Video Sequence Prediction and Generation with Advanced Deep Learning Architectures

Muhammad Anas Farooq
(21I-0813) *Department of Computer Science FAST NUCES*
Islamabad, Pakistan
i210813@nu.edu.pk

Daniyal Ahmed
(21I-2493) *Department of Computer Science FAST NUCES*
Islamabad, Pakistan
i212493@nu.edu.pk

Muhammad Usman Azam
(21I-0653) *Department of Computer Science FAST NUCES*
Islamabad, Pakistan
i210653@nu.edu.pk

***Abstract*—This project focuses on developing deep learning models to predict and generate future frames in video sequences using the UCF101 dataset. Through short input sequences, the models learn to simulate ongoing actions and thus produce coherent and realistic continuations of videos. This project develops three deep learning approaches toward this goal: ConvLSTM, PredRNN, and a Transformer-based model. The produced frames are compiled into video segments, and a user interface is developed to represent the input, predictions, and full sequences. The performance of the models is tested using metrics like Mean Squared Error (MSE) and Structural Similarity Index (SSIM).**

I. Introduction

A. Problem Definition

The task of video prediction entails predicting future frames in a given video sequence from a given number of past ones. Superimposing this to video requires going one step further, not just finding ways to generate the frames so that they are related spatially, but also time-. Since future frames need to be predicted, the model must capture spatial as well as temporal dependencies from the input sequence as continued action.

B. Importance of Video Sequence Prediction

Video sequence prediction is a promising concept with applications in the field of video compression, auto-mobiles, and robots. For example, in the autonomous driving case, object movement prediction may enhance the actions the system makes. Within video production, missing frames can be produced to help alleviate time consumption and improve the general editing process. This paper elaborates that deep learning methods, which mimics human-liked visual understanding, provide a way to enhance the authenticity of the synthesized video sequences.

C. Related Work

Several DL methods have been proposed to solve the problem of VFP, and all of them are tailored to explore the ST connections in different manners:

ConvLSTM: It will integrate the Convolutional operation with LSTM cells so as to capture both learned spatial and temporal contexts. In general, it performs reasonably well but has problems with the long dependencies because of its sequential work.

PredRNN: Expands ConvLSTM with incorporating a dedicated spatiotemporal memory layer that can address long-range dependencies of motion patterns.

Transformer-based Models: Conduct self-attention to model long term temporal relationships and have proven effective in the application of video prediction that requires complex motion and lengthy sequences.

In this work, we compare these methods for video sequence prediction tasks.

II. Proposed Approach/Implementation Details

A. Overview of Approach

Three deep learning models are implemented for video prediction: ConvLSTM, PredRNN, and Transformers. These models are the prediction models learnt on UCF101 dataset, which is one of the most frequently used datasets on action recognition and will yield the future frames of a video input given a sequence.

B. Loading and Preprocessing the Dataset

To prepare the data for model training, several preprocessing steps are performed:

Frame Resizing: This is done in order to minimize the amount of computations during training where all videos are resized to 64x64. Augmenting these smaller frames allows for faster training and often quicker prediction when computational resources are minimal.

RGB Conversion: The input size stays the same because we keep color data in the model using the RGB color space. This approach guarantees integration of all the three channels of the color space that include Red, Green and Blue so as to enable the model fully take advantage of the details of color .

Subsequently, RGB conversion retains overall color data which may be significant for analyzed tasks and requiring careful analysis of visual information.

Scaling: Pixel values are scaled by dividing by the maximum pixel value to a range of [0,1] so that the function converges during training. This step aids to make the model in a position to process the data tested and can converge faster.

Normalization: The pixel values were scaled and normalized channel-wise using the mean of [0.5, 0.5, 0.5] and a standard deviation of [0.5, 0.5, 0.5]. This brings the pixel values from the whole scale of [0, 1] (assuming that the images had been normalized to that before being passed through this layer) to [-1, 1]. The centralization and normalization of the values for every color frequency (Red, Green, Blue) around zero promote the stability and improvement of the model's learning phase.

C. Frame Selection using Stride

Another pre-processing method used is stride pre-processing technique, which is used to determine how frames in the sequence are selected and from which the 'strips' are created. It can be used to simplify cases that are approached by the model to decrease the number of frames in a video but retrieve only important frames, especially those representing the action dynamics of the video.

Stride Explanation:

Stride on the other hand means how many frames are skipped to come to each selected frame in a sequence. For instance, the stride function at 2 means that the model samples every second frame from the original video. It assists with down sampling frames and consequently decreases the intensity or demands, for instance, for many frames in a lengthy video. Choosing to skip frames does not eliminate the ability to identify important temporal changes in the sequence; thus, the training can be performed much faster. For example, if there are 30 frames in a given video, the use of stride 2 will limit the selected frames to; 1, 3, 5, 7, …., 29 thus cutting down the frame totals in half.

Example:

For the purpose of explaining we can think of a video sequence as a video with 100 frames. This is the case if the stride value is set to 4, in this case the model will take frames 1, 5, 9, 13, …; 97. As it proposed, you can change the stride to regulate the amount of temporal resolution you get your model to capture. Though a larger stride may lose some minuet temporal features, as it only processes fewer frames and accelerates the model learning.

The total number of sequences can be calculated using the following formula:

$$\textbf{Total Sequences} = \left(\frac{\text{Total Frames} - \text{Sequence Length}}{\text{Stride}}\right) + 1$$

III. Model Architectures

A. ConvLSTM Architecture

ConvLSTM is an architecture that processes sequences of spatiotemporal data by incorporating CNN at spatial dimensionality and LSTMs at the temporal dimensionality. It lies in following one or more ConvLSTM layers, and the final convolution layer for generating the predicted frames.

Key Features:

• **Spatial and Temporal Learning:** While each ConvLSTM unit covers the spatial features of the video it also maintains temporal dependences.

• **Limitation:** The ConvLSTM is good when used for short-term forecasting since its computing approach is sequential.

B. PredRNN Architecture

The proposed PredRNN is an enhancement of ConvLSTM and the incorporation of the specific developed Spatio-Temporal LSTM recurrent unit to enable the model to capture richer temporal patterns over long sequences. These enhancements contribute to Improves complex motion control and transition between frames.

Key Features:

• **Spatiotemporal Memory Flow:** The memory flow between the frames also become an advantage in learning long-term dependencies for the video domain. Enhanced Temporal

• **Modeling:** Compared with ConvLSTM, another similar structure, PredRNN is more suitable to capture long-term dependency.

C. Transformer Architecture

Transformers have been incorporated into video prediction problems since they are designed to incorporate long-range dependencies using self-attention strategies. This architecture splits the video into spatial patches and fed to multiple Transformer blocks to comprehended spatial as well as temporal correlation.

Key Features:

• **Attention Mechanism:** Self-attention can be employed to attend to this long-range context over the temporal dimension and is thus suitable for video analysis in applications that require long temporal loops.

• **Scalability:** Comparing with the ConvLSTM and PredRNN, Transformers can be easily extended to handle large scale data and long sequence.

IV. Experimental Setup and Training

A. Training Process

For all the models, the Adam Optimizer is chosen with an

initial learning rate which has been kept at 1e-3. The overfitting is handled by using of early stopping technique where regularization training is stopped the moment the validation set accuracy fails to improve. The value of the batch size was selected considering the amount of memories that would be required in the future. We also viewed how much memory a batch was taking and adjusted the batch sizes up and down to figure out which would provide the greatest performance with out using to much memory. A smaller batch size of 16 was kept constant throughout all the experiments of models.

Moreover, we included a learning rate scheduler that meant the learning rate decreases per a fixed number of epochs. It means that VERSION learning rate can increase or decrease with the increase or decrease of the loss or SSIM scores. This is to ensure that when the performance is increasing the model converges faster than when the performance just stagnates. The training durations vary for each model:

ConvLSTM: Trained for 8 epochs.

PredRNN: Trained for 20 epochs.

Transformer: Trained for 100 epochs.

V. Performance Evaluation

A. *Evaluation Metrics*

The primary metrics used to evaluate the model's performance are:

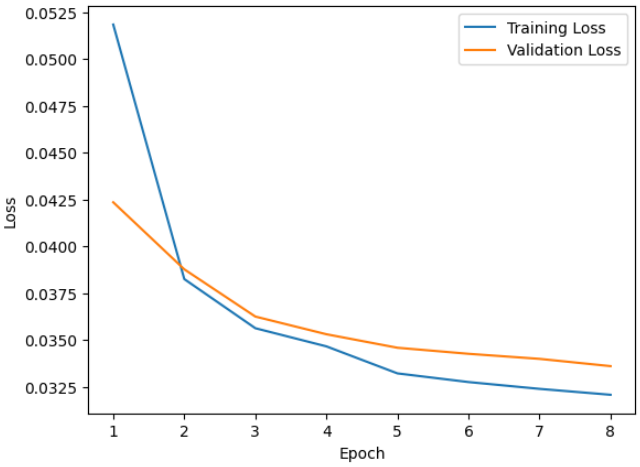
- **Mean Squared Error (MSE):** This calculates the squared error as to how exactly pixel by pixel the prediction frames are to the ground truth frames; the lower the better.

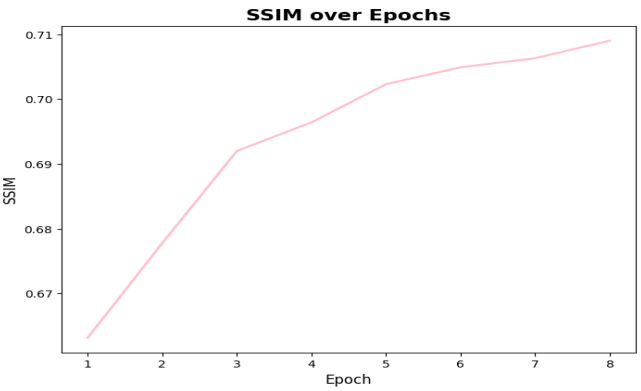
- **Structural Similarity Index (SSIM):** SSIM measures the perceptual quality of the compressed video comparing the quality of the predicted frames with the original frames in terms of luminance, contrast and structure. The measured SSIM values are higher when the predicted frames are apparently identical to the true frames.

VI. Result

A. *ConvLSTM Model Performance*

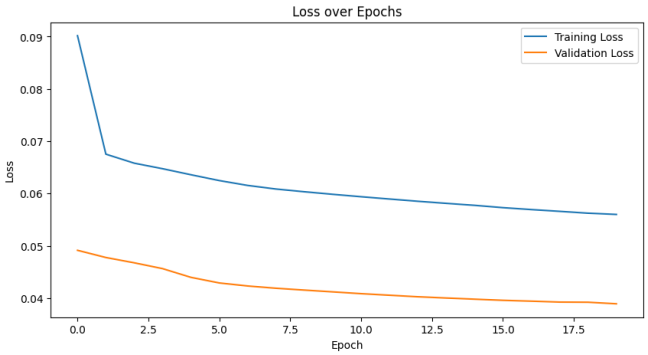
The ConvLSTM model was quite a success on short sequences of videos with fair frame predictions. In this case, the authors also argued that by maintaining a simpler model they could train the model faster, the generated frames are spatially consistent but failed to learn complex, temporal relationships.

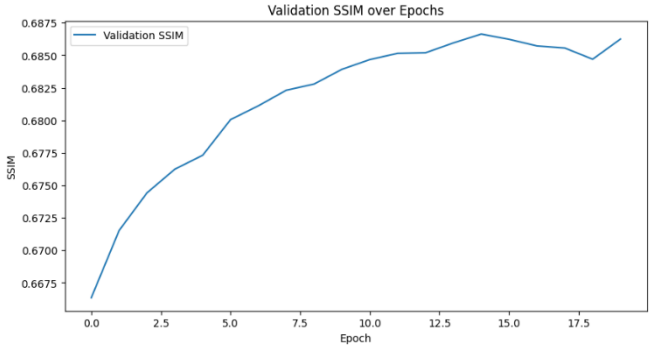


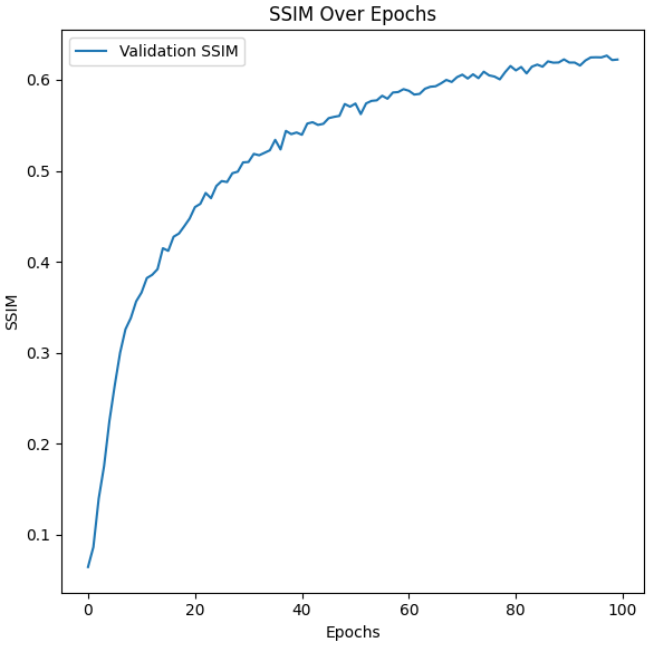


B. *PredRNN Model Performance*

For the current work, longer sequences displayed that PredRNN offered significantly better temporal coherence than ConvLSTM. The model transitioned between frames more smoothly and had better motion representation over the multiple time-step than the previous model, however, it had a higher computational overhead and training time.

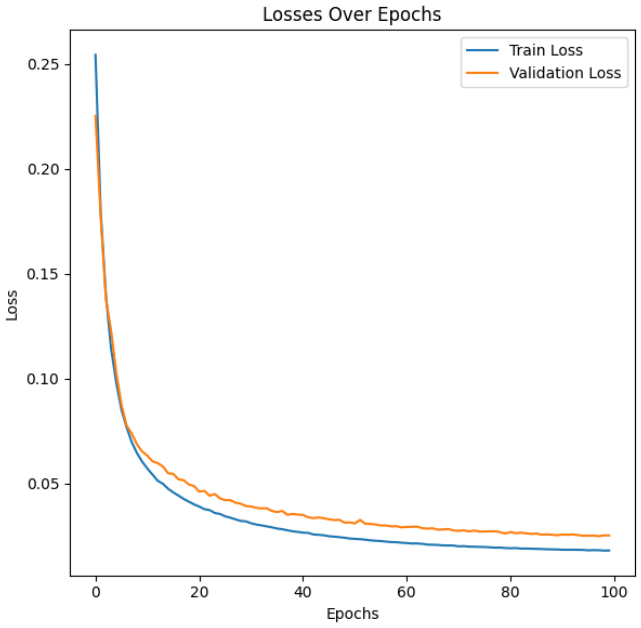






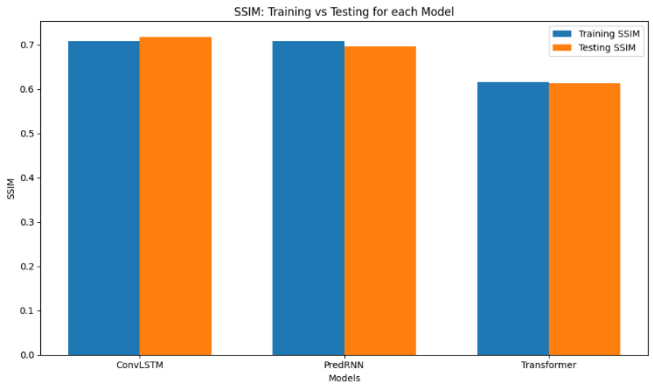
C. *Transformers Model Performance*

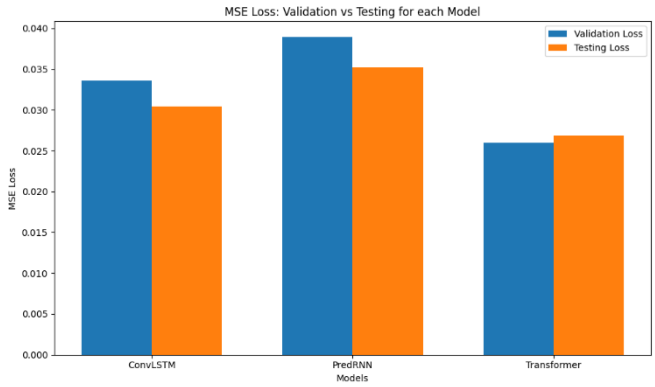
While the Transformer based models brought significant improvement in terms of learning long term dependencies it was not the best at this when this experiment was attempted. Nonetheless, the Transformer model learned to produce high-quality translations when trained to 100 epochs but failed to be stable across a number of shorter sequences. It needed a lot of computational power and training time; thus it was not very usable in many-few -resource/time situations. In fact, it learned patterns that consist with long-range dependencies; however, the overall prediction did not yield higher SSIM and visual coherency compared to other models.



D. *Comparison of Results*

Based on the performance comparison it was quite clear that PredRNN was able to perform exceedingly well in the case of temporal continuity and in the aspect of generating visually plausible frames through video sequences. Even though more computationally expensive than ConvLSTM, PredRNN was able to handle sequences of larger length and produced smoother frame transitions. On the other hand, it was shown that the ConvLSTM algorithm is appropriate for the prediction of shorter video sequences as it provides good results with low computational complexity. While both the LSTM and the Transformer model are capable of capturing long-term dependency the latter was the most computationally expensive and less accurate of the two in terms of the validation set's accuracy and training time when trained with the same constraints.





VII. Conclusion

This project also provides evidence of using deep learning models for the prediction of the video frames. Three methods were compared, namely ConvLSTM, PredRNN, and Transformer in terms of the generated future frames of the video. The proposed PredRNN achieved the highest accuracy, especially in cases with intricate temporal structures and less jerky motion in longer sequences. In comparison, short-length signals were predicted effectively with help of ConvLSTM leading to nearly optimal balance in terms of accuracy and computations. On the same note, although the Transformer models are accustomed to LSTM by capturing long-term dependencies, these models had the lowest performance in this experiment, this is due to the high request in resources and computational times. From such analysis, this work affirms that computing resources and sequence length are essential factors before choosing the suitable model to work on video prediction problems.

VIII. References

1) GitHub - Thuml/Predrnn-Pytorch: Official Implementation for NIPS'17 Paper: PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs." *GitHub*, https://github.com/thuml/predrnn-pytorch. (Accessed: 28 November 2024).
2) *Datasets & DataLoaders — PyTorch Tutorials 2.5.0+cu124 Documentation*. https://pytorch.org/tutorials/beginner/basics/data_tutorial.html. (Accessed: 26 November 2024).
3) Desai, P. (no date) *Next frame prediction using ConvLSTM*, *iopscience*. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012024/pdf (Accessed: 27 November 2024).