SuperTran: Reference Based Video Transformer for Enhancing Low Bitrate Streams in Real Time

Tejas Khot* Nataliya Shapovalova† Silviu Andrei† Walterio Mayol-Cuevas
Amazon

Abstract

This work focuses on low bitrate video streaming scenarios (e.g. 50 – 200Kbps) where the video quality is severely compromised. We present a family of novel deep generative models for enhancing perceptual video quality of such streams by performing super-resolution while also removing compression artifacts. Our model, which we call SuperTran, consumes as input a single high-quality, high-resolution reference images in addition to the low-quality, low-resolution video stream. The model thus learns how to borrow or copy visual elements like textures from the reference image and fill in the remaining details from the low resolution stream in order to produce perceptually enhanced output video. The reference frame can be sent once at the start of the video session or be retrieved from a gallery. Importantly, the resulting output has substantially better detail than what has been otherwise possible with methods that only use a low resolution input such as the SuperVEGAN method. SuperTran works in real-time (upto 30 frames/sec) on the cloud alongside standard pipelines.

* Work done while at Amazon
† Equal contributions

Figure 1: Qualitative result comparison. Comparison of results on two scenarios at low bitrates (50 and 200 kbps). The H264 frame is fed as input to SuperVEGAN[3] and SuperTran. SuperTran also consumes the reference frame. SuperTran is able to borrow details from the reference frame under change of scale and orientation.
1. Introduction and Related Art

Video transmission accounts for over 75\% of global IP traffic\cite{9} and is only expected to increase with time. This paper is addressing the problem of transmitting high resolution videos (\(\geq 720p\)) under low bitrate (e.g. \(< 200Kbps\)) settings. As the adoption of mobile phones having high-quality cameras continues to increase, such low-bandwidth, high-demand circumstances occur quite often, spanning geographies.

Video super-resolution and enhancement has been a long studied task with a rich history of work. We only discuss representative works here and refer interested readers to\cite{4, 18, 24} for excellent surveys of techniques for super-resolution and overall enhancement of videos.

**Video super resolution.** Deep learning based techniques have dominated benchmarks on this task. Recent work has shown that significant improvement can be made by accounting for motion implicitly\cite{28} or explicitly\cite{12, 19, 3}. Generation of perceptually realistic details in the outputs\cite{8, 16, 17, 3} has been with use of GANs\cite{10}.

**Video Enhancement.** Works on enhancement have tried to tackle correction of compression related spatial and temporal artifacts such as blocking, ringing, flickering, and color changes from reduced color depth with decreasing bitrate. Deblocking has been tackled using design of targeted correction filters\cite{13, 14}, Kalman filters\cite{15}, and optical flow\cite{6, 25}. Some works performed video enhancement in conjunction with super-resolution by incorporating frame alignment modules\cite{21}, offline detection of non-PQF’s (Peak Quality Frames)\cite{27} and combination of learned upsampling filters with GANs\cite{3}.

**Reference-based methods.** Adjacent to the previous approaches, there is a body of work focused on harnessing additional reference imagery in order to improve performance on tasks like colorization and super-resolution. Alignment based approaches\cite{22, 29, 33, 29} focus on aligning the reference and low-resolution videos via a registration procedure\cite{29}, via warping based feature synthesis\cite{22} or using optical flow\cite{33}.

Some methods\cite{7, 31, 32} adopt a patch-based alignment procedure to extract detail from reference images via local matching. In addition to the pixel space, this can also be done on image gradients\cite{7} and semantic features from CNNs\cite{32, 20, 26}. These prior works are restricted to super-resolution and operate on single images, not videos. SuperTran is aimed at improving upon video enhancement with aid from reference imagery.

2. Methodology

A strategy for dealing with low bitrates is to transmit videos at a reduced resolution and compensate for the quality by performing enhancement later\cite{3}. To the best of our knowledge, SuperTran is the first model to jointly tackle super-resolution, video deblurring and overall video enhancement with a reference-based approach.

Given a sequence of \(2k + 1\) consecutive low-resolution, low-quality frames \(I_{in}^{t-k:t+k}\) from a video stream, the goal of video enhancement is to estimate a high-resolution, high-quality frame \(I_{out}^{t}\) corresponding to \(I_{in}^{t}\). SuperTran uses a reference frame \(I_{ref}^{t}\) as additional input. Our system to achieve this task is illustrated in Figure 2. SuperTran consists of two main modules, namely the Backbone network and the Multi-Scale Texture Transformer network (MSTT).

2.1. Backbone Network

The backbone network is the central CNN architecture making up SuperTran. This network takes as input one or more low resolution, compressed input images (video frames) and produces as output an enhanced version of the central input frame. The backbone network receives as additional input the features processed by the Transformer network at various stages. Consuming neighboring frames as input allows the model to learn temporal patterns in data.

The design of SuperTran is optimized for high inference speed. Thus, all operations are performed at the low input image resolution. As shown in Figure 3 SuperTran consists of a series of stacked residual blocks with MSTT modules interleaved periodically. Features at multiple points in the network are collected and processed jointly to obtain the final upscaled output with a pixel shuffle layer.

2.2. Multi-Scale Texture Transformer

The goal of the texture transformer module is to extract textures from the reference image such that they closely match with the textures in the input image. Since the reference frame is of higher visual quality, the extracted features, when transferred, can aid the backbone network in
producing improved output. The texture transformer module consumes $I_t^{	ext{in}}$, reference image $I^{	ext{ref}}$ and its downscaled version $I^{	ext{ref}}_↓$. For rescaling images, we use bicubic interpolation. Note that while the Backbone network receives a set of input frames as input, the Transformer network only receives the current frame in addition to the reference frame.

**Learnable Texture Extractor.** The Learnable Texture Extractor (LTE) is a CNN that consumes an image and produces high dimensional feature representations for it at multiple scales. These features can be thought of as textures. We utilize weight sharing for LTE and learn a joint feature space for $I_t$, $I^{	ext{ref}}$ and $I^{	ext{ref}}_↓$. The outputs from LTE are treated as the query $Q$, key $K$, and value $V$ for the next step and are the key ingredients of the transformer module.

**Feature matching.** We perform patch-level similarity matching between $Q$ and $K$ by first extracting $d \times d$ size patches from both. The similarity (relevance) score $R_{i,j}$ is computed using a normalized inner product between patches $Q_i$ and $K_j$.

$$R_{i,j} = \left( \frac{Q_i}{\|Q_i\|} \cdot \frac{K_j}{\|K_j\|} \right)$$

**Attention Mechanism.** Using the relevance scores, we transfer textures from reference image in two ways. We utilize hard attention to transfer only the most relevant (best match) textures and soft attention to transfer a synthesized blend of relevant textures. For hard attention, we extract the indices $H_i$ for the most relevant matches and use them to obtain the transferred features $T_i$ by performing a selection operation on the patches $V$.

$$H_i = \arg \max_j R_{i,j} \quad T_i = V_{H_i}, \forall i$$

The soft attention map $S_i$ is computed from the relevance scores to indicate confidence for the best matching texture.

$$S_i = \max_j R_{i,j}$$

As a final step, we concatenate the features from $I_t^{	ext{in}}$ and the transferred features and apply elementwise soft attention to obtain blended features. The texture matching (TM) components described above along with texture transfer (TT) at multiple scales are shown in Figure 3.

### 2.3. Loss Function

We train SuperTran in a multi-objective optimization setting with a loss function consisting of 4 components: pixel-wise reconstruction loss $\mathcal{L}_{\text{rec}}$, adversarial loss $\mathcal{L}_{\text{adv}}$, perceptual loss $\mathcal{L}_{\text{per}}$, and texture loss $\mathcal{L}_{\text{tex}}$. For the definitions below, we consider $N = C \times H \times W$ where $C, H, W$ are the number of channels, height and width respectively.

$$\mathcal{L} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{per}} \mathcal{L}_{\text{per}} + \lambda_{\text{tex}} \mathcal{L}_{\text{tex}}$$

**Reconstruction Loss.** This is measured with the per-pixel $L_1$ distance.

$$\mathcal{L}_{\text{rec}}(I^{gt}, I^{out}) = \frac{1}{N} \| I^{gt} - I^{out} \|_1$$

**Adversarial Loss.** We use WGAN-GP[11] as our GAN loss which uses a gradient norm penalty in addition to the Wasserstein GAN formulation[5]. The loss is given by,

$$\mathcal{L}_{D} = \mathbb{E}_{\tilde{x} \sim P_{\tilde{x}}} [D(\tilde{x})] - \mathbb{E}_{x \sim P_{\text{real}}} [D(x)] + \lambda \mathbb{E}_{\tilde{x} \sim P_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2],$$

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\tilde{x} \sim P_{\tilde{x}}} [D(\tilde{x})],$$
where $D$ is our discriminator, $x$, $\tilde{x}$ are samples from real and generated distributions respectively and $\mathbb{P}$ denotes the corresponding probability distribution.

**Perceptual Loss.** Our perceptual loss consists of two terms measuring similarity in the feature space. In addition to the typical loss in the VGG19 feature space for output and ground truth image, we use an additional loss between the LTE transferred features and those of the output image. This ensures that $I^{out}$ retains the transferred features $T$.

$$L_{per} = \frac{1}{N} \left\| \phi^{vgg}(I^{out}) - \phi^{vgg}(I^{gt}) \right\|^2_2 + \frac{1}{N} \left\| \phi^{lte}(I^{out}) - T \right\|^2_2$$

(8)

Here, $\phi$ represents features extracted from intermediate network layers for VGG and LTE networks.

**Texture Loss.** Similar to the perceptual loss, we compute this loss in the VGG19 space, but with a model that was trained for the task of texture classification on the MINC dataset. We observe that this loss brings stability to outputs.

$$L_{tex} = \frac{1}{N} \left\| \phi^{minc}(I^{out}) - \phi^{minc}(I^{gt}) \right\|^2_2$$

(9)

Here, $\phi$ represents features extracted from intermediate network layers of the MINC trained VGG network.

3. Experiments

**Datasets.** For training, we collected 10 video sequences spanning 17 minutes each, recorded at 1080p resolution (i.e. 1920 x 1080px). These videos depict outdoor scenery and are recorded from a device different from the test videos. From these videos, we sample 20,000 frames and resize them to 1280 x 720px to make up the ground truth for training dataset. For testing, we have curated a Talking Heads dataset containing a collection of high quality videos of people speaking into the camera (on laptop or a handheld mobile device) in a casual format akin to a video calling setup. These videos have been recorded under a diverse range of settings including varying lightning conditions, camera motion, surroundings, camera orientation, video resolution and the subjects (talking heads) in the video. The dataset consists of a total of 71 videos spanning over 1 hour 36mins with an average video length of 1 min 11 sec.

To create paired data, we first bilinearly downscale the videos to 320 x 180px and encode them using the H264 implementation in Gstreamer[11], modifying only the bitrate parameter of the encoder. The downsampling factor (here, 4x) can be easily modified to derive multiple SuperTran variants. Note that our framework is not restricted to the H264 video codec and can adapt to other codecs with corresponding training.

**Training Setup.** We train using randomly sampled 96 x 96px patches from the low-resolution, compressed image dataset and use a window of 5 frames as input. The model is tasked to produce a 4x upsampled output corresponding to the middle input frame. For the loss function, we set $\lambda_{te}$, $\lambda_{adv}$, $\lambda_{per}$, $\lambda_{tex}$ to 1, 5e^{-4}, 1e^{-2} and 1e^{-2} respectively. We initialize the learning rate to 1e^{-3} and train the model for 500 epochs halving the learning rate after 300 epochs. For the first 20 epochs, we train using only reconstruction loss and enable other losses thereafter.

**Metrics.** We adopt metrics used in other works: PSNR, SSIM[23] and LPIPS[30]. PSNR and SSIM are the most common metrics used for assessing image quality, but they do not fully capture the perceptual quality of the image and favor blurriness. We mainly target the LPIPS metric, which is a learned metric in the CNN feature space and is optimized for high correlation with human judgement of perceptual details in images. The LPIPS metric was also found to correlate well for human evaluation of videos[3].

**Results.** We report quantitative results averaged over all videos in our test dataset in Table 1. For comparison, we consider the H264 codec with bicubic upscaling as the baseline and SuperTran as the representative state-of-the-art technique. We perform evaluation using inputs of resolution 320 x 180px and compute metrics at the 4x upsampled output resolution 1280 x 720px. For illustrative purposes, we show qualitative results on outdoor scenes in addition to our test data in Figure 1. Please refer to the supplementary material for video results.

| Method  | Bitrate | PSNR↑ | LPIPS↓ | SSIM↑ |
|---------|---------|-------|--------|-------|
| H264    | 50      | 27.43 ± 3.56 | 0.43 ± 0.11 | 0.82 ± 0.08 |
| SVEGAN[3] | 50      | 27.25 ± 3.37 | 0.23 ± 0.11 | 0.81 ± 0.08 |
| SuperTran | 50      | 26.32 ± 2.82 | 0.18 ± 0.11 | 0.82 ± 0.08 |
| H264    | 100     | 29.00 ± 2.66 | 0.29 ± 0.08 | 0.84 ± 0.06 |
| SVEGAN[3] | 100     | 28.71 ± 2.42 | 0.18 ± 0.07 | 0.84 ± 0.06 |
| SuperTran | 100     | 27.6 ± 1.94  | 0.13 ± 0.06 | 0.84 ± 0.06 |
| H264    | 200     | 30.03 ± 2.48 | 0.26 ± 0.07 | 0.86 ± 0.02 |
| SVEGAN[3] | 200     | 29.52 ± 2.17 | 0.15 ± 0.05 | 0.85 ± 0.05 |
| SuperTran | 200     | 28.14 ± 1.78 | 0.11 ± 0.04 | 0.86 ± 0.05 |

Table 1: Results at 50, 100 and 200kbps bitrate. For PSNR and SSIM, higher is better. For LPIPS, lower is better. SuperTran obtains up to 27% relative improvement in LPIPS over SVEGAN.

4. Conclusion

We introduce a novel video transformer for enhancing low bitrate video streams with the aid of reference imagery. Our proposed model architecture, SuperTran, is able to borrow visual detail from a single reference image and remove artifacts from video streams significantly better than with previous methods. SuperTran is domain agnostic in that it is not restricted to a specific application (e.g., human faces) unlike recent work[2]. Our cloud-based implementation of SuperTran works on live video at up to 30fps at 720p resolution and can support a variety of video streaming, processing and storage applications.
References

[1] Gstreamer. https://gstreamer.freedesktop.org/. Accessed: 2021-04-15.

[2] Nvidia Maxine. https://developer.nvidia.com/maxine. Accessed: 2021-04-15.

[3] Silviu S Andrei, Nataliya Shapovalova, and Waltero Mayol-Cuevas. SUPERVEGAN: Super resolution video enhancement GAN for perceptually improving low bitrate streams. IEEE Access, 9:91160–91174, 2021.

[4] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. arXiv preprint arXiv:1904.07523, 2019.

[5] Vivek Boominathan, Kaushik Mitra, and Ashok Veeraraghavan. Improving resolution and depth-of-field of light field cameras using a hybrid imaging system. In ICCP, 2014.

[6] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, Jo Younghyun, Oh Seoung Wug, Kang Jaeyeon, and Kim Jong Beom Ra. A deblocking filter with two separate modes and motion compensation. In CVPR Workshops, 2017.

[7] Jo Younghyun, Oh Seoung Wug, Kang Jaeyeon, and Kim Seon Joo. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. CVPR, 2018.

[8] Yuanbo Rao and Leiting Chen. A survey of video enhancement techniques. Journal of Information Hiding and Multimedia Signal Processing, 3(1):71–99, 2012.

[9] Yunbo Rao, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Learning cross-scale correspondence and patch-based synthesis for reference-based super-resolution. In BMVC, 2017.

[10] Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super-resolution network using cross-scale warping. In ECCV, 2018.