SimVPv2 for Image Sequence Interpolation and Extrapolation

Joseph Mills
University of Sussex
jm662@sussex.ac.uk

Abstract

Recent advancements in machine learning architecture and computational efficiencies has created a catalyst to widening access to State Of The Art (SOTA) algorithms. SimVPv2 is a powerful SOTA architecture that requires far less training and computational resource whilst maintaining very high performance on multiple spatio-temporal learning benchmarks. Here, the architecture is fitted to a bespoke problem that requires the interpolation and extrapolation of images in a sequence. It is shown that despite low volumes of training data and limited computational runtimes; that the model performs exceptionally well for interpolation and extrapolation problems in a low data environment.

1 Introduction

A machine learning system that can be used for the interpolation and extrapolation of sequences of images can be extremely useful when applied to real-world problems.

Interpolation is where images or frames are inferred between a sequence of existing frames. This functionality, for example, can be used in smoothing the movement or increasing the frame rate rate of video, replacing damaged frames or identifying intermediary states, for example when tracking weather patterns from satellite imagery which may have long periods of time between each frame [2].

Extrapolation is where the sequence of images is extended. This could also be useful in weather forecasts or identifying potential hazards in autonomous vehicles, or even editing footage in TV and film.

In this paper a machine learning model is applied to a specific problem. Images taken from MNIST [1] have been transformed into a sequence of images. The sequence depicts an autoregressive process where rotation, translation and intensity-based transformations are applied to the initial frame. Two frames are removed from the sequences and the model is evaluated on the performance on predicting those missing frames. The machine learning system is subsequently tested on an out-of-sample set and through various manipulations of the input sequence into the model.

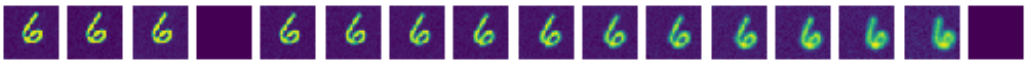

Figure 1: Example Data

This paper primarily revolves around the implementation of Simple but better Video Prediction (SimVP), a Convolutional Neural Network (CNN) architecture introduced by Tan et al. [6, 5] with the variant of the later version including a Gated Spatio-temporal Attention (gSPA) mechanism, both

MRes Advanced Artificial Intelligence (University of Sussex 2024).

of which compose into (SimVPv2) [5, 6]. The rationale of model choice is explored throughout the background and methods sections.

2 Background

Many video prediction algorithms are exceptionally computationally expensive to train. This was a key consideration when choosing an appropriate learning and inference mechanism for this task.

Until recent years, with the introduction and adoption of transformer and attention mechanisms, image sequence prediction focused largely on Recurrent Neural Network (RNN) based architectures and their variants, such as Long Short Term Memory (LSTM) RNN's. However, despite some recent developments, RNN's are non-parrellisable given the requirement to Back-Propagate Through Time (BPTT).

Other methods were explored, particular with consideration to low data environments such as Generative Adversarial Networks [7], Temporal-Difference Variational Auto Encoders (TD-VAE) [4, 3] as well as more transformer based architectures such as PredFormer [9].

SimVPv2 is the second iteration of the Simple but better Video Prediction model [6]. And evidently circumnavigates the constraints poised by many other architectures.

3 Methods

3.1 SimVPv2

SimVPv2 removes the U-Net architecture as seen in the previous version in Figure 2 and replaces the functionality with gSTA enabling a better receptive field coverage through large kernel convulsions and dilated convolutions. This change also increases the computational efficiency. When in development of the experiments of the model, SimVP achieved *22it/s* vs. *35it/s* for SimVPv2.

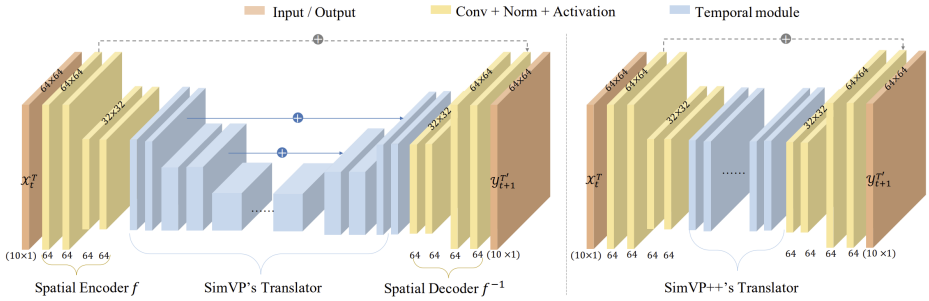

Figure 2: SimVP and SimVPv2 Framework. source: Tan et al. [6]

The standard implementation consisted of the following hyper parameters, T, T', hid_S, hid_T, N_S and N_T. The model was trained using Mean Square Error 2 and Mean Absolute Error 3. The model used a batch size of 16 and ran for 500 epochs with adam optimisation.

Dataset	***n6***	***n6-all***
T	4	4
T'	4	4
hid_S	64	64
hid_T	256	256
N_S	2	2
N_T	8	8
epoch	500	500

Table 1: Hyper-parameters of the trained models.

3.2 Data

The original dataset used for training consisted of 400 sequences of a sample of MNIST [1] of the number "6", referred to as ***n6-train***. The sequence already had the 4th and 16th image removed and these images are within ***n6-test***.

To test the model's ability to generalise; an additional dataset, ***n3-train*** and ***n3-test***, were used. This data comprised of 100 sequences and the model is evaluated, out of sample, on this data. *Note: n6-test was also not used to train the model.*

Figure 3 shows the data processing steps taken to the data to create synthetic data samples and to test the models ability in handing missing and noisy sequences.

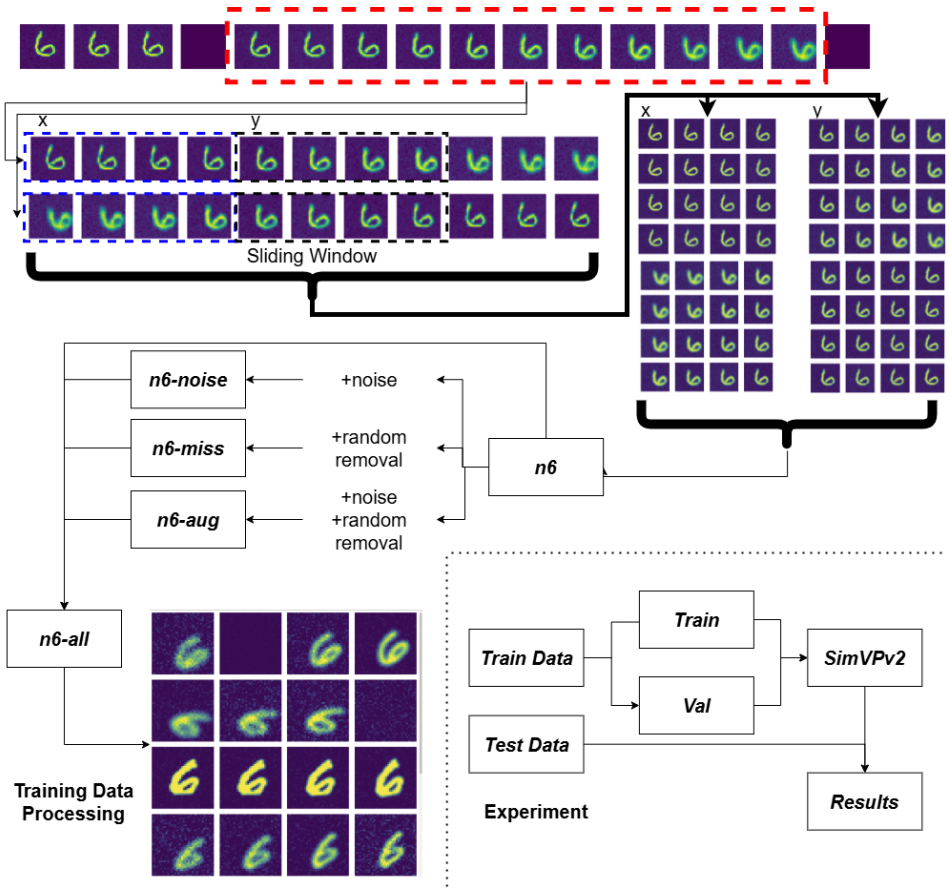

Figure 3: Training Data Processing and Simple Experiment Design Graphic

Given the limited amount of data available for training, synthetic data was generated from the original ***n6-train***.

n6-all comprises of the middle section of the original ***n6-train*** sequences both in their original order and reversed, these sequences are duplicated and separate training examples are generated consisting of additional random noise, random missing images, and random noise and missing images.

Given that the task has a complete sequence of 11 images, these were selected and partitioned into sequence of 8 images. The first 4 images act as the input sequence and the subsequent 4 images as the target, or output, sequence. 4 images each was chosen to retain as much information presented to the model whilst allowing for the possibility to increase the number of training example via a sliding window function 3.

For the test sets, ***n6-test*** and ***n3-test***, images 5-10 were reversed and images 11-15 were used to predict the intermediary image 4 and the extrapolated image 16, respectively. These sequences

undergo the same transformation as **n6-noise**, **n6-miss** and **n6-aug**. The output, or target, sequence remains unaltered from the original ***n6-test*** and ***n3-test***. The results are listed in Table 2 and Table 3.

3.3 Evaluation

For the quantitative evaluation MSE, MAE and Structural Similarity Index Measure (SSIM) were used. This is in-line with previous research and is standard for many spatio-temporal learning problems [5, 6, 8].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

where:

- n is the number of data points
- y_i is the actual value of the i-th data point
- $\hat{y}_i$ is the predicted value of the i-th data point
- μ_x is the average of x
- μ_y is the average of y
- σ_x^2 is the variance of x
- σ_y^2 is the variance of y
- σ_{xy} is the covariance of x and y
- c_1 and c_2 are two variables to stabilize the division with weak denominator. Here c_1 and c_2 are set to 0.01 and 0.03 respectively

4 Results

4.1 Quantitative Analysis

Table 2 depicts the average results against the various cuts of testing data, for image 4 and 16 combined, trained only on the original ***n6*** sequences.

As expected, the results show that the model doesn't perform incredibly well on the the synthetic data input sequences.

Test Data	N	MSE x10 ↓	MAE ↓	SSIM ↑
n6	400	48.5	11.7	0.61
n6-noise	400	136.0	16.4	0.32
n6-miss	400	77.4	14.4	0.43
n6-aug	400	157	18.5	0.26
n3	100	45.5	11.5	0.61
n3-noise	100	124.0	15.9	0.31
n3-miss	100	73.5	14.1	0.42
n3-aug	100	146.3	17.9	0.25

Table 2: SimVPv2 Results from ***n6*** training, 500 Epochs

What is of particular interest, however, is the performance of the model across ***n6*** and the out-of-sample ***n3*** are almost identical. This shows that the model isn't over-fitting to the ***n6*** training data and is capturing the auto-regressive processes that have been applied to the sequences.

Given the model was only trained for 500 epochs, sparse training data, multi-modal ability and no manual changes have been made to shape the underlying architecture to the problem; the performance metrics are very impressive, particularly evidenced in the qualitative section.

Test Data	N	MSE x10 ↓	MAE ↓	SSIM ↑
n6	400	87.7	14.8	0.56
n6-noise	400	99.9	15.5	0.54
n6-miss	400	87.7	14.8	0.56
n6-aug	400	100.0	15.5	0.54
n3	100	80.1	14.4	0.57
n3-noise	100	91.2	15.0	0.55
n3-miss	100	80.0	14.4	0.57
n3-aug	100	88.5	14.9	0.55

Table 3: SimVPv2 Results from ***n6-all*** training, 500 Epochs

After training SimVPv2 on the augmented data, ***n6-all***, Table 3 shows that the performance for all of the augmented data, except for ***n6-miss*** and ***n3-miss***, have improved. The performance dip in ***n6*** is understandable given the now equal distribution of training examples passed to the model with different synthetic augmentations. The performance decrease in ***n6-miss*** and ***n3-miss***, however, can likely be attributed to the nature of the gSTA unit of the model and because only 1 image out of the sequences of 4 was randomly removed, it can be expected that the pattern presented to the model is being effectively learned.

4.2 Qualitative Analysis

Despite promising results from the quantitative analysis. It was deemed important to visually inspect the actual prediction output by the model for the ***n6***, ***n3*** and the augmented data, here ***n6-aug*** is used, to visually inspect the models robustness to noise and missing data.

The subsequent figures comprise of the input image sequence presented to the model (top-left, top-right). Below each input sequence, from left to right: True test image, Predicted test image and the difference (Test-Predicted) for each input sequence.

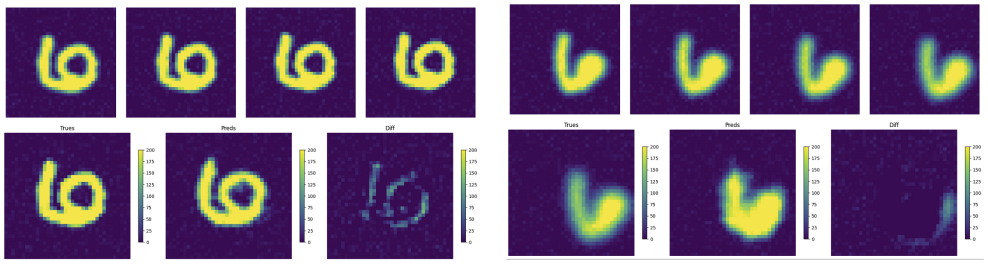

Figure 4: Example of strong SSIM scores on test for ***n6*** sequences trained on ***n6-all***

In Figure 4 and Figure 5 it appears that model performs better for sequences that have a less aggressive autoregressive process. From looking at multiple samples, this appears to be true.

Crucially, Figure 6 shows, both in the good and poor example, that the model has been able to generalise well and capture the autoregressive process of the original input image for the out-of-sampl ***n3*** data.

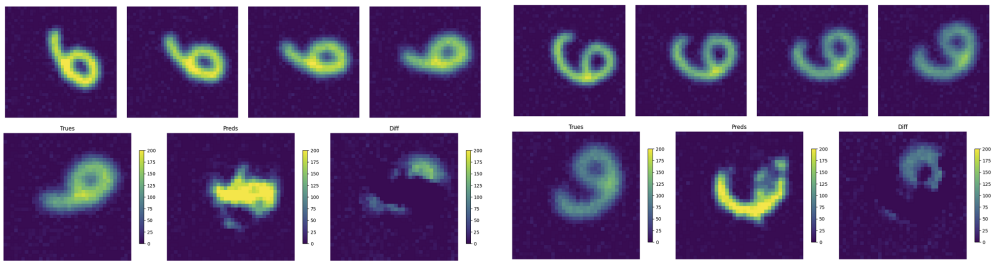

Figure 5: Example of poor SSIM scores on test for ***n6*** sequences trained on ***n6-all***

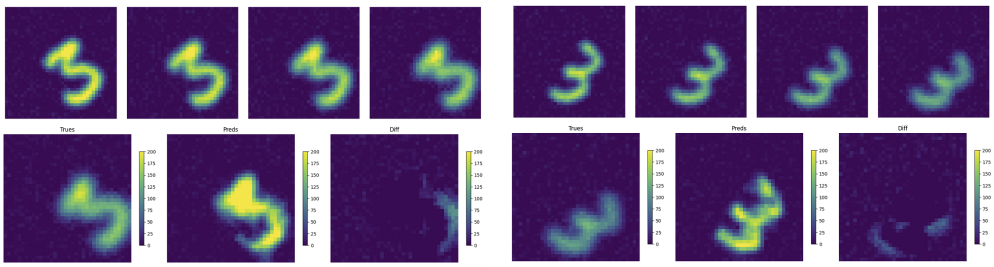

Figure 6: Good (left) and poor (right) SSIM scores on test for ***n3*** sequences trained on ***n6-all***

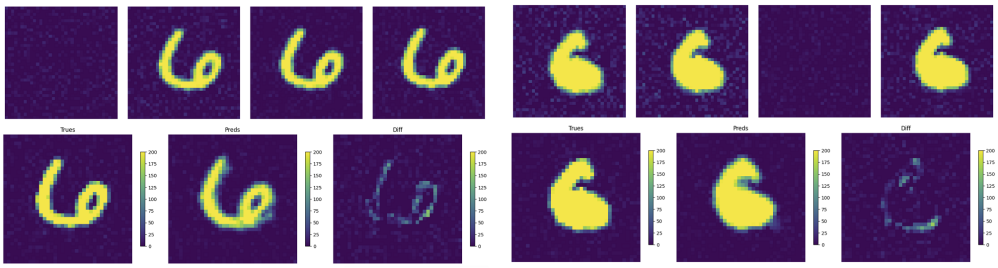

Figure 7: Example of strong SSIM scores on test for ***n6-aug*** sequences trained on ***n6-all***

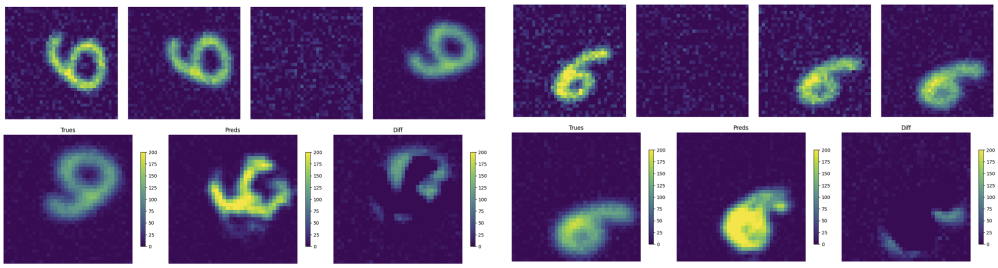

Figure 8: Example of poor SSIM scores on test for ***n6-aug*** sequences trained on ***n6-all***

As observed in the quantitative analysis, Figure 7 and Figure 8 show surprising performance, effectively capturing a lot of the information the input sequence presented, despite the missing data and noise.

5 Conclusion

SimVPv2's performance holds strong for interpolating and extrapolating image sequences by presenting the standard model with variations of the original input sequences. It was evident that the model

performed worse on predicting image 16 in the sequence which was the most transformed out of all. The performance also dipped for aggressive auto-regressive processes.

To address the performance issues it would be wise to balance the training data to reflect the different velocity of changes in the image sequences. This could be done, for example, using trigonometry on the centre-of-mass, or centroid, of the images to bring each sequence back to the centre of the 36x36 grid. In addition, finding the average pixel value of each quarter of the image could enable sufficient information, together with the scale of translation, to balance effectively.

In future work, fine-tuning the architecture and hyper-parameters together with allowing the model to train for longer will undoubtedly be able to increase the performance. The original SimVP and SimVPv2 implementation used 2000 epochs [6, 5]. It would also be prudent to explore using other frameworks such as PredFormer [9], StyleGAN [7], VAE's [4, 3] and other models that are available via OpenSTL [8] that are top performing models on spatio-temporal benchmarks, as in [6].

References

[1] Deng, Li(2012): *The MNIST Database of Handwritten Digit Images for Machine Learning Research*.

[2] Shi, Xingjian u.a.(2015): *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*.

[3] Bepler, Tristan / Zhong, Ellen D. / Kelley, Kotaro / Brignole, Edward J. / Berger, Bonnie(2019): *Explicitly disentangling image content from translation and rotation with spatial-VAE*.

[4] Gregor, Karol / Papamakarios, George / Besse, Frederic / Buesing, Lars / Weber, Theophane(2019): *Temporal Difference Variational Auto-Encoder*.

[5] Gao, Zhangyang / Tan, Cheng / Wu, Lirong / Li, Stan Z.(2022): *SimVP: Simpler yet Better Video Prediction*.

[6] Tan, Cheng / Gao, Zhangyang / Li, Siyuan / Li, Stan Z.(2022): *SimVPv2: Towards Simple yet Powerful Spatiotemporal Predictive Learning*.

[7] Sauer, Axel / Schwarz, Katja / Geiger, Andreas(2022): *StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets*.

[8] Tan, Cheng / Liu, Siyuan / Gao, Zhangyang / Guan, W. / Wang, Zedong / Liu, Zicheng / Wu, Lirong / Li, Stan Z.(2023): *OpenSTL: A Comprehensive Benchmark of Spatio-Temporal Predictive Learning*.

[9] Tang, Yujin / Qi, Lu / Xie, Fei / Li, Xiangtai / Ma, Chao / Yang, Ming Hsuan(2024): *PredFormer: Transformers Are Effective Spatial-Temporal Predictive Learners*.