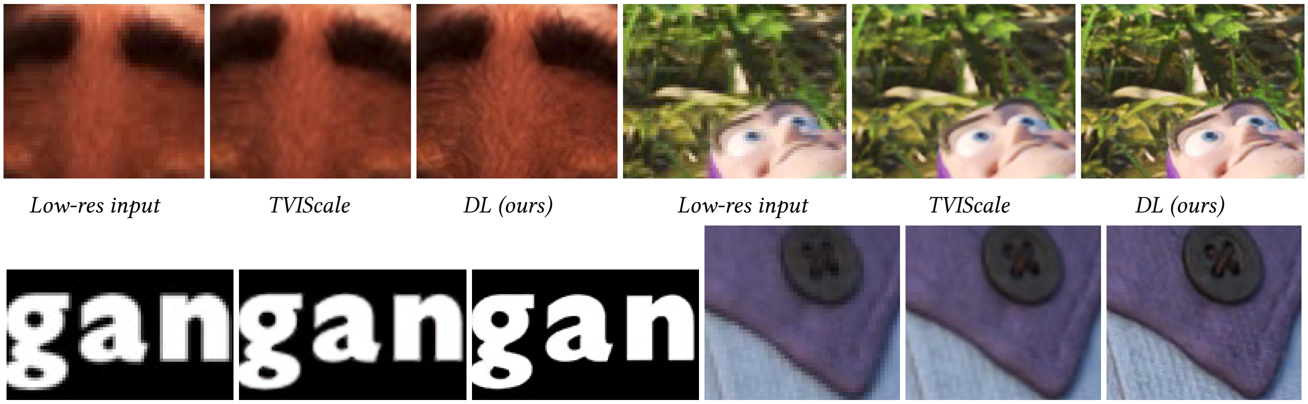
Deep Learned Super Resolution for Feature Film Production

Vaibhav Vavilala
Pixar Animation Studios
vibe@pixar.com

Mark Meyer
Pixar Animation Studios
mmeyer@pixar.com



Low-res input *TVIScale* *DL (ours)* *Low-res input* *TVIScale* *DL (ours)*

Low-res input *TVIScale* *DL (ours)* *Low-res input* *TVIScale* *DL (ours)*

Comparing a low-res input (left) with a TVIScale enhanced output (center) and deep learned enhanced image (ours, right). ©Disney/Pixar

ABSTRACT

Upscaling techniques are commonly used to create high resolution images, which are cost-prohibitive or even impossible to produce otherwise. In recent years, deep learning methods have improved the detail and sharpness of upscaled images over traditional algorithms. Here we discuss the motivation and challenges of bringing deep learned super resolution to production at Pixar, where upscaling is useful for reducing render farm costs and delivering high resolution content.

CCS CONCEPTS

• **Computing methodologies → Computer vision tasks**.

ACM Reference Format:
Vaibhav Vavilala and Mark Meyer. 2020. Deep Learned Super Resolution for Feature Film Production. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '20 Talks), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407334

1 BACKGROUND & RELATED WORK

To increase the resolution of images, several techniques are commonly used such as nearest-neighbor, bilinear, and bicubic interpolation. Total Variational Inpainting has proven fruitful for upscaling in the film industry via the Nuke node, TVIScale. In recent years, deep convolutional neural networks have demonstrated even greater reconstruction quality by learning the low-resolution (LR) to high-resolution (HR) mapping from a high volume of data. With the introduction of generative adversarial networks (GANs) and perceptual loss functions in the seminal SRGAN work, upscalers can now produce images with details and sharpness indistinguishable from the ground truth. The aim of our work at Pixar is to put GANs into production for upscaling.

2 TRAINING DATA

Training a super resolution model requires pairs of high-resolution and low-resolution frames. Most prior research relies on pre-built datasets comprised of high-resolution photographs representing the HR, and LR photos obtained by employing a downsampling operator (typically bicubic). However, in our testing, pre-trained super resolution models trained on bicubic downsampled pairs do not generalize to novel data (i.e. no new details are synthesized) because real-world degradation operators are much more complex than bicubic downsampling. At Pixar, we have a renderer to synthesize the pairs of images, high quality scenes (with diverse shaders, geometry, and lighting conditions), and a datacenter fine-tuned to render at tremendous scale.

To obtain training data, we render 1K-2K pairs of production images using RenderMan, with shots randomly sampled from *Coco*, *Incredibles 2*, *Toy Story 4*, *Onward*, and *Soul*. As a given shot includes multiple passes (e.g. solid elements, volumes, and sky) that are composited at the end of the pipeline, we gather all elements excluding deep layers. We correct for known resolution-dependent parameters such as the micropolygon length and the camera ray differential multiplier. Despite this, we still occasionally notice incoherent pairs and exclude them from the training set. We note that we often cannot re-use previously finaled 2K renders because shots often change even after finaled due to production churn. We

SIGGRAPH '20 Talks, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7971-7/20/08.
https://doi.org/10.1145/3388767.3407334

also obtain pre-existing 4K renders from the UHD releases and add any coherent 2K-4K pairs to the dataset (typically a small fraction of each show is rendered at 4K, and the rest is upscaled from 2K with TVIScale, as large-scale 4K rendering is cost-prohibitive). We end up with a dataset containing 3700 pairs.

3 TRAINING PIPELINE

We observe in the literature that the state-of-the-art super resolution deep learning models use deep residual networks with adversarial training [Wang et al. 2018]. Minor differences in the architecture or loss functions differentiate these works. We begin with an open source training pipeline and network architecture. We then create a PyTorch development environment, and prepare a Linux instance with two 24GB NVIDIA Quadro P6000 GPUs.

4 PRODUCTION FEATURES

Most work in super resolution does not take into account high dynamic range (HDR) imagery, but doing so is crucial in the film industry. HDR data is typically represented with floating point intensities exceeding 1, stored in a file format like OpenEXR. As neural networks perform best with input data normalized between $[-1, 1]$ or $[0, 1]$, we apply the following range compression function to accommodate HDR data [Bako et al. 2019]: $T_y = \kappa log(1+\mu y)/log(1+\mu)$. We set $\kappa = 0.6$ and $\mu = 5000$, providing range up to luminance values near 300. We then convert our range-compressed dataset into a high-performance data structure, the lightning memory-mapped database, accelerating training speeds by about 1/3 over reading EXR images directly.

We experiment with a novel on-the-fly data augmentation strategy, whereby we introduce random color shifts to the (LR,HR) pairs to make the network more robust to diverse lighting conditions. We do so by adding a random color $(c_r, c_g, c_b) \in [-0.01, 0.01]$ patchwise to each of the color channels: $R' = R + c_r$, $G' = G + c_g$, $B' = B + c_b$. We find that this improves generalization. We note that the training pipeline we adopt also performs random flips and rotations of the (LR,HR) pairs, which further improves robustness.

A key challenge was addressing unwanted color shifts in the generated image. We introduce an additional loss term that penalizes the $\ell 1$ loss between the downsampled generated image and the input LR image. With this additional loss term, we have not observed any color shifts in the trained network output. We perform hyperparameter tuning experiments to maximize the sharpness of the synthesized image while avoiding excessive noise artifacts that commonly accompany GANs. Our latest trained network occasionally produces edge artifacts on text (such as end credits) which we anticipate obtaining more training data with text will help eliminate. The PSNR-only network (no GAN) does not produce edge or noise artifacts, but blurs some high frequency details such as film grain or small highlights.

Our network is pretrained with a PSNR-only loss term for $215k$ iterations, then trained with GAN, color shift loss, and a perceptual (feature) loss term for $340k$ iterations. We importance sample 192x192 patches based on intensity with minibatch size 20. The weights for the loss terms are {PSNR, colorShift, feature, GAN} $= \{1, 5, 0.1, 0.0005\}$ Our training time is 108 hours for PSNR-only and 208 hours with GAN (316 hours total training time). Our inference time from 2K to 4K is around 15 seconds on the GPU, and we emphasize the model can upscale at any input resolution (limited by GPU/CPU RAM). Later in development we started training on NVIDIA's DGX2 supercomputer, which accelerated training speeds by around 5x and enabled running multiple experiments concurrently.

5 RESULTS

We have trained and deployed a production-quality super resolution model that consistently produces high-quality, artifact-free upscaled images in most cases - especially where TVIScale synthesizes insufficient detail and a 4K render would have been required. The quality is consistent even on scenes with depth of field, motion blur and/or volumes, as these phenomena were represented in the training set. Elements of the *Onward* UHD disc were upscaled with our model, and broader usage is planned on *Soul*. Our work is under active testing for other use-cases such as promotional work and theme park deliverables. Further, our latest trained model shows promise towards a pipeline where we can render at 1K and upscale to 2K, which would save 50-75% of the studio's renderfarm footprint if used for all intermediate renders.

6 FUTURE WORK

Our super resolution model only supports single-frame upscaling of the RGB color channels, making our work useful only at the end of the pipeline. For usage in intermediate renders, upscaling alpha is required, which is non-trivial given that pre-trained discriminator networks are generally RGB-only. While our current model produces temporally coherent results without taking in cross-frame inputs as in [Bako et al. 2017], doing so is still expected to help. We anticipate accounting for auxiliary feature buffers such as normals and albedo will further enhance the quality of synthesized images. Additionally, we expect our training time and inference speed to improve with network quantization and pruning. Finally, while we have focused on 2x upscaling so far, 4x upscaling has shown great promise in prior work and could be helpful for high-resolution promotional renders and delivering content for 8K TVs recently entering the market. Towards this end, running a 2x network for multiple passes has shown potential in our testing.

ACKNOWLEDGMENTS

The authors thank Christopher Schroers, Jeremy Newlin, Jack Zhao, Freeson Wang, Dominic Glynn, Eric Pearson, Mark Dinicola, John Halstead, and Steve May for contributing to this project.

REFERENCES

Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 527–542.

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* 36, 4 (2017), 97–1.

Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. 2018. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 864–873.