Deep Learning Project Report*

*Video Prediction Using Deep Learning: Frame Generation with UCF101 Dataset

21I-0603 Hamna Sadia Rizwan
Computer Science Department
FAST-NUCES

21I-0470 Aliza Ibrahim
Computer Science Department
FAST-NUCES

21I-0572 Kissa Zahra
Computer Science Department
FAST-NUCES

***Abstract*—This paper presents a video prediction model capable of generating future frames based on short input sequences from the UCF101 dataset. By leveraging advanced deep learning models—ConvLSTM, PredRNN, and Transformer-based models—we explore spatial-temporal patterns to simulate realistic ongoing actions. The project includes a comprehensive evaluation of these models, video generation through frame synthesis, and a user interface for visualization. Results highlight the effectiveness of each model in generating coherent video sequences, with potential applications in video synthesis and animation.**

***Index Terms*—Video Prediction, Deep Learning, ConvLSTM, PredRNN, Transformer, UCF101 Dataset, Frame Generation.**

I. Introduction

- **Background:** Introduce the problem of video frame prediction and its applications in video synthesis, animation, and scene prediction.
- **Objective**: To develop a model that predicts multiple consecutive frames, thereby generating a coherent video continuation.
- **Scope**: Use the UCF101 dataset focusing on five actions with consistent motion, preprocess the data, and implement three deep learning models for comparative analysis.

II. ConvLSTM

A. Introduction

Video frame prediction is a crucial task in action recognition, where the goal is to predict the next frame in a sequence of frames from a video. This problem is particularly challenging due to the complex temporal relationships between frames. In this section, we focus on using **ConvLSTM** (Convolutional Long Short-Term Memory) networks to model spatiotemporal dependencies in video data. ConvLSTM networks are ideal for sequence prediction tasks because they combine the strengths of convolutional neural networks (CNNs) and LSTM networks, enabling the model to capture both spatial and temporal patterns in video frames.

B. Methodology

*1) **Model Overview** :*

- The **ConvLSTM** model is a type of LSTM that integrates convolutional layers to process spatial features within video frames. ConvLSTM replaces the standard fully connected layers in LSTM with convolutional operations, enabling the model to capture spatial dependencies, making it ideal for video prediction tasks where spatial and temporal features must be learned together.
- In this project, the ConvLSTM model is designed to predict the next frame in a video sequence based on previous frames. The model processes video frames that are resized to 64×64 pixels, and the temporal dependencies are captured using stacked ConvLSTM layers.

*2) **Model Architecture** :* The ConvLSTM model consists of the following layers:

1) **Input Layer**: The input shape is defined as $(\text{None}, 64, 64, 1)$, where `None` allows for variable sequence lengths, and 64×64 is the spatial resolution of each video frame, with 1 channel representing grayscale images.
2) **ConvLSTM Layers**:
   - The first ConvLSTM layer uses 64 filters with a 5×5 kernel, which learns spatial features from the video frames. Here, we use ReLU, an activation function that introduces non-linearity and helps the model understand complex patterns. This layer returns sequences (i.e. maintains temporal information across the frames). [5]
   - The second ConvLSTM layer uses 64 filters with a smaller 3×3 kernel, continuing to learn temporal and spatial features at a finer level.
   - The third ConvLSTM layer also uses 64 filters but with a 1×1 kernel, further refining the learned spatial and temporal features.
3) **Batch Normalization**: This is applied after each ConvLSTM layer to stabilize learning and accelerate convergence by normalizing the activations in each mini-batch.
4) **Output Layer**: The output is a 3D convolutional layer (`Conv3D`) with a $3 \times 3 \times 3$ kernel and a sigmoid activation. This layer generates the predicted frame, and the sigmoid activation function is used for pixel-wise binary classification, where pixel values range between 0 and 1.

*3) **Activation Functions** :*

- **ReLU (Rectified Linear Unit)**: Used in the ConvLSTM layers, ReLU allows the model to learn complex patterns and introduce non-linearity, which helps in handling high-

dimensional video data. Because of its simplicity, non-saturating nature, and ability to effectively address the vanishing gradient problem—which results in faster training and better performance in deep neural networks—it is frequently preferred over alternative activation functions. [5]

- **Sigmoid**: Used in the final layer to output a pixel-wise probability for the predicted frame, which is between 0 and 1.

*4) **Training Setup** :*

- **Optimizer**: Adam optimizer is used, known for its adaptive learning rates and efficiency in handling large datasets.
- **Loss Function**: Binary Crossentropy loss is used, which is suitable for frame prediction tasks where the output consists of pixel-wise binary classification.
- **Batch Size**: 8, chosen for balancing computation time and memory usage.
- **Epochs**: 20 epochs, with early stopping and learning rate reduction strategies.
- **Callbacks**:
  - **EarlyStopping**: Monitors validation loss and stops training if no improvement is seen for 10 consecutive epochs.
  - **ReduceLROnPlateau**: Reduces the learning rate if validation loss plateaus for 5 epochs.

*5) **Dataset and Preprocessing** :*

- The dataset consists of videos showing human activities such as Jumping Jack, Pull-ups, Skydiving, Apply Lipstick, and Boxing Punching Bag.
- Each video frame is resized to 64 x 64 pixels and converted to grayscale. The dataset is split into training, validation, and testing sets.

C. Results

*1) **Summary of Evaluation Metrics** :* The model performance was evaluated using two key metrics: **Mean Squared Error (MSE)** and **Structural Similarity Index (SSIM)**. Below are the results for the ConvLSTM model across the five action classes.

TABLE I
MSE AND SSIM VALUES FOR DIFFERENT CLASSES EVALUATED USING CONVLSTM

Class	MSE (Mean ± Std)	SSIM (Mean ± Std)
Jumping Jack	6.2109 ± 5.7136	0.9925 ± 0.0165
Pull-ups	8.6922 ± 2.7559	0.9970 ± 0.0010
Skydiving	7.7204 ± 10.2854	0.9740 ± 0.0677
Apply Lipstick	8.2810 ± 3.2638	0.9938 ± 0.0084
Boxing Punching Bag	10.4379 ± 4.1909	0.9947 ± 0.0080

D. Evaluation

The ConvLSTM model demonstrates strong performance across all action classes. Most predictions achieved MSE values below 10, which is considered excellent. The SSIM scores were close to 1 for all action classes, indicating high structural similarity between predicted and ground truth frames. The model excelled in capturing periodic actions (e.g., Jumping Jack, Pull-ups), while performance was slightly more variable for actions with complex motion (e.g., Skydiving, Boxing Punching Bag).

TABLE II
EVALUATION OF PREDICTIVE ACCURACY AND RESULTS FOR DIFFERENT CLASSES USING CONVLSTM

Class	Evaluation
Jumping Jack	Excellent predictive accuracy and similarity.
Pull-ups	High accuracy and near-perfect SSIM.
Skydiving	Good accuracy but slightly variable results.
Apply Lipstick	Strong predictions with high SSIM values.
Boxing Punching Bag	Accurate structure; room for pixel-wise improvement.

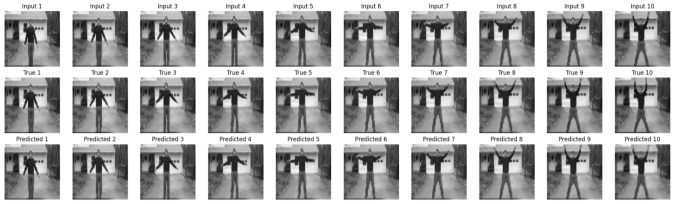

Fig. 1. JumpingJack with ConvLSTM

E. Challenges and Insights

- **Challenges**
  - The model struggled with irregular and fast motion patterns, particularly in the Boxing Punching Bag class, where MSE values was higher due to the rapid changes in the scene.
  - Training required significant computational resources, and processing large video sequences took considerable time.
- **Insights**
  - The ConvLSTM model is effective for structured actions but may benefit from enhancements to capture irregular movements better. PredRNN, with its more complex architecture, could address these challenges by modeling motion over longer time horizons.
  - The use of batch normalization helped stabilize training, particularly with deep ConvLSTM layers.

F. Conclusion

The ConvLSTM model has shown strong performance for video frame prediction, excelling in tasks with regular temporal patterns. However, there is room for improvement in capturing more complex, irregular motions. In the upcoming sections we will involve comparing ConvLSTM with PredRNN and Transformer models to explore ways to enhance prediction quality and efficiency.

III. PREDRNN

A. Introduction

Predictive RNN (Recurrent Neural Network) is a deep learning model designed for spatiotemporal sequence modelling,

specifically for the task of video prediction and sequencing.[2] Using the magic of convolutional LSTMs, spatial and temporal dependencies are both captured for more accurate future frame prediction.In this project, PredRNN was implemented from scratch to generate predicted video frames using the UCF101 dataset, showing its ability to predict motion patterns.

B. Methodology

*1) **Model Overview**:* At its core, PredRNN extends the concepts of Convolutional LSTMs by adding a mechanism to retain memory of spatial and temporal representations over multiple time steps.[2]

Unlike the traditional **ConvLSTMs** that focus on frame-level dependencies, PredRNN integrates spatiotemporal features through its unique memory-sharing strategy, allowing the model to process hierarchical information effectively[2].In this project, the PredRNN model is implemented to predict frames in video sequences from the UCF101 dataset. Each video frame is resized to 64×64 pixels, and the model uses stacked ConvLSTM layers to capture spatial and temporal dependencies over time.

*2) **Model Architecture** :* The PredRNN model consists of the following components running on 20 epochs:

- **Input Layer** : The first layer, the input layer has an input shape of (None,64,64,1). It is a single channel representing grayscale images.
- **ConvLSTM 2D** : This is for the learning of spatial and temporal features. It uses 64 filters with a kernel size (3,3) with same padding and using of relu activation function.
- **ConvLSTM 3D** : This is the second ConvLSTM layer which is 3D for the modelling of the temporal evolution of the sequence by learning spatiotemporal features across adjacent frames. It has 64 filters of kernel size (3,3,3) with a relu activation function.
- **Second ConvLSTM 3D** : This is the second 3D ConvLSTM layer for the modelling of the temporal evolution of the sequence by learning spatiotemporal features across adjacent frames. It has 64 filters of kernel size (3,3,3) with a tanh activation function.

*3) **Activation Functions** :*

- **ReLU (Rectified Linear Unit)**: Used in the ConvLSTM layers, ReLU allows the model to learn complex patterns and introduce non-linearity, which helps in handling high-dimensional video data.
- **Tanh**: Used in the final layer to output a pixel-wise probability for the predicted frame. The tanh function allows the model to predict negative and positive deviations in pixel values, which is crucial for accurately reconstructing fine details in grayscale images. It outputs values centered around zero, which helps with better gradient flow and easier learning of long-term dependencies. [1]

*4) **Training Setup** :*

- **Optimizer**: Adam optimizer is used, known for its adaptive learning rates and efficiency in handling large datasets.
- **Mean Squared Error(MSE)**:Measures the average squared difference between predicted and actual pixel values which is suitable for tasks where the goal is to predict continuous values.
- **Batch Size**: The model processes data in batches of size 8 to balance memory usage and training speed.
- **Epochs**: The model is trained for 20 epochs, meaning the entire training dataset is processed 20 times.
- **Output Sequence**: The target sequence length is set to 10 meaning the model is trained to predict the next 10 frames for every input sequence.

*5) **Dataset and Preprocessing** :*

- The dataset consists of videos showing human activities such as Jumping Jack, Pull-ups, Skydiving, Apply Lipstick, and Boxing Punching Bag.
- Each video frame is resized to 64 x 64 pixels and converted to grayscale. The dataset is split into training, validation, and testing sets.

C. Results

*1) **Summary of Evaluation Metrics** :* The model performance was evaluated using two key metrics: **Mean Squared Error (MSE)** and **Structural Similarity Index (SSIM)**. Below are the results for the ConvLSTM model across the five action classes.

TABLE III
MSE AND SSIM VALUES FOR DIFFERENT CLASSES EVALUATED USING PREDRNN

Class	MSE (Mean ± Std)	SSIM (Mean ± Std)
Jumping Jack	0.018566 ± 0.00615	0.5916 ± 0.11338
Pull-ups	0.02145082 ± 0.00894	0.56537 ± 0.12046
Apply Lipstick	0.026789 ± 0.012258	0.553737 ± 0.09516
SkyDiving	0.02116 ± 0.00866	0.48066 ± 0.12393
Boxing Punching Bag	0.036114 ± 0.012519	0.4449 ± 0.12567

TABLE IV
EVALUATION OF PREDICTIVE ACCURACY AND RESULTS FOR DIFFERENT CLASSES USING PREDRNN

Class	Evaluation
Jumping Jack	Good predictive accuracy and evaluation scores
Pull-ups	Good accuracy and variable results
Skydiving	Good accuracy but slightly variable results
Apply Lipstick	Good predictions with fair evaluation scores.
Boxing Punching Bag	Good accuracy, needs improvements on pixel accuracy

D. Evaluation

The PredRNN model demonstrates a good performance across all action classes. The predictions achieved MSE values below 10, which is considered to indicate high accuracy. The SSIM scores were however, highly variable for the action classes, indicating fluctuating structural similarity between predicted and ground truth frames. The model was successful in capturing periodic actions (e.g., Jumping Jack, apply lipstick), while performance was slightly more variable for actions with complex motion and resulted in more blurs. (e.g., Skydiving, Boxing Punching Bag).

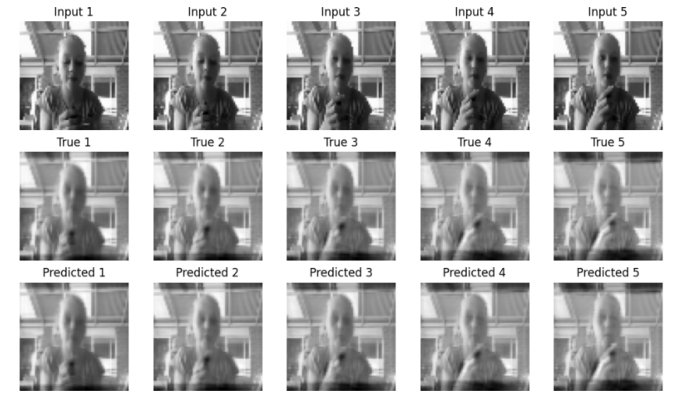

Fig. 2. ApplyLipstick with PredRNN

E. *Challenges and Insights*

- **Challenges**
  - The model struggled with irregular and fast motion patterns, particularly in the Skydiving and Boxing Punching Bag classes, and had fluctuated results between predicted frames and ground truth frames
  - Training required significant computational resources, and processing large video sequences took considerable time and GPU resources.
- **Insights**
  - While PredRNN enhances ConvLSTM and produces better MSE scores with its added retention of long term dependencies, it still struggled with more complex actions. A transformer based model could further refine and optimize this problem.

F. *Conclusion*

The PredRNN model has shown relatively strong performance for video frame prediction, excelling in tasks with repetitive motion patterns such as applying lipstick. However, there is room for further improvement in capturing more complex, irregular motions. In the upcoming sections we will involve comparing Transformer with PredRNN and ConvLSTM models to explore ways in which efficiency and accuracy, as well as quality can be enhanced.

IV. TRANSFORMER

A. *Introduction*

The Transformer model, originally introduced by Vaswani et al. in "Attention Is All You Need" (2017), has become a cornerstone in deep learning due to its ability to model long-range dependencies in sequential data. [2] They were initially designed for natural language processing, and have demonstrated exceptional performance in capturing long-range dependencies and modeling sequential data. [4]

Video frame prediction is a challenging task that requires understanding both the temporal dynamics and spatial features within a sequence of frames. Traditional models such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) often struggle to efficiently handle long-term dependencies or require significant computational resources for capturing such relationships. The self-attention mechanism of the Transformer addresses these limitations by allowing the model to focus on all frames in a sequence simultaneously, enabling efficient modeling of temporal dependencies.

B. *Methodology*

*1) **Model Overview** :* The Transformer-based model for video frame prediction leverages the power of attention mechanisms to model long-range temporal dependencies. Unlike traditional convolution-based models, which process data locally, the Transformer uses self-attention to capture global relationships across the entire sequence of frames. This model uses the following components:

- **Input Embedding**: Each video frame is embedded into a higher-dimensional space using a linear layer, and positional encodings are added to preserve the temporal order.
- **Transformer Encoder-Decoder**: The Transformer consists of both encoder and decoder layers, where the encoder processes the input frames, and the decoder generates the predicted future frames. This architecture is ideal for sequential tasks such as video prediction.
- **Output Generation**: The output of the decoder is passed through a fully connected layer to generate the predicted frames.

*2) **Model Architecture**:* The Transformer Video Predictor model is designed as follows:

- **Input Layer**: The input consists of a sequence of video frames, resized to 64×64 pixels. The frames are flattened and passed through an embedding layer. Each frame is represented by a vector, and positional encoding is added to each vector to maintain the temporal order.
- **Transformer Layers**: The encoder-decoder structure of the Transformer is composed of multiple layers. In this case, the model has two layers of both encoder and decoder, each with four attention heads. The Transformer processes the sequence of input frames and decodes them into a sequence of future frames.
- **Fully Connected Output Layer**: After decoding, the output is passed through a fully connected layer to produce the predicted video frames, reshaped to their original spatial dimensions.

*3) **Training and Inference**:* The model is trained using a sequence of input frames to predict the next set of frames. For training, mean squared error (MSE) is used as the loss function to minimize the difference between predicted and ground truth frames. During inference, the model generates predictions autoregressive, where each prediction is used as input for the subsequent step. **Training parameters are:**

- Epochs: 1000.
- Batch Size: 4.
- Device: CUDA for accelerated training.

Fig. 3. JumpingJack class with Transformer model

*4) **Dataset and Preprocessing**:*

- **Dataset:** UCF101 action recognition dataset.
- **Preprocessing Steps:**

1) Frames were extracted from videos and resized to 64x64.
2) Normalized pixel values to the range [0, 1].
3) Ensured sequence length compatibility with the model by padding or trimming videos.

C. Results

TABLE V
MSE AND SSIM VALUES FOR DIFFERENT CLASSES EVALUATED USING TRANSFORMERS

Class	MSE (Mean ± Std)	SSIM (Mean ± Std)
Jumping Jack	35.1501 ± 2.3131	0.6993 ± 0.0135
Pull-ups	62.4665 ± 5.2173	0.7702 ± 0.0468
Apply Lipstick	46.6417 ± 11.2729	0.4777 ± 0.0220
Sky Diving	79.6387 ± 2.2315	0.3704 ± 0.0134
BoxingPunchingBag	62.6181 ± 1.7763	0.5382 ± 0.0172

TABLE VI
EVALUATION OF PREDICTIVE ACCURACY AND RESULTS FOR DIFFERENT CLASSES USING TRANSFORMERS

Class	Evaluation
Jumping Jack	Acceptable with moderate MSE and decent SSIM.
Pull-ups	Good with moderate MSE and high SSIM.
Skydiving	Fair with high MSE and low SSIM.
ApplyLipstick	Fair with high MSE and low SSIM.
BoxingPunchingBag	Average with moderate MSE and low SSIM.

*1) **Summary of Evaluation Metrics** :* The model is trained using a sequence of input frames to predict the next set of frames. For training, mean squared error (MSE) is used as the loss function to minimize the difference between predicted and ground truth frames. During inference, the model generates predictions autoregressive, where each prediction is used as input for the subsequent step.

*2) **Evaluation** :* When tested in several classes, the Transformer model exhibits variable performance. Classes like "Pull-ups" and "Jumping Jack" show acceptable to good results, exhibiting dependable reconstruction quality and structural similarity, with modest MSE values and SSIM scores above 0.7. However, classes such as "Apply Lipstick," "Sky Diving," and "Boxing Bag" perform worse. These classes have higher MSE values, which indicate more significant reconstruction errors, and low SSIM scores (below 0.6), which indicate decreased structural fidelity. Given that the Transformer model struggles to generalize to some complicated or visually diversified tasks, this mismatch indicates that it needs to improve its feature learning and flexibility.

D. Challenges and Insights

*1) **Challenges**:*

- **Handling Fast and Complex Movements**: The Transformer model struggled with actions involving rapid and complex motions (such as rapid boxing punches). These high-speed motions resulted in slightly higher MSE values in certain cases.
- **Training Time and Resources**: Training the Transformer model required significant computational power, especially for processing long video sequences, and training time was relatively high compared to other architectures like ConvLSTM.

*2) **Insights**:*

- **Effectiveness in Predicting Structured Movements**: The Transformer model excelled in tasks with predictable motion patterns. Its ability to capture global dependencies via the attention mechanism allowed it to model long-range temporal dependencies well.

E. Conclusion

The Transformer model demonstrates strong potential for video frame prediction, particularly in structured actions where temporal dependencies are crucial. Its attention-based architecture effectively captures long-term temporal relationships, making it suitable for action recognition tasks. However, further improvements are necessary to handle rapid or irregular movements, and exploration of hybrid models combining Transformers with other architectures like ConvLSTM or PredRNN may lead to better performance in complex scenarios.

V. COMPARATIVE ANALYSIS

ConvLSTM, PredRNN, and Transformers exhibit distinct strengths and weaknesses in spatiotemporal modeling. ConvLSTM achieves the highest SSIM values across all classes (mean ¿ 0.97) indicating superior structural similarity and visual quality but its moderate MSE values and variablity suggest less stability for complex motions like Skydiving. PredRNN excels with the lowest MSE values (¡ 0.04), highlighting its reconstruction accuracy but its significantly lower SSIM (0.44–0.59) reflects poorer perceptual quality. While transformers are competitive in some cases (e.g. Pull-ups with SSIM of 0.77), show the highest MSE values (¿ 35) and inconsistent performance across classes, particularly struggling with complex dynamics like Skydiving. Overall, ConvLSTM offers the best balance of accuracy and quality, PredRNN is ideal for minimizing errors, and Transformers require further optimization for spatiotemporal tasks.

REFERENCES

[1] Cole Stryker , "recurrent-neural-networks" IBM [Online] Available: https://www.ibm.com/topics/recurrent-neural-networks. [Accessed: Dec. 2, 2024].

[2] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, P. S. Yu, and M. Long, "PredRNN: A recurrent neural network for spatiotemporal predictive learning," \textit{IEEE Transactions on Pattern Analysis and Machine Intelligence}, vol. XX, no. X, pp. 1–5, Mar. 2021.

[3] A. Vaswani *et al.*, "Attention Is All You Need," 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[4] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, "Transformers: 'The End of History' for Natural Language Processing?," *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 677–693, 2021, doi: https://doi.org/10.1007/978-3-030-86523-$8_41 https://doi.org/10.1007/978-3-030-86523-8\_41$.

[5] GeeksforGeeks, "Why ReLU Is Better Than the Other Activation Functions?," *GeeksforGeeks*, Feb. 14, 2024. https://www.geeksforgeeks.org/why-relu-is-better-than-the-other-activation-functions/