Milestone - 2 - Artificial Intelligence Project

Video Action Recognition Using Frame-Based Deep Learning Models

Project Overview: Video Action Recognition

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.

Architectural Approaches & Objectives

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.

CNN-only Models

Focus on spatial features within individual frames.

RNN-based Models

Model temporal dependencies across frame sequences.

CNN-RNN Hybrid Models

Combine spatial and temporal learning for comprehensive analysis.

Learning Objectives

1. **Fundamentals of Frame-Based Recognition**

   Understand spatio-temporal learning.

2. **CNNs for Spatial Features**

   Interpret their role in extracting features from image frames.

3. **RNN Variants for Temporal Dependencies**

   Apply RNN, LSTM, and GRU to model sequences.

4. **Deep Learning Architecture Design**

   Implement CNN-only, RNN-based, and CNN–RNN hybrid models.

5. **Data Preprocessing**

   Organize pre-extracted video frames for deep learning workflows.

6. **Model Training & Evaluation**

   Train and evaluate models using appropriate metrics.

7. **Model Comparison & Selection**

   Justify the best suitable architecture.

8. **Practical Insights**

   Gain understanding into real-world frame-based video analytics.

Dataset Description: UCF101 Frames

Download the UCF101 Frames Dataset from the following link:

https://www.kaggle.com/datasets/pveogam/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 x 100 videos = 1,000 videos

Project Tasks: Data Preparation

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.

Project Tasks: Model Implementation

3. Baseline Model (CNN)

Implement a CNN-based model for frame-level classification. Train and evaluate using individual frames.

4. Sequence Modeling (RNN Variants)

Extract features with CNN, then implement RNN, LSTM, and GRU for temporal dependencies. Train each independently.

5. CNN–RNN Hybrid Model

Integrate CNN for spatial and RNN (LSTM/GRU) for temporal modeling. Train with frame sequences and compare results.

Project Tasks: Evaluation & Analysis

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

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**