

Video Frame Rate Doubling Using Generative Adversarial Networks

Ashwin R Bharadwaj¹, Hardik Gourisaria² and Hrishikesh Viswanath³

PES University, Bangalore, India
ashwinrb7799@gmail.com, hardik.g@outlook.com,
hrishi.vish@outlook.com

Abstract. This work explores a Deep Neural Network based approach to double the frame rate of a video by adding synthetically generated frames between two consecutive frames in the original video. The Neural Network is a Generative Adversarial Network that consumes two consecutive frames from the original video, interpolates them into one image, and generates a synthetic intermediate frame. Apart from the Neural Networks, Statistical and Image Processing techniques have been used to further enhance the generated frame. The final generated frame is an amalgamation of the interpolated frame and generated frame which are merged through masking. This has various applications including but not limited to remastering legacy videos with low frame rates to improve their quality. Finally, this work compares the results of the above technique against various pre-existing techniques.

Keywords: Neural Networks, Generative Adversarial Networks, Video Frames, Frame Interpolation, Frame Duplication, Digital Image Processing, CycleGAN, Histogram Matching, Structural Similarity Index

1 Introduction

Efficient video streaming is a requirement of many internet users to reduce network load and buffering time. On the server end, the streaming service could reduce the frame rate of the videos to improve streaming speed and on the consumer end, boost the frame rate to get back the original quality. If the frame rates could be boosted on the user side, then ideally, each video would only need half as many frames, reducing the size of the video on the server end.

This work presents a method to double the frame rate of a video. Lower frame rate videos can be reconstructed into videos with twice the frame rate. The process involves the addition of frames (synthetic frames) into the original video in order to increase the frame rate. Synthetic frames are produced by extracting information from the adjacent frames in the video.

There is some amount of noise present in the high frame rate video that has been constructed from the lower frame rate video [1]. Noise is introduced due to the synthetic frames not having the exact color scheme of the original frames, which leads to flickering. However, with sufficient training of the Generative Adversarial Neural Network

followed by applying the image processing techniques mentioned ahead, it is possible to minimize such noise.

Traditional methods of performing video frame rate boosting involve video frame duplication and video frame interpolation [2]. The former technique makes no improvement to the visual quality of the video while the latter generates frames that have high mean squared error with respect to the desired frames in videos with rapid movements. This work aims to address these issues.

2 Related Work

Sharma et al. [2] suggest a method to increase the frame rate of videos by interpolation. Spatio-temporal medial filtering approach was implemented to improve the quality of the interpolated frames. Goodfellow et al. [3] proposes a method to estimate generative models through an adversarial process, essentially constructing a GAN. McCarthy et al. [4] suggests that viewers were less susceptible to changes in frame rate but were more sensitive to changes in resolution. This information is used to implement a model that processes videos of known resolution. Xiao et al. [1] used a Variational Auto Encoder to boost the frame rate of videos. The problem of flickering persisted in the output. Flickering can be reduced by a significant amount using techniques described in the following sections. Feng et al. [5] proposes techniques for transmitting videos over the network by compressing them. A client-side buffer is utilized to smoothen the frame rate of the video.

Xu et al. [6] suggest a method to improve the frame rate of videos through an efficient sub pixel convolution neural network. Higher resolution is realized by combining motion estimation between the adjacent frames with the CNN. Further amplification of frame rate was done by frame interpolation. This was done by calculating the image optical flow between the frames. The method was proven to have advantages in the quality of video restoration. Clark et al. [7] suggest a method to produce longer videos of higher resolution than the original using a Dual Video Discriminator GAN. The assumption is that the pixels of each frame don't depend directly on other pixels of the video. The model utilizes two discriminators Spatial Discriminator and Temporal Discriminator to resolve the issue of generating large videos. Spatial Discriminator discerns the contents of a single frame while the temporal discriminator signals movement. Ying et al. [8] propose a method to generate previously unseen frames of a video for the purpose of predicting the future to aid intelligent agents in prediction. The process of predicting is aided by synthetic inter frame difference. Two paths were designed - Coarse Frame generator, to determine the coarse details of the future frame and Difference Guide Generator, to generate the difference image that contains complementary details. In their paper, Qi et al. [9] explore a method to reduce bandwidth consumption by reducing the frame rate of the videos and further increase them through interpolation on the receiver side. Wang et al. [10] have developed a deep learning model to denoise the images. The model performs operations such as contrast enhancement and motion

awareness. The denoising model requires tuning parameters that have large variance across various settings.

Li et al. [11] have discussed a method to upscale the frame rate of 3D videos. The color values are encoded in parallel to increase the speed of interpolation. Aigner et al. [12] have used PGGAN to predict the future frames of the video. New layers are added while training to accommodate larger datasets. Chen et al. [13] have implemented a GAN based model to translate videos. The model processes each frame and translates it to match the style of the target video. Janzen et al. [14] performed a study to understand whether frame rate was more important to viewers than latency. Frame Rate affects the ability of the viewers to recognize objects. Latency affected the viewers' perception only when the frame rate was abysmally low.

3 Proposed Method

Three methods are discussed in this paper to improve the frame rate of videos. First one uses a simple Pix2Pix GAN [15] similar to Xiao et al. [1] to improve the frame rate of videos. The second method, which is supposed to provide an improvement over GAN, uses CycleGAN [16] to reduce flickering that occurs with the usage of a regular GAN. Our proposed method combines GAN and interpolation techniques [2] to fully exploit the two, each of which is able to work well in situations that the other model failed. GAN technique is shown to work better in videos with rapid movement while interpolation has better performance when applied to videos with limited movement.

3.1 Data Collection

The videos used to train the models are required to have a high degree of variation to provide a dynamic set of frames. Failing to do so will cause the model to over-fit to that particular type of video and will not generalize well. The data collected include and is not limited to videos of nature, animated movies and racing scenes.

3.2 Dataset Preparation

Three sets of frames were extracted from each video and stored separately with two labels - input and output. The input data set contained the endpoint frames while the output contained the intermediate frames. This was done to separate the input from the desired output of the model. The two input frames were then interpolated by pixel-wise averaging. To tackle the hardware limitations of training a GAN model, the image sizes were resized to 256 x 256.

3.3 Model Definition and Training

Pix2Pix Generative Adversarial Network [3] consists of a Generator network and a Discriminator network. The Generator is trained to generate a synthetic frame from the

given input comprising two interpolated frames. The Discriminator is trained to discern if a frame is real or synthetically generated. Both of these models act as adversaries of each other and in turn, are used to improve the performance of the other one. The generator aims to generate frames that are indistinguishable from the ground truth. The discriminator compares the features of the synthetic and real images to differentiate between the two.

The generator is an U-Net [15] model with 7 encoder layers and 7 decoder layers. The discriminator has 6 Convolution Layers in a sequence and takes the output of the generator and the target frame as the input. The output of the discriminator helps identify the degree of deviation of the generated image from ground truth. Each of the convolution layers performs 2d convolution between the image and the filter. A 4x4 filter is chosen to mask the image, which is padded with zeroes. Convolution operation between two 2d matrices is given by the following equation

$$y[i, j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m, n] \times x[i - m, j - n] \quad (1)$$

wherein, y is the activation map matrix. h is the filter matrix and x is the input matrix. Convolution is done to extract important features of the image and reduce the dimensions of the said image. The resulting dimension of the activation map y after applying a single filter is given by the formula

$$A = \frac{I - F + 2P}{S} + 1 \quad (2)$$

wherein, I is the dimension of the input image, F is the filter size and P is the padding. S represents the stride, the amount by which the filter moves after each convolution operation. The final dimension of the activation map will be (A, A, n) , where n is the number of filters applied to the image. Each filter captures a specific feature of the image.

The output of each convolution layer, including the hidden layers, is batch normalized to reduce co-variance shift and to speed up the learning process. Co-variance shift refers to the shift in the mapping between input and output when the input distribution changes. By normalizing the values, higher learning rates can be used since the values don't fluctuate and conform to a distribution. It prevents over fitting by adding noise to the activation of hidden layers, similar to regularization.

Normalization is done by first determining the mean of a batch of input denoted by $I = \{x_1, x_2, \dots, x_N\}$. where x_1, x_2, \dots, x_N are activation values.

$$\mu_I = \frac{1}{N} \sum_{k=1}^N x_k \quad (3)$$

The normalized value of x is given by

$$\hat{x}_I = \frac{x_I - \mu_I}{\sqrt{\sigma_I^2 + \epsilon}} \quad (4)$$

ϵ is an arbitrary constant and σ is the standard deviation of the batch. The normalized output is finally scaled and shifted in the following manner

$$y_i = \gamma \hat{x}_I + \beta \quad (5)$$

The process of learning happens through minimizing the loss, which is calculated using cross entropy with logits. Logits are matrices that are zeroes for synthetic output and ones for real output and are denoted by x . The image matrix is denoted by z .

$$loss = z(-\log \sigma(x)) + (1 - z)(-\log(1 - \sigma(x))) \quad (6)$$

The loss is minimized with Adam Optimizer, which aims to dynamically alter the learning rate by using a weighted sum of previous gradients, momentum and Nesterov acceleration gradient. Nesterov Acceleration gradient provides a look ahead to slow down the momentum as the weights approach the target value and prevent the value from crossing the convergence value. Momentum, does the exact opposite. It increases the rate of gradient descent with each epoch, essentially, speeding up the convergence.

$$W_{t+1} = W_t - \frac{\eta V_t}{\sqrt{S_{dw} + \epsilon}} \quad (7)$$

$$V_t = \beta V_{t-1} + (1 - \beta)g_t \quad (8)$$

$$S_{dw} = \beta S_{dw-1} + (1 - \beta)g_t^2 \quad (9)$$

In equation (7), V_t represents the momentum at epoch t as a weighted average of the previous momentum and the previous gradient. ϵ is an arbitrary constant to prevent the denominator from becoming zero. β is a value between $[0,1]$, used to perform exponential smoothing. As β approaches 1, the momentum will be an average of all the data points. At lower values of β , the momentum reduces. g_t represents the gradient at epoch t .

In equation (9), S_{dw} represents the weighted average of the previous gradients, with the most recent gradient contributing the highest to the next gradient.

The model code, when written in Tensorflow, requires large amounts of training data and epochs to converge to the optimum. The model was trained for 200000 steps before it started providing acceptable results with low mean squared error and structural similarity values closer to 1. The same model was implemented using Keras and due to the internal optimizations in Keras, the model converged after 15000 steps.

The model was trained on a system with Intel i7, Hex-Core Processor, and an Nvidia GeForce 1050Ti Graphics Card.

3.4 Histogram Matching

As presented in Rakwatin et al. [17], Histogram matching is a technique that compares two Cumulative distribution functions and aims to map the source histogram to the target histogram.

The distribution of the image generated by the generator is represented as a histogram. This output is modified to fit the distribution of the previous frame.

However, this method does not significantly improve the quality of the output because the auto encoder, which is the primary component of the Generator, is expected to match the distribution of the images. The generator works by minimizing a linear combination of mean squared error and Kullback-Leibler Divergence (KL Divergence) [18] between the input and the output.

KL Divergence is a representative of how different the two distributions are. Lower values of KL Divergence indicate that the histograms are similar.

3.5 Filtering Noise in Synthetic Frames

Various filters were implemented to remove the additional noise [19] introduced in the synthetic frame generated by GAN [1]. These include a Mean filter, Median Filter and Histogram Equalization. Median Filter outperformed the other filters as indicated by MSE values.

4 CycleGAN

CycleGAN is a model that enables unsupervised training of images. It is primarily used for the translation of images [16]. The network mainly learns the mapping between images. The motivation for using CycleGAN began with the flickering that was present in the output video of the Pix2Pix GAN. The initial idea was to use CycleGAN to translate noisy or rather flickering frames into expected frames. There was however no considerable improvement in the quality of the output as determined by mean squared error and structural similarity index.

5 Region of Interest

Another approach to resolve the issue of flickering in videos is to extract the region of interest from images and use these regions to selectively pass to GAN and interpolation model.

Region of interest refers to the region that changes rapidly between frames. Any fast-moving object is considered to be in the region of interest while the background, which is generally static between images, is discarded while training.

While generating synthetic images, static regions are not translated by GAN but are interpolated from the adjacent frames, thereby reducing a significant amount of flickering that the original GAN generated. GAN however, translates the region of interest. Interpolation performs sub optimally on rapidly changing videos.

As mentioned by Wang et al. [20], if regions of interest are used instead of the entire image, the number of bits required to encode the image is lesser and consequently, the model has to learn fewer sets of weights, reducing the possibility of overfitting.

Furthermore, the transmission of videos across networks with low bandwidth becomes easier if the video is broken down into a region of interest on the server end and sent to the client, where it is converted back to its original structure.

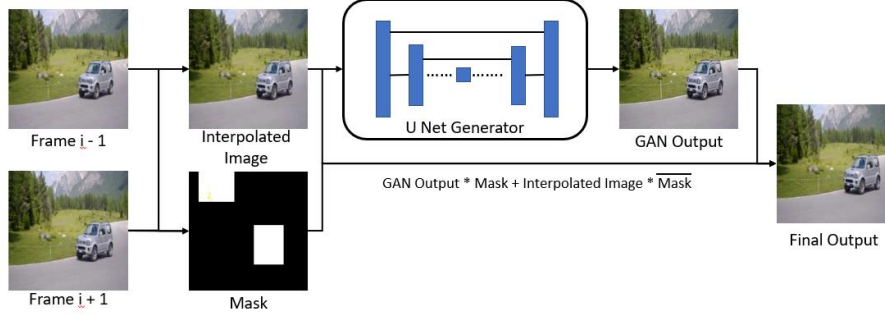


Fig. 1. Proposed Model Architecture.

6 Evaluation

Setting aside the qualitative analysis of the model's functionality by the visual inspection of the output, it is important to compute and compare the results quantitatively.

High FPS videos are downgraded to half their FPS by removing alternate frames. The FPS of the downgraded video is doubled by inserting synthetically generated frames. The output video is compared against the original high FPS video by pixel wise differencing of the generated and the expected frame. Mean Squared Error is computed for the output generated by all of the methods demonstrated. Structural Similarity Index values are determined to perform a two-fold evaluation of the output.

Mean squared error is given by the following equation

$$MSE = \frac{1}{fnm} \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^f (Y_{i,j,k} - \hat{Y}_{i,j,k})^2 \quad (10)$$

Structural Similarity Index is a measure of how similar the two images in question are. If the SSIM value of two images is 1, that implies that the two images are exactly the same. By comparing the synthetic images with real images, we can conclude whether the two are indistinguishable. For multi-dimensional images (RGB), SSIM compares luminance (L), structure (S) and contrast (C).

$$SSIM(x, y) = [L(x, y)]^\alpha \times [C(x, y)]^\beta \times [S(x, y)]^\gamma \quad (11)$$

$$L(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (12)$$

$$C(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (13)$$

$$S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (14)$$

In the above equations, μ terms refer to mean and σ terms refer to standard deviation σ_{xy} is cross-covariance. SSIM is calculated between each synthetic and ideal frame of

the video. The average value of SSIM of every synthetic frame is a representative of how similar the synthetic video is to the original one.

7 Performance and Results

Given below are three examples of the outputs used for visual inspection. As observed, the synthetic frame is pretty close to the target frame. We have used MSE and SSIM to prove that that is true.

The MSE between the ground truth video and the video generated by GAN model is lesser than the MSE obtained by frame duplication. However, frame interpolation has a lower MSE than the GAN network. This is because GAN introduces additional background noise which can be reduced by further training and focusing on the region of interest.

CycleGAN has lower training time than Pix2Pix GAN but the performance is found to be slightly worse than GAN. This difference was not discernable in most of the videos. Occasionally, higher values of MSE are obtained, but it is visually not observable.

A combination of GAN and interpolation outperforms the other models when used individually. The generated videos are visually pleasing.

The video is split into fast-moving scenes and slow-moving scenes. This is done by extracting the region of interest and determining the MSE between consecutive regions of interest. Fast-moving scenes are fed to the GAN while slow-moving scenes are fed to the interpolation model. The resulting frames are impossible to identify as being synthetically generated. The only situation where the synthetic images are imperfect is when an extremely dark frame transitions into a very bright frame.









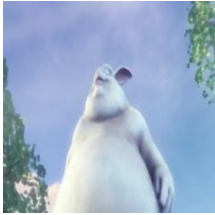
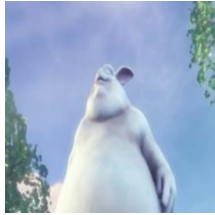
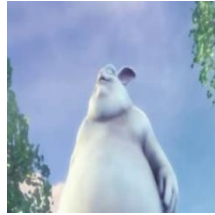
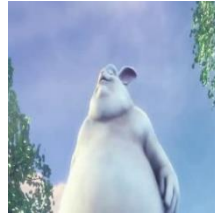
A final approach to remove flickering is to determine the difference in average RGB values of the color palette of the synthetic image and the real image. Based on this difference, the brightness of the synthetic images is dynamically increased. This difference in average color palette values is consistent and is easy to scale.

It is not necessary to focus on individual pixels because any minor discrepancies are taken care of with median filters.

Table 1. Performance of the Models

Method Used	Pixel-Wise Mean Squared Error	Average SSIM
Frame Duplication	0.01372	1.00000
Frame Interpolation	0.00918	0.88873
GAN Model	0.00936	0.84048
CycleGAN	0.00966	0.86300
GAN + Interpolation	0.00929	0.88250

Table 2. Sample Results

Frame i-1	Frame i+1	Interpolation	Final Output
			
			
			

8 Future Work

More advanced models can be implemented and tested for improving the accuracy of the synthetic frames generated. Due to hardware limitations, the input dimension is restricted to $256 \times 256 \times 3$. Future work involves optimizing this model to handle high-resolution videos without resizing them.

9 Limitations

Hardware limitations restricted the training process to small datasets. The models were repeatedly retrained on batches of small datasets. The video resolution was restricted to 256×256 for the same reason.

Acknowledgement

The authors would like to thank Dr. Natarajan Subramanyam who provided valuable insight and feedback. The work was done while the authors were with PES University.

References

1. T. Xiao, R. Puri, and G. Kesineni, "Frame rate upscaling with deep neural networks," Term Paper for CS294-129 Deep Neural Networks, Fall, 2016.
2. R. K. Sharma, R. Hazra, and A. Kasai, "Method and apparatus for increasing video framerate," Feb. 20 2001. US Patent 6,192,079.
3. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
4. J. D. McCarthy, M. A. Sasse, and D. Miras, "Sharp or smooth? comparing the effects of quantization vs. frame rate for streamed video," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, (New York, NY, USA), p. 535–542, Association for Computing Machinery, 2004.
5. W.-c. Feng, "Critical bandwidth allocation techniques for stored video delivery across best-effort networks," in Proceedings 20th IEEE International Conference on Distributed Computing Systems, pp. 56–63, IEEE, 2000.
6. M. Xu, D. Wang, and X. Du, "A video frame resolution and frame rate amplification method with optical flow method and espcn model," in Proceedings of the 2020 3rd International Conference on Image and Graphics Processing, pp. 91–95, 2020.
7. A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," arXiv preprint arXiv:1907.06571, 2019.
8. G. Ying, Y. Zou, L. Wan, Y. Hu, and J. Feng, "Better guider predicts future better: Difference guided generative adversarial networks," in Asian Conference on Computer Vision, pp. 277–292, Springer, 2018.
9. X. Qi, Q. Yang, D. T. Nguyen, and G. Zhou, "Context-aware frame rate adaption for video chat on smartphones," in Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, pp. 111–114, 2013.
10. Z. W. Wang, W. Jiang, K. He, B. Shi, A. Katsaggelos, and O. Cossairt, "Event-driven video frame synthesis," in Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 0–0, 2019.
11. Y. Li, L. Sun, and T. Xue, "Fast frame-rate up-conversion of depth video via video coding," in Proceedings of the 19th ACM international conference on Multimedia, pp. 1317–1320, 2011.
12. S. Aigner and M. Köhner, "Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans," arXiv preprint arXiv:1810.01325, 2018.
13. J. Chen, Y. Li, K. Ma, and Y. Zheng, "Generative adversarial networks for video-to-video domain adaptation," arXiv preprint arXiv:2004.08058, 2020.
14. B. F. Janzen and R. J. Teather, "Is 60 fps better than 30? the impact of frame rate and latency on moving target selection," in CHI'14 Extended Abstracts on Human Factors in Computing Systems, pp. 1477–1482, 2014.
15. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134, 2017.

16. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, pp. 2223–2232, 2017.
17. P. Rakwatin, W. Takeuchi, and Y. Yasuoka, "Stripe noise reduction in modis data by combining histogram matching with facet filter," IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 6, pp. 1844–1856, 2007.
18. T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," IEEE Transactions on Information Theory, vol. 60, no. 7, pp. 3797–3820, 2014.
19. C. Mori and S. Gohshi, "Real-time non-linear noise reduction algorithm for video.," in ICETE (1), pp. 487–493, 2018.
20. Z. Wang, A. C. Bovik, and L. Lu, "Wavelet-based foveated image quality measurement for region of interest image coding," in Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205), vol. 2, pp. 89–92, IEEE, 2001.