Flow-enhanced Super-resolution and Frame interpolation

20170151 JaeHyun Kim, 20170432 Woongro Youn, 20170831 Hyunjae Lee
https://github.com/jlee335/cs470_Team7_VideoEnhancer

Introduction

The importance of video has increased more than ever. We interact with others through streamed video platforms such as zoom, and we consume massive amounts of content through Youtube. However, there are many cases where quality of video becomes a concern. Zoom for example has bandwidth limitation and hence sends video in low quality and framerate. Youtube videos that are from the early 2000s are in appalling quality.

In this report, we would like to increase the quality of a video, both in terms of frame-rate and resolution. We have divided the quality issue into two sub-problems; screen resolution and frame-rate, and tried to tackle this by using two separate neural networks. One network tries to increase screen resolution, while the other tries to generate in-between frames of a video.

Our Super-resolution network will try to improve upon existing image super-resolution networks **by considering the previous and next frames, adding a Flow-detecting network alongside the well-used ResNet architecture for super-resolution.** For this, we have made gradual improvements of our network and divided them into “Attempt 1”, “Attempt 2”, and “Attempt 3”. Our frame-interpolation network is a small deviation from existing flow-detection architectures.

Related Works

To begin with, we searched for single-image super-resolution networks for reference, as videos are sequences of individual frames. Early approaches of SR use deep residual-network architectures, and were very successful [1].

A key difference between videos and images is the existence of previous and next frames that can be used for reference. Hence, the vector of motion each pixel takes, so called “Flow” of an image can be calculated. Mehdi S. M. Sajjadi et al shows extracting optical flow by simple network with pooling and bilinear upscaling, and it surely contributed to SR performance[2].

GAN training, and use of higher-dimensional perceptual loss, can be beneficial to SR as well. In Christian Ledig et al[3], the author proposes a perceptual loss function which takes content loss and adversarial loss into consideration. In a GAN training environment, adversarial loss trains the network of generator and discriminator and content loss allows the model to learn perceptual similarity. **We decided to use GAN to improve upon existing network implementations.**

Overview of Networks

SR Attempt 1

Attempt 1 is made up of Fnet and residual network structure illustrated in Figure 1.

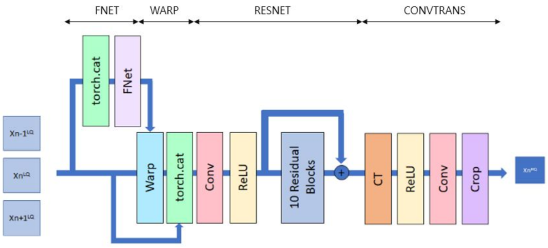


***Figure 1**. Basic network for 2X upscaling*

We tried to use separate optical-flow estimators provided by OpenCV and NVIDIA DALI library. However, OpenCV’s results had too much noise, quickly becoming a bottleneck, while DALI’s version did not work with Colab. Hence, we decided to use an optical flow network and train it end-to-end with the residual network.

Flow estimation method

The paper FRVSR[2] shows that a network with convolutional encoder-decoder structure could get optical flow. We used a similar network but added the next frame as an input. We warped the original image with the output of FNet to reflect the optical flow to the present frame.

$$F^{LR} = FNET(X_{n-1}^{LR}, X_{n}^{LR} . X_{n+1}^{LR}) \in [-1, 1]^{H\times W\times 2}$$

$$S_n^{LR} = WARP(F^{LR}, X_n^{LR}) \in [0, 1]^{H\times W\times 6}$$

Residual network

The residual network consists of 10 small residual blocks. Each residual block has 2 convolutional layers, relu activation function between them and skip connection from the input of the network to output.

$$X_n^{HR} = RESNET(S_n^{LR}) \in [0, 1]^{H\times W\times 64}$$

Upscaling

We use convolution transpose to make double scaled images.

$$S_n^{HR} = CONVTRANS(S_n^{LR}) \in [0, 1]^{(2\times H)\times(2\times W)\times 3}$$

SR Attempt 2

Results from Attempt 1 were good, but we observed substantial noise on pixels with extreme color. Attempt 2 is an initial attempt to mitigate this by adding a long skip-connection. Other aspects of this network are identical to attempt 1.

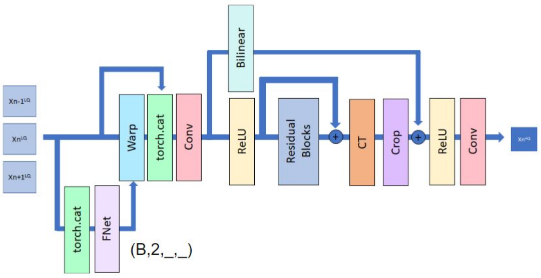


***Figure 2**. Network with bilinear skip connection*

SR Attempt 3

Attempt 3 is another modified variant of attempt 1, after we observed attempt 1's flow-net struggling to train. Instead of using warp to take optical flow into consideration, attempt 3 adds the result of FNet right before residual blocks in hope of increasing gradient flow.

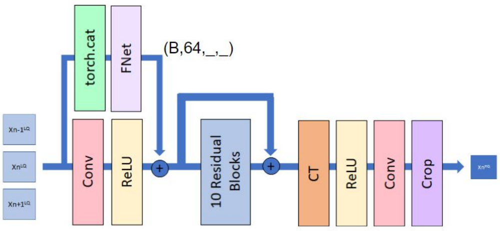


***Figure 3**. Network with modified Fnet*

GAN

Discriminator Network

The structure of discriminator is based on SRGAN[3]. Discriminators get a ground truth frame and a generated high resolution frame, and are trained to determine which is real and which is generated.

Loss function

We use three loss terms to train our model. The MSE loss is the basic loss to calculate pixel-wise difference and the adversarial loss is for training of the GAN network. MSE loss could make lack of high dimensional detail. So, we add the perceptual loss to calculate distance between feature maps propagated through VGG19.

$$L_{total} = L_{MSE} + L_{adversarial} + L_{perceptual}$$

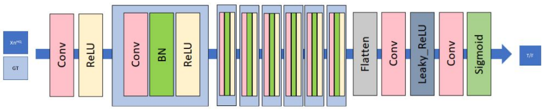


***Figure 4**. Discriminator network*

Frame Interpolation Network

We have modified the Fnet by adding skip connections to see if this can handle frame-interpolation tasks. Each ConvRelu layer is formed with 3 sets of convolution layers right after a leaky relu layer. The network resembles a U-net architecture, as we needed both spatial and contextual information.

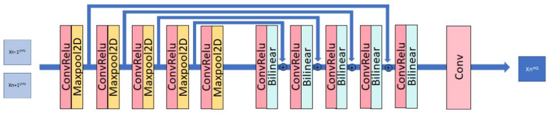


***Figure 5.** Frame-interpolation network*

Experimental Results

Dataset & Training details

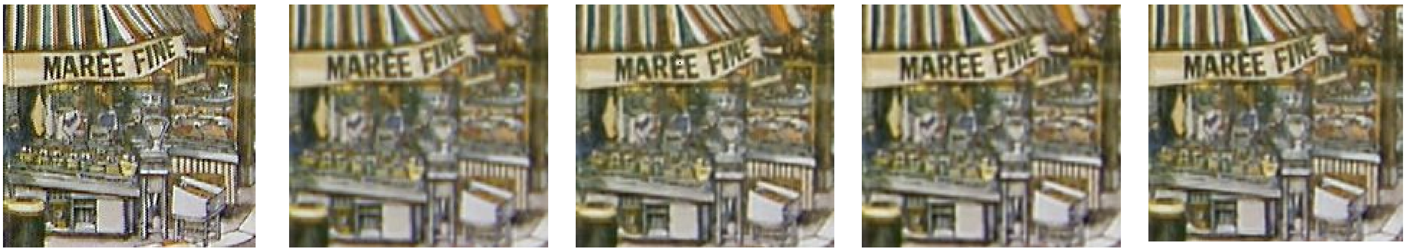
For training, we have downloaded a variety of videos from Youtube using pytube library. We noticed that the large noise from Attempt 1 was due to lack of videos with strong (especially white or black) colors, and thus tried to get videos with a variety of context and range of color. The training dataset was composed of around 40 videos, with a variety of length and color.

After downloading videos, we encoded them to have constant frame rate as NVIDIA DALI framework specifically required this type of encoding. We used the DALI pipeline to create batches of three sequential frames. The batches are collected by random from our database and are additionally random-cropped to size.

Our model needs 3 frames in a row. At first we tested attempt 1~3 and frame interpolation network with Middlebury university video interpolation dataset. However, as we cannot find out benchmark results of other networks with that dataset, we tried Vid4 dataset which has benchmark results for 2X SR in another paper[4]. The low resolution images are made in Matlab function *imresize* in bicubic mode which is the same method as we compared below.

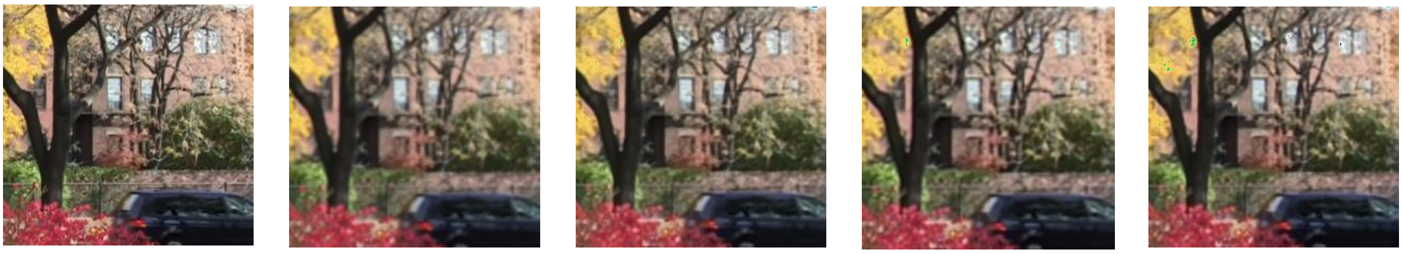
Qualitative results

The quantitative results are on the Figure 6&7 below. From right to left, there are Original HR images, image upscaled by bilinear interpolation of OpenCV, upscaled by using Attempt 1~3 network.



Original HR Bilinear interpolation Attempt 1 Attempt 2 Attempt 3

Figure 6. *Benchmark of Vid4 calendar image*



Original HR Bilinear interpolation Attempt 1 Attempt 2 Attempt 3

Figure 7*. Benchmark of Vid4 foliage image*

Quantitative results

	Attempt 1	Attempt 2	Attempt 3	Frame Interpolation
PSNR	33.56043	31.77315	32.92885	21.9282
SSIM	0.97917	0.98269	0.98899	0.94308

Table 1. *PSNR, SSIM value using the Middlebury frame interpolation dataset.*
(https://vision.middlebury.edu/flow/data/)

	Bilinear	Attempt 3
PSNR	28.42	24.62006
SSIM	0.866	0.88012

Table 2. *PSNR, SSIM value using Vid4 dataset*

The other results of state-of-the-art models are written in the paper by Wang, Longguang, et al[4].

Analysis

Figure 6 and 7 shows out attempts were much better than bilinear interpolation. Attempt 3 was a bit clearer than others but these models did not show significant difference, especially in 2x scale.

From the benchmark results of the Middlebury frame-interpolation dataset, we could notice that attempt 3 has the highest SSIM. It is related to the result that attempt 3 is the best picture when we look at it, since SSIM represents how good the picture seems from a human perspective. Therefore, we tried to benchmark attempt 3 again with a more widely-used 2x SR dataset. Results are displayed in Table 2.

In a quantitative perspective, we noticed our model did not perform as much as other state-of-the-art models. One reason behind this is the drastically different context of data we had compared to Vid4, where we focused on recently taken HDR high resolution videos while data on Vid4 was dated. Another reason we speculate is the lack of training time due to many network modifications and unstable Colab environment. In spite of noise, our model performed better than bilinear interpolation. In Figure 8, original and reconstructed images are compared. You can see that there is noise on pixels that are strongly white.

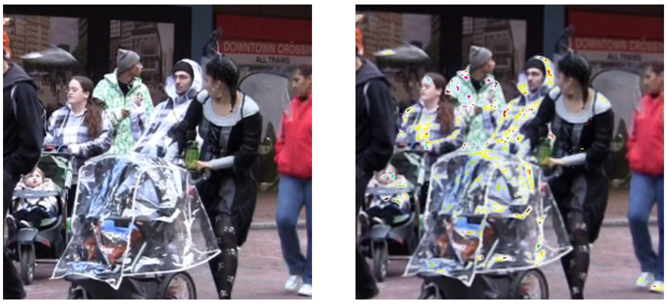
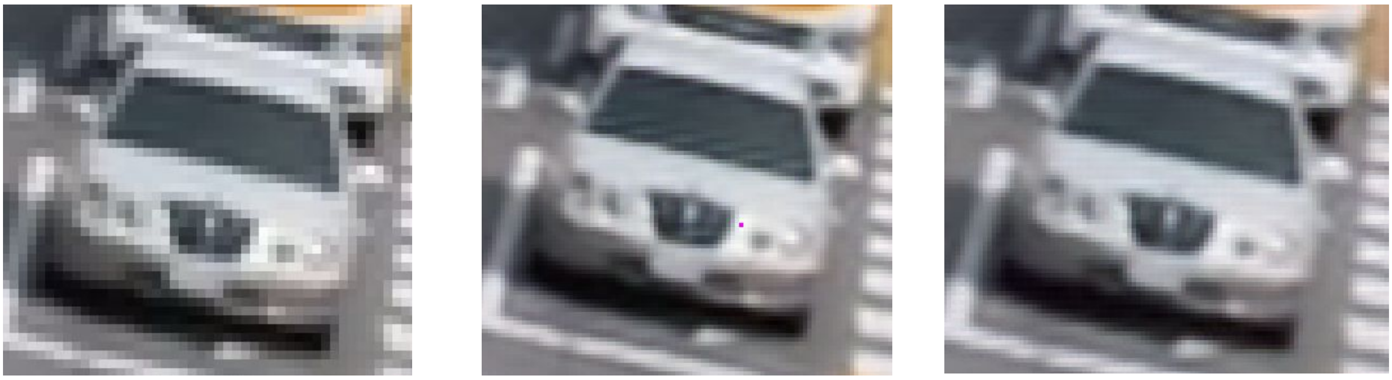


Figure 8*. Illustration of pixel noise. Left figure is original HR image while the right is the image reconstructed through our network.*



Original LR

Inputting 3 same frames

Inputting 3 different frames in motion

Figure 9. *Illustrating different results from inputting 3 same frames versus 3 sequential frames.*

Attempt 3 had better success in utilizing the Fnet architecture that processes previous and current frames. Figure 10 is a result we extracted from getting the saliency map with respect to one pixel in our reconstructed image. This way, we can observe which parts of the input image the resultant image was affected by.

Also, we compare the upscaled result when the input is the same 3 frames or 3 sequence frames. 3 consecutive frames have better performance. On Figure 9 the left one is upscaled with the same 3 frames and the right one is upscaled with consecutive 3 frames. We could find out 3 consecutive frames was useful to reduce stair shape stripe on upscaled frames.

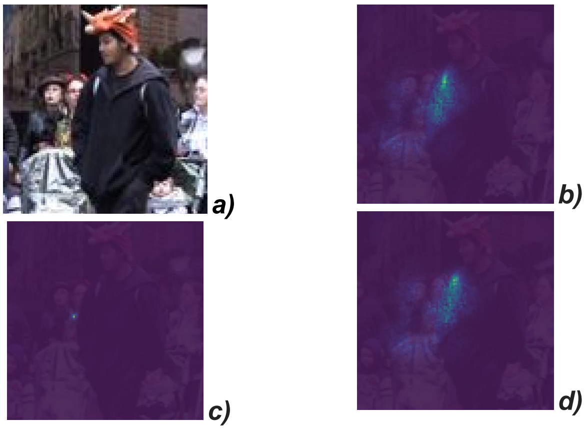


Figure 10. *Saliency map*
a) is the original image, b) is previous frame,
c) is current frame, and d) is next frame

Figure 11 illustrates our training progress using GAN. In this particular run on Attempt 3, we observed that meaningful improvement was done through GAN. On (b), we see PSNR values increase by non-GAN training, but meet a dead end. The graph retreats as we start GAN epoch at 0, and we see PSNR increase after GAN training. We can see from (c) and (d) the generator and discriminator interacting with each other, and (e) the total loss we defined from SRGAN reducing.

Conclusion and Future improvements

Conclusion

Super-resolution has shown some success, showing results clearly better than bilinear interpolation. For frame interpolation, we concluded we needed a different network architecture, as the simple U-net architecture struggled to converge the two scenes.

We have made meaningful observations on the use of optical flow and how the adjacent frames help super-resolution, as seen from our saliency map and some reconstruction results.

Other than that, we have created a pipeline and dataset of videos that can be easily re-purposed for a variety of video-related tasks.

Future Improvements

Since we have a straightforward training pipeline that we can utilize for frame interpolation and super-resolution, we can improve our project in various areas.

To begin with, we can investigate clearly why the ‘stairway’ phenomenon happens on inputting three same images. Making it clear whether our addition of flow positively benefits reconstruction results.

There are many good techniques for frame-interpolation that we may implement. Xiang, Xiaoyu et al[5] used deformable convolution as a frame-interpolation method, being very effective even at large-motions. Bao, Wenbo et al[6] also used depth estimation to get better flow estimation and overall contribute to frame interpolation.

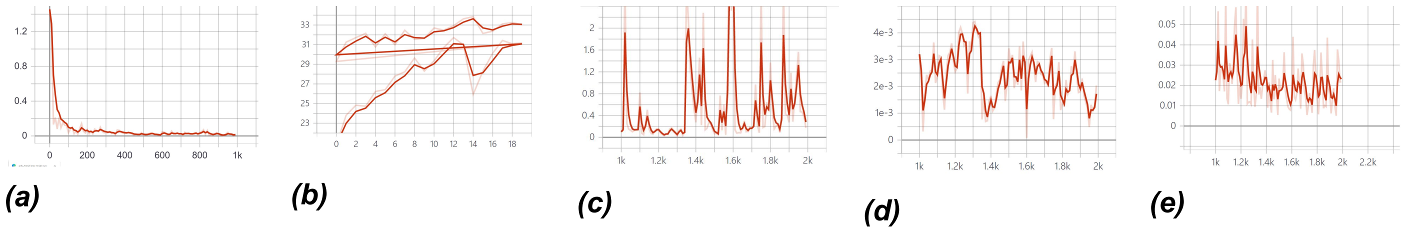


Figure 11. *Graphs of training results*
Graph (a) is Iteration/Loss(L2) of our pre-training process before using GAN, Graph (b) is change of PSNR Value per epoch for both pre-training and GAN training processes (note epoch is reset to 0 when starting GAN phase). Graph (c) is discriminator loss of GAN training, Graph (d) is adversarial loss of GAN training. Graph (e) is Iteration/Loss graph of combined Loss.

For our networks to be applicable in real-time scenarios like zoom, we would need to consider performance. Xiang, Xiaoyu et al[5] combined both frame interpolation and super-resolution in one network, achieving large performance benefits.

We have identified many ways to improve our training. For our GAN training methodology, we have trained the discriminator network by running countless iterations through the same code, hoping for meaningful adversarial training, yet, it was hard to match the generators and discriminator's learning status this way. More considerate methods such as progressive training may yield more consistent results. We may apply different weights for our GAN loss function as well as weighing skip connections suggested from Lim,Bee et al[1]**.** Finally, considering our difficulty in letting the flow-net convey useful information, we may attempt to load pre-trained weights next time.

Expanding our super-resolution network to handle 4x or 8x resolution increase may be possible as well. Considering the scale of super-resolution only differs at the end, we can initialize the majority of weights using our pre-trained 2x network.

Individual Contributions

All members of the team, through real-time collaboration, contributed to the basic pipeline creation, dataset collection and making baseline code for Attempt 1. This allowed everyone to have a good understanding about our program. Woongro implemented Attempt 2, and the flow-net, Jaehyun implemented end-to-end video enhancer program and was a presenter, Hyunjae implemented Attempt3, benchmark code, Frame interpolation, and generation of saliency maps.

References

1. Lim, Bee, et al. "Enhanced deep residual networks for single image super-resolution." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017.

2. Sajjadi, Mehdi SM, Raviteja Vemulapalli, and Matthew Brown. "Frame-recurrent video super-resolution." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

3. Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

4. Wang, Longguang, et al. "Deep video super-resolution using HR optical flow estimation." IEEE Transactions on Image Processing 29 (2020): 4323-4336.

5. Xiang, Xiaoyu, et al. "Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

6. Bao, Wenbo, et al. "Depth-aware video frame interpolation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.