Predicting and Generating Video Sequences Using Deep Learning

Siyam Haider

FAST, NUCES, Islamabad, Pakistan

Abstract. This project develops a video prediction model to generate future frames from short input sequences using the UCF101 dataset, which captures various human activities. The model predicts multiple consecutive frames to create coherent video clips, simulating the continuation of actions. By learning motion patterns, the model supports applications in video synthesis, animation, and scene prediction. A user-friendly interface visualizes generated sequences, and performance is evaluated using ConvLSTM, PredRNN, and Transformer-based architectures. Metrics such as MSE, SSIM, and PSNR are employed to assess frame quality. The Transformer-based model outperforms others, achieving significantly lower MSE and higher SSIM and PSNR.

Keywords: ConvLSTM, PredRNN, Transformer, MSE, SSIM

1 Introduction

Video prediction is a challenging domain in computer vision with applications in video synthesis, animation, and predictive surveillance. This project develops a deep learning-based model to generate future frames from short input sequences using the UCF101 dataset, featuring diverse human activities. The model aims to predict multiple consecutive frames, simulating realistic motion patterns. Additionally, a user-friendly interface visualizes input and predicted frames, enhancing accessibility. Three architectures—ConvLSTM, PredRNN, and Transformer-based models—are explored and evaluated for performance.

2 Methodology

2.1 Convolutional LSTM

The methodology involves preprocessing video data, constructing a ConvLSTM model, and evaluating performance. Videos from UCF101 are preprocessed by extracting, resizing, and converting frames to grayscale. A fixed number of frames are sampled, stored as NumPy arrays, and normalized to $[0, 1]$. The dataset is split into training, validation, and test sets, with labels encoded categorically.

The ConvLSTM model comprises stacked ConvLSTM layers with batch normalization, followed by fully connected layers. It is trained using categorical cross-entropy loss and the Adam optimizer, with early stopping and checkpointing. Performance is assessed using accuracy, MSE, SSIM, and PSNR.

2.2 PredRNN

PredRNN is employed for action recognition, preprocessing videos similarly to ConvLSTM, resizing frames to 64×64 and converting to grayscale. Frames are padded if necessary and stored as NumPy arrays. PredRNN's spatiotemporal recurrent layers capture temporal dependencies, enhanced by convolutional and recurrent units, dropout, and a softmax output. The model is trained with categorical cross-entropy loss, using early stopping and checkpointing. Performance is evaluated using accuracy, MSE, SSIM, and PSNR.

2.3 Transformer-based Model

The Transformer-based model preprocesses videos by resizing frames to 64×64 and storing them as .npy arrays. A `VideoFrameDataset` class handles frame sampling, splitting sequences into input and target frames. The `VideoTransformer` model uses patch-based embeddings and Transformer layers with residual connections, predicting future frames via a decoder. Training employs MSE loss, AdamW optimizer, and validation-based checkpointing. Predicted frames are assembled into videos, with performance evaluated using MSE, SSIM, and PSNR.

3 Results

3.1 Convolutional LSTM

- Test Accuracy: 36%
- MSE: 105.1418
- SSIM: 0.0114
- PSNR: 7.49 dB

3.2 PredRNN

- Test Accuracy: 17%
- MSE: 105.7063
- SSIM: 0.0084
- PSNR: 10.34 dB

Table 1. Evaluation Metrics

Model	MSE	SSIM	PSNR (dB)
ConvLSTM	105.1418	0.0114	7.4948
PredRNN	105.7063	0.0084	10.3468
Transformer	0.0058	0.7796	22.3548

3.3 Transformer-based Model

- Training Loss: 0.0074
- Validation Loss: 0.0075
- MSE: 0.0058
- SSIM: 0.7796
- PSNR: 22.35 dB

4 Comparison of Models

ConvLSTM combines convolutional and LSTM layers to capture spatial and temporal dependencies but struggles with long-term dependencies. PredRNN improves temporal modeling with spatiotemporal memory blocks, handling longer sequences but requiring higher computational resources. Transformer-based models leverage self-attention to model global dependencies in parallel, offering superior performance but demanding significant memory and data. Table 1 summarizes evaluation metrics, highlighting the Transformer's dominance.

5 User Interface

A Streamlit-based interface enables users to upload videos or select from the UCF101 dataset, choose a model (ConvLSTM, PredRNN, or Transformer), and generate predictions. The interface displays input frames, predicted frames, and evaluation metrics (MSE, SSIM). Steps include selecting a video, processing it, and viewing predicted frames. Error handling ensures robustness against invalid inputs.

6 Insights

- ConvLSTM exhibits high MSE (105.1418) and low SSIM (0.0114), indicating poor frame quality.
- PredRNN shows similar issues, with high MSE (105.7063) and low SSIM (0.0084).
- The Transformer model achieves low MSE (0.0058), high SSIM (0.7796), and high PSNR (22.35 dB), demonstrating superior frame quality and accuracy.

7 Challenges

7.1 ConvLSTM

Challenges include out-of-memory errors during data loading, time-consuming training, and difficulties in matching tensor dimensions for sequential data.

7.2 PredRNN

Preprocessing, capturing temporal dependencies, training stability, and computational complexity pose challenges. Error propagation in long-term predictions and model interpretability are additional hurdles.

7.3 Transformer-based Model

Managing large sequences, computing self-attention matrices, and creating meaningful patch embeddings are challenging. Long training times and feature extraction from attention mechanisms further complicate implementation.