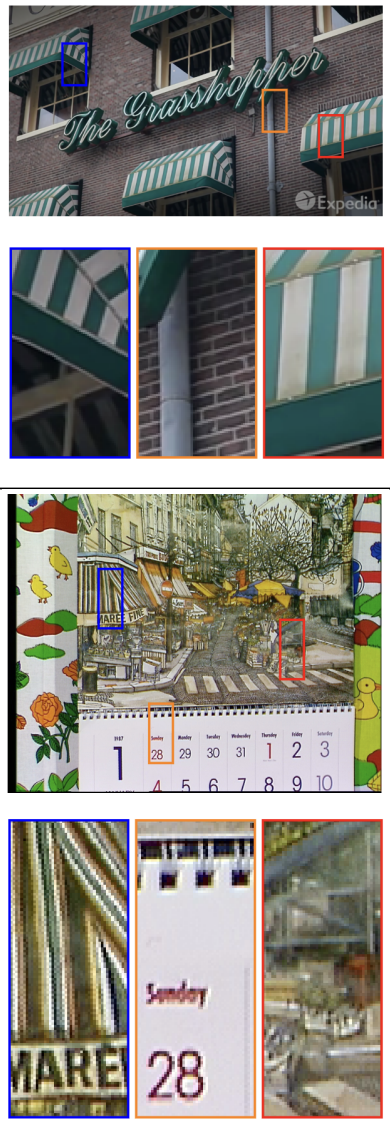
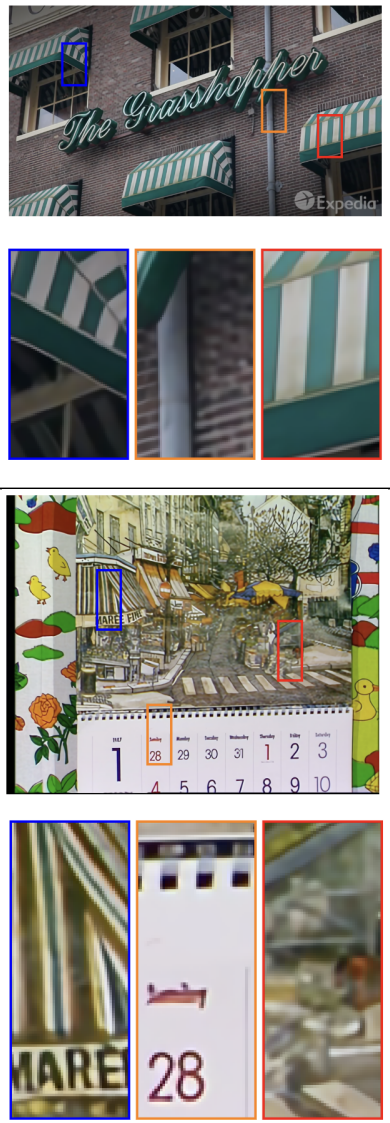
SISR upscale: $H^l_{t-n-1} = \mathrm{Net}_{sisr}\left(L_{t-n-1}; \theta_{sisr}\right)$

MISR upscale: $H^m_{t-n} = \mathrm{Net}_{misr}\left(M_{t-n}; \theta_{misr}\right)$

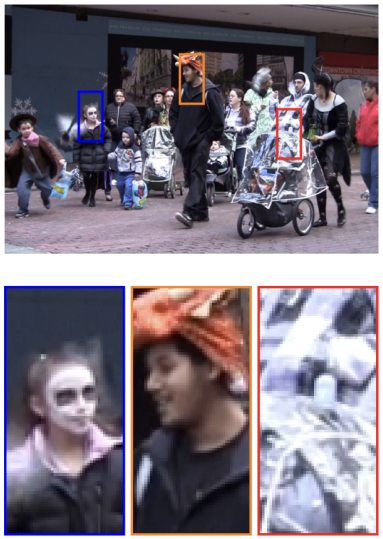
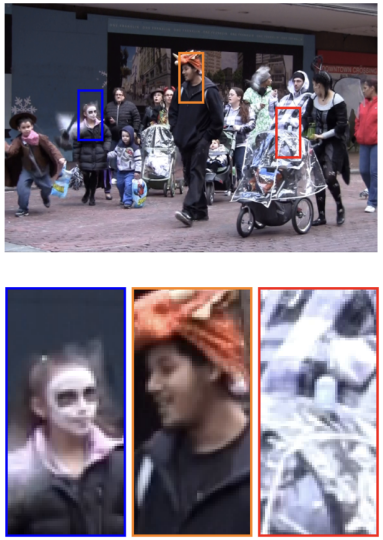
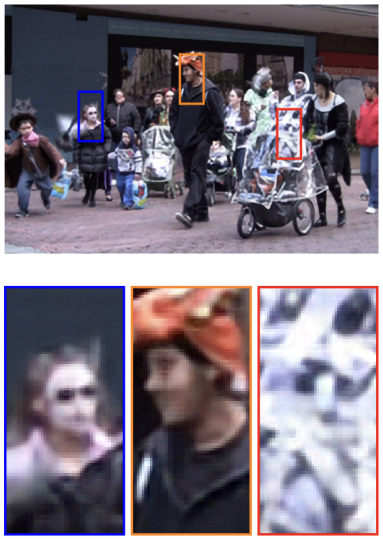
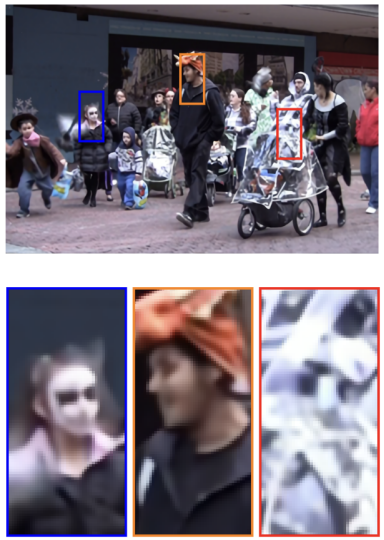
Residual: $e_{t-n} = \mathrm{Net}_{\mathrm{res}}\left(H^l_{t-n-1} - H^m_{t-n}; \theta_{\mathrm{res}}\right)$

Output: $H_{t-n} = H^l_{t-n-1} + e_{t-n}$

Results

Ground Truth	RBPN+ MAE loss	RBPN+ MSE loss	RBPN+ 4 fold loss
			

Results(cont.)

Ground Truth	RBPN+ MAE loss	RBPN+ MSE loss	RBPN+ 4 fold loss
			

Results(cont.)

Test Video	Metric	RBPN + MAE Loss	RBPN +MSE Loss	RBPN + Four-fold-loss
AMVTG	PSNR	27.45	23.85	25.89
	SSIM	0.860	0.668	0.790
	VMAF	86.01	69.41	82.17
CACT	PSNR	33.49	29.50	33.05
	SSIM	0.950	0.893	0.949
	VMAF	98.3	88.27	97.93
Calender	PSNR	22.27	20.42	22.17
	SSIM	0.809	0.704	0.803
	VMAF	86.53	65.04	84.01
CITY	PSNR	26.18	24.73	26.15
	SSIM	0.814	0.712	0.809
	VMAF	82.40	66.82	80.30
Foliage	PSNR	24.69	23.14	24.52
	SSIM	0.790	0.710	0.785
	VMAF	87.10	75.44	85.91
Jvc	PSNR	28.52	26.11	28.30
	SSIM	0.911	0.838	0.906
	VMAF	88.82	73.41	86.93
Walk	PSNR	29.21	26.86	29.00
	SSIM	0.915	0.866	0.913
	VMAF	93.52	81.26	92.47

Our Contributions

- We trained the RBPN model from scratch once by using MSE loss in place of the MAE loss originally used in training the RBPN.
- We then trained the RBPN by incorporating the Perceptual, MSE and Total Variation losses on top of the MAE loss.
- We trained for 4 epochs on the Vimeo dataset and evaluated on the Vid4 and SPMCS datasets.
- Since training time for every epoch was close to 6 hours, we did not train for more epochs.
- We observed that the results we got were worse than the pretained RBPN model with MAE loss on most of the video sequences across the three datasets.

Conclusion

- The results that we got with our four fold loss were in general slightly better than the results we got by training with MSE loss alone.
- We tried to fine tune the pretrained RBPN model with MAE loss on the Vimeo dataset using the other three loss functions that we defined, but the results did not seem to improve.
- We used the pre-trained weights of the RBPN for evaluating it with MAE loss.
- We analysed a spatio-temporal approach to VSR that uses recurrent-generative backprojection networks.
- RBPN enables the model to generate superior SR images by combining spatial and temporal information from the input and neighboring frames.

Future Work

We could introduce an adversarial loss component by using the RBPN network as the generator and include a discriminator network for classifying the generated super resolution frames as real or fake which could further improve the performance by reducing the blur in the generated frames.

References

- Haris, Muhammad and Shakhnarovich, Greg and Ukita, Norimichi Recurrent Back-Projection Network for Video Super-Resolution,CVPR(2019)
- H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model, IEEE Transactions on Image Processing (2015)
- Justin Johnson,Alexandre Alahi and Li Fei-Fei, Perceptual Losses for Real-Time Style Transfer and Super-Resolution, CoRR(2016)