DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Deep upscaling for video streaming

A case evaluation at SVT

FREDRIK LUNDKVIST

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Deep upscaling for video streaming: a case evaluation at SVT.

Fredrik Lundkvist
KTH Royal institute of technology
Stockholm, Sweden
flundkvi@kth.se

ABSTRACT

While digital displays have continuously increased in resolution, video content produced before these improvements is however stuck at its original resolution, and the use of some form of scaling is needed for a satisfactory viewing experience on high-resolution displays. In recent years, the field of video scaling has taken a leap forward in output quality, due to the adoption of deep learning methods in research. In this paper, we describe a study wherein we train a convolutional neural network for super-resolution, and conduct a large-scale A/B video quality test in order to investigate if SVT video-on-demand viewers prefer video upscaled using a convolutional neural network to video upscaled using the standard bicubic method. Our results show that viewers generally prefer CNN-scaled video, but not necessarily for the types of content this technology would primarily be used to scale. We conclude that the technology of deep upscaling shows promise, but also believe that more optimization and flexibility is need for deep scaling to be viable for mainstream use.

Author Keywords

Deep learning; Super-resolution; Video streaming; public service; perceptual evaluation; ESPCN.

CCS Concepts

•**Computing methodologies** → *Neural networks;* **Image processing;** •**Human-centered computing** → *Empirical studies in HCI;* User studies;

INTRODUCTION

Since the mid-2000s, high-definition displays have become cheaper and more prevalent, and can now be found in televisions, computer monitors, and smartphones. Simultaneously, video streaming services have increased in popularity, resulting in video being expected to account for more than 80% of all internet traffic by 2022[4]. While content production has adapted to improvements in display technology by simply recording at higher resolutions, content created before these resolution shifts is stuck at its original resolution, and does not look good on modern displays without the use of some form of scaling.

Super-resolution (SR) is the somewhat ill-posed problem of reconstructing or creating a high-resolution (HR) image from a low-resolution (LR) input. While the idea of this operation being trivial is common in popular culture, it is an extremely hard task in reality. This difficulty is primarily due to the fact that there is no guaranteed one-to-one relationship between low- and high-resolution images, as multiple HR images could theoretically yield the same LR image when subsampled. Although the problem itself is challenging, solutions that are deemed to be "good enough" are used in fields such as medical imaging, high-resolution display technology, video processing, and photography.

Before the machine learning revolution of the 2010s, most SR tasks were performed using bespoke algorithms; in recent years, researchers have shifted their attention towards using deep neural networks instead, primarily making use of Convolutional neural networks (CNNs) or Generative adversarial networks (GANs). This approach has already found mainstream use in fields such as photo editing[10] and video games[8].

The promise of deep super-resolution is higher quality upscaling of resolution-limited content –such as old TV shows or movies for which the source material might not exist at higher resolutions, or re-scanning at higher resolutions is deemed too expensive or time consuming– to enable improved viewing experiences on contemporary high-resolution displays. Further into the future, performant super-resolution models could be used for real-time scaling on viewing devices, in order to reduce payload sizes for streaming services, thus reducing distribution costs for service providers.

At the time of writing, SVT provides legacy content to viewers under the "öppet arkiv" (open archive) brand[1] via their video-on-demand (VOD) service SVT play. Low-resolution content, such as old broadcast recordings in PAL resolution (720 × 576 pixels) is upscaled using bicubic interpolation[19] video filters in FFMPEG[9], with Lanczos[15] filtering being used in some experiments. Using a deep learning model for this task could improve upsampling speed as well as output quality, as shown in multiple papers[25][12][14]. While these papers have shown impressive results on objective metrics such as PSNR or SSIM, there has to our knowledge not been

ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

any studies conducted to verify that the outputs of these deep models are also *perceived* as better-looking by actual humans by using large-scale subjective evaluation.

This report concerns a case study wherein we implement a slightly modified version of Shi et. al's ESPCN model[25], and train it using frames extracted from source files in the SVT video archives. Our model is then used to scale a number of video clips, which are used for a large-scale, double-blind A/B test in order to determine if SVT play viewers prefer video scaled using deep learning to video scaled using bicubic interpolation.

Problem definition

The main question investigated in this paper is *Do SVT play viewers prefer video upscaled using neural networks to video upscaled using bicubic interpolation?* We attempt to answer this question by conducting a large scale double-blind A/B test of perceived video quality, based on the ITU standard for video quality assessment [5]. Based on our results, we also discuss the feasibility of using deep upscaling in a video processing flow for a video streaming platform.

METHODS AND STATE OF THE ART IN SUPER-RESOLUTION

As explained briefly in section 1, super-resolution is the problem of constructing a high-resolution image *HR* from a low-resolution image *LR*. This can be accomplished in different ways: the most common approach is to use interpolation techniques such as bicubic interpolation, or other bespoke algorithms. With the deep learning revolution, research focus has shifted towards super-resolution using deep neural networks. To our knowledge, no major video streaming service uses deep upscaling beyond a few experiments. Returning to the world of research, several network architectures have been proposed in the past few years, of which we will present and discuss some of the more important in section 2.2. In this section, we will begin by quickly presenting the most commonly used conventional methods (Bicubic interpolation and Lanczos upsampling), before presenting different deep learning approaches to the super-resolution problem.

Conventional methods

Super-resolution is another term for supersampling, specifically supersampling of a digital image signal; thus, any supersampling algorithm could theoretically be used for super-resolution. In theory, a sinc filter (also known as the Whittaker-Shannon formula) would provide the best possible reconstruction or supersampling of a signal. However, this filter makes theoretical assumptions that are not necessarily true for real-world digital images. In practice, Lanczos resampling [15] –which is an approximation of the sinc filter technique– provides very good results, but is also computationally expensive. At the time of writing, the most commonly used approach for image upscaling is Bicubic interpolation[19], which can be seen as an approximation of the Lanczos technique, as well as an improvement of bilinear interpolation. While less computationally intensive than Lanczos, Bicubic interpolation is still computationally expensive and slow; using bicubic upscaling as a part of an ML model has been shown to slow down performance considerably [14].

In general, the use of conventional upscaling methods faces the user with a choice between quality and speed, and one should choose an appropriate method based on their specific use case. Because of this inherent trade-off nature of video scaling, there exists an incentive to find alternate solutions that are ideally both faster and provide better results. In the following subsections, we will present new approaches towards super-resolution utilizing deep learning.

Deep learning methods

Since the early 2010s, machine learning methods have become more and more prevalent, in research as well as in industry. This is due to a number of factors, including, but not limited to: faster, more capable, and cheaper hardware in the form of GPUs, theoretical advances, as well as impressive results when applied to open problems in a number of fields such as computer vision and natural language processing. Today, neural networks are used for tasks such as translation, image classification, autonomous vehicles, and medical diagnosis.

In this subsection, we will present the two major architecture families for super-resolution applications: Convolutional neural networks and Generative Adversarial networks, and highlight some of the more influential models.

Convolutional neural networks

Convolutional Neural network (CNN) architectures are commonly used for tasks that use signals as input data, such as images, audio, and video. In essence, CNNs can be viewed as a series of signal convolutions, with kernel parameters being learned during model training. SRCNN [13] is the first notable example of a CNN designed for super-resolution, and achieved impressive performance compared to standard approaches. It is worth noting that SRCNN uses a bicubically upscaled version of the LR image as input data, which slows down the model considerably. The authors of the original paper would later revisit their work to improve performance by using a deconvolution filter for upscaling, rather than bicubic interpolation. This –along with other changes– resulted in FSRCNN[14]. Shi et al. adopted a similar approach –learning all convolutions *before* upscaling rather than after– for their model ESPCN, and made use of a novel pixel-shuffling layer to produce the final HR images. This approach improved computational efficiency, reducing the amount of parameters and allowing their model to achieve real-time video upscaling on a single NVIDIA K2 GPU [25]. While newer models have surpassed ESPCN in terms of distortion measures such as PSNR and SSIM, it is still relevant today as a lightweight model that produces very good results for its size (see table 1 for comparison to other models).

Since the publications of SRCNN, FSRCNN and ESPCN, deeper CNN architectures such as VDSR[21], DRCN[20], and EDSR[23] have been proposed. By making use of novel techniques such as residual connections, they manage to outperform the previously mentioned models in PSNR and SSIM measurements. However, these newer models are very large, and as Yang states: "*...it is still difficult to deploy these models to real-world scenarios, which is mainly due to massive parameters[sic] and computation*"[26]. The parameter growth of newer models is illustrated in table 1.

As such, ESPCN and FSRCNN remain two of the most attractive architectures for real-world use cases such as video streaming services, where large amounts of data need to be processed by the model often, and performance is crucial.

Perceptual loss functions & Generative adversarial networks

The models presented in section 2.2 normally use objective mathematical loss functions during training, with mean-square error (MSE) being used as the baseline in Yang et. al's direct comparison [26], and in most papers. However, research has shown that using non-standard loss functions can yield perceptually impressive results. By using layers of pre-trained models –such as VGGnet– as a *perceptual loss function*, Johnson et al achieved notable results in style transfer and super-resolution tasks[17]. While fine detail was worse than models trained with conventional loss functions, image features such as edges and color transitions were more pronounced, which could be argued to be a perceptually better scaling.

The idea of perceptual loss functions is closely related to the concept of Generative Adversarial Networks (GANs). In essence, a GAN model consists of two networks, a *Generator* and a *Discriminator*, with the generator continuously trying to fool the discriminator, with both networks learning from each attempt. At the time of writing, the best performing super-resolution models are primarily GANs, with SRGAN[22] being the most notable example. However, it is important to acknowledge that these models have drawbacks that limit their use in practice. GANs often fail to converge during training [7][6][24], and are susceptible to mode collapse[7][16]. In addition to these issues, GANs often require large amounts of parameters, since they consist of multiple networks, with one or both often being very large alone as well. These issues make GANs less suitable for research where the network itself is not the main subject, as training would take far more time than a conventional fully-connected network or CNN, even if no complications are encountered. Nevertheless, GANs are a subject of intense research (at the time of writing), and hold a lot of promise once a reliable solution to the training issues are found.

Video super-resolution

The models and methods presented in the previous subsections have been created for the purpose of *single image* super-resolution, i.e for scaling of still frames. These methods are still applicable for video, as one can simply upscale individual frames sequentially or in parallel, depending on the decoder used. However, this approach fails to consider the temporal dimension inherent to the medium of video, which could theoretically lead to performance gains. Kappeler et al proposed VSRNet[18], a model that processes multiple frames together to account for the temporal aspect as well; adding motion compensation to the model was shown to improve performance. Like SRCNN, VSRNet upscales the LR images using bicubic interpolation before using them as model input, an approach that increases processing time and computational costs. This combined with an inefficient motion compensation algorithm results in VSRNet only achieving 0.016 frames per second on LR video with lower than standard definition[1] resolution [12]. Caballero et al [12] combined the ESPCN architecture[25] with a spatial-temporal network to exploit intra-frame correlations, resulting in VESPCN achieving higher PSNR and better performance than that of VSRNet. Despite the results of VESPCN, research on video-specific SR models has not received much attention, with single-image models being the primary focus of the field; perhaps due to the fact that single-image models can be used for both stills and video.

Metrics for visual quality

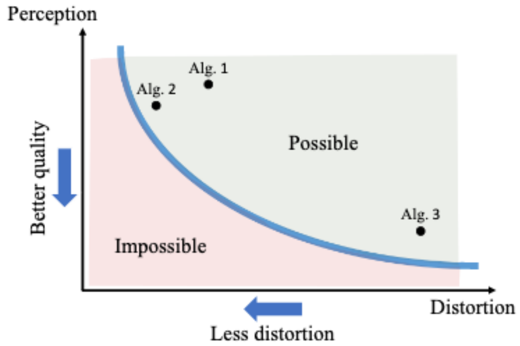


Figure 1. The perception-distortion relationship visualized. The graph shows that no reconstruction algorithm can achieve optimal perceptual quality and minimal amounts of distortion simultaneously. Source: [11]

Within the field of image processing, different approaches and metrics are used for assessing the quality of upscaled images and video. Due to ease of comparison, objective distortion measures such as Peak signal-to-noise ratio (PSNR) is the most commonly used in research, sometimes accompanied by structural similarity image metric (SSIM). It is also customary to include samples of upscaled images in published reports, to allow readers to make their own qualitative assessment of the results. However, it is uncommon to conduct large-scale perceptual studies to assess output quality.

While most published research in the field share results in the form of PSNR or other distortion values, it is important to note that low distortion does not necessarily mean that images and video are also *perceived* as better looking by human viewers. On the contrary, Blau and Michaeli showed that there is an inherent tradeoff between maximizing perceived quality and objective distortion measurements[11]. This relationship is visualized in figure 2. From this, one can deduce that super-resolution is not only an ill-posed problem, but also might be impossible to solve optimally. The fact that finding an optimal solution might not be possible should however not be seen as discouragement to study super-resolution, as there may still be improvements to be made.

As a counterweight to the distortion measures, perceptual quality of upscaled images should be evaluated as well, using quantitative methods where possible. This can for example be done by using perceptual metrics such as Netflix's VMAF [2], or by conducting A/B-tests to compare methods in pairs,

[1] 480 (NTSC) or 576(PAL) pixels in height, depending on broadcast region. Training data included both.

Model	PSNR (4x upscale)	Dataset	Parameters
SRCNN_EX	30.49	ImageNet subset	$57 * 10^3$
ESPCN	30.90	ImageNet subset	$20 * 10^3$
VDSR	31.35	G200 + Yang91	$665 * 10^3$
DRCN	31.53	Yang91	$1.77 * 10^6$
EDSR	32.62	DIV2K	$43 * 10^6$

Table 1. Comparison of PSNR and number of parameters of different models for super-resolution. Data from Yang et al [26].

a method commonly used for qualitative assessment of television picture quality[5]. Interesting to note is that no one has conducted such a study yet (to our knowledge). For our study, we chose to conduct a comparative A/B test between a known efficient and well-performing deep super-resolution model (ESPCN) to the commonly used bicubic scaling method, in order to gain insight on whether audiences perceive one of the methods as better-looking.

METHOD

As mentioned in sections 1 and 2.3, a knowledge gap exists on whether or not the output of super-resolution models are *perceived* as better looking by human audiences. In this paper, we intend to fill part of this gap by comparing output from a standard super-resolution algorithm (bicubic interpolation) to the output from a deep neural network (ESPCN) via double-blind A/B testing –a commonly used method for assessing television image quality[5]– to investigate which method is preferred by SVT video on demand (VOD) viewers. In this section, we will explain our choice of model, as well as how it was implemented and trained. We will also describe our evaluation process and participant selection.

Model and training

Model

For our study, we decided to implement ESCPN[25] using pytorch[2], with the model configuration used in the original paper (3 convolutional layers, $64 \to 32 \to c^2$ output channels with a final pixel-shuffle layer), and train it to perform super-resolution. Our implementation differs from the original in that we train our model on the full YUV image, rather than just the Y channel, which was done in [25], with the chroma channels being bicubically scaled. Although many newer models outperform ESPCN in output PSNR, they are larger by orders of magnitude (see table 1). In the case of GANs, there is also the issue of unstable training[7][6], making implementing and training a GAN deemed to be too time-consuming for this study. ESPCN is a faster model, both due to it's smaller size as well as its general structure. A simpler architecture is also easier to both implement and debug, further justifying our choice of model.

As the model is not the main focus of our study, we chose to use a smaller model to ensure that time was spent on evaluation rather than debugging.

It is important to note that ESPCN is a *single-frame* super-resolution model, as stated in section 2, meaning that it will not be able to take the temporal dimension into account. Once again, ease of implementation influenced our decisions, as using VESPCN would have required more configuration and training.

Training and datasets

Training was conducted largely in the same manner as [25], with some alterations. While the original model only learns super-resolution in the luma (Y) channel (scaling U and V using bicubic interpolation), our model was configured to super-resolve all three.

The training and validation data consisted of still frames extracted from high-quality source files selected from the SVT archives. These video source files are of higher quality than those served to users, resulting in better preservation of textures and fine details. Due to SVT storing source material in an interlaced format, all videos were de-interlaced before extracting between 40-90 frames, depending on the length and visual variety of the file. A downsampled copy of each frame was produced; each frame (both low and high resolution) was then divided into patches of 32×32 pixels for the low-resolution versions, with the corresponding high-resolution patches being $s * 32 \times s * 32$ pixels, with s being the model scale factor. The patch size was decided after running an experiment wherein a 3× upscaling model was trained for 100 epochs on the general100 dataset introduced in [14] with varying patch sizes. A patch size of 32 pixels produced the highest PSNR on the validation set, and was thus chosen for training the final models. The model used for the main study had an upscaling factor of

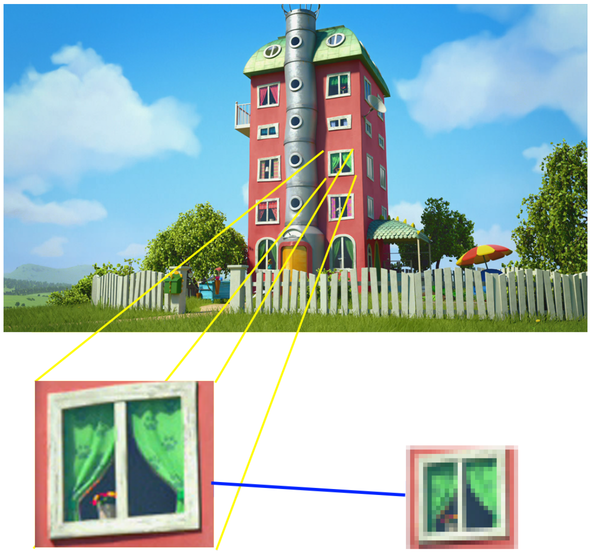


Figure 2. Visualization of our data pre-processing. Patches are extracted from still frames and downscaled.

[2] https://pytorch.org/

2. Since ESPCN learns scaling using a set factor rather than direct transforms, using patches allowed us to iterate faster, as well as using source material with varying resolutions and aspect ratios; our dataset included content with 16:9 aspect ratio (1920 × 1080 and 1280 × 720 pixels) as well as 4:3 (720 × 576 pixels). While all training videos were fetched from SVTs archives, there was no guarantee that training data would be similar to test data, due to the large variety of content present in the archives. To accommodate for this, we attempted to extract data from as many different genres and types of content as possible, using news broadcasts, documentaries, period dramas, children's shows, and more. We also chose to train the model specifically for live-action content. While animation is a common video genre, the content is too dissimilar to live-action for a model to be able handle both well. To support our argument of separating live action and animation, it is worth mentioning that SVT uses different encoding profiles for live action and animated video due to this difference in content and thus required encoder settings.

Our model was trained until no improvement in PSNR on the validation set had been found for 100 epochs, with the best performing model being selected for use; making use of early stopping in this way allowed us to avoid overfitting. Training the model used for evaluation took roughly 5 days on a single nvidia quadro GPU.

Evaluation

In this subsection, we will present the evaluation methods used for this study. We will start by discussing the content used in the evaluation, before explaining the evaluation process itself, and finally discussing participant selection and possible biases.

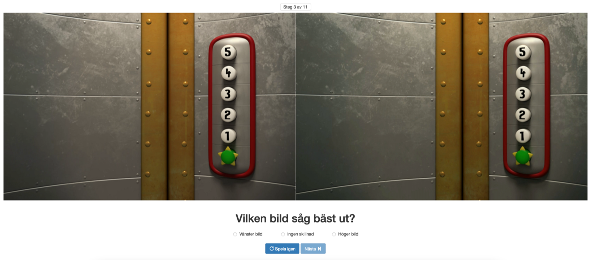
Content for evaluation

Ten shows were selected for evaluation as a representative sample of content available on SVT play. The selection included various genres such as drama, documentary, studio broadcast, stage performance, and children's programming. We also made sure to include some legacy content in our selection. The full list of content used for evaluation can be found in table 2.

From each selected show, a random episode was selected; from each episode, a one-minute clip was deinterlaced and extracted for use in our evaluation process. Each extracted clip was encoded using the lossless variant of the h.264 video codec. Clips were selected based on visual content, to ensure that participants were exposed to a variety of upscaling scenarios. The non-legacy clips were scaled down to a resolution of 960 × 540 pixels. Legacy clips were not downsampled, and were instead upscaled from their original resolution of 720 × 576 pixels (PAL resoultion). All low-resolution clips were then upscaled using two methods: FFMPEG's built-in video scaling filter using bicubic interpolation, as well as our trained model using a custom script. The upscaling was conducted without bitrate limits, and the output files were encoded using the the lossless version of the h.264 video codec. After upscaling, both variants were once again encoded using h.264, this time using the standard encoding profile for SVT play, resulting in an average bit rate of 3.1Mbit/s and a resolution of 1920 × 1080 pixels. This was done in order to match the circumstances viewers would normally be watching SVT play content in, as well as reducing file sizes in order to allow participants to do the test online. By encoding all evaluation clips with the same profile, we also ensured that the two methods were presented to participants under equal circumstances. It is worth noting that SVT uses genre-specific encoder profiles for their VOD service, and that we specifically chose to encode all videos with their standard "program" h.264 profile rather than using matching genre profiles for each clip. We argue that using genre-specific profiles would have added a variable to our evaluation, as differences in encoder settings such as deblocking filters, entropy coding,buffer size, or quantization strength could lead to differences being more or less pronounced than they would have been using another profile. By using the same encoder profile for all clips, we could ensure identical encoding and compression circumstances.

Double-blind A/B testing

In order to conduct our evaluation, we modified an existing web application used by SVT for assessing perceived video quality to support playing two videos in sync simultaneously. This allowed us to avoid any generational losses that would have been introduced if we had merged the separate versions into one clip for comparison. When taking the survey, participants were presented with two versions of each clip, presented side by side on screen. They were prompted to watch the videos carefully, and decide which version looked better to them, or if they found them to be of equal quality. This process was repeated for each of the 10 clips used for evaluation. The interface of the web application is shown in figure 3. Clip variants for each screen position (right/left) were independently randomly selected, meaning that four combinations were possible: CNN/CNN, CNN/bicubic, bicubic/CNN, and bicubic/bicubic, all with equal probabilities of occurring. The aim of the web app was to loosely adhere to the ITU standard for subjective assessment of television picture quality[5]. By conducting evaluation using a web application, we could reach a larger number of participants than in-person evaluation would have allowed; the latter approach would also have been made more difficult by the ongoing COVID-19 pandemic. One drawback of this approach was that we did not collect any comments from the users, as that would have made exiting the survey page harder when partaking from a mobile device (an option that would not be used anyway); thus, the only data we had to analyze was video comparison answers.



Vilken bild såg bäst ut?

Figure 3. A screenshot from the survey web app. Users are prompted to select which version looks best (left, right, or no difference). The user can restart the playback, and is allowed to move on to the next clip when an answer has been provided.

Show	Genre	Original Resolution	Legacy content
Bolibombpa: drakens trädgård	Children	1920 × 1080	No
Livets mirakel	Documentary	720 × 576	Yes
Rapport	Studio news	1280 × 720	No
Atlantic Crossing	Drama	1920 × 1080	No
Melodifestivalen	Music	1920 × 1080	No
Rederiet	Drama	720 × 576	Yes
Trädgårdstider	Lifestyle	1920 × 1080	No
Diagnoserna i mitt liv	Stage performance	1920 × 1080	No
Leif och Billy	Comedy	1920 × 1080	No
En bild berättar	Art	1920 × 1080	No

Table 2. Shows used for evaluation. Source material courtesy of SVT.

Participants, selection, and bias

When using people's opinions as evaluation data, it is important to ensure that the participants in the study are as representative of the intended target group (if not a general population) as possible. While target group definitions might vary between different studies, the primary target group for our study was SVT play users. While this group is perhaps not representative of a general population of VOD viewers, it is the exact group that would benefit from improved video quality on the platform. Participants were recruited by sending out a survey via the SVT play web app to 15% of all users over a time period of two weeks. Participation was voluntary, and no identifying data was saved by the web application.

Since the test population was self-selecting, our aim was to recruit a large number of participants, in order to mitigate any self-selection bias, leading us to decide that 1000 participants would be the minimum required amount.

Before partaking in the survey, participants were informed of how the test was structured, what they were expected to do, and what data would be saved; however, they were *not* notified of what the test was meant to evaluate, beyond *"Potential video quality improvements for SVT play"*; this was done purposefully, in order to ensure that participants would not behave differently due to knowing what to look for, and thus focus on specific parts of the videos that they might not have paid attention to otherwise.

RESULTS

Survey results

In total, we received 3292 responses from the web application. Due to only collecting data regarding which variants were shown, as well as participant's responses, display resolution, Operating system (OS), and OS version, we do not have any deeper information regarding the demographics of the respondents. Analysis of variance (ANOVA) was conducted in order to assess the respondents' preferences. For seven of the ten clips used for assessment, viewers showed a significant preference for the version upscaled using our CNN ($p < 0.01$) when comparing it with the bicubically upscaled variant. For the remaining three clips, no preference could be found regarding upscaling method; it is worth noting that both legacy clips were in this group of no-preference content. This implies that viewers do not prefer deep upscaling for the type of content it would primarily be used for if SVT were to start using it in their existing video processing flow.

Our results showed stronger preferences for the CNN version in clips with prominent textual content such as news headlines, and clips with prominent textures such as tree bark. While skin texture and other fine details were also sharper in the CNN-scaled versions, this did not seem to impact results as much.

Output comparison

When comparing video files that are scaled using the different methods side by side[3], some differences are immediately obvious, while others require more critical viewing. The most obvious difference is contrast: videos scaled with our CNN consistently have slightly higher contrast than the bicubically scaled versions; we reason that this is due to the averaging nature of bicubic scaling not being able to create or "hallucinate" high-frequency data in the same way as our CNN. Prominent edges also seem slightly sharper; this is most noticeable when looking at text, for example in the form of news headlines. In general, the CNN seems to be able to scale high-frequency content slightly better.

When zooming in, more differences become apparent. Fine detail and texture are clearer in the CNN version, as well as borders that might be smudged out by the bicubic scaling. Examples of this can be seen in figures 4 and 5. Note the edges on the grater in figure 4 and the separation between the leaves of grass in figure 5.

At first glance, it might seem as though some of the output from the CNN is slightly color-shifted. However, we verified that this was not the case by comparing the waveforms of the original high-res clips with both scaled versions, finding no differences except for the slightly higher contrast mentioned earlier.

As mentioned in section 4.1, audiences did not seem to perceive any differences between the different variants when viewing legacy content. Indeed, when comparing the two versions side by side, the differences are not as prominent as they were for the non-legacy content. It seems that noise that may be due to legacy mediums or codecs used becomes more prominent in the CNN upscaled variant (see figure 6), which might be a consequence of the models tendency to preserve high-frequency content (i.e sharp edges) when scaling. The model would also produce artifacts when scaling some frames, see figure 7. Much like the watercolor effect shown in figure 6

[3] we recommend using https://svt.github.io/vivict/ for this.

it seems that artifacts were more likely to occur when scaling legacy content.

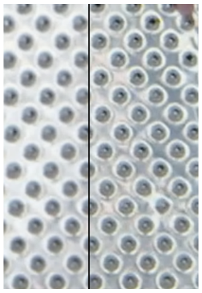


Figure 4. Comparison crop of upscaled video. Left: bicubic scaling. Right: Deep scaling.

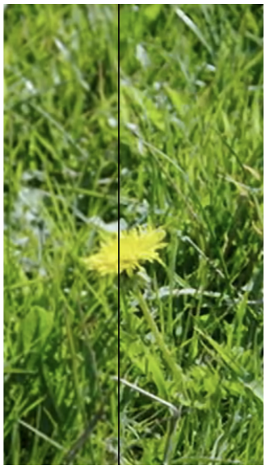


Figure 5. Comparison crop of upscaled video. Left: bicubic scaling. Right: Deep scaling.

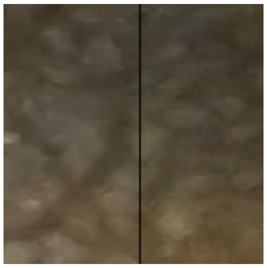


Figure 6. Comparison crop of upscaled legacy video. Left: bicubic scaling. Right: Deep scaling.

DISCUSSION

In this section we discuss the results presented in section 4, and present our theories about why these results were obtained. We also discuss limitations of deep using scaling in practice, as well as participant selection for our evaluation.

Impact of content type

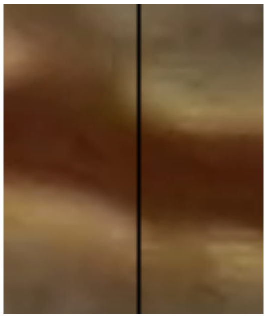


Figure 7. Comparison crop of upscaled legacy video. Left: bicubic scaling. Right: Deep scaling.

As mentioned in section 4, the clips scaled with our CNN were favored in the majority of cases, but not always. Further, the preference seemed to be greater when the video in question prominently featured text or detailed textures such as tree bark or cheese graters. On the other hand, we found that no noticeable preference could be found for upscaled legacy content. Our results thus indicate that the payoff of using DL-based scaling varies depending on the content being scaled, and that adopting DL scaling might not be a good universal solution (see section 5.3 for further discussions).

Interestingly, no clear preference was found regarding the legacy content, i.e the videos with low original resolution (576 pixels vertical). We believe that there could be several reasons for this: the first possibility is that the chosen content did not feature many elements where the impact of CNN scaling is the most noticeable, such as prominent text, sharp texture, or other high-frequency image content. The second possible reason is intermediate scaling; since the input file resolution was not a divisor of the target resolution (576 pixels and 1080 pixels in vertical resoultion respectively), slight downscaling was applied in the final transcoding step. The software profiles used bicubic methods for scaling, and as such the version presented to participants had very slight bicubic downscaling applied. It is possible that this negatively impacted the quality of the clip; however, this impact was deemed negligible by us. Still, it is possible that it had an effect on our results. Another factor might be the methods used for storing legacy content. These files were encoded using the DVPro video codec, a codec for tape media that is older and less efficient than the DNxHD codec used for non-legacy files. We reason that compression artifacts, blurring, or softening may have been introduced when transferring the content from tape to digital, or when encoding it, leading to lower source quality, meaning that there was not as much detail for the scaling methods to find or preserve. It could also be that the scaling function found by our model comes out on the wrong side of the perception/distortion curve (fig. 2) presented by Blau and Michaeli [11]. Our network is trained with an objective loss function (MSE), and as mentioned in [11] as well as in section 2.3, low distortion does not necessarily indicate output that is perceived to look good. Regardless, the CNN seemed to handle these files a bit worse than others, see section 4.2.

Perceiving video

While the results from papers on super-resolution models [25][22][23] would lead one to think that images scaled using deep learning would *always* be perceived as better looking, our results indicate that this assumption is not necessarily true.

We speculate that this is due to a discrepancy between objective measures of video quality and the way we as humans view and perceive video. As mentioned in section 2.3, and explored in [11], maximizing these objective measurements does not guarantee that human audiences will *perceive* the material as better looking. When watching streaming video, users generally do not crop, zoom, or otherwise inspect the material critically; if the video quality is not disastrous, many users do not pay it any mind (and they certainly do not try to assess stream bitrate or codec settings). However, the models discussed are usually optimized to enhance features where a difference is barely noticeable if one is not watching critically, meaning that the perceptual returns quickly become diminishing in the context of casual consumption. One possible way to avoid these diminishing returns is to instead use perceptual optimization (as recommended by Blau), but this method risks losing detail compared to standard methods.

In general, ensuring good video quality is a game of tradeoffs, wherein one must find a balance between file size, video quality, and computing resources. So while this approach might not be the most flexible, and requires a bit more CPU time than bicubic scaling, some may deem that the increase in quality is worth it; our results, and the limitations mentioned here indicate that choosing to use DL-based scaling is not always the optimal choice for every situation.

Practical limitations of deep upscaling

Our results indicate that streaming audiences at SVT play prefer CNN upscaled video to the bicubic option in the majority of cases. One could argue that this means that SVT (and perhaps even video streaming providers in general) should switch to using some deep learning-based scaling method in their video processing flows. However, this is not necessarily true, as DL-based scaling has a number of limitations that would complicate such a transition.

The first (and biggest) of these limitations is (lack of) flexibility. As an example, ESPCN only supports scaling by integer factors, due to limitations in the pixel-shuffling layer. This is an issue, since there is no guarantee that the target resolution is an integer multiple of the original resolution. For example, the PAL standard has a vertical resolution of 576 pixels, while the Full HD standard has a vertical resolution of 1080 pixels. This results in an scaling factor of 1.875, which ESPCN cannot handle, introducing a need for intermediate scaling which would then use bicubic or lanczos methods. Continuing on the theme of flexibility, most common models for super-resolution learn one specific transformation, either in resolution or in scaling factor; this means that in order to support different input resolutions, multiple models would need to be trained. Additionally, one might need different models for different types of content; while the upper bound for this number is theoretically infinite, we believe that at least two different content models are needed: live action and animation. This is due to the vast differences between these two forms of video content. This would mean that more resources need to be spent on setting up training data, allocating computing resources, etc., increasing the cost of using DL-based upscaling compared to using the bicubic approach.

The second limitation is ease of integration. Bicubic and Lanczos scaling is included in most video transcoding software, while DL-based scaling would require developing a custom plugin for the transcoding software used, or adding a new step to the existing video processing flow. This should prompt potential adopters to think twice, and investigate if the potential increase in perceived quality among viewers is worth the headaches of integration, as well as the extra computing time and resources needed.

In summary: while using deep scaling does lead to a noticeable improvement in most cases, the lessened effectiveness with legacy content, combined with a number of limitations that complicate integration into existing video flows leads us to conclude that the technology is not ready for use at streaming services quite yet. However, it could prove useful in other situations, such as archival, or presenting archived material in educational settings, for example at museums.

Participant selection

Because our participant population was self-selecting (since they participated by voluntarily clicking a link), one should keep in mind that it might not be representative of the overall SVT play user base. In order to compensate for this, we needed a large amount of participants, as the diversity within the group could act as a counterweight to the self-selection bias. In the end, we managed to collect more than 3000 responses, an amount we believe is large enough for this study. It is worth noting that our responses still cannot be seen as representative for the general population, but could be considered representative for the population of SVT play users, at least the population that uses the web application. We would have liked to include mobile users in the test as well, and had prepared our evaluation app for it. However, issues with testing on multiple devices, as well as possible collisions with other experiments resulted in delays, and by the time the app was ready to go, we had already collected a satisfactory amount of responses from the web version. Our mobile adaptation efforts were luckily not in vein, as SVT can use this tool for video quality assessment in future experiments, on both mobile and web.

SUSTAINABILITY AND ETHICS

When discussing deep learning, one cannot gloss over the issue of power consumption. In general, training deep models takes a lot of time, and therefore also a lot of power. While the process can be sped up by using designated hardware (like a GPU), it is still time and power-consuming. As an example, our model took roughly 5 days to train. In many cases, this might be worth it; using a deep model to speed up a time consuming process might be a net gain in the long run, when factoring in savings made by reducing computation time. A great example of this is the transcoding service mux using deep learning to speed up per-title encoding [3], saving a lot of power from being spent on "unnecessary" encodes. Deep learning can also be used to effectivize power grids or appliances, in order to reduce energy consumption.

In the case of this study though, we're investing adding a new feature to an existing process, that might not be immediately useful, or reduce power consumption. In theory, running the deployed model as an FFMPEG filter in SVTs transcoding flow *might* be faster than doing the same scaling using bicubic interpolation, but we do not have the data to back that claim up; the inflexibility of the model may also make usage so rare that getting a net positive from this change could take several years.

When discussing digital modification, enhancement, and remastering of art, it is also important to discuss the ethical implications of these modifications. It can be argued that any modification of the source material warps or distorts the author's vision; for example, converting a video that was originally in 4:3 aspect ratio to widescreen may affect the viewing experience significantly, especially if the content was created with one specific aspect ratio in mind, with some aesthetic and creative choices made based on this knowledge. This discussion becomes even more relevant when using machine learning, as we know very little about what exactly a model does to a given input, processing-wise; when the processing is a black box, the use becomes more dubious, as the creators or editors have less control. Concern has been raised that AI enhancement of historic material could present a inaccurate or even completely false representation of the past, for example by coloring monochrome material incorrectly due to biases introduced during training. Discussions on this topic are ongoing, and no clear consensus or conclusion has been reached. Still, it is important for developers of deep enhancement technology such as super resolution to consider the ethical and artistic aspects of the use of their work. We believe that super-resolution has great technological potential, but that ethical discussions are needed before mainstream adoption.

CONCLUSION

In this paper, we have implemented and trained a deep model for super-resolution, and evaluated if video streaming service users prefer video scaled using this model compared to video upscaled using bicubic interpolation, by conducting a large-scale A/B-test. Our results show that SVT play viewers prefer the version scaled with deep learning to the bicubically scaled version in a majority of cases, but not always. We conclude that while usage of deep learning for video scaling provides increased video quality, lack of flexibility in the models, as well as the trade-off nature of video processing for streaming services means that the decision to use deep learning to scale video needs to be made on a case-by-case basis, depending on the content.

ACKNOWLEDGMENTS

We would like to thank SVT for their support, especially in providing hardware, training and evaluation clips, and letting us use their SVT play platform to recruit participants. In particular, we want to thank the video core team, which was our main collaborator on this study. Olof Lindman (SVT) and Haibo Li (KTH) are thanked for their guidance and supervision of the project, which would not have been completed without their help. We also thank Mike Carmody for his invaluable help with excel, which the project would have failed without.

REFERENCES

[1] Öppet arkiv | oppetarkiv.se. (????). `https://www.oppetarkiv.se/`

[2] 2016. Toward A Practical Perceptual Video Quality Metric | by Netflix Technology Blog | Netflix TechBlog. `https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652`. (June 2016). accessed 2021-01-25.

[3] 2018. Better Video Quality through Deep Learning | Mux blog. `https://mux.com/blog/better-video-quality-through-deep-learning/`. (April 2018).

[4] 2018. *Global Internet Growth and Trends Source: Cisco VNI Global IP Traffic Forecast*. Technical Report.

[5] 2019. BT.500 : Methodologies for the subjective assessment of the quality of television images. (October 2019). `https://www.itu.int/rec/R-REC-BT.500`

[6] 2019. GAN Training | Generative Adversarial Networks | Google Developers. `https://developers.google.com/machine-learning/gan/training`. (April 2019). accessed 2021-01-25.

[7] 2020. Common Problems | Generative Adversarial Networks | Google Developers. `https://developers.google.com/machine-learning/gan/problems`. (February 2020). accessed 2021-01-25.

[8] 2021. Deep Learning Super Sampling (DLSS) Technology | NVIDIA. (May 2021). `https://www.nvidia.com/en-us/geforce/technologies/dlss/?nvid=nv-int-gfhm-55050`

[9] 2021. FFmpeg. (March 2021). `https://ffmpeg.org/`

[10] 2021. From the ACR team: Super Resolution. (March 2021). `https://blog.adobe.com/en/publish/2021/03/10/from-the-acr-team-super-resolution.html`

[11] Yochai Blau and Tomer Michaeli. 2017. The Perception-Distortion Tradeoff. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (nov 2017), 6228–6237. DOI: `http://dx.doi.org/10.1109/CVPR.2018.00652`

[12] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2016. Real-Time Video Super-Resolution with Spatio-Temporal Networks and Motion Compensation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (nov 2016), 2848–2857. `http://arxiv.org/abs/1611.05250`

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016b. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (feb 2016), 295–307. DOI: `http://dx.doi.org/10.1109/TPAMI.2015.2439281`

[14] Chao Dong, Chen Change Loy, and Xiaoou Tang. 2016a. Accelerating the super-resolution convolutional neural network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9906 LNCS. Springer Verlag, 391–407. DOI: `http://dx.doi.org/10.1007/978-3-319-46475-6_25`

[15] Claude E. Duchon. 1979. LANCZOS FILTERING IN ONE AND TWO DIMENSIONS. *Journal of applied meteorology* 18, 8 (aug 1979), 1016–1022. DOI: http://dx.doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2

[16] Ian Goodfellow. 2016. NIPS 2016 tutorial: Generative adversarial networks. (dec 2016). http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9906 LNCS (mar 2016), 694–711. http://arxiv.org/abs/1603.08155

[18] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K. Katsaggelos. 2016. Video Super-Resolution With Convolutional Neural Networks. *IEEE Transactions on Computational Imaging* 2, 2 (mar 2016), 109–122. DOI: http://dx.doi.org/10.1109/tci.2016.2532323

[19] Robert G. Keys. 1981. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29, 6 (1981), 1153–1160. DOI: http://dx.doi.org/10.1109/TASSP.1981.1163711

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2015. Deeply-Recursive Convolutional Network for Image Super-Resolution. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (nov 2015), 1637–1645. http://arxiv.org/abs/1511.04491

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. *Accurate Image Super-Resolution Using Very Deep Convolutional Networks*. Technical Report. 1646–1654 pages.

[22] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2016. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (sep 2016), 105–114. http://arxiv.org/abs/1609.04802

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. *Enhanced Deep Residual Networks for Single Image Super-Resolution*. Technical Report.

[24] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. 2019. Recent Progress on Generative Adversarial Networks (GANs): A Survey. *IEEE Access* 7 (2019), 36322–36333. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2905015

[25] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (sep 2016), 1874–1883. http://arxiv.org/abs/1609.05158

[26] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing Hao Xue, and Qingmin Liao. 2019. Deep Learning for Single Image Super-Resolution: A Brief Review. (dec 2019). DOI: http://dx.doi.org/10.1109/TMM.2019.2919431