Study on Reference-based Video Super-Resolution Using Deformable Convolutional Networks

Guo Zirui

Abstract

In this paper, we review the super-resolution reconstruction tasks. Specifically, video super-resolution. Image super-resolution aims to recover natural and realistic textures for a high-resolution image from its degraded low-resolution counterpart. Existing super-resolution methods can be broadly categorized into two types: single image super-resolution (SISR) and multi-frame super-resolution (MFSR). However, Since the downsampling process is not reversible, the high frequency information is permanently lost. Thus, traditional super-resolution methods can lead to a blurry visual effect. Recently, reference super-resolution methods have been widely studied. The input is not just a low-resolution image or video, but a low-resolution image and a high-resolution image used as reference. Reference-based methods aim at transferring HR textures from a given Ref image to produce visually pleasing results. In this paper, we purposed a novel reference super-resolution method using deformable convolution and compared our model with a baseline for quantitative and visual estimation. We conducted experiments at a $\times 4$ scaling factor and found that our proposed method can recover high-quality super-resolution images from very low-resolution levels with higher speed and less memory usage.

Index Terms

Video super-resolution, reference frame. deformable convolution.

I. Introduction

CONVOLUTIONAL, neural network (CNN) based methods have achieved significant improvements on super-resolution. Super-resolution is a technique that converts a low-resolution (LR) image to a high-resolution (HR). It has been widely used in consumer electronics. The existing methods can be divided into two categories, single-image super-resolution (SISR) and multi-frame super-resolution (MFSR). SISR only uses a single LR image. Thus, it has an economic advantage. However, it suffers blurry results due to the lost information of the HR image. Some generative adversarial networks (GANs) based methods have been proposed to tackle this problem, but the final result may not be the same as the actual situation, that is, the result has some false textures, although the image is clear in visual effect, the image details are false.

To address the drawbacks, another method has been proposed, which is multi-frame super-resolution. Multi-frame super-resolution takes more than one image as input. A wild known task in MFSR is video super-resolution that takes consecutive frames as input. Another task in MFSR is reference-based super-resolution. There are two input images for this model, one blurry and one clear. The model takes real high-frequency information from the clear image and then synthesizes it into a blurry image. The high resolution reference image could be selected from adjacent frames in a video, images from web retrieval, an external database, or images from different viewpoints. Zheng et al. [1] purposed an optical flow based approach to adopt RefSR. However, due to the structural limitation of its module which calculates the optical flow called FlowNet [2], the length and width of the input image must be an integer multiple of 16. Zhang et al. [3] then purposed a patch-matching based network called SRNTT. But the patch matching stage is time consuming as the operator executed on this stage is not implemented on GPU. Recently, Yang et al. [4] purposed transformer based network called TTSR. However, the transformer consuming lots of video memory.

To address these problems, we propose a **R**efSR **w**ith **D**eformable **C**onvolutional **N**etwork (RwDCN). This approach adopts deformable convolution for aligned LR image and Ref image and then further for texture transfer. Our method accepts input of arbitrary size, runs quite faster than current methods with much less memory usage.

II. Related Works

In this section, we review four related works: single image super-resolution (SISR), video super-resolution (VSR), reference-based super-resolution (RefSR) and deformable convolution. Besides, we introduced the latest paper which shows the usage of deformable convolutional networks in super resolution.

A. Single Image Super-Resolution

Super-resolution is viewed as a traditional image processing task. In recent years, deep learning based research on super-resolution has achieved great improvements over traditional non-learning based methods.

The very beginning of deep learning methods is SRCNN which was proposed by Dong et al. [5]. It is an end-to-end image super-resolution approach.

Pixel shuffle layer [6] is another great improvement on SISR. Deconvolution will lead to check board artifacts, which can be circumvented by pixel shuffle.

In 2017, Timofte et al. [7] introduced a high resolution dataset DIV2K which consisting of 1000 2K resolution RGB images. The current state of the art networks are trained on DIV2K, such as EDSR [8] and RCAN [9].

Residual block was later introduced into SISR. EDSR [8] and RCAN [9] are representative methods. Deeper networks are designed to reconstruction SR images on the feature domain. We can also find this design principle in video super-resolution methods where ResBlock is widely used on the reconstruction stage.

Methods mentioned above are using pixel loss, such as mean square error (MSE) and mean absolute error (MAE), as their objective function. however, with the introduction of perceptual loss [10], more and more people are paying attention to human perceptions. GAN loss is also used in SR tasks.

B. Video Super Resolution

Video super-resolution differs from single image super-resolution in that it can make use of aggregating information from related but misaligned frames. Video super-resolution can be divided into categories, methods that require alignment and methods that do not require alignment e.g. [11] [12]. Here We will focus on the alignment methods.

The alignment method is divided into two stage: alignment of different frames and reconstruction. The align different frames stage usually inputs consecutive frames and then outputs an aligned feature map for reconstruction. There are mainly three methods for alignment. Optical flow based methods, represented by RBPN [13] and BasicVSR [14]. This kind of method is estimating optical flow field between center frame and its neighboring frames. The neighboring frames then are warped according to the motion fields. However, accurate optical flow is hard to estimate under the circumstances given large motion. Dynamic filter [11], representing by DUF [11] is used to tackle this problem. Another method widely used method is deformable convolution, representing by TDAN [15] and its successor EDVR [16].

C. Reference-based Super-Resolution

As SISR research has hit a bottleneck, some have begun to look to RefSR. The main challenge in RefSR is to align the LR image and its Ref image. Optical flow, represented by CrossNet [1], is adopted. However, calculating optical flow is time-consuming. Besides, the reconstruction quality largely depends on the performance of alignment stage. Deformable convolutional networks are also used in RefSR tasks. Representing by [17]. Recent work SRNTT [3] is using patch matching methods between VGG features of the LR image and Ref image. TTSR further addresses the problem that SRNTT feeds all the features equally. The latest work TTSR [4] adopted transformer for texture transfer and get great improvement.

D. Deformable Convolution

It is CNNs' limitation to model geometric transformations due to their fixed kernel configuration. Dai et al. [18] [19] proposed a deformable convolution operation. It has been used in tackling high-level vision tasks such as object detection and semantic segmentation. TDAN was the first method use deformable convolution to align the center frame and its neighboring frames in a video super-resolution task.

E. Deformable Convolutional Networks and Video Super Resolution

Although Deformable convolutional networks have been widely used in video super-resolution tasks, there is no specific explanation about how it works. Latest paper [20] shows that deformable alignment can be formulated as a combination of feature-level flow-warping and convolution. Here we would like to introduce deformable convolutional networks first, then show how it works in video super resolution.

A deformable convention kernel is given of $K = k \times k$ sampling locations. w_k and p_k are the weight and offsets for the $k-$th location. For each position p_0 in feature map, we can get a shift Δp_k for each convolution kernel p_k. For instance, a 3×3 kernel is defined with $k = 3$ and $K = 3 \times 3 = 9$. Here, $\mathbf{p}_k \in \{(-1,-1),(-1,0),\cdots,(1,1)\}$. In [18] the operation of deformable convolution is defined as:

$$y(\boldsymbol{p}) = \sum_{k=1}^{n^2} w_k \cdot x\left(\boldsymbol{p} + \boldsymbol{p}_k + \Delta \boldsymbol{p}_k\right) \tag{1}$$

Here, x and y is the input and output features, respectively. We can denote the feature warped by the $k-$th offset by $x_k(\boldsymbol{p}) = x\left(\boldsymbol{p} + \boldsymbol{p}_k + \Delta \boldsymbol{p}_k\right)$. From Eqn. 1, we have:

$$y(\boldsymbol{p}) = \sum_{k=1}^{n^2} w_k \cdot x_k(\boldsymbol{p}) \tag{2}$$

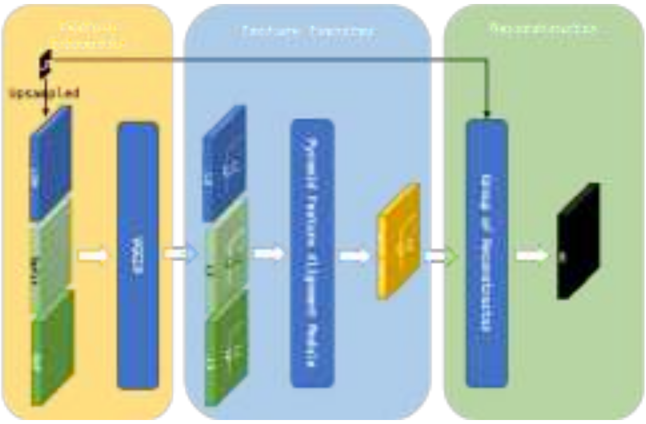


Fig. 1. The structure of RwDCN. LR refers to low resolution image needed to be restored; Ref refers to high resolution reference images; LR↑ refers to upsampled LR image; Ref↓↑ refers to sequentially downsampled and upsampled Ref images.

which can be viewed as a $1 \times 1 \times 1 \times n^2$ standard 3D convolution.

In paper [20], they also extracted offsets from pretrained EDVR model and compared with optical flows. After quantitatively studying the correlation between the offsets and optical flows, they found over 80% of the estimations have a difference smaller than one pixel from the optical flow.

III. The Proposed Method

In this section, we introduce the proposed method, **R**eference-based Video Super Resolution **w**ith **D**eformable **C**onvolution **N**etwork (RwDCN).

We first do a brief introduction to our approach in III-A. And then introduce our feature extraction and alignment method using deformable convolution network in III-B and III-C. Finally, we introduce how we reconstruct a HR image from aligned feature map in III-D.

A. Overview

The structure of our proposed method is shown in Figure 1. Our proposed RwDCN is aiming to transfer texture from Ref image to LR image. Our main idea is to align texture in the feature domain and transfer textures in multi-scale for it enables our model to study transfer scaled or rotated texture from a reference image.

Our architecture can be divided into three stages: feature extraction (yellow part in fig. 1), texture transfer (blue part in fig. 1) and reconstruction (green part in fig. 1). Our feature extraction module uses VGG19 to extract feature as Papers discussing style transfer [21] show that VGG19 is able to extract texture from an image. In texture transfer stage, we use deformable convolution to transfer texture. Traditional optical-flow based methods are time-consuming. Latest transformer based methods [4] consumes a huge amount of memory during calculate relevance embedding. Thus, we use deformable convolution to transfer texture which consumes less time and memory. In Reconstruction stage we use residual blocks to reconstruct image as it is shown in [9] very deep trainable network helps to recover image.

Here, we use LR↑ representing the $4\times$ bicubic-upsampled LR image. For Ref image, we sequentially apply bicubic down-sampling and up-sampling with the scale factor $4\times$ to obtain Ref↓↑. We use VGG19 to extract feature maps from LR↑, Ref↓↑ and Ref respectively.

B. Feature Extraction

Texture extraction from the Ref image is crucial in the RefSR task, as accurate and appropriate texture information helps to restore SR images. Here VGG19 is adopted as our feature extraction as it is used for image classification and has shown the ability to extract high level perceptual information so it is widely used in style transfer tasks which also concerning about transfer texture.

Texture transfer stage is aiming to find the similar part in LR image and Ref image. As it is hard to compare feature between a clear image (Ref) and a blurry image (LR↑) because they have different frequency band, we would like to compare blurred Ref, which is Ref↓↑, with LR image.

Here we extract the 2nd, 5th and 10th layers in VGG19 whose spatial size is original HR spatial size, $\frac{1}{2}$ HR spatial size and $\frac{1}{4}$ HR spatial size respectively. The extract feature maps are denoted as L_3, L_2, L_1 in later texture transfer and reconstruction stage.

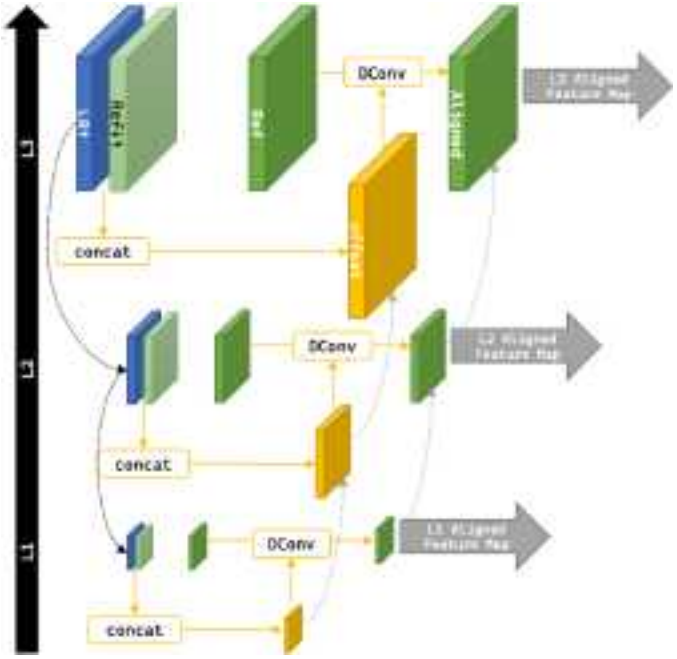


Fig. 2.

C. Texture Transfer

This module is inspired by EDVR and TTSR. Optical-flow based methods usually align images on image domain, while deformable convention based methods usually apply alignment on feature domain. We first introduce the use of deformable convolution for alignment, that is, aligning feature maps extracted from LR image and Ref image.

The normal convolution operation applies the convolution kernel to the feature map, but deformable convolution applies the convolution kernel to the feature map and an extra offset. The offset has the same spatial size as the feature map, but its channel number is defined as follow:

$$C_{\text{offset}} = k^2 \times C_{\text{feature map}} \tag{3}$$

Here, C denotes to the number of channels, k denotes to kernel size, usually set to 3.

We use concatenated features of upsampled LR image and Ref image to initialize the learnable offset Δp_k:

$$\mathbf{\Delta P} = f\left([F_{\text{Ref}}, F_{\text{LR}}]\right) \tag{4}$$

where f is a general function consisting several convention layers, and $[\cdot, \cdot]$ denotes the concatenation operation. F_{Ref} is the ref feature map, F_{LR} is the LR feature map.

We initialize the offset using LR↑ and Ref↓↑. But when calculate aligned feature map, feature map extracted from Ref image is used to transfer texture which is lost in LR image.

The LR image and Ref image do not have same viewpoint. To address large parallax problems in alignment, we align and reconstruct the image based on well-established principles: pyramidal processing. The feature maps we get from the feature extraction stage have pyramid spatial size: HR size, $\frac{1}{2}$ HR size and $\frac{1}{4}$ HR size. The L_i aligned feature map is generated as follow:

$$F_i = \text{DConv}[F_{\text{Ref}_i}, e[\text{concat}(\text{offset}_i, {\text{offset}_{i-1}}^{\uparrow 2})]] \tag{5}$$

where Dconv$(\cdot)$ is the deformable convolution operator, $(\cdot)^{\uparrow 2}$ refers to upsampling by a factor 2 using bilinear interpolation. and e is a general function with several convolution layers. The output of the alignment stage is a pyramid spatial size aligned feature maps.

D. Reconstruction

Our reconstruction module is implemented by combining multiple aligned feature maps into a deep generative network corresponding to different scales. The overview of our reconstruction module is shown in fig.3. After passing through one reconstructor, the feature map is restored to 2× spatial size. Here we use three reconstructor to restore images. The details of one reconstructor is shown in fig. 4.

$$\psi_l = [\text{Res}(\text{concat}[\psi_{l-1}, F_{l-1}]) + \psi_{l-1}] \uparrow_{2\times} \tag{6}$$

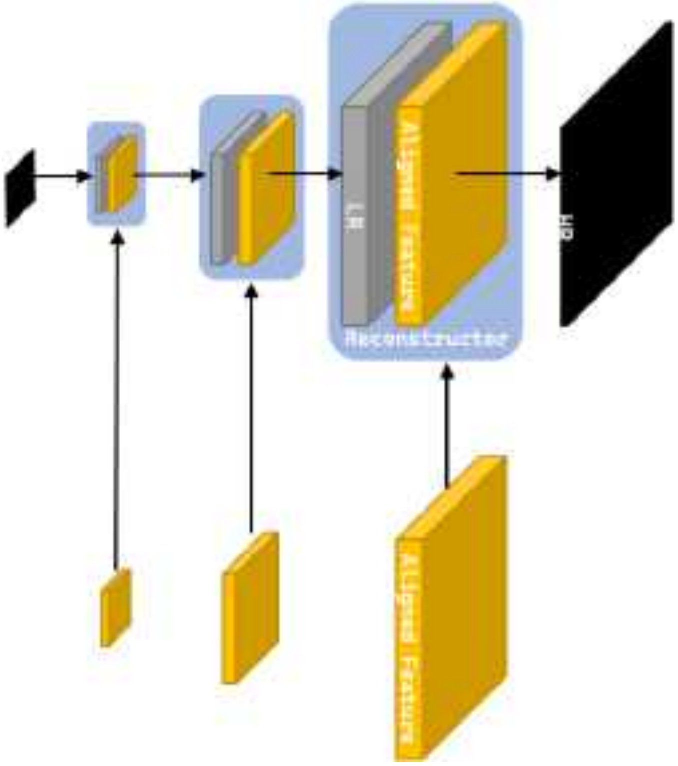


Fig. 3. The reconstruction module. Our reconstruction module is implemented by combining multiple-scaled aligned feature maps into a deep generative network.

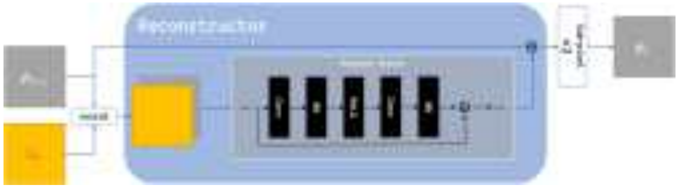


Fig. 4. One reconstructor in reconstruction module.

For the $l-$th reconstructor, we first conctenate feature map ψ_{l-1} get from $(l-1)-$th reconstructor and $l-1$ level aligned feature map F_{l-1}. Secondly, we send concatenated feature map into a sequency of residual blocks. Finally, add up ψ_{l-1}, then send to a pixel shuffle block to get ψ_l.

IV. Experiments

A. Datasets

We first describe the dataset using in our work. We use REDS [22] dataset, a newly proposed high-quality video dataset. The REDS datasets contains 240 training clips and 30 validation clips. Each clip consists of 100 consecutive frames. We re-grouped them to get 270 datasets and from which 4 clips are selected as validation datasets. The validation datasets is denoted by REDS4 as what they did in EDVR. Specifically, REDS4 contains the 000, 010, 015 and 020 clips. We let the $4+10\cdot i$ frame in a set be the reference frame for the $[10\cdot i, 10\cdot i+9]$ frame, *i.e.* 10 frames share a single reference frame. So for one epoch we have 23940 sets of images. Due to the lack of memory, we crop the LR images to 128×128 during training. Here we use random cropping for data augmentation.

We use Vid4 for validation. Vid4 [23] contains 4 clips of video. However, as the CrossNet requires the height and weight should be the of 16, So we crop the dataset. For calendar clip and city clip, we crop them to 144×176; For foliage and walk, we crop them to 112×176. The center frame in each clip serves as the reference image for the whole clip. As our proposed method is a reference-based method, so we use new proposed dataset CUFED5 proposed in [3] as one of our validation dataset. CUFED5 defines four similarity levels from high to low, i.e. L1, L2, L3 and L4, according to the number of best mathes of SIFT features. The testing set contains 126 groups of samples.

TABLE I

QUANTITATIVE COMPARISON ON **REDS4** FOR 4× SR. RED AND BLUE INDICATES THE BEST AND THE SECOND BEST PERFORMANCE.

		SISR Method	RefSR Methods		
Clip	Bicubic	RCAN	CrossNet	SRNTT	RwDCN(Ours)
Vid4	20.41/0.520	20.00/0.513	18.63/0.372	19.04/0.488	20.00/ 0.530
REDS	25.93/0.724	28.15/0.802	22.28/0.504	27.61/0.779	28.27/ 0.815
CUFED5	22.92/0.632	24.21/0.712			
CUFED5_level1			20.18/0.465	24.12/0.717	24.239/0.7274
CUFED5_level2			20.06/0.451	24.09/0.715	24.237/ 0.7273
CUFED5_level3			20.09/0.454	24.09/0.715	24.230/ 0.7268
CUFED5_level4			20.04/0.448	24.06/0.714	24.227/ 0.7268
CUFED5_level5			20.07/0.448	24.09/0.714	24.239/ 0.7270

B. *Baseline Models*

We compare our proposed RwDCN with several state-of-the-art methods on SR. We also took bicubic interpolation into account. We would like to introduce baseline models below.

1) Single-Image SR Methods: We use the SOTA single image super-resolution architecture RCAN [9] dataset as one of our baselines. For its pretrained model is not trained on REDS dataset, in fairness, we retrain them on our own. We train this model for 50 epochs.

2) Ref SR Methods: We use CrossNet and SRNTT as our RefSR baselines. We use the pretrained model provided on GitHub.

C. *Loss Function*

There are 2 component in our loss function, reconstruction loss and perceptual loss.

$$L = \lambda_{\text{rec}}\ \mathcal{L}_{\text{rec}} + \lambda_{per}\ \mathcal{L}_{\text{per}} \tag{7}$$

Here we set λ_{rec} to 1 and λ_{per} to 0.1.

1) Reconstruction Loss: Charbonnier penalty function is first introduced in [24] and is widely accepted as loss function in SR task for its robust and ability to handle outliers. Here, we use Charbonnier penalty function as our reconstruction loss for achieving higher PSNR.

$$\mathcal{L}_{\text{rec}} = \sqrt{\left\|I^{GT} - I^{SR}\right\|^2 + \varepsilon^2} \tag{8}$$

ε is set to 1×10^{-3}.

2) Perceptual Loss: Perceptual loss has shown its ability in [10] for achieving better visual quality.With perceptual loss, we can enhance the similarity in feature space.

$$\mathcal{L}_{\text{per}} = \sum_{I} \frac{1}{C_i H_i W_i} \left\|\phi_i^{vgg}\left(I^{SR}\right) - \phi_i^{vgg}\left(I^{HR}\right)\right\|_2^2 \tag{9}$$

here,$\phi_i^{vgg}(\cdot)$ denotes the $i-$th layer's feature map of VGG19. We use the relu1_1, relu2_1, relu3_1 layer. (C_i, H_i, W_i) denotes the shape of the feature map at the $i-$th layer.

D. *Implementation Details*

Each mini-batch contains 4 LR frames with size 128×128 along with 4 Ref frames with size 512×512. We used gradient accumulation training strategy to get bigger mini-batch size. Accumulation step is set to 4. So the real mini-batch size is 16.

We train our model with Adam optimizer by setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 4×10^{-4}. We implement out models with the PyTorch framework and train them using 2 NVIDIA RTX 2080 GPUs.

V. EXPERIMENTAL RESULTS

A. *Quantitative Evaluation*

In this section, we compare both quantitative and qualitative results. To evaluate our proposed method, we report the PSNR and SSIM scores of every test dataset in table I. In table I, the red color represent the highest score in the comparison while the blue color represents the second one. Our proposed method obtained highest scores on all test dataset. Considering with the averaged evaluation, SRNTT is the second best. RwDCN surpasses SRNTT by 0.66 dB on REDS4. When compared with CrossNet, RwDCN obtains a marked improvement of 5.99 dB on REDS4.

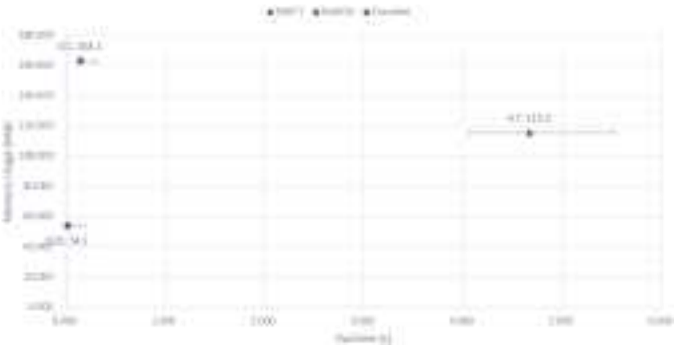
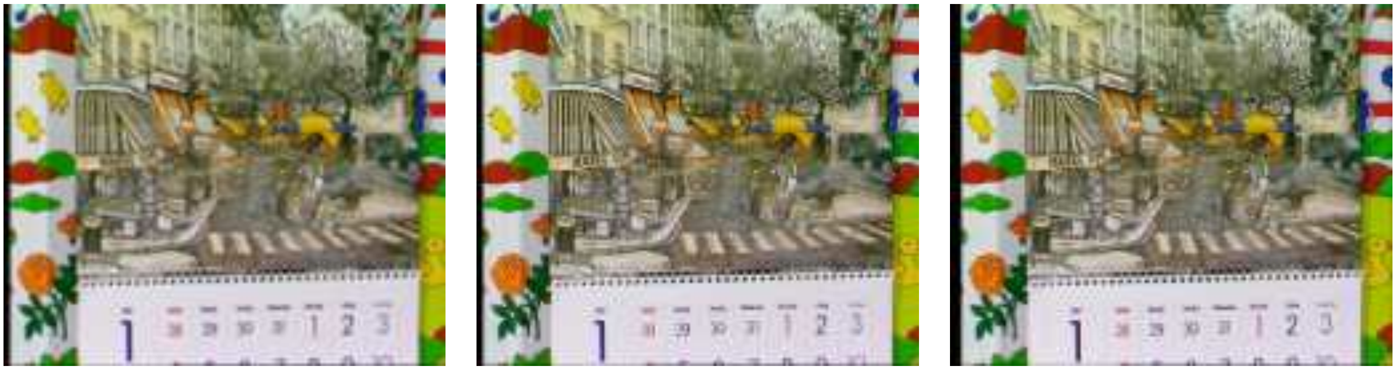


Fig. 5. The runtime and memory usage comparison.



(a) bicubic 17.15/0.45 SRNTT 16.40/0.44 RwDCN 17.31/0.49

Fig. 6. The 1st frame of Vid4's calendar clip. Although bicubic achieved higher PSNR and SSIM scores, but SRNTT has restored clear texture. This is a example shows that, SSIM and PSNR sometimes fail to evaluate texture reconstruction results.

However, on Vid4 test dataset, bicubic achieved second best performance. As it is shown in figure 6, bicubic achieved higher PSNR and SSIM scores, but SRNTT has restored clear texture. This is a example shows that, SSIM and PSNR sometimes fail to evaluate texture reconstruction results. Thus, we would like to visual comparison.

As it is shown in figure 7, our proposed RwDCN architecture can reconstruct parts of building's windows frame while RCAN failed to recover correct texture. But in figure 8, although our proposed RwDCN achieved better performance than RCAN in recovering building, but SRNTT is better in recovering leaves. In figure 9, SRNTT recovered texture of stone but generated fault pattern of bricks. In figure 10, SRNTT recovered the shape of chairs while the others do not. From the example above, SRNTT successfully recovered in recovering fine texture from references at the most of times, but sometimes recover fault texture. Our RwDCN architecture gets smoother images than SRNTT and can recover correct pattern when significant information is lost.

We also tested the speed and video memory usage of each model on an NVIDIA GTX 1080 Ti on Vid4 dataset. We can see in the figure 5 that RwDCN's average runtime is 0.0138 second, CrossNet's average runtime is 0.146 second and SRNTT's average runtime is 4.666 second. SRNTT takes 30 times longer to compute than RwDCN and CrossNet takes 1.5 times longer to compute than RwDCN. As for memory usage, RwDCN's average memory is 54.186 MiB, CrossNet's average memory is 163.132 MiB and SRNTT's average memory is 115.533 MiB. SRNTT requires twice as much video memory as RwDCN. CrossNet requires 3 times as much video memory as RwDCN. Without bells and whistles, RwDCN beats CrossNet and SRNTT on both runtime and memory usage.

B. Subjective Evaluation

We also conducted subjective evaluation. Totally 20 people participated in our subjective evaluation. Our questionnaire contains 63 pairs of images. The 63 pairs of images were divided into three groups of 21 pairs of images each. These three groups compare CrossNet and RwDCN, RCAN and RwDCN, and SRNTT and RwDCN, respectively. The content of each group can be found in table II.

We can find that our method RwDCN is substantially ahead of CrossNet. when comparing with RCAN, our method is better than RCAN in about 70% of scenarios. But when comparing with SRNTT, most people think that SRNTT is better than our method.

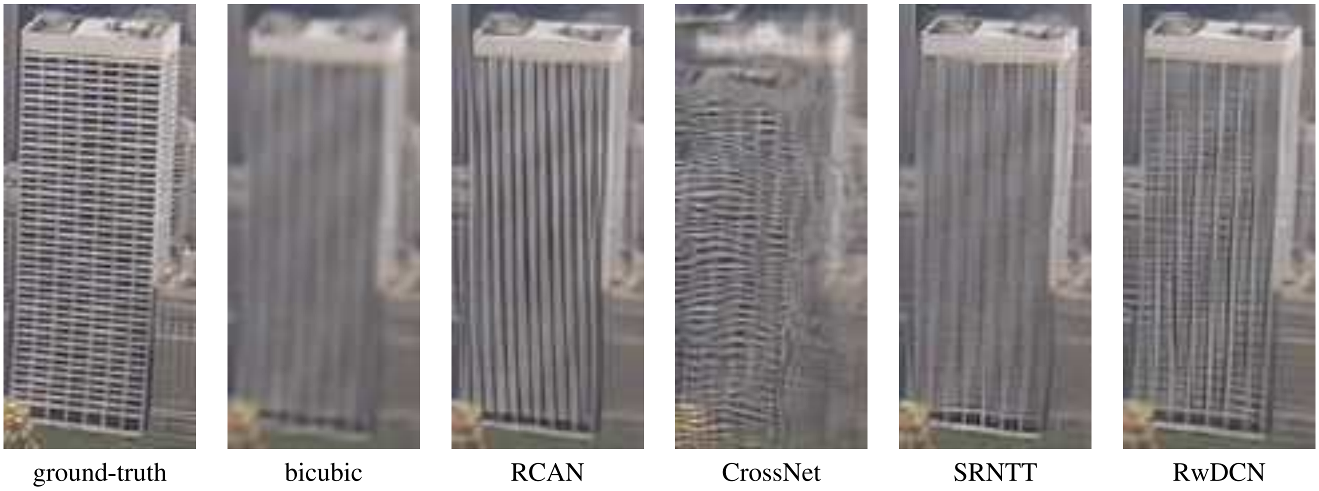


Fig. 7. The 1st frame of Vid4's city clip.

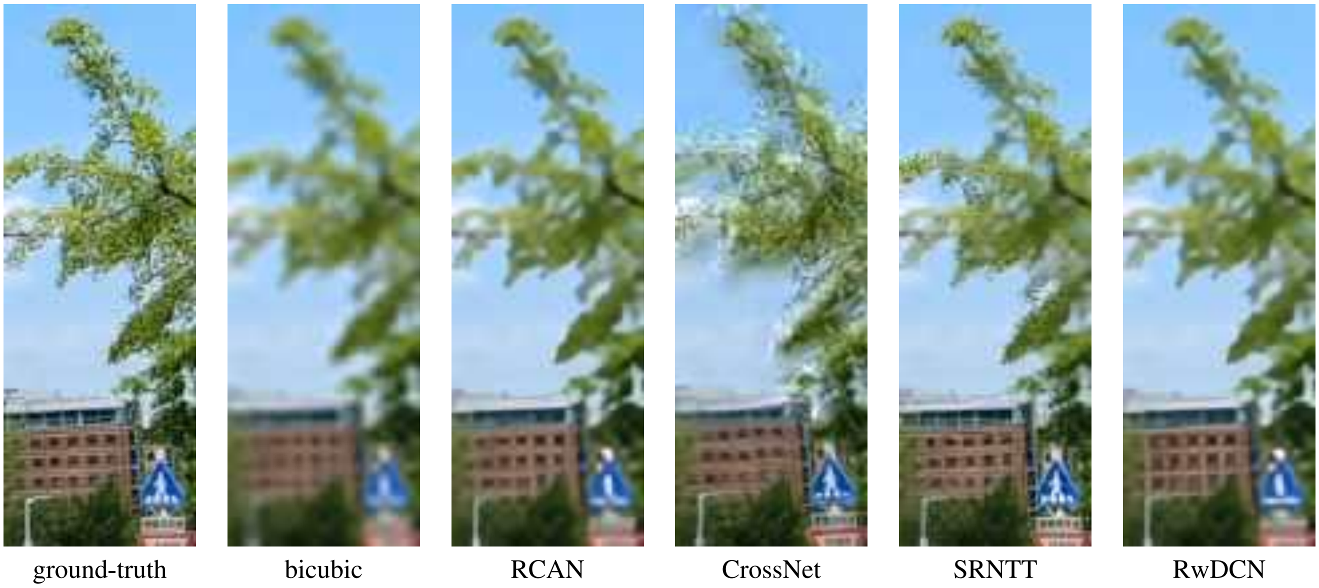


Fig. 8. The 26th frame of REDS' 000 clip.

We asked some experimental participants that there is a big difference between CrossNet and RwDCN. there is not much difference between RCAN and RwDCN visually. They indicated that the images recovered by SRNTT were sharper in some cases and well distinguished from RwDCN.

VI. Conclusion

A. Conclusion of This Paper

In this paper, we proposed a reference-based super-resolution method using deformable convolutional networks. By adopting deformable convolution, we successfully made use of the high frequency information in reference frame.

TABLE II
The content of one group of images in our questionnaire.

	The Number of Pairs	Description
CUFED5	5	The 1st to 5th pairs' level 1 images.
REDS4	8	For each clip, choose the 1st and 6th frames.
Vid4	8	For each clip, choose the 1st and the one that closest to the reference frame.
Sum	21	

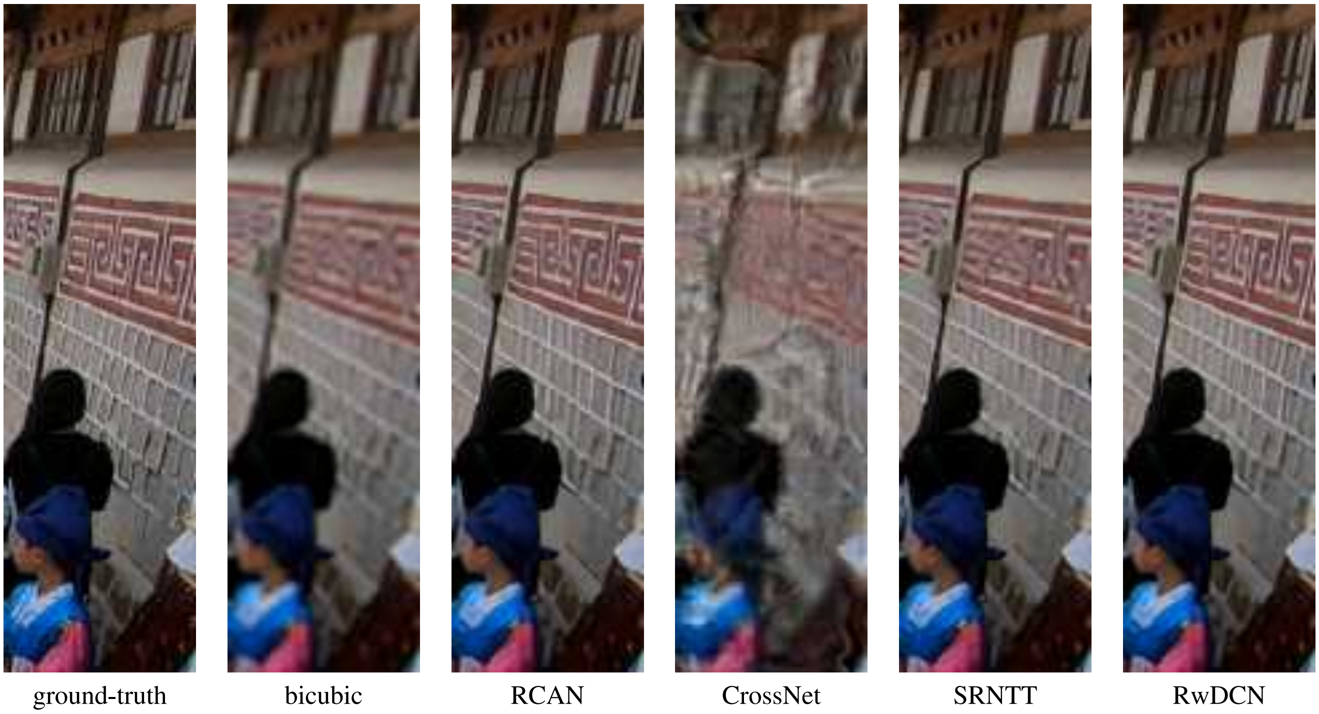


Fig. 9. The 35th frame of REDS' 010 clip.

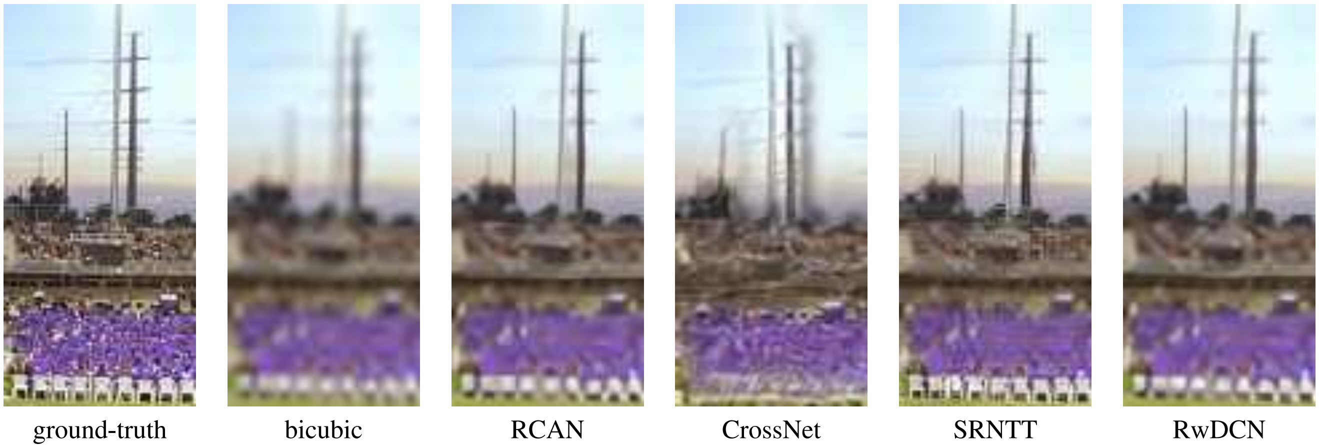


Fig. 10. The 1st frame of CUFED5's level_1 image.

Our purposed model can be divided into three stages. The first stage is to extact feature maps from LR image and Ref image. The second stage is to transfer texture from Ref feature map to LR feature map and get the aligned feature map. The third stage is to recover SR image from aligned feature map.

In the experiments, we compared the proposed model with other state-of-the-art reference-based models, and our model got the highest scores in PSNR and SSIM. Our model run much faster and requires much less video memory than optial flow based methods.

TABLE III
THE RESULTS OF OUR SUBJECTIVE EVALUATION. THE NUMBER INDICATES THE PERCENTAGE OF PEOPLE WHO THOUGHT OUR METHOD RWDCN WAS BETTER.

	CrossNet v.s. RwDCN	RCAN v.s. RwDCN	SRNTT v.s. RwDCN
CUFED5	93%	78%	30%
REDS4	96%	65%	52%
Vid4	90%	64%	52.5%
Average	92.3%	67.9%	46.9%

B. Future Works

Several problems remain to be addressed in our study.

Firstly, when evaluating CUFED5 datasets, it is logical that level 5 should have worse performance than level 1, but we found that there are no different amount levels in quantitative metrics. Thus, we would like to compare the output images directly. After examining several pairs, we found that there is no big difference between different levels. This problem is not only found in our approach, but also the others. The RefSR method has two functions, super-resolution, and texture migration. But it is hard to say which part helps to recover images from the visual aspect. In the future, maybe we need to revisit these two functions in RefSR and clarify the superior of RefSR on texture migration over traditional super-resolution methods.

Secondly, the importance and effect of deformable convolutional networks in super-resolution have been demonstrated. However, the role of deformable convolutional networks in texture migration has not been verified. In the future maybe we need extra experiments to address the role deformable convolutional networks play in texture migration.

Thirdly, both our method and SRNTT performed well on migrating grid-like textures, like window frames. But on other natural textures like wood, they do not perform well. In the future maybe we need to figure out the deeper reasons.

References

[1] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "Crossnet: An end-to-end reference-based super resolution network using cross-scale warping," in *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 87–104.

[2] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[3] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7974–7983.

[4] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5790–5799.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 184–199.

[6] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.

[7] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1122–1131.

[8] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1132–1140.

[9] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 294–310.

[10] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 694–711.

[11] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3224–3232.

[12] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," *Advances in neural information processing systems*, vol. 28, pp. 235–243, 2015.

[13] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3892–3901.

[14] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Basicvsr: The search for essential components in video super-resolution and beyond," 2021.

[15] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3357–3366.

[16] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. C. Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1954–1963.

[17] G. Shim, J. Park, and I. S. Kweon, "Robust reference-based super-resolution with similarity-aware deformable convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8425–8434.

[18] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[19] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9300–9308.

[20] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Understanding deformable alignment in video super-resolution," *arXiv preprint arXiv:2009.07265*, vol. 4, p. 3, 2020.

[21] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2414–2423.

[22] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. M. 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)*, 2019, pp. 1996–2005.

[23] C. Liu and D. Sun, "On bayesian adaptive video super resolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 346–360, 2013.

[24] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.