Milestone - 2 - Artificial Intelligence Project

Video Action Recognition Using Frame-Based Deep Learning Models

Project Overview:

This project focuses on **video action recognition using pre-extracted image frames as input**, eliminating the need for video processing and allowing you to directly work with sequential visual data. You will use frame datasets derived from videos, where each action is represented as an ordered sequence of images capturing both appearance and motion cues. **Convolutional Neural Networks (CNNs)** will be applied to learn spatial features from individual frames, while **Recurrent Neural Network (RNN) variants such as LSTM and GRU** will be used to model temporal dependencies across frame sequences.

Multiple architectures including CNN-only models, sequence-based RNN models, and CNN-RNN hybrid approaches will be implemented, trained, and evaluated on the same frame dataset. The objective is to compare model performance using suitable metrics and select the architecture that best captures spatio-temporal patterns from frame sequences, providing practical insight into deep learning–based video understanding and frame-level video analytics.

Learning Objectives:

By the end of this project, you will be able to:

- Understand the fundamentals of frame-based video action recognition and spatio-temporal learning.
- Interpret the role of CNNs in extracting spatial features from image frames.
- Apply RNN variants (RNN, LSTM, GRU) to model temporal dependencies across frame sequences.
- Design and implement CNN-only, RNN-based, and CNN–RNN hybrid deep learning architectures.
- Preprocess and organize pre-extracted video frames for deep learning workflows.

- Train and evaluate multiple deep learning models using appropriate performance metrics.
- Compare model performance and justify the selection of the most suitable architecture for the dataset.
- Gain practical insight into real-world frame-based video analytics applications such as action recognition and activity analysis.

Dataset Description:

Download the UCF101 Frames Dataset from the following link:

https://www.kaggle.com/datasets/pevogam/ucf101-frames

- Select any **10 action categories** from the **UCF101 Frames dataset.**
- For each selected category, use up to **100 videos** (i.e., frame sequences derived from 100 videos) for model training and evaluation.
- If a category contains more than 100 videos, randomly sample 100 to maintain dataset balance.
- 10 categories × 100 videos = **1,000 videos**

Project Tasks:

1. **Dataset Selection & Sampling**
    - Select **10 action categories** from the UCF101 Frames dataset.
    - Sample **100 videos per category** to create a balanced dataset.
    - Organize frame sequences category-wise and video-wise.
    - Document the selected categories and sampling strategy.
2. **Data Understanding & Preprocessing**
    - Analyze the structure of frame sequences for each video.
    - Resize and normalize image frames for model input.
    - Ensure consistent frame sequence length (padding or truncation if required).

- Encode class labels appropriately.
- Split the dataset into training, validation, and testing sets.

3. **Baseline Model Implementation (CNN)**
   - Implement a **CNN-based model** to perform frame-level classification.
   - Train the model using individual frames as input.
   - Evaluate baseline performance using accuracy and loss metrics.
4. **Sequence Modeling Using RNN Variants**
   - Extract feature vectors from frames using a CNN backbone.
   - Implement **RNN, LSTM, and GRU** models to learn temporal dependencies across frame sequences.
   - Train and evaluate each sequence model independently.
5. **CNN–RNN Hybrid Model Development**
   - Integrate CNN for spatial feature extraction and RNN (LSTM/GRU) for temporal modeling.
   - Train the hybrid model using frame sequences.
   - Compare results with CNN-only and RNN-only approaches.
6. **Model Evaluation & Comparison**
   - Evaluate all models using appropriate metrics such as accuracy, loss, and confusion matrix.
   - Plot training and validation performance curves.
   - Compare models based on performance, generalization, and computational efficiency.
   - Select the best-performing model and justify the choice.
7. **Results Visualization & Analysis**
   - Visualize sample predictions for different action classes.
   - Analyze misclassifications and model limitations.
   - Summarize insights gained from model comparisons.
8. **Conclusion & Future Scope**
   - Summarize key findings from the project.
   - Discuss challenges faced during frame-based modeling.

- Suggest potential improvements such as data augmentation or advanced architectures.

Deliverables:

Source Code Repository

- Frame preprocessing scripts
- CNN model implementation
- RNN / LSTM / GRU model implementations
- CNN–RNN hybrid model code
- Model evaluation and visualization scripts

Dataset Folder

- Sampled dataset (10 categories × 100 videos)
- Organized frame sequences per video
- Train / validation / test splits

Trained Model Files

- Saved CNN model
- Saved RNN / LSTM / GRU models
- Saved hybrid model

Project Report (PDF)

- Dataset description and project objective
- Sampling strategy and preprocessing steps
- Model architectures and design decisions
- Performance comparison and analysis
- Final model selection and justification
- Limitations and future enhancements

Evaluation Rubric:

Evaluation Criteria	**Marks**
Dataset Selection & Sampling	15
Data Preprocessing & Frame Handling	15
CNN Model Implementation	15
RNN / LSTM / GRU Modeling	20
CNN–RNN Hybrid Model & Selection	20
Results Analysis & Documentation	15
Total	**100**