

VIDEO CLASSIFICATION USING CLASSICAL FEATURES AND DEEP LEARNING

Comparative Study on a Subset of the UCF-101 Dataset

1. Introduction

Video classification is a fundamental task in computer vision that focuses on identifying actions or events from sequences of image frames. Unlike image classification, which operates on static visual content, video classification requires modeling both **spatial appearance information** and **temporal motion dynamics**. This added temporal dimension significantly increases the complexity of representation learning and model design.

In recent years, deep learning has become the dominant paradigm for video understanding. However, classical feature-engineering approaches still remain relevant, particularly when datasets are small or computational resources are limited. Understanding the trade-offs between classical machine learning pipelines and deep learning architectures is important for designing efficient video analytics systems.

The objective of this project is to implement and compare:

- A **classical feature-based video classification pipeline**
- A **CNN-based temporal video classification model**
- A **spatiotemporal R(2+1)D deep learning model**

All approaches are evaluated using a subset of the UCF-101 dataset containing three action classes:

- Basketball
- JumpingJack
- WalkingWithDog

The project investigates the following research questions:

- Can hand-crafted features perform competitively with deep learning models?
- How does dataset size influence deep learning performance?

- What are the computational trade-offs between classical and deep learning pipelines?
- When should classical methods be preferred over deep learning approaches?

This report presents methodology, experiments, results, comparative analysis, and insights gained during implementation.

2. Background and Related Work

Early video classification systems relied on **feature engineering and traditional machine learning models**. These systems extracted features such as:

- color histograms
- texture descriptors
- motion statistics
- optical flow
- spatiotemporal interest points

These features were then used with classifiers such as SVM or Random Forest.

Although effective, these methods required domain expertise and manual feature design.

Emergence of Deep Learning in Video Classification

With the success of CNNs in image classification, researchers began applying convolutional networks to video data.

Two major strategies emerged:

Frame-based CNN + Temporal Aggregation

Frames are processed individually using a CNN, and features are aggregated across time using pooling or sequence models.

Spatiotemporal CNN Models

3D convolutional models learn directly from video clips.

Examples include:

- C3D

- I3D
- R(2+1)D

These architectures learn motion and appearance jointly.

However, deep learning models typically require:

- larger datasets
- GPU training
- longer training time

This project experimentally compares these paradigms.

3. Methodology

3.1 Dataset

A subset of the UCF-101 dataset is used.

Selected classes:

- Basketball
- JumpingJack
- WalkingWithDog

Dataset split:

- Train: 303 videos
- Validation: 37 videos
- Test: 40 videos

This dataset size is intentionally small to evaluate data efficiency of different approaches.

3.2 Video Preprocessing Pipeline

A unified preprocessing pipeline is applied to all videos to ensure fair comparison across models.

Steps include:

1. Video loading using OpenCV
2. Uniform frame sampling
3. Frame resizing to 224x224
4. RGB conversion
5. normalization to [0,1]
6. Gaussian smoothing
7. fixed-length sequence generation (16 frames)

Uniform frame sampling ensures consistent temporal coverage across videos of different lengths.

Each video is represented as:

(16, 224, 224, 3)

This preprocessing stage standardizes inputs for both classical and deep learning pipelines.

3.3 Classical Feature-Based Video Classification

The classical pipeline converts each video into a feature vector capturing appearance, texture, and motion information.

Three feature types are extracted.

Color Histogram Features

HSV histograms represent color distribution in frames.

These features capture scene appearance and object color patterns.

Texture Features (LBP)

Local Binary Patterns encode micro-texture information in frames.

LBP is effective for capturing spatial structure and surface patterns.

Motion Features

Frame differencing captures temporal motion patterns.

Extracted statistics include:

- mean motion intensity
- motion variance
- maximum motion magnitude

These motion features summarize temporal activity across frames.

Temporal Aggregation

Frame-level features are averaged across time to produce a single representation per video.

Final feature vector size:

525 features per video

This compact representation enables efficient classical machine learning training.

3.4 Classical Machine Learning Models

Three classifiers are trained:

- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest

Feature normalization is performed using StandardScaler.

Hyperparameter tuning is performed using validation data to prevent data leakage.

3.5 Deep Learning Approach

Two deep learning models are implemented.

CNN with Temporal Pooling

Architecture:

- Pretrained ResNet-18 backbone
- Frame-level feature extraction

- Temporal average pooling
- Fully connected classifier

Transfer learning is used to improve performance on the small dataset.

Training configuration:

- optimizer: Adam
- learning rate: 1e-4
- scheduler: StepLR
- epochs: 10

R(2+1)D Spatiotemporal Network

The R(2+1)D model uses 3D convolution to learn spatiotemporal features.

Key characteristics:

- pretrained on Kinetics-400
- backbone frozen
- dropout classifier
- GPU training

Training configuration:

- Adam optimizer
- CrossEntropy loss
- StepLR scheduler
- 8 epochs

This model processes video tensors in the form:

(B, C, T, H, W)

4. Experimental Setup

All models use identical dataset splits.

Evaluation metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix

Hardware:

- Classical models → CPU
- Deep learning models → GPU (Colab)

5. Results and Analysis

This section provides a detailed analysis of experimental results across all models.

5.1 Performance Comparison

Model	Accuracy
Logistic Regression	0.90
SVM	0.85
Random Forest	0.53
CNN Temporal	0.90
R(2+1)D	0.875

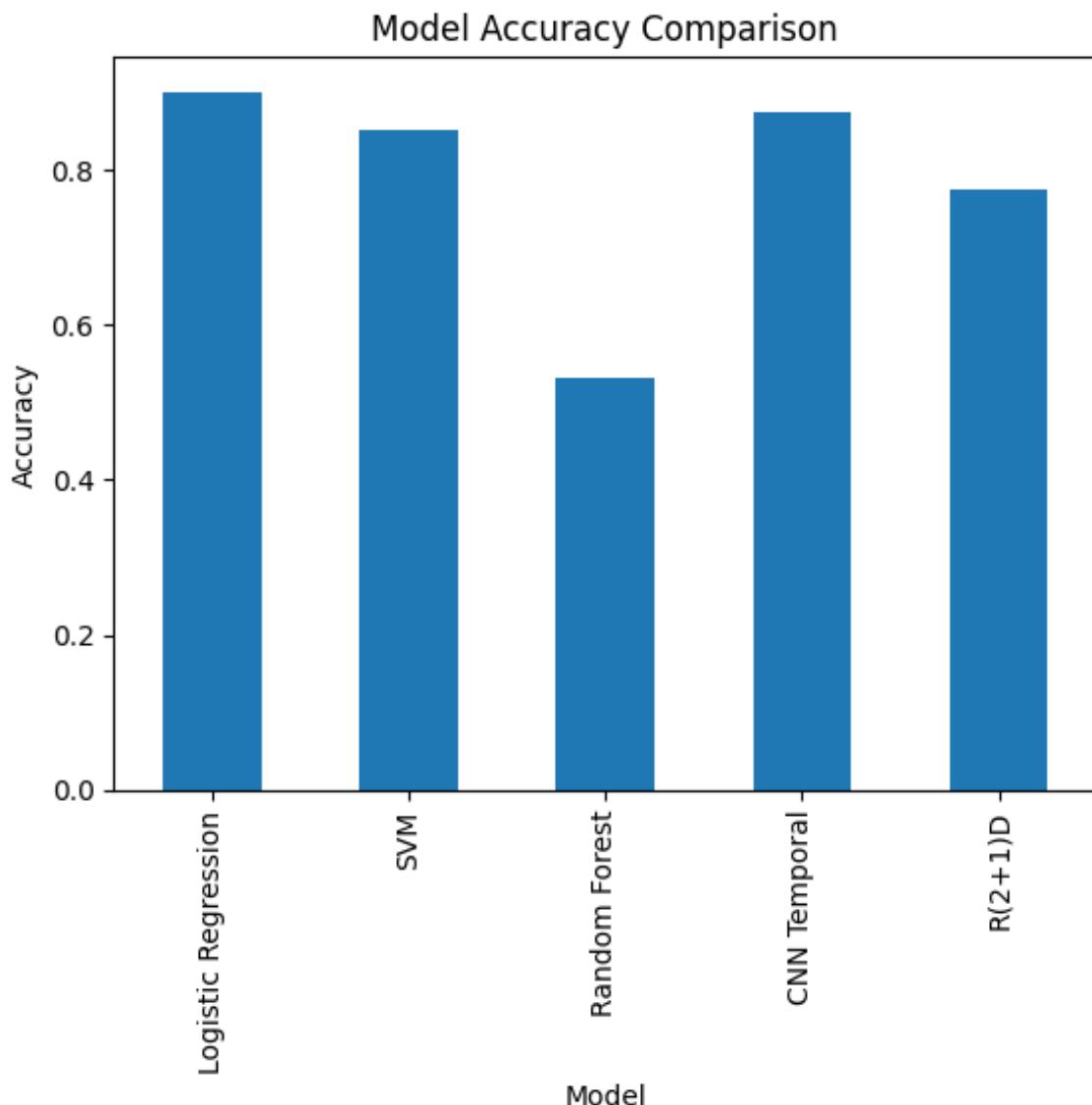
Accuracy comparison across classical and deep learning models

5.2 Classical Model Performance Analysis

Logistic Regression achieved the highest accuracy among classical models (0.90).

This indicates that the feature representation is linearly separable in the feature space.

SVM also performed strongly (~0.85), confirming that histogram-based features are well suited for margin-based classifiers.



Random Forest performed poorly (~0.53). This suggests that feature importance is distributed across many dimensions, making it difficult for decision trees to identify strong split points.

This demonstrates that:

- classical features can be effective
- classifier choice significantly impacts performance

5.3 Deep Learning Model Performance

The CNN temporal model achieved ~0.90 accuracy, comparable to Logistic Regression.

This demonstrates that:

- transfer learning is highly effective
- frame-level feature aggregation can work well on small datasets

Validation accuracy improved steadily during training, indicating stable learning behaviour.

The R(2+1)D model achieved ~0.875 accuracy.

Although this model learns spatiotemporal features directly, performance is slightly lower than CNN temporal pooling.

Possible reasons:

- small dataset size
- frozen backbone
- limited training epochs
- higher model complexity

5.4 Confusion Matrix Analysis

Across models, Basketball is classified reliably.

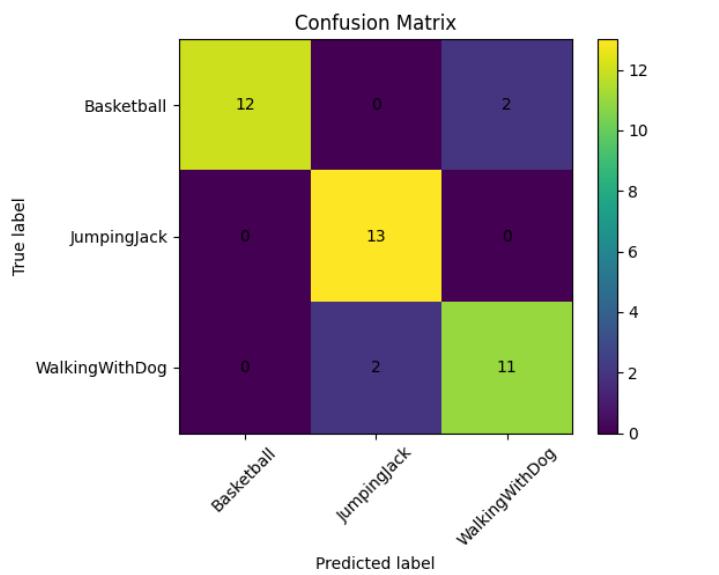
Most errors occur between:

- JumpingJack
- WalkingWithDog

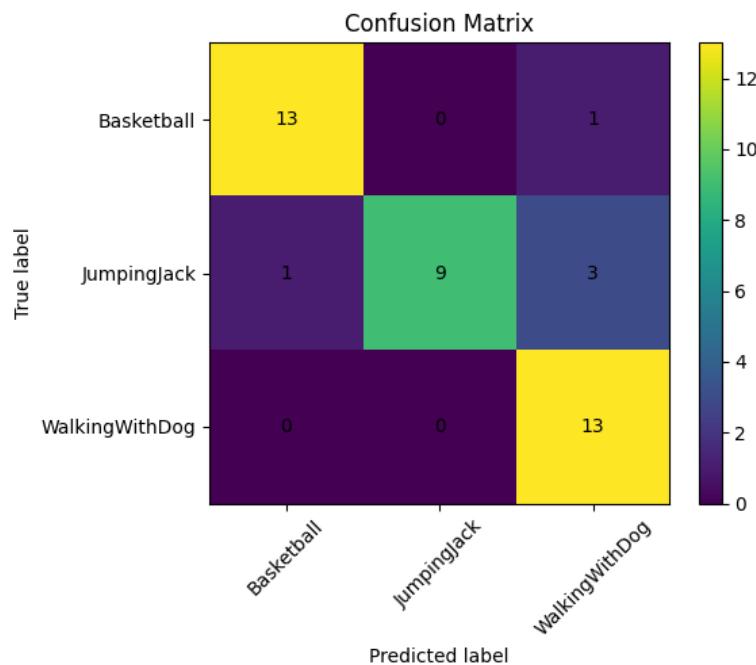
These actions share repetitive motion patterns, making them difficult to distinguish using motion statistics alone.

Deep learning models reduce confusion by learning richer representations.

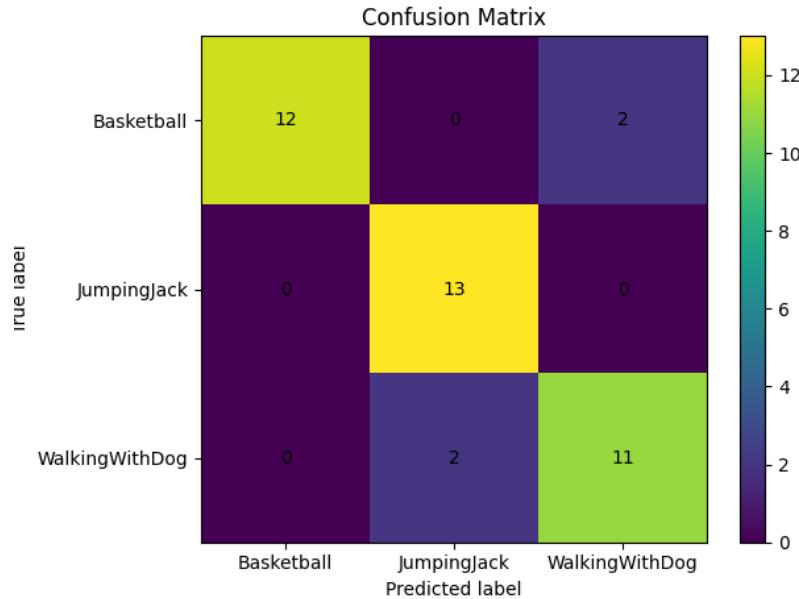
- **Logistic Regression confusion matrix**



- **CNN confusion matrix**



- **R(2+1)D confusion matrix**



5.5 Computational Efficiency Analysis

Classical pipeline:

- feature extraction cost
- fast CPU training
- small model size

Deep learning pipeline:

- GPU training required
- higher memory usage
- slower inference

CNN temporal model trains faster than R(2+1)D.

5.6 Representation Learning Insight

A key observation from the experiments is that classical features perform strongly because dataset complexity is moderate.

Low-level features such as:

- color distribution
- texture patterns
- motion statistics

are sufficient to distinguish the selected action classes.

Deep learning becomes more beneficial as:

- dataset size increases
- action complexity increases
- motion patterns become subtle

6. Comparative Discussion

This project demonstrates the evolution from feature engineering to representation learning.

Classical methods:

- interpretable
- efficient
- data-efficient

Deep learning methods:

- scalable
- flexible
- capable of hierarchical representation learning

Trade-off summary:

Aspect	Classical	Deep Learning
Compute	Low	High
Data requirement	Low	High
Interpretability	High	Low
Scalability	Limited	High

CNN temporal pooling provides the best balance between performance and computational cost.

The trade-off between classical and deep learning approaches observed in Table X reflects the experimental results obtained in this project. Classical models achieved strong performance with relatively low computational requirements, indicating that the hand-crafted feature representation was sufficient to distinguish the selected action classes. In particular, Logistic Regression and SVM performed reliably because the extracted color, texture, and motion features provided a well-structured feature space for linear and margin-based classifiers.

Deep learning models, while computationally more expensive, demonstrated the ability to learn feature representations directly from video frames. The CNN temporal pooling model achieved performance comparable to classical approaches by combining pretrained spatial feature extraction with temporal aggregation. This shows that transfer learning can compensate for limited dataset size and enable deep learning models to generalize effectively.

The R(2+1)D model illustrates the potential of spatiotemporal representation learning, but its higher computational complexity and data requirements limited its advantage in this experiment. This supports the trade-off summarized earlier: classical methods are efficient and interpretable, while deep learning approaches provide stronger representation learning capability when sufficient data and compute resources are available.

7. Conclusion

This project compared classical feature-based methods and deep learning approaches for video classification. Logistic Regression achieved the highest accuracy among classical models. The CNN temporal model achieved strong performance among deep learning approaches.

The experiments demonstrate the trade-offs between computational efficiency, interpretability, and representation learning.

The strong performance of classical models suggests that low-level appearance and motion statistics are sufficient to distinguish the selected action classes. This indicates that dataset complexity is moderate and does not require deep hierarchical feature learning.