# Machine Learning Video Upscaling

Taylor Carrington
carri320@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

## Abstract

This paper reviews a deep learning approach for video upscaling. Video upscaling is when a low resolution video is used to generate a high resolution video. This process starts by splitting each frame of the video into its own image. The upscaling then either upscales each image separately or looks at multiple surrounding images to make choices based on earlier and later images for more consistency. The paper discusses Chu, Mengyu and Xie, You and Mayer, Jonas and Leal-Taix's 2020 generative adversarial networks as an approach for upscaling with a focus on temporal and spatial consistency.

***Keywords:*** machine learning, deep learning, neural networks, Generative Adversarial Network, video resolution, video upscaling, TecoGAN

## 1 Introduction

Video upscaling is the process of changing a low resolution video to a high resolution one, as shown in animated Figure 1. The upscaling can be done in different ways but the most common is with deep learning. Deep learning is a more advanced type of machine learning and tries to mimic a human brain with artificial neural networks. There are many different techniques that use neural networks for upscaling. One of them is TecoGAN, which is Chu et al's [2] 2020 general adversarial network, with the purpose of keeping track of the temporal and spatial consistency in the upscaled video. Temporal consistency is the when the motion of objects between frames is correct and spatial consistency is when the shape of the objects themselves are correct. An example of upscaling would be upscaling low resolution video of a airplane flying across the sky. The high resolution upscaled video's spatial consistency should make sure the airplane continues to have the same shape, and the temporal consistency should make sure the motion of the airplane flying matches the low resolution video. The results of upscaling in general is fair but far from perfect, but it is quickly improving. TecoGAN's results specifically are good and usually, but not always, better than other top upscaling methods.

The paper's road map includes background, TecoGAN, conclusion and acknowledgements.

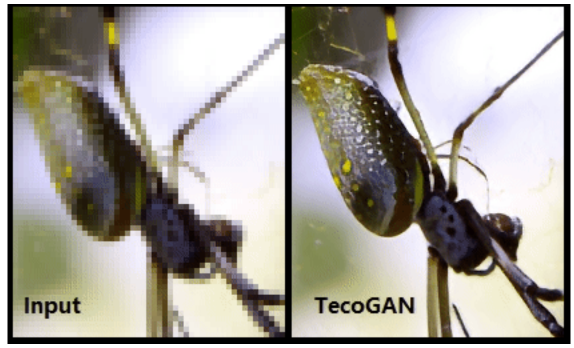


**Figure 1.** Left low resolution spider, right upscaled high resolution spider. Click to start animation [3]

## 2 Background

The following section will include background information on video, upscaling, machine learning, neural networks, loss terms, deep learning, and generative adversarial networks.

### 2.1 Video

A video can be high (HR) or low resolution (LR) and have a variety of frames rates, measured in seconds (FPS). Video resolution is measured by width times height and is often is often referred to by just the height. The most common resolutions are 480p for DVDs (720 x 480), 1080p (also known as 2K) for Blu-ray (1920 x 1080), and 2160p (or 4K) for Ultra HD Blu-ray (3840 X 2160). [9] Although these are the most common resolutions for movies, streaming services such as YouTube can have videos that are anywhere from 144p to 4320p or 8K, but most content is still in the middle range of 480p to 1080p. Additionally, streaming services will show videos with dynamic resolutions based on your internet speed, meaning if you internet speed dips the video resolution will go down until the internet picks back up. Surprisingly, most commercial 4K and 8K content is an upscale of 2K content because most CGI is rendered at 2K. This leads to movies shot on film, usually pre-2000, looking better than a brand new movie, something one might not find intuitive. Movie shot on film looks better because film can be digitally scanned up to 16K but in practice is rarely scanned over 4K. Low resolution is any video that is under 720p while high resolution is equal to or greater than that.

A video is made up of frames which are single images. FPS is then multiple images being shown per second one after

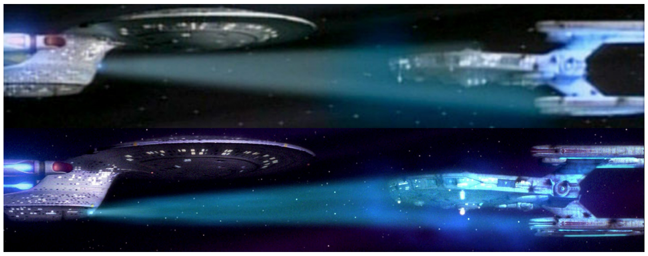

**Figure 2.** Fan upscaling of Star Trek Deep Space 9 4K from a DVD source, cropped. Top 480p DVD, Bottom 4K Upscale [1]

the other. This is perhaps easiest to understand by thinking of creating an 15 FPS stop motion film in which every 15 photos taken corresponds to 1 second of video. The standard for film is 24 FPS while American Television is 29.97 FPS, which means a 90 minute film would have 129,600 individual frames and a 22 minute sitcom would have 39,560 frames [7]. A few movies have higher FPS such as 48 or 60, but are considered controversial with most audiences disliking or not noticing.

### 2.2 Upscaling

Upscaling is taking a LR image or video and generating a HR image or video. An example, Figure 2, show the results of fans upscaling Star Trek: Deep Space 9 to 4K, as the show has only been released on DVD and will not be remastered for Blu-ray. Images are always the thing being upscaled even with Video Super Resolution (VSR), i.e. video upscaling, because the video is broken down into frames and these are upscaled. This means that some models can handle videos by separating them into frames while other models only work for individual images. For the individual image models to process videos you would manually have to split them into frames, and after the upscaling is finished you would have to manually put them back together.

Different LR videos can propose different challenges to upscaling. All LR videos are not the same with some differences being live-action vs animated, fine details vs few details, a lot of motion between frames vs little motion between frames, or the types of objects in the frame. A drastic example is if you train a model on video of buildings using it to upscale a video of a face will not work very well. Videos can have a wide variety of these differences between frames or inside a single frame, therefore good upscaling models needs to be able to handle the variety.

### 2.3 Machine Learning

Machine learning is the study of a type of artificial intelligence which improves the accuracy of a computer algorithm through the use of data and time. Machine learning algorithms take in training data, which can be any type of data, and use it to learn how to classify or make choices based on that data without having to be manually programmed.

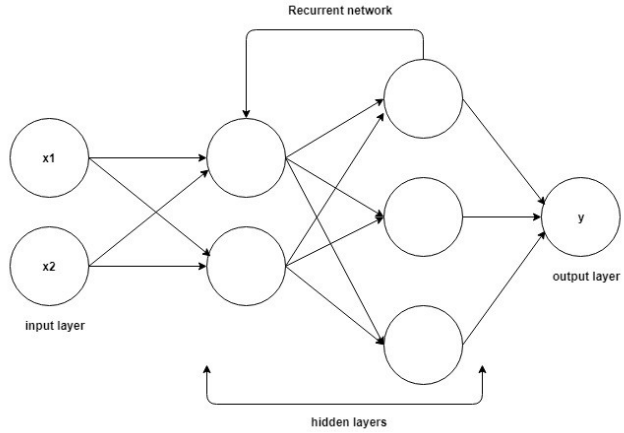


**Figure 3.** Recurrent Neural Network, circles are the nodes, arrows are the edges [4].

The applications for machine learning are numerous including internet credit-card fraud detection, marketing, theorem proving, robot locomotion, adaptive websites, and medical diagnosis. [10] A specific example of this is VSR which takes in training data of LR frames and target HR frames, and learns how to generate HR frames from the LR frames.

The upscaling models are trained, which will be explained more later, with HR images that are down-sampled to LR, which leaves the generated image to be compared with the original HR image at the end. Examples include Figure 1 and Figure 6. TecoGAN down-sampling images with Gaussian blur, a specific mathematical function that creates LR images that are worse quality then real life LR images. LR images being artificially created allows for the original HR image to be the target data. TecoGAN only uses the target data when deciding how well the upscale did, not for the upscaling itself. The original HR image is also known as the ground truth (GT) in VSR, because the upscaling target is to be as close to the GT as possible. TecoGAN is trained on harder down-sampled images because real life use cases will not have GT.

### 2.4 Neural Networks

Neural Networks (NN) uses input data, output data, target data, training, and weights. Input data can be any type of data and is used to start the process; for VSR it is LR frames. Output data is the result at the end of the process; for VSR, output data is the generated HR frames. The way NNs are able to take input data and get accurate output data is through the use of target data and training. Target data is examples of what the output data should be; for VSR this is HR frames. Training is the taking of inputs and matching them to outputs through the target data, to then be able to generate its own data that is similar to the target data.

As shown in Figure 3 NNs are made up of nodes which are the circles and edges which are the arrows. The input data starts at the input layer and makes its way through the hidden layers, node to node using the edges, that uses weights to alter the data, to end up as the output data in the output layer. Most of the time this process goes from one node to the next with no ability to go to a previous node.

Recurrent Neural Network (RNN), are the exception that allows for edges to send data back to a previous node. RNNs are needed for TecoGAN as they allow the generated HR output to be used with the input of the next frame. This is useful so that the LR input knows more about what it should look like when HR and for temporal consistency.

### 2.5 Loss Terms

Loss terms are used to evaluate the overall performance of a network. For VSR the loss terms should be as low as possible. Therefore to improve the performance of a NN, the algorithm makes changes to the weights with the goal of reducing the loss. Perceptual loss terms are loss terms for comparing two images.

### 2.6 Deep Learning

Deep learning is a more advanced type of machine learning "that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces." [6] The multiple layers used in deep learning are themselves artificial neural networks. Deep learning can use Convolutional Neural Network (CNN), is a NN that uses in at least one layer the mathematical operation of convolution which is when a third function results from the operation of two functions. A basic example is multiplying function $f$ and $g$ to get $f * g$. [5]

### 2.7 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were invented by Ian Goodfellow and his colleagues [8] and is a type of machine learning which uses two NNs, the generator and discriminator, that go against each other in a zero-sum game. The generator is fed training data with the goal being to figure out how to create new data that is practically the same as the training data while fooling the discriminator in believing that the new generated data is classified the same as the training data. The two NNs work against each other by having the generator trained on the training data to produce the generated data while the discriminator trains to better decide on whether the data is the "real" training data or "fake" generated data. This is the adversarial zero-sum game where both the generated data and discriminator tries to outsmart the other all with the goal of creating "better" generated data.

A complication of "better" data is that the generator only tries to trick the discriminator, leading to a possible situation where the discriminator keeps incentivising the wrong

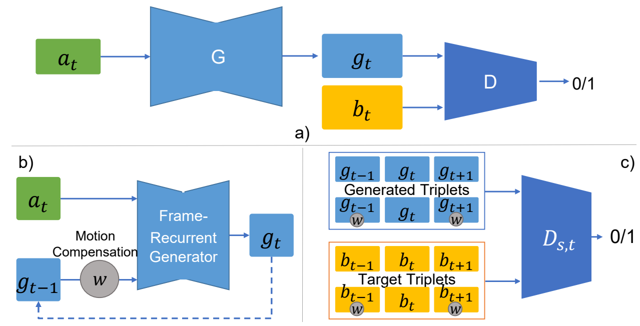


**Figure 4.** a) Short-term temporal consistency GAN. b) In depth view of the frame-recurrent generator. c) In depth view into what goes into the discriminator [2]

thing. The discriminator does not really know "right" from "wrong", and can make mistakes simply from the discriminator overgeneralizing details from the training data. An example of this overgeneralizing problem is for a small face in scenes, because most facial feature would be too blurry to make out so just the skin color is clear. These problematic results can be instinctively caught by humans but missed by machines while also being hard to correct for without complete model retraining. Continuing the previous example, a human will know it is a face and likely have knowledge of who's face it is based on more context (such as character outfit, hair style, voice). One successful applications of GANs is to have a photographic data set of faces and then being able to generate new faces that look real while also being significantly different to the originals, i.e. to create realistic faces of people who do not exist.

## 3 TecoGAN

The following section will include TecoGAN's method, metrics evaluation, and results.

### 3.1 Method

Upscaling using GANs have had a lot of success with individual photos, but have struggled with videos, having severe artifacts, or errors, caused by the difficulty of handling changes between frames. Chu et al's GAN approach [2] is trained in a new way that supervises the spatial contents as well as the temporal relationships. Additionally while their GAN supervises the short-term temporal consistency, shown in Figure 4, the long-term consistency, shown in Figure 5 is self-supervised by the GAN using a new loss formulation that they call "Ping-Pong" loss. TecoGAN is the combination of short-term consistency, long-term consistency, and VSR methods.

Chu et al's short-term temporal consistency GAN approach [2], shown in Figure 4, is made up of a standard spatial GAN for image generation which takes the a single input frame ($a_t$) of LR video and feeds it into the generator,

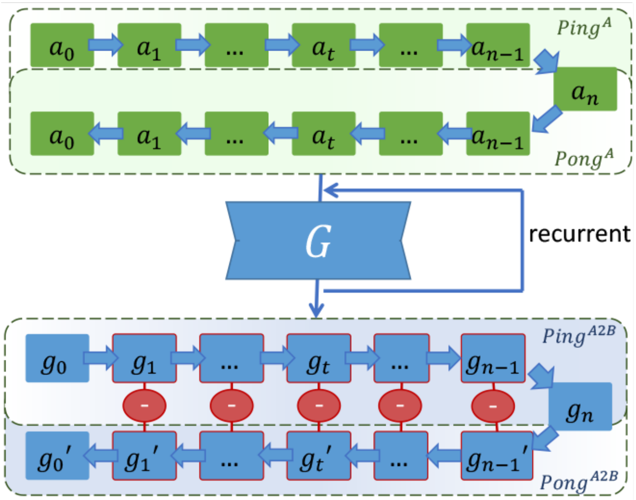


**Figure 5.** Long-term temporal consistency GAN [2]

giving an output frame ($g_t$). As shown in Part (a) of Figure 4, then the output frame ($g_t$) and the target HR frame ($b_t$) are fed into the discriminator (D) where both are compared to each other. Displayed in Part (b) of Figure 4, once there is an output then the new input is a combination of $a_t$ and $g_{t-1}$ that goes first through motion compensation before being fed into the frame recurrent generator, the RNN, which makes it recurrent. In a process called warping ($w$), the motion compensation accounts for the difference in motion between frames $g_t$ and $g_{t-1}$, so that the $g_{t-1}$ matches the motion of the current frame. Shown in Part (c) of Figure 4, the frame recurrent generator output becomes a generated triplet which is the previous, current, and subsequent frames ($g_{t-1}, g, g_{t+1}$) that with an additional target triplet ($b_{t-1}, b, b_{t+1}$) is fed into a spatio-temporal (s,t) discriminator ($D_{s,t}$). The target triplet is made of example HR frames and the discriminator is testing whether the generated triplet has the correct motion and objects in them. The warped triplets have the motion of the previous and consecutive frame being changed to match the current frame, which allows for the motion to be tracked going either forwards or backward. The triplets are only useful for short-term consistency as nothing is known of the motion of the frames just $t-2$, $t+2$ or further away from $t$.

Long-term temporal consistency is managed with a new "Ping-Pong" (PP) GAN approach as shown in Figure 5. While the short-term consistency only knows three frames, the long-term consistency takes a sequence that can have as many consecutive frames as you want. Given a video input of $a_n$, the sequence first "Pings" by going $a_0, a_1, ..., a_t ..., a_{n-1}, a_n$ and then has a reverse "Pong" that goes $a_{n-1}, ..., a_t, ..., a_1, a_0$, which allows for valid data with the motion tracked in both directions. This PP $a_n$ sequence gets fed into the frame recurrent generator, an RNN, to get a video output of sequence $g_n$. This generates a similar, but not the same, "Ping-Ponging" where the "Ping" is $g_0, g_1, ..., g_t ..., g_{n-1}, g_n$ but a reverse "Pong" that $g'_{n-1}, ..., g'_t, ..., g'_1, g'_0$. The generated "Ping" ($g_t$) is based on the forward motion of $a_n$ while the generated "Pong" ($g'_t$) is based on the reverse motion of $a_n$. This means that both $g_n$ and $g'_n$ should be exactly the same. The generated output continuously compares each frame of the "Ping" to its prime "Pong," to make sure that $g_n$ stays close to correct information and the motion going both forward and backwards is the same, if not is punished by the PP loss term. This method helps keep high-frequency details and limits drifting artifacts.

Chu et al's VSR approach [2] for all upscaling is to take LR frames to generate new HR frames. There is a need for adversarial training (a GAN) for VSR as there can be a multimodal problem where multiple structures in HR frames could come from one structure of the LR frame. This problem means some part of a LR frame could be upscaled in different ways which is a problem for spatial consistency. As an example, in LR video a small face can have small but discernible facial features, which means a human brain would know it's suppose to be a face, while the machine has no idea that a face needs to be generated. Furthermore a human is great at filling in the missing data to link a face with a specific person which makes the machine's job more difficult as it needs to generate the right person's face.

The purpose of the discriminator ($D_{s,t}$) for VSR is to teach the generator the correlation among LR inputs and HR targets. The discriminator's inputs are the LR triplet ($I_a = \{a_{t-1}, a_t, a_{t+1}\}$), the HR target triplet ($I^b_{s,t} = \{I_b, I_{wb}, I_a\}$) and the generated upscaled HR triplet ($I^g_{s,t} = \{I_g, I_{wg}, I_a\}$) all concatenated together. The generator is penalized by $D_{s,t}$ if, compared to the real inputs $I_b$, the generated input $I_g$ has unrealistic artifacts, less spatial-details, or less temporal details, allowing for the discriminator ($D_{s,t}$) to identify it as generated frames.

$$\mathcal{L}_{G,F} = \lambda_w \mathcal{L}_{warp} + \lambda_c \mathcal{L}_{content} + \lambda_\phi \mathcal{L}_\phi + \lambda_a \mathcal{L}_{adv} + \lambda_p \mathcal{L}_{PP}$$

Now that this overall architecture is set up the way they test and improve it, is with the perceptual loss function, which is the sum of five loss terms. Where the lambda's ($\lambda$) are the specific weights, that get altered. The $\mathcal{L}_{warp}$ is the warping loss which measures the difference between the input frame and the previous input frame. $\mathcal{L}_{content}$ is the content loss which measures the difference between the generated frame and the GT. $\mathcal{L}_\phi$ is the perceptual loss which measures if the specific objects from the target triplets show up in the generated triplets and uses a feature map, which keeps track of everything in the image. Feature maps keep track of basic things such as if there is a straight line or curly line, to more complex things such as if it is a car or not. $\mathcal{L}_{adv}$ is the adversarial loss which measures how well the discriminator is at judging the generated data. $\mathcal{L}_{PP}$ is the "Ping Pong" loss. Each part of the $\mathcal{L}_{G,F}$ equation breaks down further but I am only going to focus on PP loss as it is novel.

$$\mathcal{L}_{PP} = \Sigma_{t=1}^{n-1} \|g_t - g'_t\|_2$$

This loss term is shown in the bottom generated part of Figure 5. PP loss is the summation from frame ($t$) equals one to frame $n-1$ of the L2 norm of the forward generated frame ($g_t$) minus the reverse generated frame ($g'_t$). The L2 norm $\|x\|_2$ of a vector $x = (x_0, x_1, x_2, ...)$ is the sum of the square roots of the squared vector components, i.e, $\sqrt{|x_i^2|}$. This is done to make sure generated frames motion matches when going either forward or backwards and helps with the long-term temporal consistency.

### 3.2 Metrics Evaluation

To evaluate the effectiveness of TecoGAN, Chu et al used TecoGAN and other methods to upscale videos from the common Vid4 data set (Figure 7), and busy scenes from the 2011 short film Tears of Steel. The other methods TecoGAN is compared to are DUF, FRVSR, EnhanceNet (ENet), EDVR, and RBPN. DUF and FSVSR are top-end models that use upscaling methods for videos and does not have adversarial losses (no GANs), while ENet is a top-end model only for images and therefore does not pay attention to temporal changes. While TecoGAN is trained with 3 million weights, EDVR has 20 million weights and RBPN has 12 million. The number of weights matter because more weights cause the methods to take longer to process the upscale, explained more later with the Processing Time column in Table 1. TecoGAN was trained with HR images down-sampled to LR with Gaussian blur. In these tests TecoGAN could generate realistic results with improved details because of the adversarial learning. Chu et al evaluated TecoGAN using a combination of qualitatively user studies, and quantitative measurements. Their quantitative measurements are common spatial metrics and two new temporal metrics that quantify temporal coherence [2].

A traditional spatial consistency metric is PSNR, which measures how accurate the result is pixel-wise and shows the result's perceptual quality compared to the amount of it vector norm distortion. Better PSNR results have higher numbers. A earlier metric for temporal consistency, LPIPS, measures the perceptual quality, i.e. visually, how close the image is to the GT. These results when lower are better. Chu et al [2] introduced two novel temporal coherence metrics tOF and tLP. Their first new metric, tOF, measures pixel-wise closeness to the estimated motion and their second new metric tLP measures the perceptual distance between adjoining frames.

$$tOF = \|OF(b_{t-1}, b_t) - OF(g_{t-1}, g_t)\|_1$$

$$tLP = \|LP(b_{t-1}, b_t) - LP(g_{t-1}, g_t)\|_1$$

The L1 norm $\|x\|_1$ of a vector $x = (x_0, x_1, x_2, ...)$ is the sum of the absolute value of the vector components, i.e, $\sum |x_i|$. The equation for tOF measures the L1 norm ($\|\|_1$) of optical flow estimation (OF), i.e. the amount of motion in a frame, on the target frame ($b_t$) and previous target frame ($b_{t-1}$) subtracted by the OF of the generated frame ($g_t$) and previous generated frame ($g_{t-1}$). Similarly, tLP equation measures the same but with perceptual LPIPS metric (LP), i.e. how close the motion in the $g_t$ is to the GT, instead of OF. Like the other temporal consistency metric these results are better when the number is lower.

Chu et al ran multiple user studies on the Vid4 data set to compare the different models. Each user saw two methods at a time plus the GT. Until the choice of which one was better was made, each of the two methods were shown on loop and the user did not have play back options so that they could not focus on a particular frame. The methods positions, i.e. they showed up on the left or right side, was randomized. The studies encompassed 50 participants who made a total of a 1000 votes. Better results for the User Study are higher.

The last metric is the processing time required for a 90 minute LR film (129,600 frames) took to be upscaled. It is measured in hours, with the lower the time being better. The models which use more weights take longer to run than those with less weights. Recall that ENet only upscales single images, so it was left out in the processing time as one would have to manually start each frame.

Table 1 shows that TecoGAN is the best or pretty good in all metrics. Recall that TecoGAN is specifically meant to improve on temporal consistency and does in fact come in first in all three temporal consistency metrics. TecoGAN's spatial consistency is respectable, although not the best, but because of the temporal improvements it comes in first in the User Study. Just as important, TecoGAN is significantly faster than any competitive method, signifying the trade-off between great results and fast results does not apply to it.

### 3.3 Results

Table 1 shows the results for each of the metrics on each model. The results show for the spatial consistency TecoGAN is below average in PSNR but for temporal consistency TecoGAN is the best in LPIPS. The temporal consistency results shows that with the more accurate temporal consistency metrics TecoGAN is above average with tOF and the best with tLP. Next the results for the user study shows that out of the four models tested TecoGAN was the most popular. Also TecoGAN had the second fastest processing time while being significantly faster than the three models behind it.

## 4 Conclusion

TecoGAN's VSR is good but has room for improvement. In Figure 7, the lamp example show that EDVR from 2019 was better while both had problems on the building but in different ways. EDVR's lamp is a lot rounder than TecoGAN's but the the scene to the left of the lamp is more blurry. For

| Methods | PSNR↑ | LPIPS↓ ×10 | tOF↓ ×10 | tLP↓ ×100 | User Study↑ | Processing Time (HR)↓ |
|---|---|---|---|---|---|---|
| **TecoGAN** | 25.57 | **1.623** | 1.897 | **0.668** | **3.258** | 1.5 |
| **ENet** | 22.31 | 2.458 | 4.009 | 4.848 | 1.616 | - |
| **FRVSR** | 26.91 | 2.506 | 2.090 | 0.957 | 2.600 | 1.33 |
| **DUF** | **27.38** | 2.607 | 1.588 | 1.329 | 2.933 | 33.92 |
| **RBPN** | 27.15 | 2.511 | 1.473 | 0.911 | - | 18.39 |
| **EDVR** | 27.34 | 2.356 | **1.367** | 0.982 | - | 10.79 |

**Table 1.** VSR of the Vid4 data set on TecoGAN and a handful of previous models. The ↑ means the result numbers show be larger, while the ↓ means the result numbers should be lower. The embolden numbers in applicable columns in the best result out of all the tests.

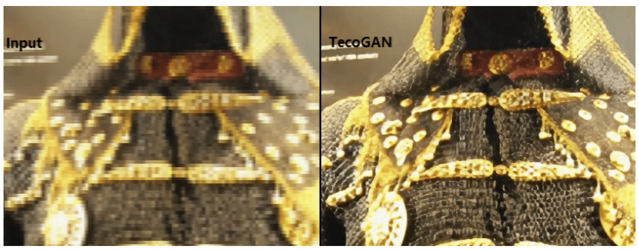


**Figure 6.** Left LR armor, right upscaled HR armor. Notice unnatural artifacts in HR half. Click to start animation [3]

the building TecoGAN is missing alot of detail and overly sharp while EDVR is overly rounding the buildings, most noticeably in the left corner. These two results highlight the difficulty in upscaling as the rounding was good for one result but not the other, showing that upscaling is not one size fits all. The user study should have included RBPN and EDVR as they were closest in terms of upscaled results and sometimes better than TecoGAN, although they take significantly longer to run. Chu et al themselves mentioned that while they were good at temporal consistency, sometimes they lacked face and text details which is less than ideal. Their paper only included static images, which is not the best way to qualitatively judge the results of VSR, especially for temporal consistency. But their supplemental materials [3] include some animated GIFs which look better than any of their static results because artifacts in specific frames can not be focused on. Figure 6, highlights this by the spatial imperfection are harder to notice than in a individual static image, although there is some temporal artifacts that could be improved. Recall, I showed one of their animated results at the start of the paper, Figure 1, which in my opinion is their best result, as it generates impressive amount of detail from the LR down-sampled input and has no noticeable artifacts. Lastly, all their examples seem to be from videos with natural movements and I would be curious to see results from unnatural movements like cutting between cameras or fading to black, as I have seen interpolation have problems with this before.

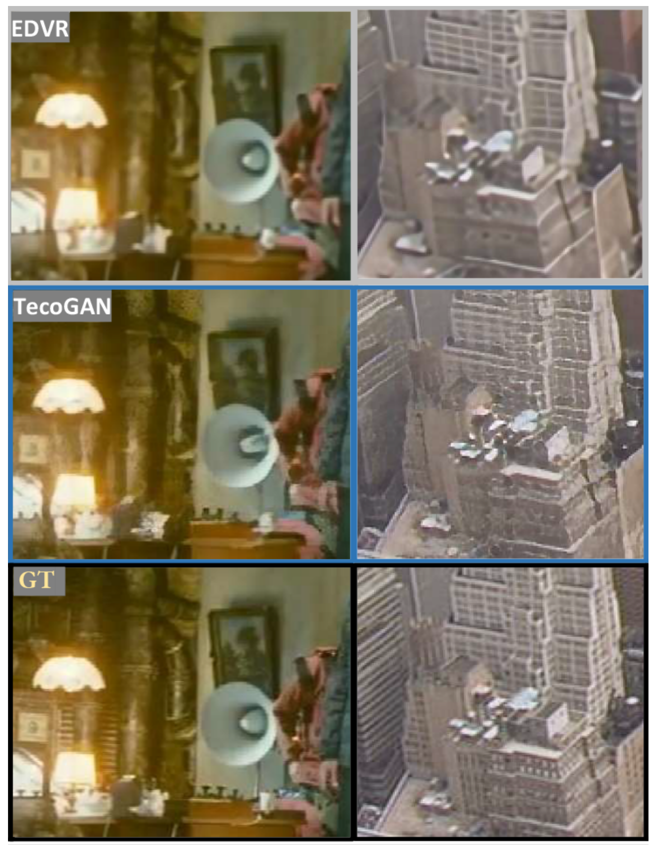


**Figure 7.** VSR of parts of Vid4 data set by EDVR and TecoGAN with the GT for reference. Shows cases where TecoGAN is not better [2].

## Acknowledgments

I would like to thank my advisor Nic McPhee, the senior seminar professor Elena Machkasova, and external reviewer Paul Friederichsen.

## References

[1] [n.d.]. https://imgur.com/a/xDe7ok1 [Online; accessed 29-April-2021].

[2] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. 2020. Learning Temporal Coherence via Self-Supervision for GAN-Based Video Generation. *ACM Trans. Graph.* 39, 4, Article 75 (July 2020), 13 pages. https://doi.org/10.1145/3386569.3392457

[3] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixe, and Nils Thuerey. 2020. Learning Temporal Coherence via Self-Supervision for GAN-based Video Generation (TecoGAN). *ACM Transactions on Graphics (TOG)* 39, 4 (2020).

[4] Debarko. 2018. RNN or Recurrent Neural Network for Noobs. https://hackernoon.com/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860 [Online; accessed 9-April-2021].

[5] Wikipedia contributors. 2021. Convolution — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Convolution&oldid=1014813607 [Online; accessed 29-March-2021].

[6] Wikipedia contributors. 2021. Deep learning — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=1009619293 [Online; accessed 13-March-2021].

[7] Wikipedia contributors. 2021. Frame rate — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Frame_rate&oldid=1010240255 [Online; accessed 13-March-2021].

[8] Wikipedia contributors. 2021. Generative adversarial network — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Generative_adversarial_network&oldid=1011184916 [Online; accessed 13-March-2021].

[9] Wikipedia contributors. 2021. List of common resolutions — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_common_resolutions&oldid=1011595540 [Online; accessed 13-March-2021].

[10] Wikipedia contributors. 2021. Machine learning — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=1011558143 [Online; accessed 13-March-2021].