IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 7, 2021, accepted June 1, 2021, date of publication June 18, 2021, date of current version July 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3090344

SUPERVEGAN: Super Resolution Video Enhancement GAN for Perceptually Improving Low Bitrate Streams

SILVIU S. ANDREI, NATALIYA SHAPOVALOVA, AND WALTERIO MAYOL-CUEVAS
Amazon, Seattle, WA 98104, USA

Corresponding author: Nataliya Shapovalova (nataliys@amazon.com)

ABSTRACT This paper presents a novel model family that we call SUPERVEGAN, for the problem of video enhancement for low bitrate streams by simultaneous video super resolution and removal of compression artifacts from low bitrates (e.g. 250Kbps). Our strategy is fully end-to-end, but we upsample and tackle the problem in two main stages. The first stage deals with removal of streaming compression artifacts and performs a partial upsampling, and the second stage performs the final upsampling and adds detail generatively. We also use a novel progressive training strategy for video together with the use of perceptual metrics. Our experiments shown resilience to training bitrate and we show how to derive real-time models. We also introduce a novel bitrate equivalency test that enables the assessment of how much a model improves streams with respect to bitrate. We demonstrate efficacy on two publicly available HD datasets, LIVE-NFLX-II and Tears of Steel (TOS). We compare against a range of baselines and encoders and our results demonstrate our models achieve a perceptual equivalence which is up to two times over the input bitrate. In particular our 4X upsampling outperforms baseline methods on the LPIPS perceptual metric, and our 2X upsampling model also outperforms baselines on traditional metrics such as PSNR.

INDEX TERMS Video super resolution, artifact removal, video enhancement.

I. INTRODUCTION

Two important computer vision problems that benefit from the ability of Generative Adversarial Networks (GANs) to work with little input data are: 1) Video Super Resolution (VSR) and 2) Video Enhancement (VE) such as when the video has gained artifacts, lost sharpness or color depth from high levels of compression. Of these problems, VSR has been studied more widely.

We tackle both of these problems jointly and concentrate on the problem of dealing with high resolution video ($\geq$720p) that is sent live at high framerates (e.g. 30 fps) and at low bitrates (e.g. $\leq$250Kbps). Low bandwidth conditions are more common than they may appear and occur for both mobile and WiFi connections outdoors and at home. Constrained bandwidth affects live video transmission where no advance encoding or buffering is possible and is accentuated by the spread of IoT devices, home security cameras, video conferencing and streaming cameras from mobile devices. Video is already the most ubiquitous type of data transmitted with IP video currently using over 82% of the overall global IP traffic, and with the average internet household expected to have generated over 117.8 gigabytes of Internet traffic per month in 2020 [1].

In this work, we develop deep learning models for the joint VSR and VE problem, as well as evaluation methodologies to allow the generation of high resolution and high quality videos that originally been affected by video compression. In particular, we focus on GAN-based models and demonstrate how they can be successfully applied to the problem of joint video super resolution and artifact removal for videos compressed at low bitrates. In addition, we leverage advances in VSR such as Dynamic Upsampling Filters [3] that extract temporal information without explicit motion modeling, which can be challenging for highly compressed videos. Our main contributions are: I) The tackling of the streaming video enhancement problem by combining super resolution and artifact removal within an end to end network. II) A family of GAN-based models (SUPERVEGANs) for the streaming video enhancement task, including high-performing real-time models. III) Adaptation of progressive training to explicitly address artifact removal and generation of fine details and

The associate editor coordinating the review of this manuscript and approving it for publication was Yun Zhang.

IV) A novel bitrate equivalency test to quantify the effective perceptual gain that a model achieves at different bitrates with both user and image metrics.

Related works are presented in Sec. II. We introduce our model SUPERVEGAN in Sec. III before presenting our bitrate equivalency study in Sec. IV. Experiments and results are in Sec. V and conclusions are in Sec. VI.

II. RELATED ART

The literature in video and image processing is vast. Here we review works that learn to enhance video, or that are precursors to such. Since to the best of our knowledge there are no works that focus on video enhancement that do simultaneous video super resolution and compression artifact removal, we discuss works that target video super resolution only or video enhancement only.

Video super resolution (VSR) increases the resolution of lower resolution videos to higher resolved ones. From forensics to medical imaging to live video streaming or high definition displays, the numerous applications of VSR merit its high interest. Recent works using deep learning have proposed a variety of models for VSR [3]–[9]. A first take for video super resolution could be to treat every frame as independent and super resolve with an image based method, e.g. [6], [10]. But some indications exist that show it is better to treat video as a temporal signal. In [5] a CNN produces video super resolution with explicit motion compensation and demonstrates better metric results using PSNR over doing super resolution a frame at a time. In the case of [3], a network with two paths one for generating upsampling filters dynamically and one that computes a residual image, converge to produce higher resolution output from a small window of video frames. That work does not require explicit motion compensation and offers an end to end solution to VSR. In the case of [8], a recurrent architecture is proposed to improve temporal consistency. Recent work on GAN architectures for VSR [9], [11], [12], has started to show further performance gains. The appeal of GANs for this task is primarily given by their ability to use reduced image input and generate realistic outputs [13], [14]. In [9], a GAN is proposed that uses a spatio-temporal discriminator and a back and forth loss function for temporal consistency improvement. As with [8], explicit estimation of motion compensation is used. Many of the above models use networks that compute flow explicitly, while this may benefit super resolution, dealing with motion when compression artifacts are present, calls for a different strategy.

Video Enhancement (VE) is a more generic task requiring spatial and temporal effect to improve detail beyond resolution alone. Video compression results in spatial and temporal artifacts which include blocking, ringing and flickering artifacts. It may also involve color changes from reduced color depth as bitrate reduces. Various works have focused on VE before the popularity of deep networks. For deblocking, some earlier works concerned with creating specific filters to use for correction [15], or to adaptively produce filters [16]. Later, deep learning methods for VE emerged. In [17], a deep network within a Kalman filter framework recursively aims to remove lossy compression artifacts. In [18], the model works to interpolate frames and do video enhancement via the integration of optic flow and interpolation kernels. In [19], authors investigated various aspects of VE. The tasks of VSR, deblocking compression artifacts and frame interpolation are tackled via a task-specific optic flow estimation. In EDVR [20] the authors presented a method that achieves state of the art results for video super resolution and video restoration by using a PCD (Pyramid, Cascading and Deformable) alignment module and a temporal and spatial attention module. Another relevant work is MFQE2 [21]. In that work, the authors show that it is possible to enhance the quality of compressed videos by detecting PQF's (Peak Quality Frames) and use them to improve the quality of neighboring non-PQF frames.

Other works have looked at end-to-end encoding as a way to remove intermediate steps e.g. [22]–[25]. However we argue there is appeal if a method can work in tandem with existing standards like H264 that are widespread, hardware implemented, and for which many existing streaming technologies are tuned for.

Combining super resolution and deblocking in a joint task has important benefits for bandwidth reduction and detail preservation. Our proposed solution follows a straight-forward strategy: a smaller video can use less bandwidth and convey detail better during motion. And if detail can be recovered as part of being upsampled and deblocked, it will measure better than a video that is compressed with existing encoders at the same bandwidth at the original resolution. Our approach can be seen as one that aims to find a balance in the perception-distortion space [26], where perceptual and direct metrics are used to generate video output.

III. SUPERVEGAN

A. PROBLEM FORMULATION AND NETWORK ARCHITECTURE

The goal of our model is to enhance quality of a video compressed at a low bitrate by jointly performing video super resolution and artifact removal (see Fig. 1). In other words, given a set of $2N + 1$ consecutive frames $\tilde{X} = X_{t-N:t+N}$ of size $H \times W$ that belong to a low resolution video compressed at bitrate b, the goal is to generate a single high-resolution frame $\hat{Y}_t$ of size $rH \times rW$ that will belong to a video of a higher bitrate b' (see Sec. IV on how we determine b'):

$$\hat{Y}_t = G_r(X_{t-N:t+N}), \tag{1}$$

where G_r is our SUPERVEGAN generator model for an upsampling rate r; t is a current time step.

Since there is a compromise to be made based on target resolution and available bitrate b, we designed two versions of our network, **SUPERVEGAN-4** (G_4) and **SUPERVEGAN-2** (G_2). SUPERVEGAN-4 performs 4x upsampling with respect to the input and SUPERVEGAN-2 performs 2x

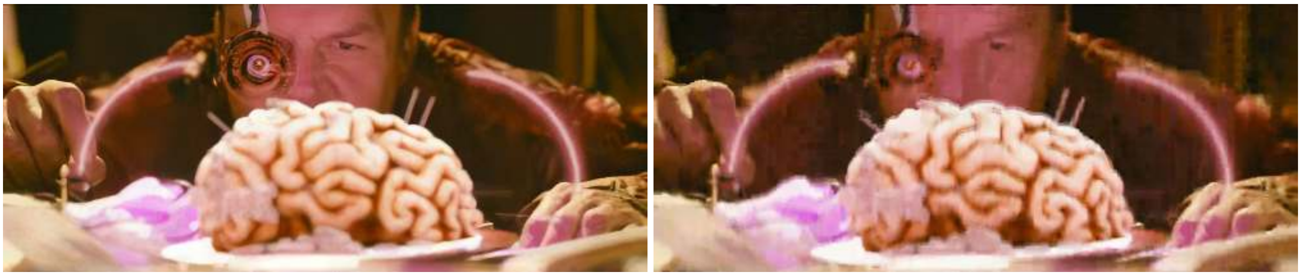


FIGURE 1. From a low bitrate (250Kbps) and low resolution (320 × 180px) input video, our method reduces artifacts and achieves high resolution (1280 × 720px) (left), compared with the original video compressed at the same bitrate with H264 (right). Image is taken from [2].

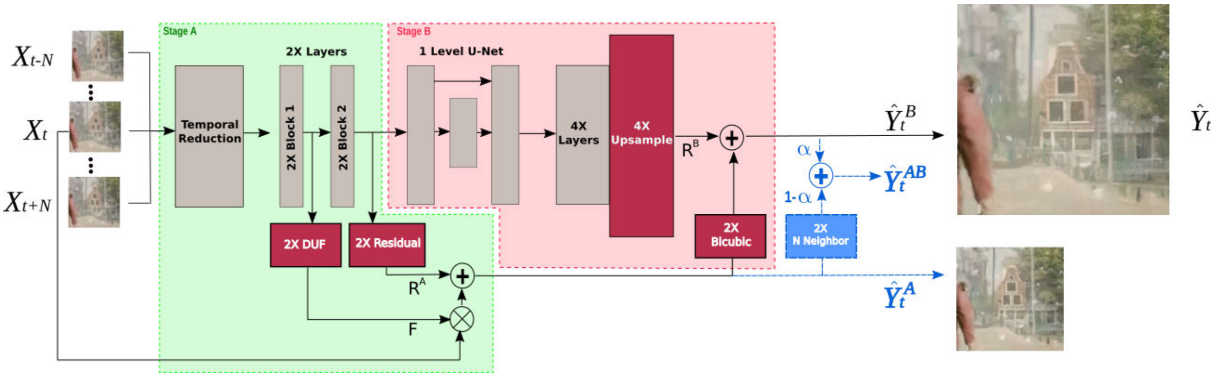


FIGURE 2. The SUPERVEGAN-4 generator architecture that jointly performs video super resolution and artifact removal. Dashed path is only used during the progressive training and not needed during inference. Stage A performs 2x upsampling and artifact removal by applying the 2x DUF filters as in Eq. 4 and then adding the 2x residual R^A as in Eq. 3. The temporal reduction layers reduce the temporal dimension to 1 by use of 3D convolutions with no padding in the temporal dimension. The 2x layers are split in two consecutive blocks, one block for computing the DUF filters and one for computing the residual. stage B adds sharpness and details by predicting a residual R^B that is added to the bicubicly upsampled output of stage A (see Eq. 2). The 1 Level U-Net from this stage is used for increasing the receptive field. A more detailed diagram is presented in Fig. 4.

upsampling. In our experimental results (see Sec. V-F) we demonstrate efficacy of such strategy: at lower bitrates it is better to use the 4x upsampling model, while at higher bitrates it is better to use the 2x upsampling model.

In the remainder of this section we describe our generic G_r and we will highlight whenever there are differences between G_2 and G_4.

1) GENERATOR WITH 2 STAGES

One of our contributions lies in the novel generator that contains 2 stages. In the first stage, we aim to reconstruct the original image as best as possible. We remove compression artifacts and reconstruct details by extracting information from multiple adjacent frames. In the second stage, we aim to synthesize missing details that cannot be recovered from the inputs alone. We do this by training this stage with an additional adversarial loss. For simplicity, we will refer to them as **Stage A** and **Stage B** correspondingly and we call the outputs of these stages $\hat{Y}_t^A$ and $\hat{Y}_t^B$ respectively. The final enhanced image $\hat{Y}_t$ is then:

$$\hat{Y}_t = \hat{Y}_t^B = \Omega_{\text{BIC}}(\hat{Y}_t^A) + R^B \tag{2}$$

where Ω_{BIC} is a 2x bicubic upsampling and R^B is a residual with high frequency details generated by Stage B in target resolution $rH \times rW$. The generator for the SUPERVEGAN-4 is depicted in Fig. 2. Note, while overall architecture for SUPERVEGAN-2 and SUPERVEGAN-4 is the same, implementation details (number of layers/filters) differ – our goal was to design networks that would have comparable running time. For more details on their differences see Sec. V-A.

2) STAGE A: DETAIL RECONSTRUCTION

This stage reconstructs the ground truth image in half resolution by removing artifacts and by extracting information from multiple input frames in a local spatio-temporal neighborhood. It is important to note that this stage only reconstructs details and does not hallucinate them as it is only trained with an MSE loss against the half-resolution ground-truth image. In this stage we generate an intermediate image $\hat{Y}_t^A$ in half the target resolution that is artifact free. To achieve this, we leverage Dynamic Upsampling Filters [3]:

$$\begin{aligned} \hat{Y}_t^A &= U_{X_t} + R^A, \\ U_{X_t} &= \Omega_F(F, X_t) \end{aligned} \tag{3}$$

where U_{X_t} is an upsampled frame X_t, Ω_F is an image upsampling function that applies dynamic filters F to the image X_t, and R^A is a residual image generated at Stage A. Networks generating F and R^A take $X_{t-N:t+N}$ as input and share most of their parameters and computations.

The filter branch predicts in total $\tilde{r}^2\, HW$ filters (one for each output pixel). The filter size we use is 5 × 5 as in the original paper [3]. Here, $\tilde{r} = r/2$ since in this stage we upsample to half the target resolution. Hence, for the case of SUPERVEGAN-4, there are $4HW$ upsampling filters and for the case of SUPERVEGAN-2 however, there are only HW filters (i.e. one filter per pixel). In case of SUPERVEGAN-4 the filters are applied as follows:

$$U_{X_t}(y\tilde{r}+v, x\tilde{r}+u) = \sum_{j=-2}^{2}\sum_{i=-2}^{2} F_{\tilde{r}}^{y,x,v,u}(j+2, i+2)X_t(y+j, x+i) \quad (4)$$

where $u, v \in \{0, 1\}$ are the offsets of the superresolved pixels corresponding to the pixel with coordinates x, y in the low-resolution input image. Since we perform 2x upsampling at this stage, each input pixel is mapped to 4 output pixels. Our 2x DUF layers predict four 5 × 5 dynamic filters F for each input pixel. Therefore, $i, j \in \{-2, -1, 0, 1, 2\}$ are offsets into a 5 × 5 filter F where the center pixel has coordinates (0, 0). During training (see Sec. III-B), we enforce MSE loss only; this allows preserving the original content and minimize distortion caused by artifacts. To extract temporal information from the given $2N + 1$ low resolution compressed frames, in the beginning we use a temporal reduction block – a series of 3D convolutions to reduce the temporal dimension from $2N + 1$ to 1.

3) STAGE B: ENHANCING DETAILS

The goal of this stage is to synthesize realistic looking details that have been lost completely and cannot be reconstructed from the inputs alone. This is achieved by training this stage using two additional losses apart from the MSE loss: an adversarial loss and a perceptual loss. In this stage, we generate a residual image R^B in $rH \times rW$ resolution that is added to the bicubicly upsampled output from Stage A to form the final image $\hat{Y}_t$, as shown in Eq. 2; see Sec. III-C for details. The architecture of this stage is simple: we add more layers on top of the feature output from Stage A. Having a small U-Net allows us to increase the receptive field for generating high fidelity details.

We want to emphasize the difference between Stages A and B. Even though they are similar in architecture (both performing upsampling and add residual on top), due to different optimization they focus on different parts of the distortion-perception tradeoff: Stage A focuses on reducing distortion, while Stage B focuses on enhancing perceptual quality. For a qualitative comparison of the outputs of the two stages, see Fig. 12 and for a quantitative analysis, see Sec. V-F

B. PROGRESSIVE TRAINING

SUPERVEGAN's generator G_r is trained in 3 phases. In the first phase, we only train Stage A targeting artifact removal for both G_2 and G_4 and 2x upsampling only for G_4. MSE loss is enforced on $\hat{Y}_t^A$. Layers that only belong to Stage B remain untrained. A diagram of our training system is depicted in Fig. 3.

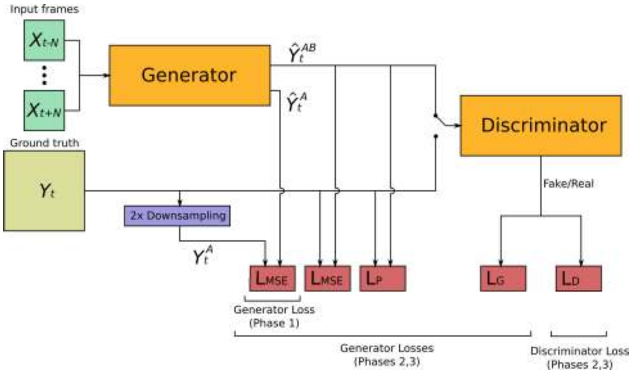


FIGURE 3. Our training system diagram. In phase 1, only MSE loss for stage A is being enforced. In phases 2 and 3, all losses are being enforced: MSE loss for stage A and MSE loss, perceptual loss, and adversarial loss for stage B; see sections III-B and III-C for more details.

In the second phase, we also enable Stage B. We gradually blend in output from Stage B with output from Stage A:

$$\hat{Y}_t^{AB} = (1-\alpha)\Omega_{\mathrm{NN}}(\hat{Y}_t^A) + \alpha\hat{Y}_t^B \quad (5)$$

where $\hat{Y}_t^{AB}$ is a blending between $\hat{Y}_t^A$ and $\hat{Y}_t^B$, Ω_{NN} is a 2x upsampling with nearest neighbor and α is a blending parameter that gradually changes from 0 to 1 over the training epochs of this phase. Note, that at inference, we set $\alpha = 1$, hence in the essence $\hat{Y}_t == \hat{Y}_t^B$, as it is indicated in Eq. 2 During this phase we also enable our discriminator. Since the discriminator is not used during the first phase, it does not need to have blended layers. The discriminator is composed of a series of convolutional downsampling blocks and a fully-connected layer at the end, refer to Sec.V-A for details.

In the third and final phase, we set $\alpha = 1$ and simply stabilize the network (both generator and discriminator). For both second and third phases, we enforce MSE, perceptual, and adversarial losses on $\hat{Y}_t^{AB}$ and MSE loss on $\hat{Y}_t^A$.

C. LOSSES FOR DISTORTION AND PERCEPTION

Recent works have highlighted the importance of metrics that are tuned for a perceptual index [27]. On the other hand traditional distortion metrics like MSE help preserve the general detail of the image without adding unnatural elements. Our architecture provides a structure to combine both metrics. We enforce the MSE loss on both $\hat{Y}_t^A$ and $\hat{Y}_t^{AB}$. Whereas, the perceptual and adversarial losses are only enforced on the $\hat{Y}_t^{AB}$.

We optimize our network with respect to 4 components of our loss function: pixel-wise MSE loss L_{MSE}, adversarial loss L_G and perceptual loss L_P:

$$\begin{aligned} L_{total} &= L_{MSE}(Y_t^A, \hat{Y}_t^A) + L_{MSE}(Y_t, \hat{Y}_t^{AB}) \\ &\quad + \lambda_G L_G(Y_t, \hat{Y}_t^{AB}) + \lambda_P L_P(Y_t, \hat{Y}_t^{AB}) \end{aligned} \quad (6)$$

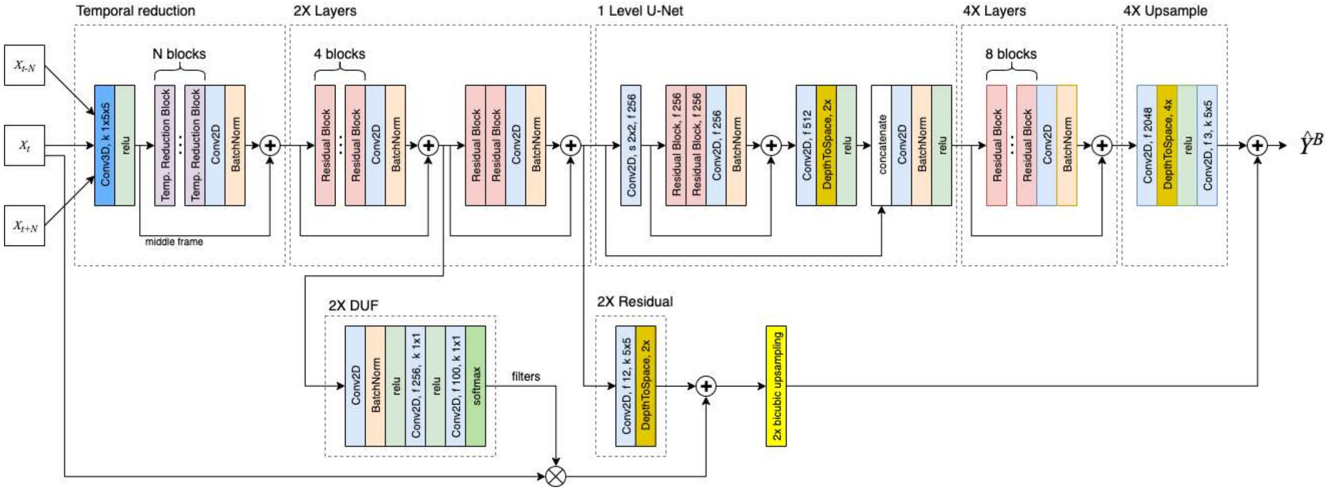


FIGURE 4. Detailed architecture of the SUPERVEGAN-4 generator. Unless specified, Conv2D blocks have 128 output channels, 3 × 3 kernel, and 1 × 1 stride. Conv3D block also has 128 output channels and stride 1 × 1 × 1. Temporal reduction block and residual block are depicted in Fig. 5.

where Y_t is the ground truth frame and Y_t^A is a 2x downsampled version of it. Given a ground truth image Y and an output image $\hat{Y}$, our MSE and perceptual losses are defined as:

$$\begin{aligned} L_{MSE}(Y,\hat{Y}) &= \textstyle\sum_{i\in Q}(Y(i)-\hat{Y}(i))^2 \\ L_P(Y,\hat{Y}) &= ||\Phi_{VGG}(Y)-\Phi_{VGG}(\hat{Y})||_2 \end{aligned} \tag{7}$$

where Q represents the image domain. Our perceptual loss L_P is in line with other works [9], [10], so Φ_{VGG} are conv54 features from pretrained VGG-19.

We use standard adversarial loss [28], where D is the output from discriminator and L_D is a loss for discriminator used in last two training phases:

$$\begin{aligned} L_G &= -\log(D(\hat{Y})), \\ L_D &= -\log(D(Y)) - \log(1-D(\hat{Y})). \end{aligned} \tag{8}$$

The combination of losses for perception and distortion along with the SUPERVEGAN architecture and progressive training results in a model that achieves video enhancement by joint super resolution and artifact removal.

IV. BITRATE EQUIVALENCY

As we discussed earlier, the goal of SUPERVEGAN is to take a video compressed at low bitrate b and produce an enhanced video that is now perceptually equivalent to a video compressed at a (hopefully) higher bitrate b'. Here we want to answer the question "How much bitrate improvement does a model provide?" and quantify the difference between b and b'. We consider two different approaches – one based on existing metrics and another one based on subjective evaluation by users. In addition, this allows us to evaluate which metrics matches user perception the most for this task.

A. ACCORDING TO METRICS

For simplicity, here we describe how to evaluate SUPERVEGAN-4 using PSNR metric w.r.t bitrate. This approach can be applied to any other model that enhances videos and using any other metric (in Sec. V-G we use PSNR and LPIPS).

We calculate PSNR on the output of SUPERVEGAN-4 and on high-resolution videos compressed by H264. For SUPERVEGAN-4, we run inference on a set of low-resolution videos compressed at different bitrates b in range from 150Kbps to 2000Kbps. Similarly, for H264, we compress the same videos but at high resolution at different bitrates in range from 150Kbps to 4000Kbps. Then, for both SUPERVEGAN-4 and H264 we build rate-distortion curves (see Fig. 7 as an example). Finally, for a given input bitrate b, we note the PSNR value for SUPERVEGAN-4 and find at which bitrate H264 gets the same value. This bitrate represents the equivalent bitrate b' according to PSNR.

B. ACCORDING TO PEOPLE

Video quality is ultimately a subjective assessment that current best metrics are still unable to capture appropriately. To better understand the gains achieved by a perceptual enhancer deep model (e.g., SUPERVEGAN-4), we use Amazon Mechanical Turk.

Our goal is to get users' answer to the question: What is the compression bit rate at which the original full resolution source appears equivalent to the model's output? Similarly as we have done when using metrics, we ran our model (e.g., SUPERVEGAN-4) on videos compressed at different bitrates b in range from 150Kbps to 2000Kbps. For each produced video, we generated a set of pairs with a corresponding high-resolution video compressed with H264 at bitrates ranging from 75Kbps to 4000Kbps (we call them reference bitrates). For each pair, we ask users a question: Which video do you prefer based on its quality? Finally, for each bitrate b we find the reference bitrate for which 50% of users prefer output of our model, and 50% of users prefer videos compressed at reference bitrate. This is our equivalent bitrate b'.

V. EXPERIMENTAL RESULTS

A. IMPLEMENTATION DETAILS

1) DATASETS

For training and validation, we have collected 42 high quality videos from where we randomly sampled 25000 frames; all the frames have been donwsampleld to 1280 × 720 resolution. For testing, we use 8 clips (1317 frames total) from the Tears of Steel (TOS) video [2] and the quality of experience database LIVE-NFLX-II dataset [29] which contains 15 clips (11146 frames total). To support cross comparison with other works, in Table 1 we provide the details of the exact scenes used from the (TOS) video [2].

TABLE 1. TOS scenes used in evaluation.

Clip	Start frame	End Frame
1-40years	1010	1094
2-battle	11355	11493
4-descent	3669	3804
5-face	9885	10050
6-holodome	5173	5337
7-rocket	214	456
8-room	4398	4547

2) BITRATES

To generate low-resolution compressed training input to our model, we used the bilinear method to downsample all the frames and then encoded them with GStreamer implementation of H264. We only set the *bitrate* parameter of the encoder and leave default values for all other parameters. For training the SUPERVEGAN-4, frames were 4 times downsampled and encoded at 250Kbps, and for the SUPERVEGAN-2 frames were 2 times downsampled and encoded at 750Kbps. For inference, we used variety of bitrates in range from 75Kbps to 2000Kbps. We want to emphasize that our framework does not exclusively depend on a particular encoder. Other encoders such as H265, VP8 and VP9, can be used instead for training and then for inference. Also note that for encoding we use a target bitrate instead of fixing for a specific encoder coefficient. This effectively allows the network to learn a range of encoding coefficients as the bitrate will affect different scenes differently. See Sec. V-F for an evaluation of bitrate training resilience.

3) TRAINING

We trained our model on 96 × 96 image patches that are randomly selected from low-resolution compressed images. Our final model takes as input a temporal window of 5 low-resolution frames to generate a single high resolution image corresponding to the center input frame. We train our network for 400 epochs in total: 100 epochs for the first phase, 100 epochs for the second phase and 200 epochs for the third phase.

4) ARCHITECTURE Of SUPERVEGANS

Detailed architecture of generator for SUPERVEGAN-4 is presented in Fig. 4 and Fig. 5. Architecture of SUPERVEGAN-2 is almost identical to the architecture of SUPERVEGAN-4. Since in SUPERVEGAN-2 we only perform 2x upsampling w.r.t. the input, our Stage A does not perform any upsampling and its purpose is purely to remove artifacts from the input image. Therefore, we only have one filter for every pixel. Also, for performance considerations and because our input is now twice as large in width and height, we use an extra 3D convolution with stride 2 in the spatial dimension in the very beginning before the temporal reduction block. This way, most of the computation will be performed at half the resolution of the input and therefore, our SUPERVEGAN-2 runs at approximately the same speed as SUPERVEGAN-4. We use the same architecture of discriminator for training all of our models; refer to Fig. 6 for details.

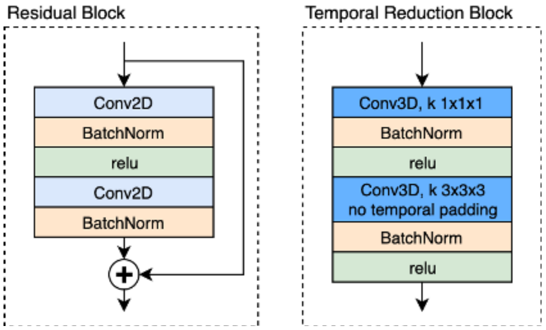


FIGURE 5. Architecture of temporal reduction block and residual block used in the SUPERVEGAN-4. Conv2D blocks use 3 × 3 kernel with 1 × 1 stride, Conv3D blocks use 1 × 1 × 1 stride.

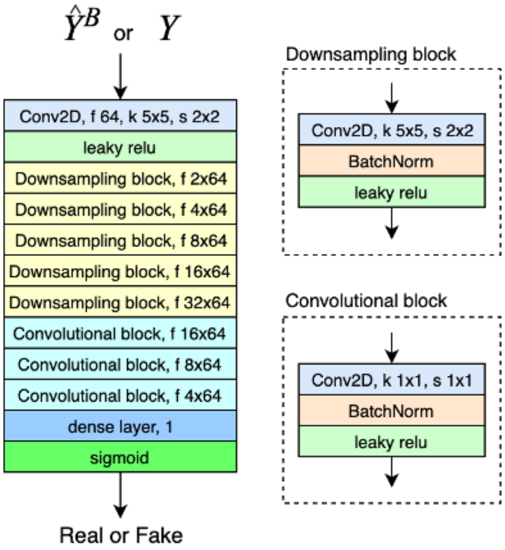

$\hat{Y}^B$ or Y
Conv2D, f 64, k 5x5, s 2x2
leaky relu
Downsampling block, f 2x64
Downsampling block, f 4x64
Downsampling block, f 8x64
Downsampling block, f 16x64
Downsampling block, f 32x64
Convolutional block, f 16x64
Convolutional block, f 8x64
Convolutional block, f 4x64
dense layer, 1
sigmoid
Real or Fake
Downsampling block
Conv2D, k 5x5, s 2x2
BatchNorm
leaky relu
Convolutional block
Conv2D, k 1x1, s 1x1
BatchNorm
leaky relu

FIGURE 6. Detailed architecture of the discriminator used for adversarial training.

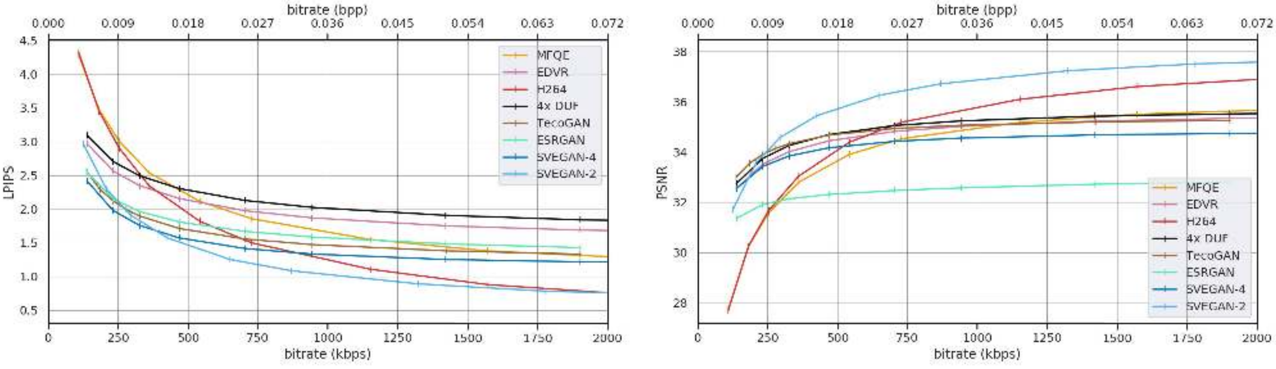
5) METRICS

We adopt metrics used in other works: PSNR and LPIPS. PSNR is the most common metric used for assessing image quality. But it does not fully capture the perceptual qual-

TABLE 2. Comparison of SUPERVEGAN-4 and other methods on LIVE-NFLX-II dataset [29] compressed at 250Kbps. Per video breakdown of LPIPS ×10 (lower is better) and PSNR. Best model is in bold, second best is underlined.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ave.
LPIPS																
H264	0.97	1.99	2.03	3.30	3.21	5.33	4.66	5.30	2.13	2.50	4.54	5.67	3.29	4.34	2.46	3.45
4x Bicubic	0.89	2.83	3.53	3.60	3.02	4.61	4.58	6.15	3.20	4.08	3.95	5.81	4.43	4.80	4.33	3.99
MFQE2.0 [21]	0.76	1.94	2.11	3.24	3.07	5.01	4.57	5.61	2.35	2.74	4.48	5.89	3.40	4.33	2.72	3.48
EDVR [20]	0.64	1.39	1.76	2.74	1.95	3.22	2.87	4.64	1.64	2.29	2.98	4.21	2.45	3.24	2.47	2.57
4x DUF-16L [3]	0.67	1.43	1.78	2.97	2.06	3.36	2.99	4.98	1.77	2.44	3.12	4.43	2.58	3.36	2.63	2.70
ESRGAN [31]	0.44	1.15	1.62	2.38	1.73	3.00	$\underline{2.32}$	**3.79**	1.37	1.86	2.43	$\underline{3.30}$	2.18	2.71	2.01	2.15
TecoGAN [9]	$\underline{0.37}$	1.09	1.48	2.33	1.74	3.02	2.41	4.02	1.19	1.81	$\underline{2.32}$	3.55	$\underline{2.06}$	2.68	1.78	2.12
$G_{\text{NO\_PROG}}$	0.40	1.07	1.46	2.36	1.80	2.94	2.37	4.20	1.22	1.83	2.38	3.65	2.14	2.69	1.77	2.15
$G_{\text{NO\_MSE\_A}}$	0.38	1.05	1.43	2.26	1.77	2.98	2.43	4.14	**1.12**	$\underline{1.70}$	2.58	3.46	2.11	2.73	1.80	2.13
SVEGAN-4	**0.36**	**1.01**	**1.28**	**1.73**	**1.63**	**2.82**	**2.27**	$\underline{3.88}$	1.14	**1.67**	**2.29**	**3.29**	**2.01**	**2.59**	**1.72**	**1.98**
Fast SVEGAN	0.41	$\underline{1.03}$	$\underline{1.40}$	$\underline{2.20}$	$\underline{1.69}$	$\underline{2.92}$	2.35	4.07	$\underline{1.13}$	1.74	2.45	3.40	2.21	$\underline{2.65}$	$\underline{1.76}$	$\underline{2.09}$
PSNR																
H264	35.97	33.09	33.91	37.75	28.24	23.69	23.67	26.97	36.14	33.49	29.41	21.38	30.90	26.62	32.54	30.25
4x Bicubic	35.09	31.86	30.11	37.15	31.58	28.48	27.48	29.59	33.62	31.00	33.32	24.71	29.81	29.70	31.07	30.97
MFQE2.0 [21]	35.90	32.79	33.84	37.59	28.38	23.81	23.71	27.06	35.57	33.68	29.54	21.47	30.92	26.77	32.67	30.25
EDVR [20]	37.44	34.96	$\underline{35.44}$	38.88	33.64	**30.36**	**29.93**	30.80	37.11	**35.19**	35.04	**25.74**	$\underline{32.88}$	$\underline{31.35}$	$\underline{34.01}$	33.52
4x DUF-16L [3]	38.03	$\underline{35.20}$	**35.49**	40.78	**33.87**	30.16	$\underline{29.84}$	$\underline{30.83}$	$\underline{37.67}$	$\underline{35.02}$	35.33	$\underline{25.61}$	**32.98**	**31.40**	**34.07**	$\underline{33.75}$
ESRGAN [31]	36.01	33.82	33.25	37.81	33.09	29.32	28.76	29.23	35.26	32.45	33.67	24.04	30.89	29.87	31.24	31.91
TecoGAN [9]	**39.54**	**36.11**	34.44	**42.49**	$\underline{33.84}$	$\underline{30.17}$	29.50	**30.85**	**38.34**	34.59	**36.00**	25.19	32.68	31.27	33.47	**33.90**
$G_{\text{NO\_PROG}}$	$\underline{38.12}$	34.81	34.19	$\underline{41.23}$	32.81	30.09	29.63	30.67	36.89	34.25	$\underline{35.40}$	25.34	32.40	31.19	33.28	33.35
$G_{\text{NO\_MSE\_A}}$	37.44	34.74	33.67	40.95	33.39	30.05	29.41	30.46	37.20	34.40	35.02	25.06	32.47	30.96	33.12	33.22
SVEGAN-4	37.86	34.92	34.85	40.44	33.49	30.16	29.63	30.59	37.15	34.67	35.21	25.26	32.59	31.18	33.42	33.43
Fast SVEGAN	37.68	34.83	34.03	39.85	33.19	29.99	29.48	30.46	36.75	34.02	35.10	25.15	31.91	30.93	32.94	33.09

Clips: 1-AirShow, 2-AsianFusion, 3-Chimera1102347, 4-Chimera1102353, 5-CosmosLaundromat, 6-ElFuenteDance, 7-ElFuenteMask, 8-GTA, 9-MeridianConversation, 10-MeridianDriving, 11-Skateboarding, 12-Soccer, 13-Sparks, 14-TearsOfSteelRobot, 15-TearsOfSteelStatic



LPIPS ×10

PSNR

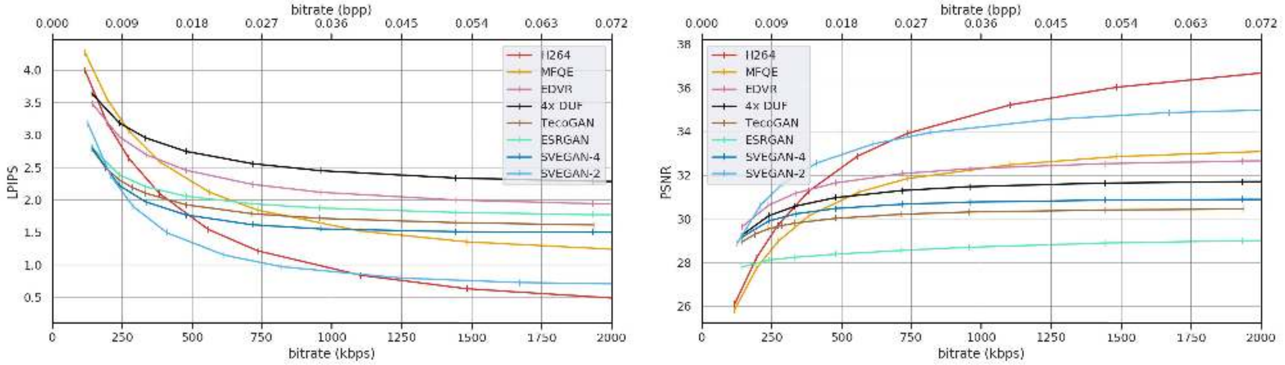
FIGURE 7. Rate-distortion curves with LIVE-NFLX-II dataset [29] at multiple bitrates. Images are best seen in digital version.

ity of the image. We use LPIPS [30], which is better correlated with how humans perceive images. We conduced user studies to confirm the performance of our models and to validate our metric choice of mainly targeting LPIPS (see Sec. V-F).

B. COMPARISON WITH STATE OF THE ART

In this section we compare our SUPERVEGAN-4 and SUPERVEGAN-2 models against DUF [3], EDVR [20], MFQE2 [21], ESRGAN [31], TecoGAN [9] and H264. From [3] we chose DUF-16L since it most closely matches our generators size in terms of running time. Since DUF, EDVR, TecoGAN and ESRGAN target super resolution only, we have retrained them with our data to achieve simultaneous super resolution and artifact removal. 4x models (DUF-16 4x, TecoGAN, EDVR, and ESRGAN) were trained with data compressed at 250Kbps, while DUF-16 2x was trained with data compressed at 750Kbps. DUF and EDVR models are optimized for distortion, while TecoGAN and ESRGAN optimize for both distortion and perception and are more comparable to our model. We target LPIPS but for easier cross-comparison with other works we also provide results for PSNR.

First, we evaluate the models on videos compressed at 250Kbps and 750Kbps. For 250Kbps we use 4x upsampling and for 750Kbps we use 2x upsampling. For both LIVE-



LPIPS ×10

PSNR

FIGURE 8. Rate-distortion curves with TOS dataset [2] at multiple bitrates. Images are best seen in digital version.

NFLX-II and TOS datasets SUPERVEGAN outperforms other approaches on LPIPS, both for 4x upsampling with input compressed at 250Kbps (Table 2 and Table 3) and 2x upsampling (Table 4 and Table 5). We provide qualitative examples in Fig. 14. In Fig. 15 we provide context images for the crops used in Fig. 14.

Second, we evaluate our model on videos compressed at different bitrates (Fig. 7 and Fig. 8). We can see that our SUPERVEGAN-4 outperforms all other methods on LPIPS in low bitrates. At the same time, our SUPERVEGAN-2 achieves the best performance in higher bitrates (greater than approx. 0.01 bpp) for LPIPS and PSNR.

C. PERCEPTION-DISTORTION PLANE

We have also evaluated SUPERVEGAN-4 using RMSE, which measures distortion, and Perceptual Index (PI), which measures perceptual quality without a reference. Perceptual Index is defined as a combination of Ma's score [32] and NIQE [33]:

$$\text{Perceptual Index}(\hat{Y}) = \frac{1}{2}(10 - \text{MA}(\hat{Y})) + \text{NIQE}(\hat{Y}). \quad (9)$$

A lower PI represents a better perceptual quality, while a lower RMSE represents a smaller distortion. Results of evaluation are presented in Fig. 9. When comparing SUPERVEGAN-4 to other approaches (Fig. 9a), we can observe from the plot that EDVR and 4x-DUF perform the best with regards to RMSE, while ESRGAN achieves the best PI. SUPERVEGAN-4 and TecoGAN strike a balance between RMSE and PI: both significantly improve PI over EDVR/4x-DUF while remaining close in RMSE; similarly, SUPERVEGAN-4 and TecoGAN achieve much better RMSE than ESRGAN. SUPERVEGAN-4 and TecoGAN have comparable results: SUPERVEGAN-4 has marginally better RMSE while TecoGAN has marginally better PI. This demonstrates that SUPERVEGAN-4 strikes a balance and belongs to the Pareto front within this graph of the perception-distortion tradeoff. Furthermore, visual results comparing SUPERVEGAN-4 vs TecoGAN and ESRGAN are provided in the video supplementary materials. Meanwhile, SUPERVEGAN-2 outperforms other methods in RMSE and is comparable to TecoGAN and ESRGAN in PI (Fig. 9a).

TABLE 3. Comparison of SUPERVEGAN-4 and other methods on TOS dataset [2] compressed at 250Kbps. Per-video breakdown of LPIPS ×10 and PSNR metrics.

	1	2	3	4	5	6	7	8	Ave.
LPIPS									
H264	3.05	4.12	3.70	2.37	2.26	4.30	2.70	2.63	3.14
4x Bicubic	5.31	5.63	5.62	4.99	3.93	5.55	3.75	5.62	5.05
MFQE2.0 [21]	3.62	4.37	4.19	2.94	2.50	4.64	2.83	2.99	3.51
EDVR [20]	3.26	3.52	3.42	2.58	2.14	3.95	1.99	3.22	3.01
4x DUF16L [3]	3.53	3.61	3.68	3.16	2.23	3.86	2.25	3.46	3.22
ESRGAN [31]	2.30	2.66	2.85	2.24	1.76	3.31	1.60	2.40	2.39
TecoGAN [9]	**2.15**	**2.62**	2.81	1.98	1.80	3.18	1.70	2.27	2.31
$G_{\text{NO\_PROG}}$	2.37	2.79	2.66	2.03	1.77	3.19	1.66	**2.11**	2.33
$G_{\text{NO\_MSE\_A}}$	2.32	2.86	2.66	2.03	1.70	3.23	1.67	2.29	2.34
SVEGAN-4	2.28	2.83	**2.54**	**1.90**	**1.62**	3.19	**1.55**	2.25	**2.27**
Fast SVEGAN	2.34	2.83	2.63	2.06	1.73	**3.07**	1.64	2.24	2.32
PSNR									
H264	30.51	22.96	27.03	32.71	30.60	24.92	29.39	29.20	28.42
4x Bicubic	27.85	24.73	27.60	29.98	29.19	27.12	28.58	26.65	27.71
MFQE2.0 [21]	30.02	22.79	26.71	31.61	29.39	24.86	28.87	28.60	27.86
EDVR [20]	**31.05**	**26.38**	**29.68**	**32.77**	**33.01**	**29.30**	**33.31**	**29.23**	**30.59**
4x DUF16L [3]	30.47	26.01	29.19	32.12	32.73	28.88	32.64	28.61	30.08
ESRGAN [31]	27.61	24.69	26.15	29.07	31.28	28.05	32.19	25.81	28.11
TecoGAN [9]	29.54	25.84	27.88	31.00	32.52	28.81	33.04	27.69	29.54
$G_{\text{NO\_PROG}}$	29.93	25.80	28.77	31.48	32.27	28.87	32.89	27.83	29.73
$G_{\text{NO\_MSE\_A}}$	29.58	25.61	28.39	31.31	32.28	28.92	33.22	27.71	29.63
SVEGAN-4	29.97	25.87	28.72	31.56	32.57	28.98	33.26	28.04	29.87
Fast SVEGAN	29.65	25.63	28.36	31.17	32.03	28.70	32.91	27.39	29.48

Clips: 1-40years, 2-battle, 3-bridge, 4-descent, 5-face, 6-holodome, 7-rocket, 8-room

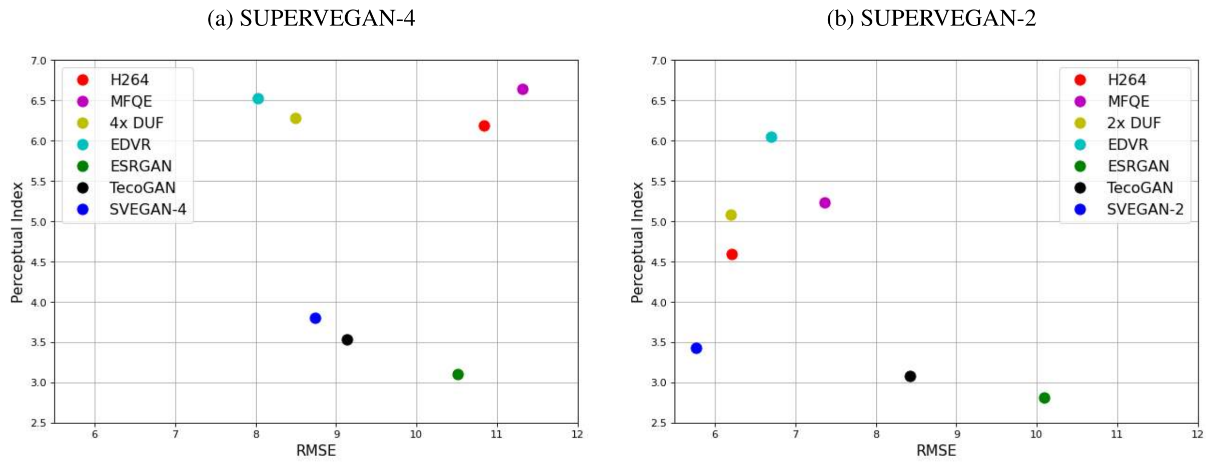
D. FAST SUPERVEGAN

We used an evolutionary strategies search (1,1)-ES [34], parallelized and within a Pareto front pool to explore hyperparameters of our network. The Pareto front used LPIPS and PSNR. After several hundred proposed architectures, we computed Pearson correlation coefficients of every parameter with PSNR, LPIPS and execution time per frame for those architectures. This guided the design to choose hyper parameters like number of filters in each layer "F"

TABLE 4. Comparison of SUPERVEGAN-2 and other methods on LIVE-NFLX-II dataset [29] compressed at 750Kbps. Per-video breakdown of LPIPS ×10 and PSNR metrics. The 2x methods, SUPERVEGAN-2 and 2x DUF-16L, were also trained with input encoded at 750Kbps.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ave.
LPIPS																
H264	0.35	0.70	0.93	2.82	1.75	2.62	1.78	3.42	1.07	1.38	2.06	3.17	1.63	2.47	1.12	1.82
2x Bicubic	0.55	1.64	1.95	2.75	1.78	2.69	2.61	4.48	2.00	2.44	2.59	3.83	2.75	3.18	2.70	2.53
MFQE2.0 [21]	0.43	0.91	1.26	2.90	1.78	2.52	1.95	4.00	1.65	2.00	2.53	3.50	1.95	2.63	1.62	2.11
EDVR [20]	0.58	1.12	1.39	2.59	1.38	2.05	1.93	3.56	1.53	1.86	2.29	3.15	1.93	2.30	1.98	1.98
2x DUF-16L [3]	0.37	0.95	0.98	2.15	1.16	2.19	1.75	3.45	0.98	1.42	1.76	2.70	1.75	2.25	1.50	1.69
ESRGAN [31]	0.36	0.90	1.29	1.96	1.21	1.91	1.64	2.78	1.30	1.65	2.38	2.23	1.82	1.90	1.71	1.67
TecoGAN [9]	0.29	0.84	1.17	**1.65**	1.20	1.90	1.68	**2.71**	1.12	1.49	1.95	2.44	1.65	1.81	1.44	1.56
SVEGAN-2	**0.24**	**0.48**	**0.71**	1.71	**0.93**	**1.70**	**1.19**	2.80	**0.64**	**0.99**	**1.51**	**2.02**	**1.34**	**1.69**	**0.85**	**1.25**
PSNR																
H264	38.71	36.50	37.26	38.83	32.83	30.58	31.44	31.43	37.65	36.67	35.33	25.93	34.69	32.06	36.01	34.39
2x Bicubic	36.79	34.31	34.42	38.41	34.12	32.22	31.08	31.55	36.10	34.32	35.36	27.28	32.63	32.45	33.78	33.65
MFQE2.0 [21]	37.61	35.47	36.33	37.97	32.76	30.82	31.41	31.28	36.41	36.00	34.80	25.99	34.13	32.15	35.62	33.92
EDVR [20]	37.80	35.49	36.11	39.20	35.18	**34.11**	32.16	32.22	37.24	36.13	36.39	27.78	33.78	33.82	34.91	34.82
2x DUF-16L [3]	36.98	35.74	37.47	40.27	34.93	33.00	32.73	32.38	38.13	36.69	36.67	28.09	34.60	33.67	35.87	35.15
ESRGAN [31]	36.24	34.09	33.69	37.90	34.18	32.03	29.66	28.95	35.30	32.77	33.30	24.85	31.20	31.38	31.51	32.47
TecoGAN [9]	40.08	36.65	34.88	**43.10**	35.12	33.72	31.00	31.83	38.53	34.81	37.03	26.60	33.04	33.72	33.97	34.94
SVEGAN-2	**40.55**	**37.61**	**38.77**	40.72	**36.08**	33.87	**33.50**	**32.96**	**39.73**	**37.54**	**37.61**	**28.32**	**35.47**	**34.39**	**36.75**	**36.26**

Clips: 1-AirShow, 2-AsianFusion, 3-Chimera1102347, 4-Chimera1102353, 5-CosmosLaundromat, 6-ElFuenteDance, 7-ElFuenteMask, 8-GTA, 9-MeridianConversation, 10-MeridianDriving, 11-Skateboarding, 12-Soccer, 13-Sparks, 14-TearsOfSteelRobot, 15-TearsOfSteelStatic


(a) SUPERVEGAN-4

(b) SUPERVEGAN-2

FIGURE 9. Perception-distortion plane for TOS dataset [2] for the SUPERVEGAN-4 (9a) and the SUPERVEGAN-2 (9b).

(see Fig. 10) with high positive correlation for compute time and low correlation with the metrics. This insight allowed us to develop a fast SUPERVEGAN that runs in 30.4ms per frame at 1280 × 720px on an NVIDIA V100 GPU with minimal degradation in quality. For comparison, the not yet optimized for speed SUPERVEGAN-4 runs in 138ms per frame. We have computed processing time for all the methods and summarized results in Table 6. Performance is averaged over 100 frames on a single NVIDIA V100 GPU.

E. USING H265 TO ENCODE INPUT

Our model is universal with respect to codec. On the main paper we have used H264 as the encoder to evaluate with. This is based on the overwhelming availability of hardware accelerated implementations as well as software infrastructure dedicated to H264. As an additional test we here consider the case of using H265, which is a more recent encoder, as the input for SUPERVEGAN. In Fig. 11 results of processing input that has been encoded with H265 are presented. As we can see SUPERVEGAN improves over H265 especially at lower bitrates for LPIPS and interestingly SUPERVEGAN-2 with H265 input is much better for PSNR too.

F. SUPERVEGAN ANALYSIS

1) ABLATION FOR SUPERVEGAN-4

Our baseline model is $G_{\text{NO\_PROG}}$ which is a SUPERVEGAN trained in a non-progressive manner with all the losses according to Eq. 6. We also analyze the impact of keeping an MSE loss for Stage A in the second and third phases of training; thus we remove this loss and get another variant of our model $G_{\text{NO\_MSE\_A}}$ Our full model is SUPERVEGAN

TABLE 5. Comparison of SUPERVEGAN-2 and other methods on TOS dataset [2] compressed at 750Kbps. Per-video breakdown of LPIPS ×10 and PSNR metrics. The 2x methods, SUPERVEGAN-2 and 2x DUF-16L, were also trained with input encoded at 750Kbps.

	1	2	3	4	5	6	7	8	Ave.
LPIPS									
H264	1.38	2.16	1.83	0.89	0.99	2.60	1.15	1.06	1.51
2x Bicubic	3.67	3.55	3.59	2.92	2.25	3.88	2.24	3.68	3.22
MFQE2.0 [21]	2.26	2.59	2.43	1.78	1.44	3.15	1.59	1.57	2.10
EDVR [20]	2.66	2.27	2.59	2.04	1.49	2.89	1.41	2.56	2.24
2x DUF16L [3]	2.18	2.12	2.16	1.50	1.14	2.43	1.24	1.99	1.84
ESRGAN [31]	2.03	2.13	2.40	1.92	1.43	2.37	1.14	2.09	1.94
TecoGAN [9]	1.79	1.85	2.30	1.60	1.41	2.25	1.19	1.91	1.79
SVEGAN-2	**1.01**	**1.50**	**1.29**	**0.79**	**0.80**	**2.03**	**0.75**	**0.92**	**1.14**
PSNR									
H264	**34.60**	27.71	31.77	**36.98**	34.93	29.72	34.74	**33.85**	33.04
2x Bicubic	30.62	27.77	30.25	32.92	32.71	29.75	31.95	29.49	30.68
MFQE2.0 [21]	32.62	27.45	30.59	33.80	32.49	29.06	32.66	31.74	31.30
EDVR [20]	31.60	28.99	31.21	33.60	34.60	31.32	34.99	30.24	32.07
2x DUF16L [3]	33.29	28.90	31.72	35.29	34.67	30.93	34.81	31.92	32.69
ESRGAN [31]	28.06	25.27	26.54	29.44	31.79	28.29	33.00	26.04	28.55
TecoGAN [9]	29.78	27.03	28.19	31.19	33.49	29.85	34.08	28.05	30.21
SVEGAN-2	33.49	**29.29**	**32.05**	35.60	**36.15**	**31.50**	**36.81**	32.89	**33.47**

Clips: 1-40years, 2-battle, 3-bridge, 4-descent, 5-face, 6-holodome, 7-rocket, 8-room

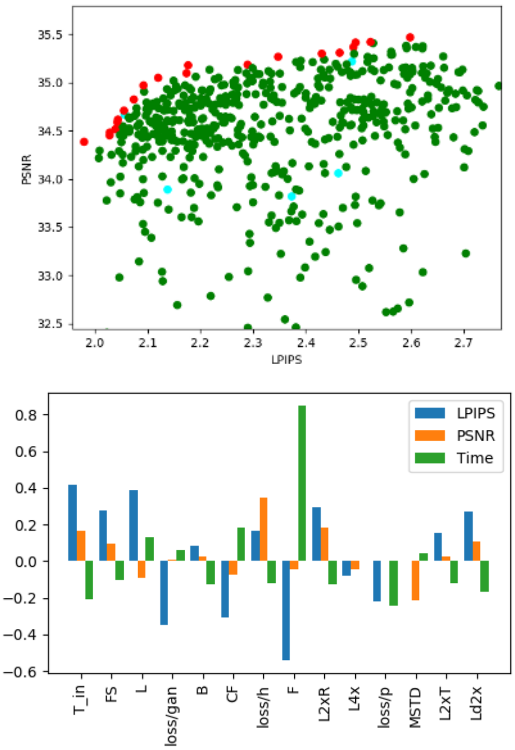


FIGURE 10. Top: Pareto front of SUPERVEGAN variations. Cyan=initial seeds, Red=pareto front, Green=others. Bottom: correlation coefficients of every hyper-parameter searched with PSNR, LPIPS and time respectively.

which includes both progressive training and keeps MSE loss for Stage A throughout the whole training. From Tables 2 and 3 we can observe that for both LIVE-NFLX-II and TOS datasets SUPERVEGAN outperforms both, $G_{\mathrm{NO\_PROG}}$ and $G_{\mathrm{NO\_MSE\_A}}$.

TABLE 6. Runtime evaluation.

Method	Processing time, ms/frame
MFQE2.0 [21] *	220 / 119
EDVR [20]	103
4x DUF-16L [3]	111
2x DUF-16L [3]	417
ESRGAN [31]	254
TecoGAN [9]	39
SVEGAN-4	138
SVEGAN-2	180
Fast SVEGAN	30

* For MFQE2.0 we report two numbers, the first one is the processing time of PQF frames, and the second one is the processing time of non-PQF frames.

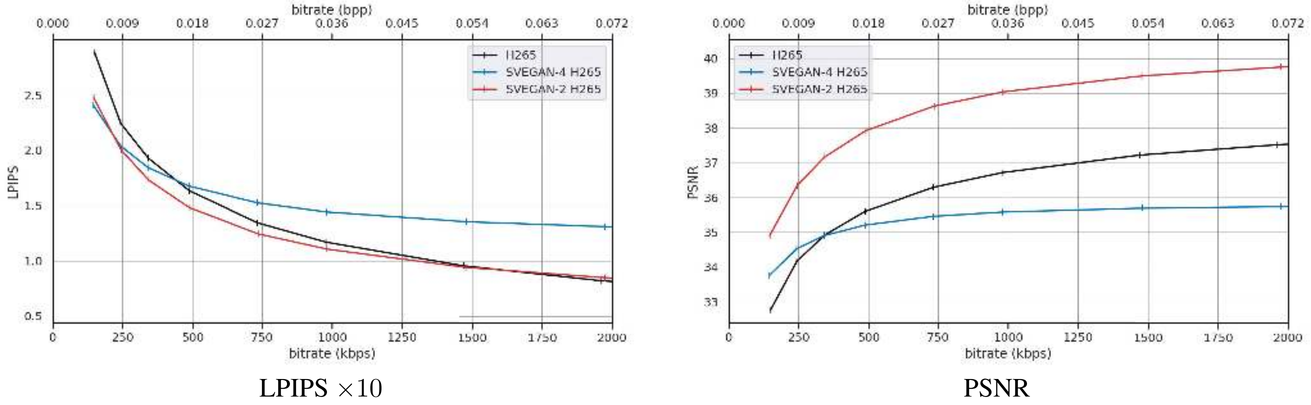
2) SUPERVEGAN DISSECTION

Here we look at validating the effect of the architectural components in our models. In order to do so, we compare encoded input to our model with the output after Stage A and the final output of SUPERVEGAN, which is equivalent to the output of Stage B. Note, that in order to conduct quantitative and qualitative evaluation and match ground-truth 1280 × 720px resolution, inputs and outputs of Stage A were upsampled with bicubic to match the ground-truth resolution.

In Fig. 12 we present a qualitative comparison of 250Kbps encoded input to our model, output after Stage A and output of the full model for SUPERVEGAN-4. As we can see, the network outputs images that qualitatively exhibit significantly less compression artifacts after the Stage A (Fig. 12b) and sharpness and higher details after the Stage B (Fig. 12c). This can also be verified by looking at the metrics in Table 7 where we see improvement (lower LPIPS) as the frame is processed through the network. However, PSNR value drops at Stage B in the case of SUPERVEGAN-4. This can be explained by the fact that PSNR favors blurry images over the images with slightly misplaced fine details, even though the latter ones look more visually pleasing. However, in the case of SUPERVEGAN-2, there is an improvement in PSNR as well as in LPIPS at both stages.

3) TRAINING FOR DIFFERENT BITRATES

We have conducted an experiment to see what is the effect of training bitrate on inference performance. For this purpose, we have trained SUPERVEGAN-2 and SUPERVEGAN-4 models on data encoded at 250Kbps, 500Kbps, 750Kbps and 1000Kbps each. Note that we do not set a specific QP value in our encoder settings like in MFQE 2.0 [21]. Instead, we specify a target bitrate. This ensures a bigger variety of compression artifacts in the training data which depends on the amount of high-frequency details and amount of motion in a particular scene. In Fig. 13 we plot the mean, min and max LPIPS value for our networks trained at the aforementioned bitrates. It can be seen that the training bitrate does have a small effect on the network performance but in high bitrates, the advantage of the SUPERVEGAN-2 over the



LPIPS ×10

PSNR

FIGURE 11. Rate-distortion curves for videos encoded with H265 on the LIVE-NFLX-II dataset. Images are best seen in digital version.

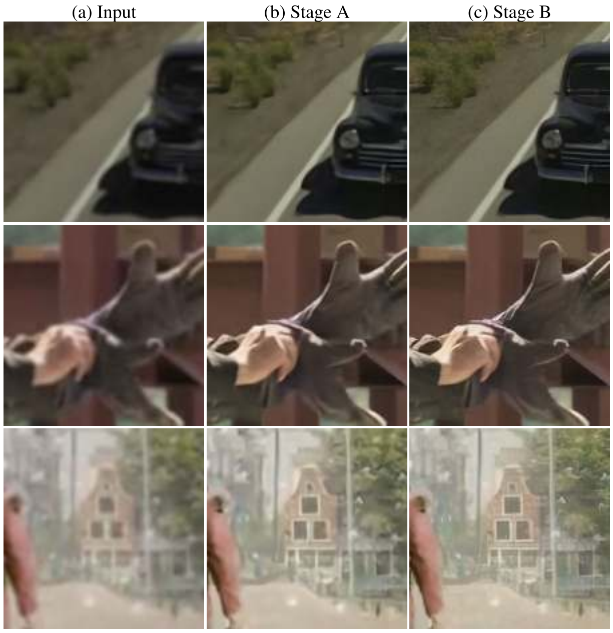


FIGURE 12. Relevance of components in SUPERVEGAN-4. After the stage A (Fig. 12b) block artifacts are removed and image is 2x upsampled. After the stage B (Fig. 12c) the image is upsampled to 4x; sharpness and fine details are added.

TABLE 7. Evaluation of different stages of SUPERVEGAN-4 and SUPERVEGAN-2. In this setting, we calculate the metrics on the input, the output of our models after stage A, and the final output (equivalent to the output of stage B).

	SUPERVEGAN-4			SUPERVEGAN-2		
	Input	Stage A	Stage B	Input	Stage A	Stage B
LIVE-NFLX-II						
LPIPS ×10	3.99	2.72	1.98	2.54	1.94	1.25
PSNR	30.97	33.72	33.43	33.64	34.55	36.26
TOS						
LPIPS ×10	5.05	3.38	2.27	3.23	2.15	1.14
PSNR	27.71	30.15	29.87	30.67	31.42	33.47

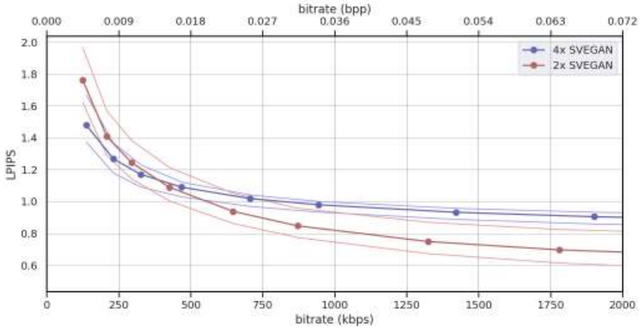


FIGURE 13. Effect of training SUPERVEGAN-2 and SUPERVEGAN-4 at different bitrates. For each model, we plot the mean, min and max of several versions of the model that were trained on data encoded at 250, 500, 750 and 1000 Kbps.

SUPERVEGAN-4 network comes mostly from the fact that it is 2x and not because of the bitrate it was trained on, since at high bitrates, even the worst performing SUPERVEGAN-2 model is better than the best performing SUPERVEGAN-4 model.

G. RESULTS OF BITRATE EQUIVALENCY STUDY

Here we describe details of the study and discuss obtained results.

1) ACCORDING TO PEOPLE

We have conducted studies on 2 datasets (TOS and LIVE-NFLX-II) and 2 models (SUPERVEGAN-4 and SUPERVEGAN-2). In total, we showed 5221 decision pairs to MTurkers and collected 44,258 responses. Given that we are interested in what users' video quality experience is, we did not ask people to focus on details or small crops but on the whole scene as they would normally experience. We positioned two videos next to each other, but placement of variants was done randomly to minimize positional bias. Only MTurkers with screen resolution $\geq 1280 \times 800$px were allowed. In addition, in order to match this resolution we have cropped videos spatially, so that users could watch them without downsampling.

Example of how we estimate bitrate equivalency is illustrated in Fig. 16. Distribution of MTurkers' votes is presented in Fig. 16a. MTurkers were asked to compare output from the

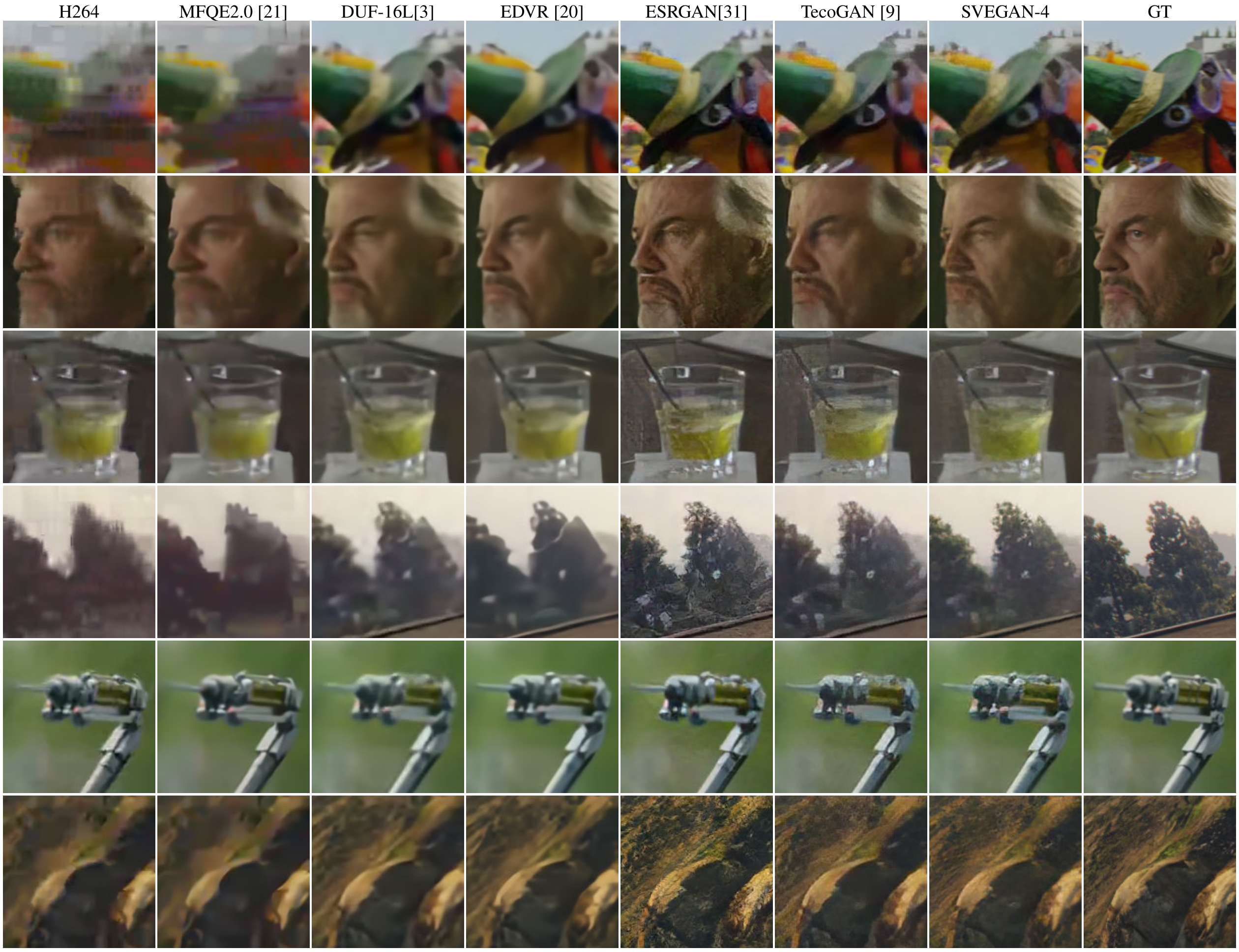
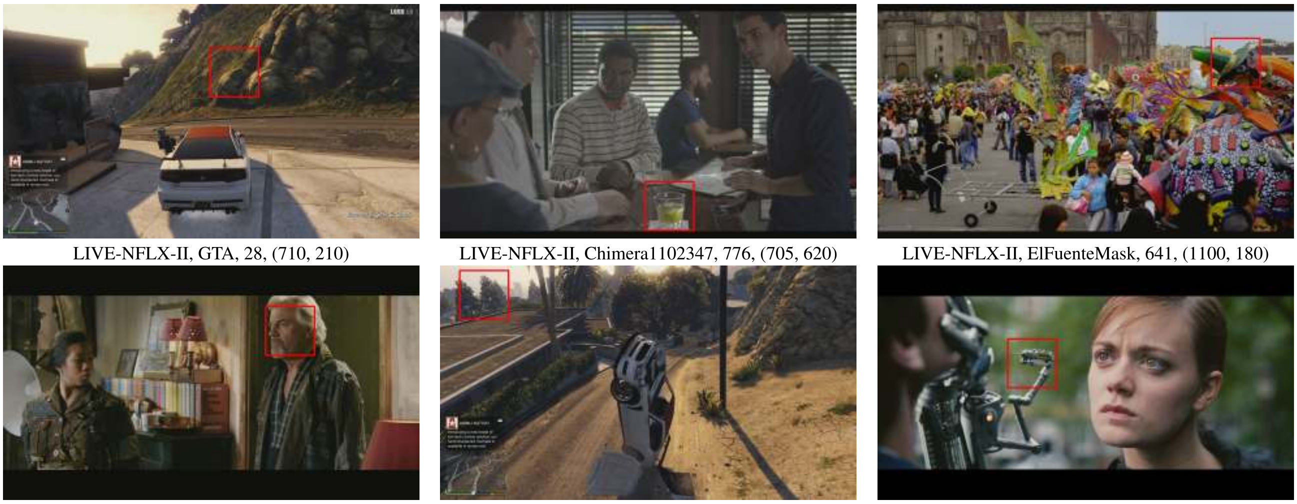


FIGURE 14. Qualitative comparison of 4x upsampling models, except MFQE2.0 and H264 which are non-upsampling methods. Input at 250Kbps and ground truth at full resolution and uncompressed. Images are taken from tears of steel (TOS) [2] and LIVE-NFLX-II [29]; see fig. 15 for context images.



LIVE-NFLX-II, GTA, 28, (710, 210)

LIVE-NFLX-II, Chimera1102347, 776, (705, 620)

LIVE-NFLX-II, ElFuenteMask, 641, (1100, 180)

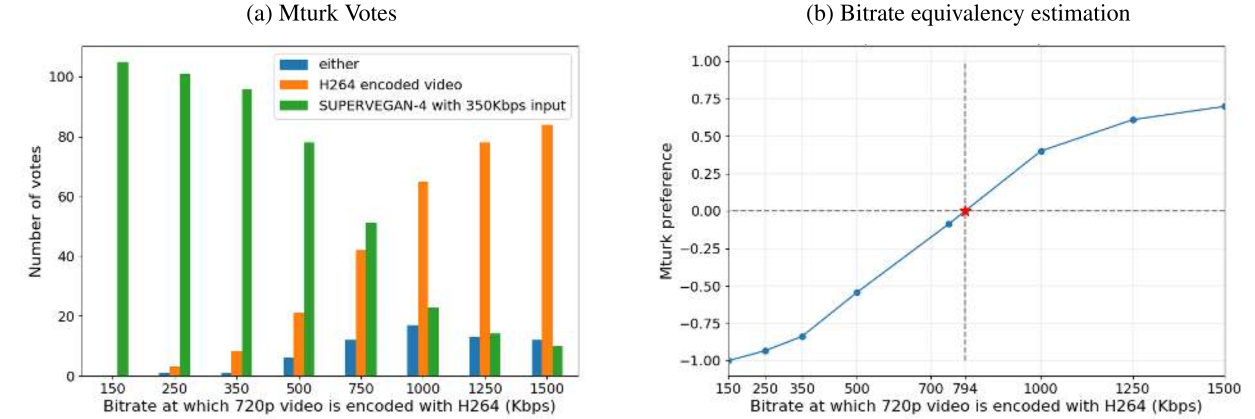
LIVE-NFLX-II, TearsOfSteelStatic, 73, (882, 200)

LIVE-NFLX-II, GTA, 321, (135, 90)

TOS, 9890, (475, 300)

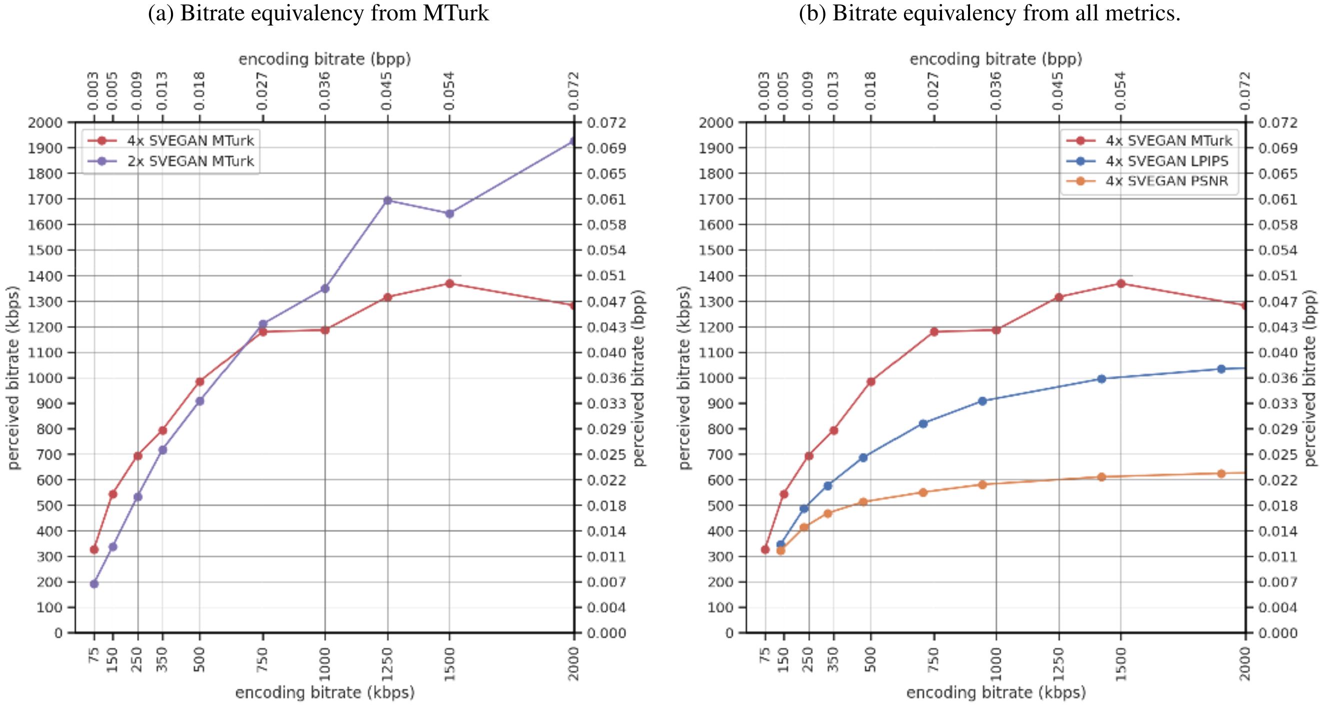
FIGURE 15. Context images from where crops are used in fig. 14. Format: dataset, video, frame number, (center x, center y). All crops are 150 × 150px.

SUPERVEGAN-4 with input video compressed at 350Kbps and high-resolution videos compressed with H264 at different bitrates. We can observe that at bitrates below 1000Kbps, MTurkers prefer output of the SUPERVEGAN-4, while at


(a) Mturk Votes

(b) Bitrate equivalency estimation

FIGURE 16. Estimating bitrate equivalency. Fig. 16a presents a distribution of MTurkers votes when shown output of SUPERVEGAN-4 with input compressed at 350Kbps and high-resolution videos compressed with H264 at different bitrates. In the fig. 16b we demonstrate how we convert these votes into equivalent bitrate.


(a) Bitrate equivalency from MTurk

(b) Bitrate equivalency from all metrics.

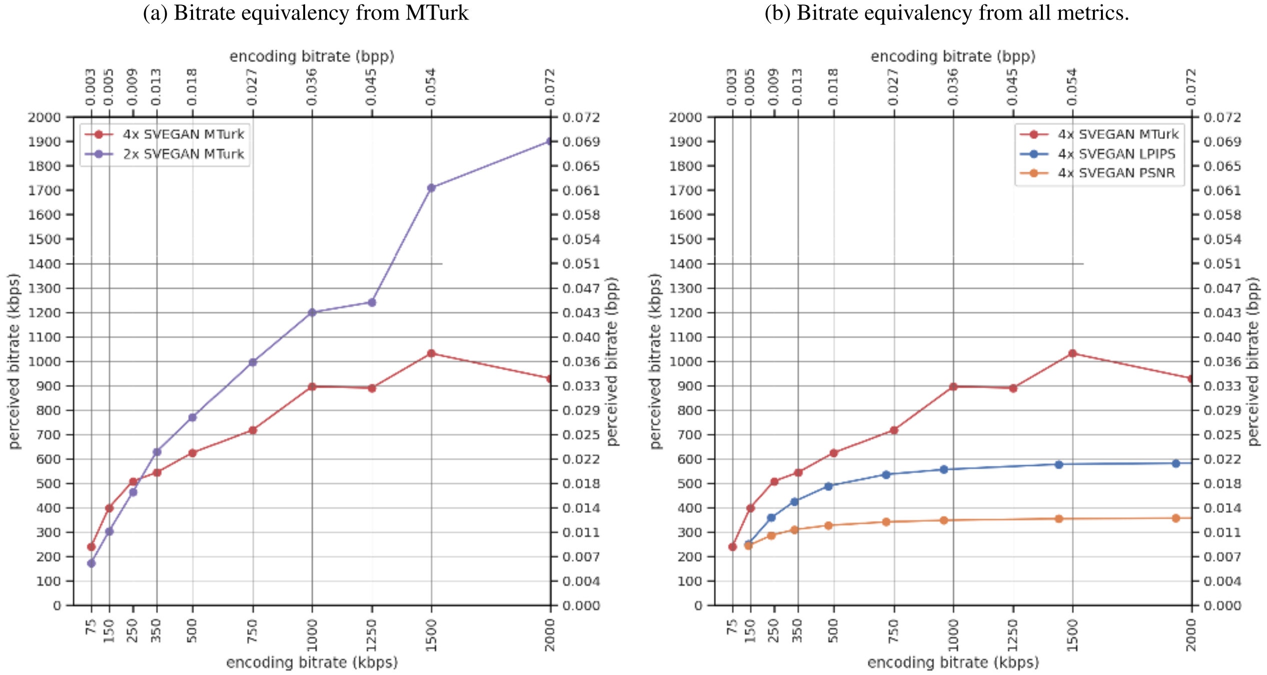
FIGURE 17. Bitrate equivalency studies for LIVE-NFLX-II dataset. For 720p and for low bitrates before about 1.5Mbps, SUPERVEGAN models show better perceived bitrate over the H264 baseline. See sec. V-G for details.

1000Kbps and above users prefer videos compressed with H264. We take this vote distribution and calculate the equivalent bitrate as depicted in Fig. 16b.

Results on LIVE-NFLX-II are presented in Fig. 17 and results on TOS are in Fig. 18. From Fig. 18a we can observe that on the LIVE-NFLX-II dataset SUPERVEGAN-4 at least doubles the perceptually equivalent bitrate for input encoded at lower bitrates, e.g. output of SUPERVEGAN-4 at 350Kbps input corresponds to 790Kbps H264 encoded video. Similarly, on the TOS dataset SUPERVEGAN-4 achieves 1.5x improvement at 350Kbps (see Fig. 18a). Meanwhile, SUPERVEGAN-2 performs the best in higher range of bitrates up to 1.5Mbps. At 1Mbps, SUPERVEGAN-2 achieves 1.34x and 1.2x improvement for LIVE-NFLX-II and TOS correspondingly. In the highest range of bitrates (above 1.5Mbps), and where SUPERVEGAN is not intended for, users prefer videos encoded with H264.

2) ACCORDING TO METRICS

From the data plots in Fig. 7 and Fig. 8 we have calculated equivalent bitrates for SUPERVEGAN-4 on LIVE-NFLX-II and TOS datasets for all the metrics (PSNR and LPIPS) as described in Sec. IV-A. Results are presented in Fig. 17b and Fig. 18b, where we also added results from the corresponding MTurk study. We can observe that: i) LPIPS is by a large margin closest to the MTurker's opinions which validates



(a) Bitrate equivalency from MTurk

(b) Bitrate equivalency from all metrics.

FIGURE 18. **Bitrate equivalency studies for TOS dataset. For 720p and for low bitrates before about 1.5Mbps, SUPERVEGAN models show better perceived bitrate over the H264 baseline. See sec. V-G for details.**

our choice for LPIPS as our main target metric, and ii) that existing metrics, including LPIPS, have a gap to cover to fully represent real user perception.

To better illustrate the proposed method, video results are provided as supplementary material.

VI. CONCLUSION

In this paper, we tackle the problems of video enhancement, video super resolution and artifact removal as a joint problem. Specifically aiming at live non-buffered streaming, we proposed a novel model family that incorporates the adversarial nature of GANs, the benefits of progressive training tuned for video enhancement and that leverages Dynamic Upsampling Filters. We proposed a bitrate equivalency process to better assess model output with real people and with existing metrics. The SUPERVEGAN architecture is shown to outperform related methods for a range of low and high bitrates. In particular the 4X upsampling model already outperforms state of the art methods on the LPIPS perceptual metric and the 2X upsampling model outperforms baselines also on the PSNR metric. SUPERVEGAN models can work with different encoding methods and the fast-SUPERVEGAN model runs at 32fps at 1280 × 720px on an NVIDIA V100 GPU with minimal degradation in quality. Overall, generative video enhancement methods that allow to recover detail under restricted bandwidth conditions benefit the increasing variety and deployment of video streaming devices.

ACKNOWLEDGMENT

(Silviu S. Andrei and Nataliya Shapovalova contributed equally to the work.)

REFERENCES

[1] Cisco. (2019). *Cisco Visual Networking Index: 2020 Global Forecast Highlights*. Accessed: Feb. 16, 2021. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2020_Forecast_Highlights.pdf

[2] (2013). *Tears of Steel*. Accessed: Feb. 16, 2021. [Online]. Available: https://mango.blender.com

[3] 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 *Proc. CVPR*, Jun. 2018, pp. 3224–3232.

[4] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *Proc. ICCV*, Dec. 2015, pp. 531–539.

[5] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. CVPR*, Jul. 2017, pp. 4778–4787.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[7] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. ICCV*, Oct. 2017, pp. 4472–4480.

[8] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. CVPR*, Jun. 2018, pp. 6626–6634.

[9] M. Chu, Y. Xie, J. Mayer, L. Leal-Taixé, and N. Thuerey, "Learning temporal coherence via self-supervision for GAN-based video generation," *ACM Trans. Graph.*, vol. 39, no. 4, Jul. 2020.

[10] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, Jul. 2017, pp. 4681–4690.

[11] A. Lucas, A. K. Katsaggelos, S. Lopez-Tapuia, and R. Molina, "Generative adversarial networks and perceptual losses for video super-resolution," in *Proc. ICIP*, Oct. 2018, pp. 51–55.

[12] E. Pérez-Pellitero, M. S. M. Sajjadi, M. Hirsch, and B. Schölkopf, "Photorealistic video super resolution," in *Proc. ECCV Workshops*, 2018, pp. 1–15.

[13] T. Wang, M. Liu, J. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Proc. NeurIPS*, 2018, pp. 1–14.

[14] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," in *Proc. ICLR*, 2019, pp. 1–18.

[15] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, “A deblocking filter with two separate modes in block-based video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 156–160, 1999.

[16] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[17] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, “Deep Kalman filtering network for video compression artifact reduction,” in *Proc. ECCV*, 2018, pp. 568–584.

[18] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, “MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Mar. 2021.

[19] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[20] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, “EDVR: Video restoration with enhanced deformable convolutional networks,” in *Proc. CVPR Workshops*, Jun. 2019, pp. 1–10.

[21] R. Yang, M. Xu, Z. Wang, and T. Li, “Multi-frame quality enhancement for compressed video,” in *Proc. CVPR*, Jun. 2018, pp. 1–10.

[22] C.-Y. Wu, N. Singhal, and P. Krähenbühl, “Video compression through image interpolation,” in *Proc. ECCV*, 2018, pp. 416–431.

[23] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learning image and video compression through spatial-temporal energy compaction,” in *Proc. CVPR*, Jun. 2019, pp. 10071–10080.

[24] Y. Xu, L. Gao, K. Tian, S. Zhou, and H. Sun, “Non-local ConvLSTM for video compression artifact reduction,” in *Proc. ICCV*, Oct. 2019, pp. 7043–7052.

[25] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “DVC: An end-to-end deep video compression framework,” in *Proc. CVPR*, Jun. 2019, pp. 11006–11015.

[26] Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in *Proc. CVPR*, Jun. 2018, pp. 6228–6237.

[27] Y. Blau and T. Michaeli, “Rethinking lossy compression: The rate-distortion-perception tradeoff,” in *Proc. ICML*, 2019, pp. 675–685.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. NeurIPS*, 2014, pp. 1–7.

[29] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, “Towards perceptually optimized end-to-end adaptive video streaming,” 2018, *arXiv:1808.03898*. [Online]. Available: http://arxiv.org/abs/1808.03898

[30] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proc. CVPR*, Jun. 2018, pp. 586–595.

[31] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, “ESRGAN: Enhanced super-resolution generative adversarial networks,” in *Proc. ECCV Workshops*, 2018, pp. 1–16.

[32] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, “Learning a no-reference quality metric for single-image super-resolution,” *Comput. Vis. Image Understand.*, vol. 158, pp. 1–16, May 2017.

[33] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a ‘completely blind’ image quality analyzer,” *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[34] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies—A comprehensive introduction,” *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

SILVIU S. ANDREI received the B.Sc. degree in computer and systems engineering from the University of Transilvania, Brasov, Romania, in 2009, and the M.Sc. degree in computer science from the Illinois Institute of Technology, Chicago, IL, USA, in 2020.

He is currently an Applied Scientist with Amazon, Seattle, WA, USA. Prior to joining Amazon, he has worked at various startup companies doing computer vision work in fields such as, medical imaging, augmented reality, and SLAM. His interests include computer vision, machine learning, augmented reality, SLAM, and video enhancement/super resolution.

NATALIYA SHAPOVALOVA received the B.S. degree in computer science from National Technical University Kharkiv Polytechnic Institute, Kharkiv, Ukraine, in 2006, the M.Sc. degree in computer vision and robotics from Heriot-Watt University, Edinburgh, U.K., in 2009, and the Ph.D. degree in computer science from Simon Fraser University, Vancouver, Canada, in 2015.

She is currently an Applied Scientist with Amazon.com, Seattle, WA, USA. Her research interests include computer vision and deep learning applications for videos such as, action recognition in videos and video super resolution and enhancement.

WALTERIO MAYOL-CUEVAS received the B.Sc. degree from the National University of Mexico and the Ph.D. degree from the University of Oxford.

He is currently a Principal Research Scientist with Amazon.com, Seattle, WA, USA, and a Professor with the Department of Computer Science, University of Bristol, U.K. His research with students and collaborators proposed some of the earliest versions of visual simultaneous localization and mapping (SLAM) and its applications to robotics and augmented reality. These include flagship humanoid robots and early commercial applications of visual mapping for wearable computing. Most recent works include novel algorithms for video enhancement, new concepts of human–robot interaction, fast computer vision methods for scene understanding, algorithms for novel visual sensors, and machine learning methods to assess skill. He was the General Co-Chair of BMVC 2013 and the General Chair of the IEEE ISMAR 2016.