Video Super-Resolution using Recurrent Back-Projection Network

Mohit Kumar
SR No. 04-01-03-10-51-21-1-19825
MTech Artificial Intelligence

Kaushik Kukadiya
SR No. 04-03-06-10-51-21-1-19762
MTech Artificial Intelligence

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. Consider an LR video source which consists of a sequence of LR video frames $LR_{t-n}, \ldots, LR_t, \ldots, LR_{t+n}$, where we super-resolve a target frame LR_t. The idea behind SISR is to super-resolve 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 seeks to address this by utilizing 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. After spatially aligning frames, missing details are extracted by separating differences between the aligned frames from missing details observed only in one or some of the frames. However, in MISR, the alignment of the frames is done without any concern for temporal smoothness, which is in contrast to VSR where the frames are typically aligned in temporal smooth order.

Traditional VSR methods upscale based on a single degradation model (usually bicubic interpolation) followed by reconstruction. This is sub-optimal and adds computational complexity. Recently, learning-based models that utilize 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). In some recent VSR methods that utilize CNNs, frames are concatenated or fed into recurrent neural networks (RNNs) in temporal order, without explicit alignment. In other methods, the frames are aligned explicitly, using motion cues between temporal frames with the alignment modules. A crucial aspect of an effective VSR system is the ability to handle motion sequences, which are often integral components of videos.

Methodology

Our method is inspired by recurrent back-projection networks (RBPNs) which utilize "back-projection" to iteratively calculate 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. Deep back-projection networks (DBPNs) use back-projection to perform SISR using learning-based methods by estimating the output frame SR_t using the corresponding LR_t frame. To this end, DBPN produces a high-resolution feature map that is iteratively refined through multiple up- and down-sampling layers. RBPN offers superior results by combining the benefits of back-projection with DBPN.

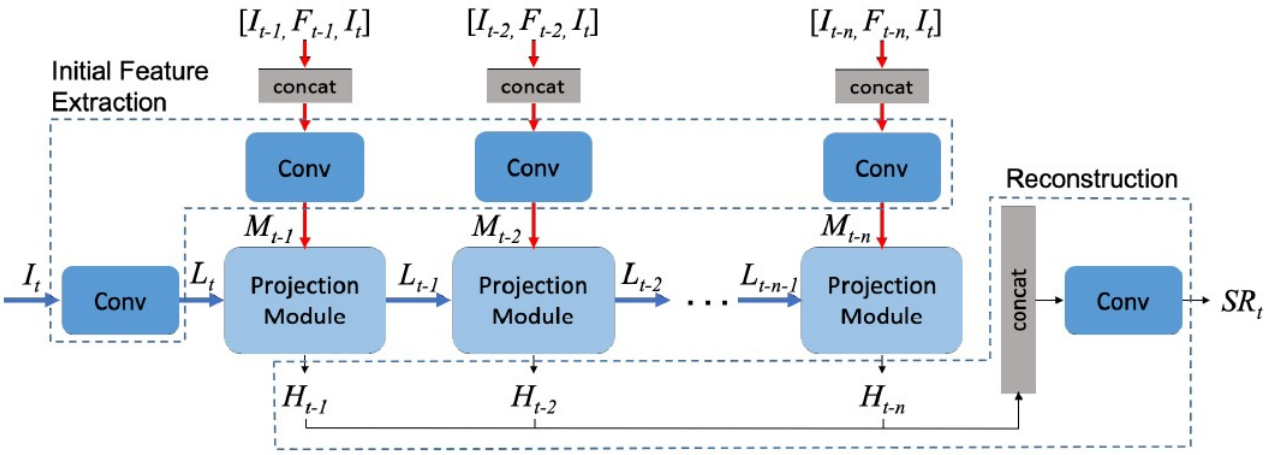


Figure 1: Model Architecture

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. Minimizing MSE encourages finding pixel-wise averages of plausible solutions that are typically overly-smooth and thus have poor perceptual quality. To address this, we adopt a four-fold (MSE, perceptual, MAE, and TV) loss function. MSE loss focuses on optimizing perceptual similarity instead of similarity in pixel space. We use a denoising loss function called TV loss. To evaluate our model we used standard datasets: Vimeo90K(a dataset containing various types of motion), Vid4 and SPMCS. When training our model, we generated the corresponding LR frame for each HR input frame by performing 4× down-sampling using bicubic interpolation. We adopted an RNN-based optical flow method that preserves spatio-temporal information in the current and adjacent frames.

Network Architecture

Fig. 1 shows the model architecture. The network has two approaches. The horizontal blue line enlarges I_t using SISR. The vertical red line is based on MISR to compute the residual features from a pair of I_t to neighbor frames $(I_{t-1}, \ldots, I_{t-n})$ and the precomputed dense motion flow maps $(F_{t-1}, \ldots, F_{t-n})$. Each step is connected to add the temporal connection. On each projection step, RBPN observes the missing details on I_t and extract the residual features from each neighbor frame to recover the missing details.

The convolutional layers that feed the projection modules serve as initial feature extractors. Within the projection modules, RBPN utilizes a recurrent encoder-decoder mechanism for fusing details extracted from adjacent frames in SISR and MISR and incorporates them into the estimated frame SR_t through back-projection. The convolutional layer that operates on the concatenated output from all the projection modules is responsible for generating SR_t.

$$\text{Encoder:} \quad H_{t-n} = \text{Net}_E\left(L_{t-n-1}, M_{t-n}; \theta_E\right)$$
$$\text{Decoder:} \quad L_{t-n} = \text{Net}_D\left(H_{t-n}; \theta_D\right)$$

The encoder module Net_E:

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

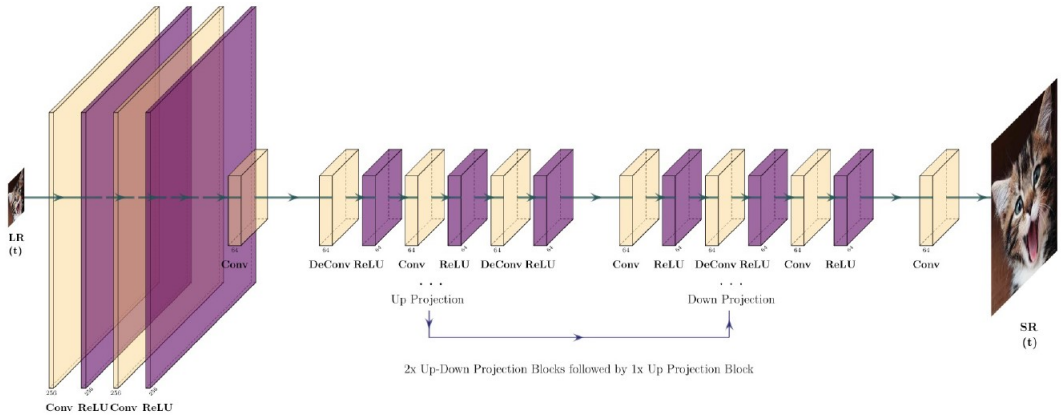


Figure 2: DBPN architecture for SISR, where we perform up-down-up sampling using 8×8 kernels with a stride of 4 and padding of 2. Similar to the ResNet architecture above, the DBPN network also uses Parametric ReLUs as its activation functions.

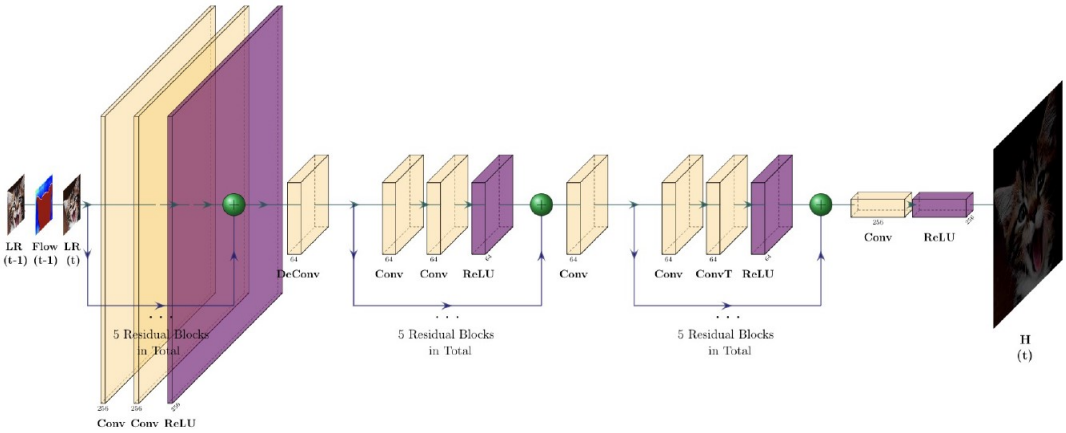
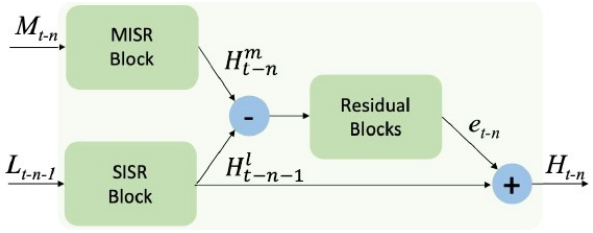
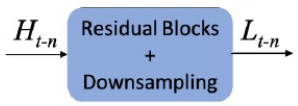


Figure 3: ResNet architecture for MISR that is composed of three tiles of five blocks where each block consists of two convolutional layers with 3×3 kernels, a stride of 1 and padding of 1. The network uses Parametric ReLUs 18 for its activations.



(a) Encoder (the back-projection)



(b) Decoder

MISR upscale: $H_{t-n}^{m} = \text{Net}_{misr}\left(M_{t-n}; \theta_{misr}\right)$
Residual: $e_{t-n} = \text{Net}_{\text{res}}\ \left(H_{t-n-1}^{l} - H_{t-n}^{m}; \theta_{\text{res}}\ \right)$
Output: $H_{t-n} = H_{t-n-1}^{l} + e_{t-n}$

Loss Functions

- **MSE loss** We use pixel-wise MSE loss for the estimated frame SR_t against the ground truth HR_t, We train the entire network jointly, end-to-end.

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

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

- **MAE loss** We use pixel-wise MAE loss for the estimated frame SR_t against the ground truth HR_t.

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

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

- **Perceptual Loss** Perceptual loss relies on features extracted from the activation layers of the pre-trained VGG-19 network in [41, instead of low-level pixel-wise error measures. We define perceptual loss as the euclidean distance between the feature representations of the estimated SR image $G_{\theta_G}\left(LR_t\right)$ and the ground truth HR_t.

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

  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.

- **Total Variation Loss** is defined as the sum of the absolute differences between neighboring pixels in the horizontal and vertical directions. Since TV loss measures noise in the input, minimizing it as part of our overall loss objective helps de-noise the output SR image and thus encourages spatial smoothness. 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.

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

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

Results

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

Conclusion

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. It combines ideas from single and multiple-frame super resolution. Temporal context is organized by a recurrent process using the idea of backprojection, yielding gradual refinement of the high-resolution features used, eventually, to reconstruct the high resolution target frame.

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. But 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 used the pre-trained weights of the RBPN for evaluating it with MAE loss. 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 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

[1] Haris, Muhammad and Shakhnarovich, Greg and Ukita, Norimichi Recurrent Back-Projection Network for Video Super-Resolution,CVPR(2019)

[2] H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model, IEEE Transactions on Image Processing (2015)

[3] Justin Johnson,Alexandre Alahi and Li Fei-Fei, Perceptual Losses for Real-Time Style Transfer and Super-Resolution, CoRR(2016)

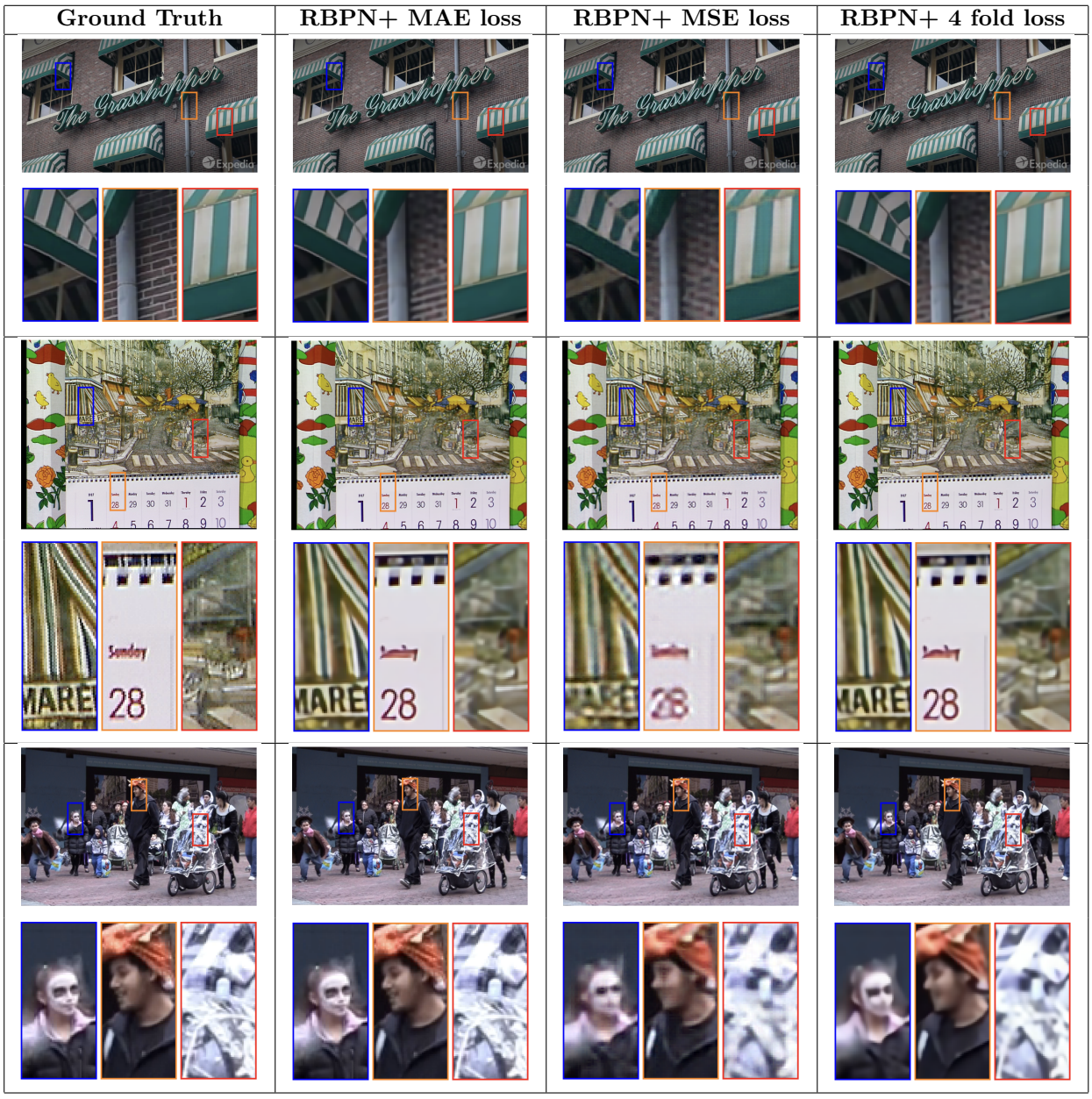


Table 1: Visual Performance Comparision