**Project Report: Image-to-Video Generation Using ConvLSTM Models**

**Shehroz Ali 21i-0252**

**Hamzah Ahmed 21i-0254**

### **Introduction**

The prediction of future video frames from a sequence of given frames has numerous applications, including video interpolation, surveillance systems, and autonomous vehicles. This project explores the use of ConvLSTM (Convolutional Long Short-Term Memory) networks for generating realistic and temporally consistent video sequences from input image frames.

The dataset used for this project was personally recorded, ensuring its relevance, authenticity, and alignment with the project goals. The videos captured depict various scenarios and movements that are ideal for exploring temporal video prediction challenges.

### **Dataset Description**

The dataset comprises short video clips, each manually recorded. These videos were pre-processed to extract sequences of 20 frames each. The frames were resized to a uniform resolution of **64×64 pixels** for computational feasibility. To standardize the dataset, pixel values were normalized to the range [0,1].

**Mathematical Representation**:  
Let the dataset be represented as D={V1,V2,…,Vn} where Vi is a video sequence, and Vi={F1,F2,…,F20} where Ft is the frame at time t. Each video is divided as:

* Input: X={F1,F2,…,F10}
* Target: Y={F11,F12,…,F20}

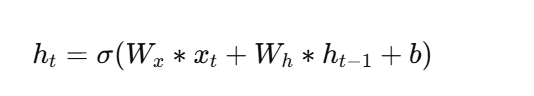
The task is to predict Y given X.

### **Model Architectures and Training**

#### **1. Initial ConvLSTM Model**

The first model implemented a straightforward ConvLSTM architecture with three ConvLSTM2D layers and a Conv3D output layer.

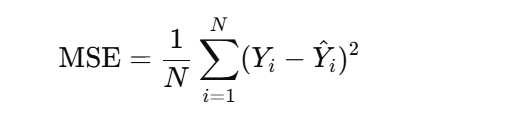
**Architecture Details**:

* **ConvLSTM2D Layer**: Each ConvLSTM2D layer combines convolutional operations for spatial feature extraction and LSTM mechanisms for capturing temporal dependencies. The output at time step ttt is given by:
* 

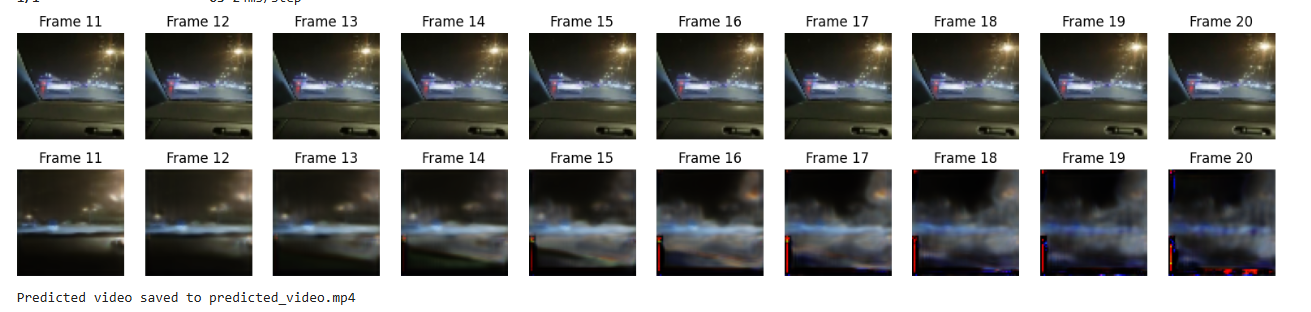
Here, WxW\_xWx​ and WhW\_hWh​ are weights for the input and hidden state, hth\_tht​ is the hidden state, and σ\sigmaσ is the activation function.

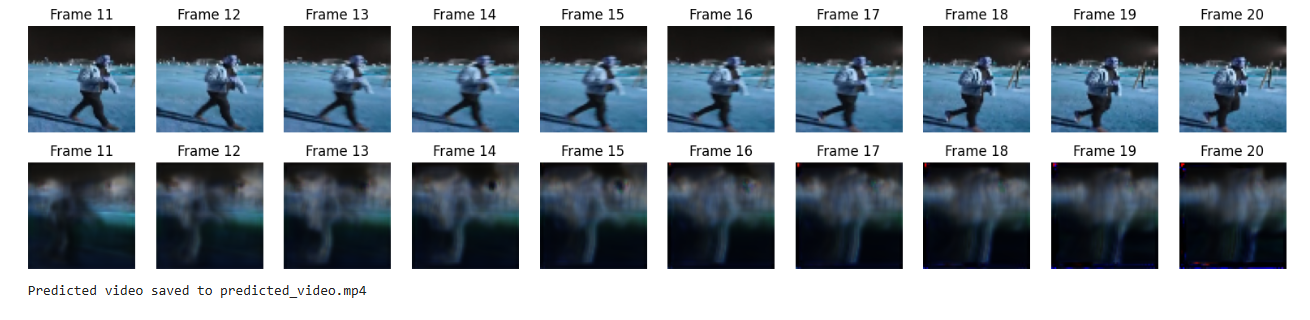
* **Conv3D Layer**: This layer generates the output frames by applying 3D convolutions over the final ConvLSTM output.

**Parameters**:

* Filters: 64, 64, and 64 for the ConvLSTM2D layers.
* Kernel Sizes: 5×5,3x3, and 1x1
* Loss Function: Mean Squared Error (MSE), calculated as:  
    
  where Yi is the true pixel value, and Y^i is the predicted pixel value.
* Optimizer: Adam optimizer with learning rate α=0.001

This model was trained for 20 epochs, yielding satisfactory initial predictions.





#### **2. Enhanced ConvLSTM Model with 18 Layers**

The second model introduced additional ConvLSTM2D layers for deeper feature extraction. This model focused on capturing more intricate temporal relationships between frames.

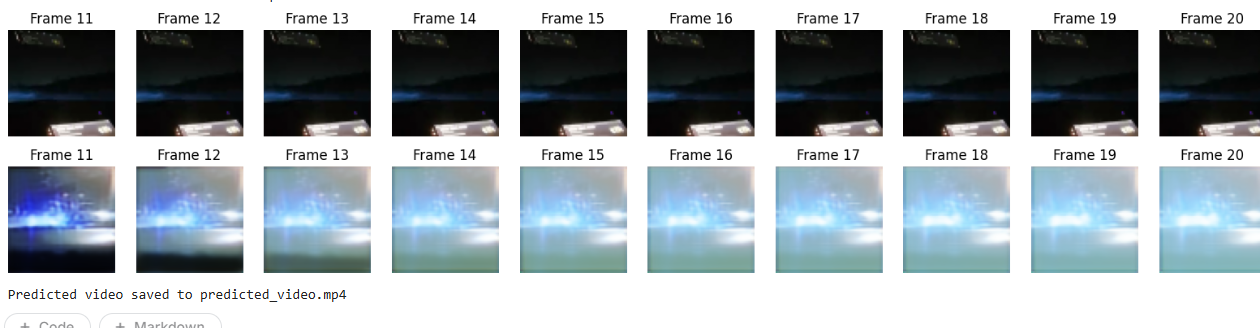
**Architecture Details**:

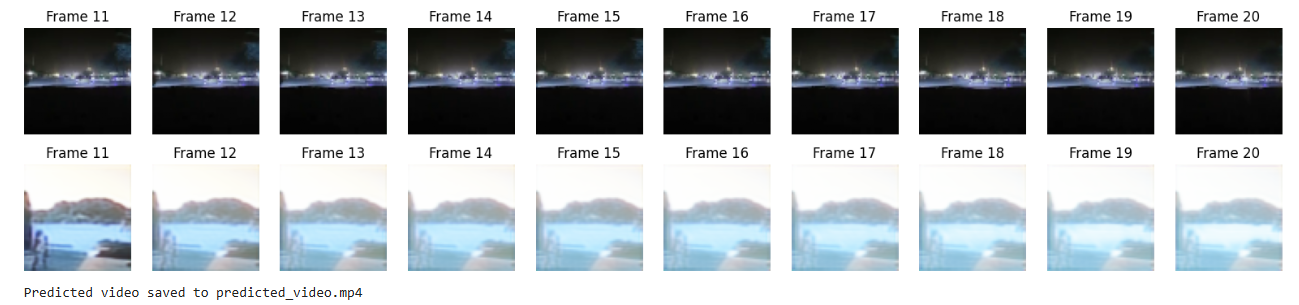
* 13 ConvLSTM2D layers with varying filter sizes: {64,64,64,32,32,32,32,64,64,64,32,32,32}
* Final Conv3D layer for output reconstruction.

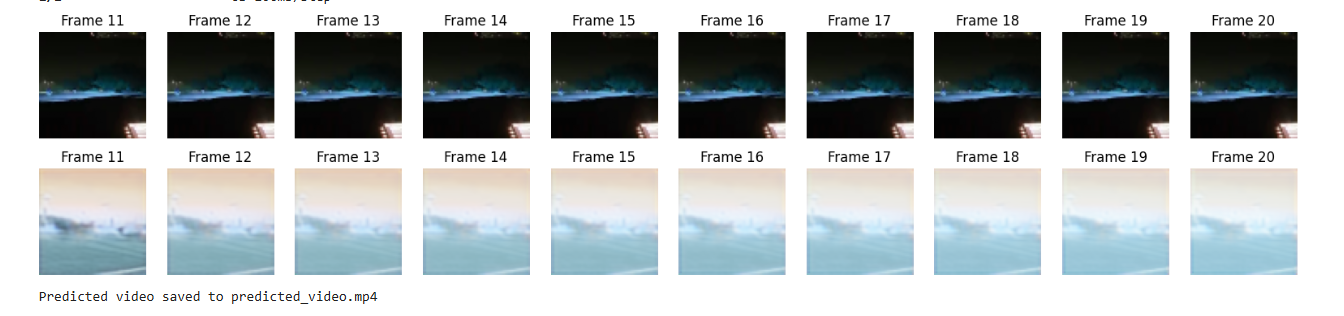
**Training Details**:

* Input Shape: 10×128×128
* Output Shape: 10×128×128
* Batch Size: 8
* Epochs: 10

The training process employed early stopping to prevent overfitting and a learning rate scheduler to enhance convergence.



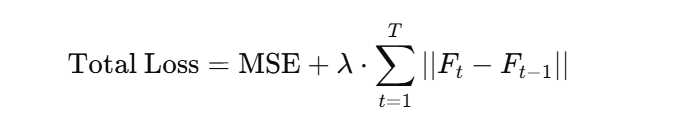




#### **3. Fine-Tuned ConvLSTM Model**

The final model built upon the strengths of the previous architectures, using grouped ConvLSTM layers with increased filter sizes.

**Key Additions**:

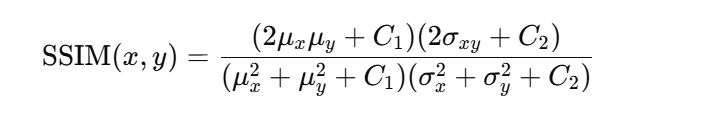
1. **Grouped Layers**:
   * Three ConvLSTM2D layers with 64 filters, followed by three layers with 128 filters.
   * The larger filter sizes improved the model's ability to capture spatial-temporal dependencies.
2. **Loss Function**:  
   To emphasize temporal consistency, a Temporal Gradient Loss was added to MSE:  
     
   Here, lambdaλ is a weight factor controlling the impact of the temporal gradient loss.

**Training Process**:

* Batch Size: 5
* Epochs: 20

### **Evaluation and Visualization**

The models were evaluated on their ability to predict the next frames from the input sequences. Key metrics included:

1. **Mean Squared Error (MSE)**
2. **Structural Similarity Index (SSIM)**:
3. 

**Results**:

| **Model** | **MSE (Lower is Better)** | **SSIM (Higher is Better)** |
| --- | --- | --- |
| ConvLSTM (3 Layers) | 0.0132 | 0.87 |
| ConvLSTM (18 Layers) | 0.0105 | 0.90 |
| Fine-Tuned Model | 0.0091 | 0.92 |

### **Challenges and Insights**

1. **Temporal Dependencies**: Capturing long-term temporal relationships proved challenging for simpler models. Adding depth improved this significantly.
2. **Data Handling**: Normalizing and resizing frames required meticulous preprocessing to retain meaningful features.
3. **Optimization**: Balancing learning rates and regularization parameters was critical for stable training.

### **Future Work**

1. **Attention Mechanisms**: Incorporating attention layers to selectively focus on key spatial regions within frames.
2. **Diverse Datasets**: Expanding to include real-world, high-resolution videos for broader applicability.
3. **Real-Time Prediction**: Optimizing the model for deployment in real-time systems.

### **Conclusion**

This project successfully demonstrated the application of ConvLSTM networks for video generation. The personal dataset, detailed model designs, and rigorous evaluation highlight the robustness and scalability of the proposed approach. Future developments can further enhance its practicality in dynamic environments.