Image and Video Super-Resolution

Dominik Chodounský

FIT CTU

chododom@fit.cvut.cz

December 31, 2021

1 Overview

The aim of this project was to use deep learning techniques to upscale the resolution of images and subsequently videos via synthetic generation.

More specifically, the project explores the usage of the U-Net and GAN architectures to achieve this goal.

2 Data

The data used comes from the Vimeo-90K Dataset. It is a large and diverse dataset of video clips from the *Vimeo* hosting platform. More specifically, I utilized the provided 6 GB septuplet version of the dataset which includes 7 824 7-frame video sequences extracted from different video clips from the Vimeo-90k. This dataset was specifically designed for video denoising, deblocking and super-resolution tasks. I trained the super-resolution models on 90 % of the data (49 287 frames) and then evaluated them on the remaining 10 % (5 481 frames).

As a bonus test, I picked a longer video and applied the super-resolution to each frame and reconstructed it to visually evaluate the effect this approach has on longer sequences. The specific video was *Seoul South Korea Low Traffic Shot*[1], which depicts a busy street with passing traffic. The video is 12 seconds long and consists of 304 frames.

3 Methods

The two chosen neural network architectures for the super-resolution task were the U-Net[4] and GAN[1]. I adapted the standard U-Net so that its contracting path accepts 64×64 px RGB images, and the expanding path upscales them a step further to output 128×128 px RGB images. I call this adaption the *SR-U-Net* and its architecture is displayed in the attached source codes and notebooks. During training, my custom data generator reads a batch of 7 images (one of the training septuplet sequences). Next, per each image, a copy of the image is scaled down to 64×64 px using cubic interpolation (this is the low-resolution input) and another is scaled down to 128×128 px to have a high-resolution ground truth to compare the network output with. Both images are normalized to range between 0 and 1 to stabilize the training. The SR-U-Net was then trained with the Adam optimizer (initial learning rate of 0.001) for 30 epochs, utilizing model checkpointing where the model with the lowest loss is saved.

As for the GAN architecture, its design was inspired by [3]. The so-called *SR-GAN* consists of a generator which accepts 64×64 px RGB images that are then run through a series of residual blocks containing double convolutions with batch normalization and a trainable parametric ReLU activation function. Following this, the features are upsampled to the desired dimensions of 128×128 px. As for the discriminator of the GAN, it accepts 128×128 px RGB images and outputs a probability that the input is a real high-resolution image rather than a fake one generated by the generator, which creates the adversarial relationship in which GANs are trained. The training of the SR-GAN is done for the duration of 10 epochs by the Adam optimizer with an initial learning rate of 0.0001.

I have experimented with two loss functions, one being a standard mean squared error (MSE) and the other is the so-called perceptual loss as described by [2]. The perceptual loss passes both the generated image and the ground truth image through a convolutional network which performs feature extraction and the mean squared error is computed on the output features. For my model, I used the *VGG19* feature extractor pre-trained on the *ImageNet* dataset. The SR-U-Net was trained using both the MSE and perceptual loss. As for the SR-GAN, the discriminator uses binary crossentropy, generator uses either the MSE or the perceptual loss and the whole GAN uses a weighted sum where either MSE or perceptual loss are weighted with 1 and the binary crossentropy of the discriminator's output is weighted by 10^{-3}, as is described in [3].

[1] *https://www.videvo.net/video/seoul-south-korea-low-traffic-shot/1458/.*

4 Evaluation and Results

After performing some pilot tests, I ended up with four configurations for evaluation: comparison of the SR-U-Net and SR-GAN architectures and comparison of MSE and perceptual loss for both of them. Each configuraton was first evaluated on the test set (all frames in the 10 % of the set aside septuplet sequences), calculating the peak signal-to-noise ration (PSNR), structural similarity index measure (SSIM) and mean squared error (MSE) of the generated super-resolution image and the original high-resolution image. All three statistics are commonly used for image reconstruction tasks and each evaluates a slightly different criterium, so all three are useful for a proper evaluation. The PSNR is measured in decibels and the higher it is, the better the image generation. For SSIM, we also look for higher numbers which range from -1 to 1 and with MSE, we strive for results as close to 0 as possible.

To test the models on a video that is longer than just a sequence of 7 frames, I also calculated the statistics per each frame of the footage of the passing traffic which was mentioned in Section 2. The frames were then combined back into a dynamic sequence and evaluated visually in comparison to the original video.

The recorded results can be seen in Table 1. As we can observe, all models performed much better on the test data that came from the Vimeo 90k dataset. Although the test images were different from the trianing ones, they may have had similar parameters that made them easier to perform super-resolution on rather than the chosen sample video. In this experiment, the SR-U-Net trained with MSE loss achieved the best final MSE. As for the other reconstruction quiality metrics PSNR ans SSIM, the best results were achieved by the SR-GAN trained with MSE loss. I would rate this model to be the best performing on this data, because MSE isn't such a reliable way to compare images and this is especially so when it itself was the criterium of optimization.

If we now look over to the second experiment with the sample video, we can see that SR-U-Net trained with MSE loss achieved best results in all three metrics with SR-GAN with MSE being just slightly behind. It seems that the models trained with perceptual loss are lagging behind, with the SR-GAN being at least quite comparable with the successfull models on the video example, but for example the SR-U-Net trained with perceptual loss performed terribly in all tests. An example of its deficiency can be seen in Figure 1, where we notice small regular squares covering the whole image, which are the likely cause of the model's high error rate. An example of the perceptual SR-GAN's failure can be seen in Figure 2, where the synthetic generation creates unnatural colors in a part of the image. However, with the SR-GAN, this is a relatively rare case and most images looked fine, which is why the results aren't as bad as is the case with the perceptual SR-U-Net.

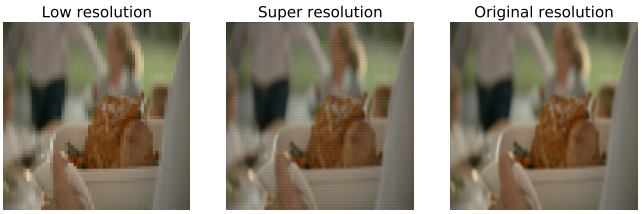

Figure 1: Result of super-resolution performed by SR-U-Net trained with perceptual loss.

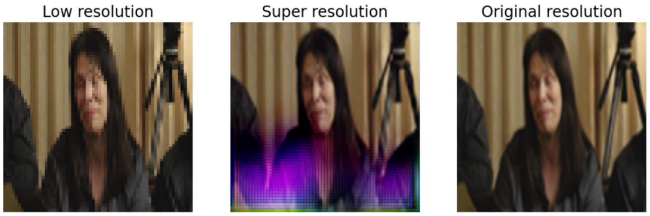

Figure 2: Faulty image generated by SR-GAN trained with perceptual loss.

As for the MSE-trained models, we can see an example of their performance in Figures 3 and 4 for the SR-U-Net and SR-GAN respectively.

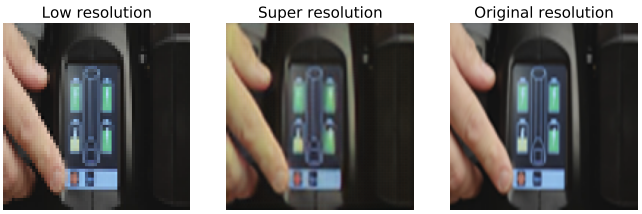

Figure 3: Result of super-resolution performed by SR-U-Net trained with MSE loss.

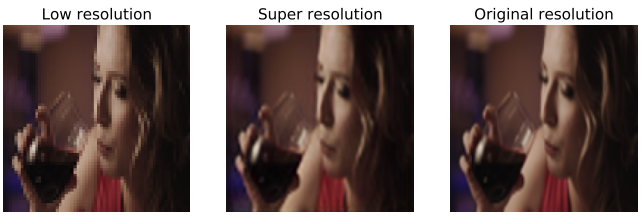

Figure 4: Result of super-resolution performed by SR-GAN trained with MSE loss.

The SR images certainly have better resolution and don't look as pixelated as the low-res versions, but we notice more blurryness and perhaps a slight color change to the original high resolution version. This could perhaps be caused by an incomplete convergence of the generator during the training, as for example the MSE-trained SR-GAN's loss progression shows in Figure 5. This leads me to believe that letting the model train for a longer period of time would stabilize it and perhaps lead to even better results.

As is clear from the calculated matrics in Table 1, the experiment did not perform so well on the chosen video, which was somewhat expected as it came from a different domain. The best result was achieved by the SR-U-Net trained with MSE loss and this video is included in the Gitlab repository of the project. We can see an increase in resolution, but there is slight flickering and color change.

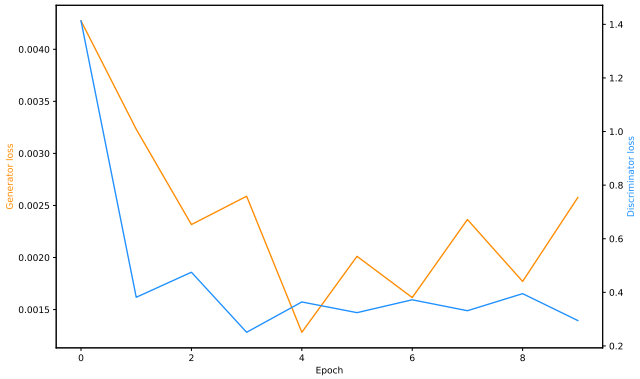

Figure 5: Generator and discriminator loss of the SR-GAN during MSE training.

5 Conclusion

Overall the experiment was a success, I managed to create several models which perform the task of image and video super-resolution to various degrees of success. I found that using MSE loss led to better results than the more commonly applied perceptual loss, but I believe that this would change if I had more time and computational resources to train the models and give them more time to converge and perhaps even train them to perform lagre upscaling for example to 256×256 px.

I have several ideas on how to improve the results aside from longer training and more diverse training datasets. I believe that in the case of video super-resolution, utilizing some recurrent layers in the network architecture could stabilize the generated images throughout the temporal dimension, because as of now, each image is generated with some amount of noise and artificial artifacts and because there is no link along the temporal dimension, each image has slightly different artifacts and this may be the cause of the recorded flickering.

Another idea that may improve the perceptual-loss-based models could be training the VGG19 network on the dataset at hand. The one used in this task was trained on higher resolution images in the ImageNet dataset and could have extracted slightly different features than in the downsampled images used in my experiments.

Lastly, there is another super-resolution approach which uses the U-Net architecutre. The idea is to use standard cubic interpolation to upscale the resolution and then train the U-Net to correct, sharpen and re-color the image to be as close to the original high-resolution version as possible.

On the basis of my research and experimentation, I would say that performing super-resolution is a very doable task and given more computational resources, we could even upscale extremely pixelated videos to life-like resolutions.

References

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[2] Xiaodan Hu, Mohamed A. Naiel, Alexander Wong, Mark Lamm, and Paul Fieguth. Runet: A robust unet architecture for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[3] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. 2017.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

Table 1: Results of training the SR-U-Net and SR-GAN with various losses on the Vimeo test set and the selected video example.

Model	Loss	Vimeo test set			Video example		
		MSE	PSNR	SSIM	MSE	PSNR	SSIM
SR-U-Net	MSE loss	**0.00214**	27.92390	0.89966	**0.02137**	**16.91301**	**0.51773**
	VGG19 loss	7107.86470	3.07393	0.43865	394342.75	-53.91510	0.023940
SR-GAN	MSE loss	0.00502	**29.96834**	**0.91631**	0.02736	15.92611	0.51506
	VGG19 loss	0.12397	16.88628	0.64679	0.06501	13.90345	0.43450