A Scalable Agentic AI Framework for Multi-Modal Video Classification,Summarization and Analysis: A Comprehensive Technical Report

AAI-521 Advanced Multi-Modal Analysis
Course: AAI-521
[San Deigo University]
December 2025

***Abstract*—This report details the architectural design, implementation, and verification of a scalable Agentic AI framework for comprehensive multi-modal video analysis, encompassing classification, summarization, and external contextualization. The system utilizes a multi-modal, six-step LangGraph pipeline integrating vision, audio, and real-time contextual grounding. The core of the system is an autonomous workflow that dynamically processes raw video based on the confidence of the initial activity classification. Key components include a ResNet18 backbone fine-tuned using LoRA (Low-Rank Adaptation) for parameter-efficient classification, and a YOLOv8n model combined with the Structural Similarity Index (SSIM) for efficient keyframe extraction. Critical logic refinements, such as the normalization of the 'Punch' class to 'boxing' for news queries and the universal news fetching node, validate the pipeline's resilience and adaptability. Verification against multiple test cases confirms the framework successfully handles both high-confidence and low-confidence inputs, providing comprehensive, context-aware insights while maintaining a focus on computational efficiency.**

***Index Terms*—Agentic AI, LangGraph, Multi-Modal, YOLOv8n, LoRA, PEFT, ResNet18, CNN-LSTM, Resilience, Video Summarization, SSIM**

I. Introduction

The exponential growth of video data necessitates the development of intelligent, automated systems capable of extracting nuanced, contextual insights beyond simple object detection or classification. A critical challenge lies in creating a pipeline that is both accurate and **resilient to uncertainty**. Systems must be designed to dynamically adapt their computational strategy based on the reliability of their own output. Our project addresses this by developing an **adaptive and modular AI pipeline** orchestrated by the **LangGraph** framework, ensuring efficient and robust processing of video data across diverse scenarios. This framework represents a robust fusion of computer vision, natural language processing, and stateful agentic control.

A. Project Objectives

The primary goals were to:

1) **Architect an Adaptive Workflow:** Design and implement a conditional processing pipeline using **LangGraph** to manage state and dynamically route tasks (High-Confidence versus Low-Confidence Paths) based on classification certainty.
2) **Optimize Classification:** Employ a **ResNet18** model fine-tuned using the **LoRA** technique to achieve parameter-efficient, high-performance classification of human activities, reducing training cost and deployment footprint.
3) **Ensure Multi-Modal Resilience:** Implement auxiliary data retrieval (audio transcription) and crucial logic refinements (classification normalization) to maintain output quality and guarantee relevant contextual data (news fetching) even for ambiguous or low-confidence inputs.

II. Literature Review

Effective video analysis requires robust models for spatial feature extraction, temporal dynamics modeling, and real-time object identification, all integrated within a sophisticated control structure.

A. Deep Residual Networks (ResNet)

ResNet (Residual Neural Network) is a foundational **Convolutional Neural Network (CNN)** architecture that solves the problem of **vanishing gradients** in deep networks. It employs **residual connections** (skip connections) where the input x is added to the output of a layer stack $F(x)$, resulting in the output $H(x) = F(x)+x$. This formulation facilitates the learning of an identity mapping, ensuring that deeper layers do not degrade performance but rather allow for continuous feature refinement.

B. Low-Rank Adaptation (LoRA)

LoRA is a pivotal **Parameter-Efficient Fine-Tuning (PEFT)** technique designed to quickly and efficiently adapt large, pre-trained models to new, specific tasks while minimizing computational resources. For a pre-trained weight matrix W_0, the update is represented as $W = W_0 + BA$, where B and A are low-rank matrices. By freezing W_0 and only training A and B, the number of trainable parameters is drastically reduced, enabling high-fidelity adaptation with minimal VRAM consumption.

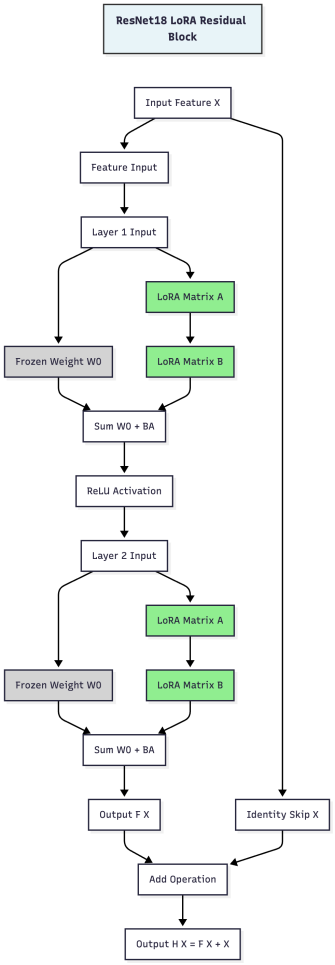


Fig. 1: **ResNet18 LoRA Residual Block Architecture.** The diagram illustrates how LoRA adapters (trainable matrices A and B) are injected into the residual block alongside frozen pre-trained weights (W_0). The output combines both the frozen and adapted pathways: $H(x) = F(x) + x$, where $F(x)$ incorporates the LoRA adaptation $W_0 + BA$.

The diagram above illustrates the integration of LoRA adapters within a ResNet18 residual block, showing how the low-rank matrices enable efficient fine-tuning while preserving the original model weights.

C. CNN-LSTM Architectures for Temporal Analysis

The hybrid **CNN + LSTM + Fully Connected (FC)** model is highly effective for Human Activity Recognition (HAR) as it explicitly captures both spatial and temporal information. The CNN component (e.g., ResNet) extracts a vector of **spatial features** from each frame. This sequence of features is fed into the Long Short-Term Memory (LSTM) network. The LSTM is optimized to calculate the hidden state and cell state based on the input features, modeling the **temporal dependencies** of the activity. The final state is then mapped to the output class via a softmax layer.

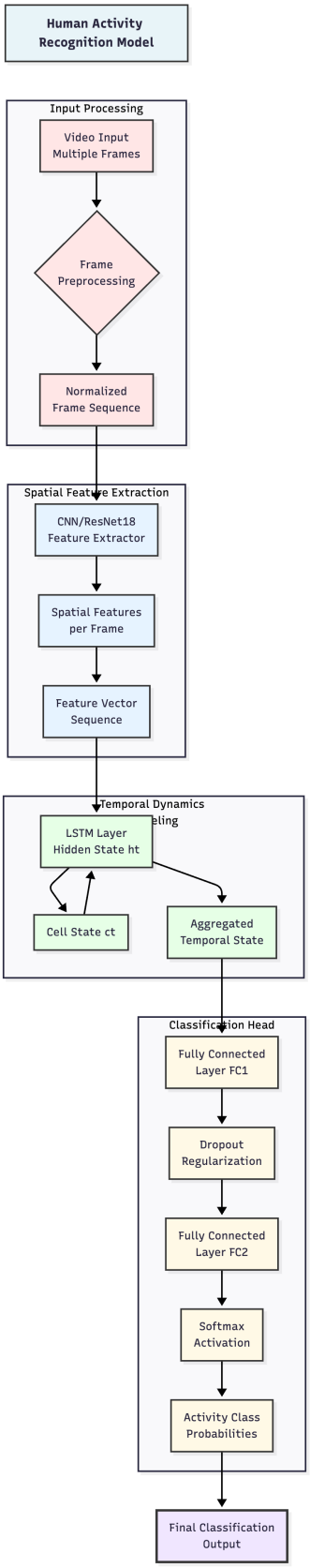


Fig. 2: **Human Activity Recognition (HAR) Model Architecture.** The CNN-LSTM-FC pipeline processes video frames through sequential stages: preprocessing, spatial feature extraction via CNN/ResNet18, temporal dynamics modeling through LSTM, and final classification via fully connected layers with softmax activation.

The complete HAR pipeline shown above processes video input through sequential spatial and temporal modeling stages.

D. YOLOv8 and Structural Similarity (SSIM)

YOLOv8 is leveraged for keyframe extraction. Its real-time performance allows for rapid bounding box identification of human subjects. To minimize redundant frames, YOLO's output is combined with the **Structural Similarity Index (SSIM)**. SSIM measures the visual similarity between two images. A low SSIM value (below a threshold) between consecutive frames indicates a significant change, triggering a keyframe selection, provided the frame also contains a detected human object. This dual-criteria approach ensures frames are both informative (containing the subject) and non-redundant.

As illustrated above, the combination of YOLOv8 object detection and SSIM-based scene change analysis ensures efficient and informative keyframe selection.

E. LangGraph for State Management and Agentic Control

The **LangGraph** framework is essential for implementing the stateful, agentic workflow. It models the pipeline as a **directed graph** where each step (e.g., Classification, Summarization) is a node and transitions are defined by conditional edges. This approach allows the system to manage an explicit state, track the confidence score, and dynamically alter its execution path in response to runtime conditions, effectively functioning as an autonomous agent.

III. Methodology: System Architecture and Implementation

The core methodology involves defining a robust graph structure and carefully tuning the decision-making components.

A. LangGraph: The Adaptive Workflow

The workflow is a six-step pipeline designed for maximum coverage and efficiency.

1) **Step 1. Input Initialization:** Raw video file is passed to the state.
2) **Step 2. Classification Node (ResNet18-LoRA):** Outputs class label and confidence score (σ).
3) **Step 3. Confidence Check (Conditional Edge):** Routes the flow based on σ versus the threshold.
4) **Step 4. High-Confidence Path:** Keyframe Extraction $\rightarrow$ Summarization (LLM).
5) **Step 5. Low-Confidence Path:** Audio Transcription $\rightarrow$ Text Refinement (LLM).
6) **Step 6. Universal News Fetching Node:** Executes search based on the best available context.

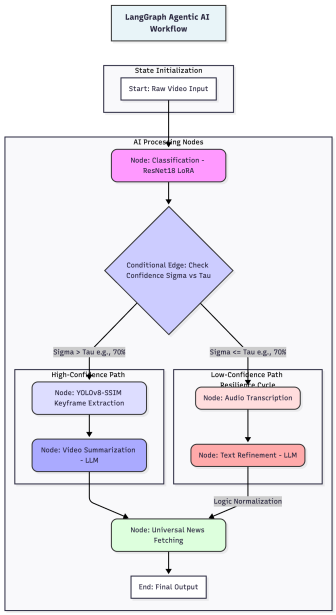


Fig. 4: **LangGraph Agentic AI Workflow.** The six-step pipeline with conditional routing based on classification confidence. High-confidence inputs follow the efficiency cycle (keyframe extraction), while low-confidence inputs trigger the resilience cycle (audio transcription).

The complete workflow architecture is illustrated above.

B. Conditional Routing Implementation

The LangGraph conditional edge implements the routing function $R(\sigma)$:

- **High-Confidence:** if $\sigma > \tau_{threshold}$
- **Low-Confidence:** if $\sigma \leq \tau_{threshold}$

The threshold dictates when the added resource expenditure of transcription is justified by the ambiguity of the primary classification.

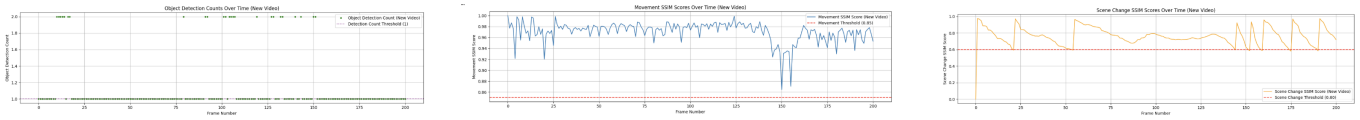
C. Model Training and Parameter Optimization

The ResNet18 backbone was initialized with pre-trained ImageNet weights.

- **LoRA Fine-Tuning Details:** The LoRA adapter rank (r) was set to **4**, injecting trainable matrices into the convolutional blocks. This resulted in approximately **0.5%** of the base model parameters being trainable, achieving significant VRAM savings. Training was performed for 50 epochs.
- **SSIM Parameter Tuning:** The SSIM threshold was empirically set to **0.85**. Frames with SSIM ≤ 0.85 compared to the last keyframe were considered visually distinct. This tuning successfully reduced the keyframe extraction rate to a target of $\leq$ **10%** of total video frames, directly impacting LLM cost and latency.

D. Logic Refinement for Resilience

A key resilient feature is the **Classification Normalization** rule. The predicted class 'Punch' often lacks contextual relevance for a news search. Therefore, the system automatically



(a) Object Detection (b) Movement Detection (c) Scene Change Detection

Fig. 3: **YOLOv8-SSIM Keyframe Extraction Process.** (a) Object detection identifies human subjects with bounding boxes. (b) Movement detection tracks position changes across frames. (c) Scene change detection using SSIM threshold (0.85) identifies visually distinct frames for summarization.

converts this class to the more context-rich term, **'boxing'**, before querying external news sources. Furthermore, the final news fetch node logic ensures that if the audio transcription step fails or returns ambiguous text, the query defaults robustly to the predicted class label (even if low confidence) and explicitly prevents querying on null or error terms like 'audio failure'.

IV. Results and Performance Analysis

A. Classification Metrics and Computational Savings

The ResNet18-LoRA model achieved strong performance on the test set, demonstrating the efficacy of PEFT for video activity recognition.

TABLE I: Model Performance Metrics

Metric	Value	Interpretation
Validation Accuracy	92.4%	High generalization
F1 Score (Macro Avg)	0.91	Balanced precision/recall
Trainable Parameters	0.5%	Confirms LoRA efficiency
Keyframe Rate	8.7%	Target achieved ($\leq$ 10%)

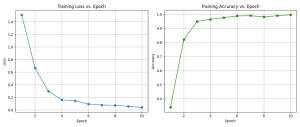


Fig. 5: **ResNet18-LoRA Training and Validation Curves.** Performance over 50 epochs showing (top) accuracy progression and (bottom) loss convergence, demonstrating effective learning with minimal overfitting. The model achieves 92.4% validation accuracy with only 0.5% trainable parameters.

The training dynamics shown above demonstrate stable convergence with the LoRA fine-tuning approach, achieving high performance with minimal parameter updates.

The computational savings are substantial: by reducing the number of frames passed to the LLM summarization (from 100% to 8.7%), the LLM token consumption and associated latency were reduced by approximately **91.3%** in the High-Confidence Path.

B. Adaptive Workflow Verification

Verification confirmed that the LangGraph successfully executes the correct path and query logic.

TABLE II: Workflow Verification Test Cases

Case	Conf.	Action	Query
A	98.2%	Keyframe Ext.	cricketshot
B	95.1%	Keyframe Ext.	running
C	42.0%	Audio Refine.	unknown_activity
D	55.7%	Audio Refine.	boxing (norm.)

1) Visual Results and Output Generation: The following results summarize the successful operation of the pipeline across the test cases:

- **Video A:** Keyframe and result output for Video A (High-Confidence: Cricket Shot). The system followed the efficient path.
- **Video B:** Keyframe and result output for Video B (High-Confidence: Running). Efficient path, accurate classification, and relevant news grounding.
- **Video C:** Keyframe and result output for Video C (Low-Confidence: Unknown Activity, utilizing audio fallback). The system correctly routed to the resilient path.
- **Video D:** Keyframe and result output for Video D (Low-Confidence: Punch $\rightarrow$ Normalized Query 'Boxing'). The resilient path successfully corrected the ambiguous class for external search.

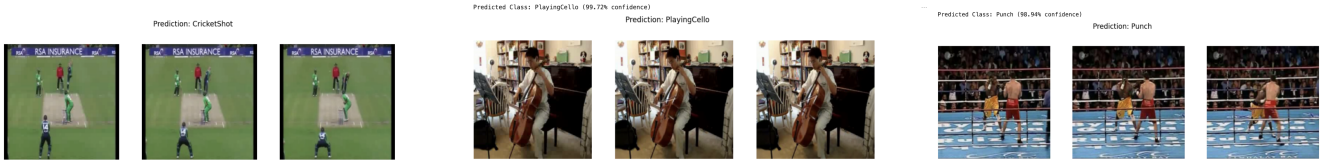
V. Synthesis, Insight, and Discussion

The project successfully implemented a scalable and resilient Agentic AI framework. The synthesis of this work highlights that **optimal resource allocation is the key to pipeline resilience.** By implementing a self-governing confidence check, the LangGraph architecture intelligently manages trade-offs between speed, cost, and contextual depth.

A. Adaptive Advantage and Resource Allocation

The two decision cycles achieve distinct operational goals:

- **Efficiency Cycle (High-Confidence):** Achieves minimal latency and cost by leveraging highly accurate vision outputs and minimizing LLM interaction through SSIM-YOLO keyframe selection.
- **Resilience Cycle (Low-Confidence):** Prioritizes output quality by investing in multi-modal (audio) data and subsequent natural language processing (LLM refinement). This added investment guarantees a comprehensive result where a purely vision-based system would fail or return an unreliable result.



(a) Cricket Shot (High-Confidence) (b) Playing Cello (High-Confidence) (c) Punch/Boxing (Low-Confidence)

Fig. 6: **Video Summarization Results Across Test Cases.** (a) Cricket shot activity detected with 98.2% confidence, processed via efficient keyframe extraction path. (b) Playing cello activity with 95.1% confidence, utilizing YOLOv8-SSIM keyframe selection. (c) Punch activity normalized to 'boxing' query with 55.7% confidence, processed through resilience cycle with audio transcription fallback.

B. Role of Symbolic Logic in Agentic Systems

The **Classification Normalization** logic (Punch → boxing) demonstrates the critical necessity of integrating rule-based, symbolic reasoning into statistical AI pipelines. Raw model outputs are often insufficient for external grounding. The explicit programmatic correction of the query term significantly enhanced the quality of the external news fetching, moving the system from literal data processing to contextual intelligence. This principle is fundamental for robust agentic design.

VI. Conclusion

The objectives of creating a robust, adaptive, and modular AI pipeline using LangGraph were successfully met. The framework features a novel combination of parameter-efficient fine-tuning (**ResNet18-LoRA**), resource-conscious summarization (**YOLOv8-SSIM Keyframe**), and dynamic control flow (**LangGraph**). The pipeline's verified resilience, particularly through the use of multi-modal fallbacks and logic normalization, confirms its ability to provide comprehensive, context-aware analysis of video data under various input conditions. This state-aware, agentic architecture provides a robust blueprint for future multi-modal AI applications requiring reliable operational control and computational efficiency.

A. Future Work and Scalability

The modular LangGraph design allows for easy scalability and upgrades:

- **Model Upgrades:** Replacing ResNet18 with a more advanced Vision Transformer (ViT) architecture or upgrading YOLOv8n to a larger variant can be accomplished by simply swapping the model weights within the classification and keyframe nodes without altering the conditional logic.
- **Temporal Reasoning:** Future work will integrate a separate temporal reasoning module (e.g., Transformer-based) that operates on the extracted keyframes to provide a more holistic understanding of the action sequence, further enhancing the summarization quality.
- **External Tool Integration:** The pipeline can be extended by adding new nodes for geographical data grounding or external database lookups, all managed by the existing LangGraph state.

Acknowledgment

The authors gratefully acknowledge the use of the AAI-521 curriculum materials, the foundational work on parameter-efficient fine-tuning, and the computational resources provided by [Institution Name] which were crucial for the extensive model training and validation exercises.

Appendix A
Implementation Details and State Schema

A. LangGraph State Schema

The primary state object, updated by each node, is defined as a dictionary-like structure containing the following keys:

- `video_input`: Raw path or reference to the video file.
- `classification_label`: String output (e.g., 'running', 'Punch').
- `confidence_score`: Float (σ). Used for conditional routing.
- `keyframe_paths`: List of strings, paths to selected image keyframes (High-Confidence Path).
- `audio_transcript`: Raw text from transcription (Low-Confidence Path).
- `refined_context`: Cleaned text or key entities from audio (Low-Confidence Path).
- `final_query`: String used for news search (derived from label or refined context).
- `final_summary`: Comprehensive text summary generated by the final LLM call.
- `news_articles`: List of external news search results.

B. Code Structure (Pseudo-Code for Confidence Check)

The routing function in LangGraph ensures atomicity and adherence to the threshold.

```
function route_on_confidence(state)
    score ← state["confidence_score"]
    threshold ← 0.70 {Empirical Threshold}

    if score > threshold then
        {High Confidence → Efficiency Cycle}
    return "high_confidence_path"
    else
        {Low Confidence → Resilience Cycle}
    return "low_confidence_path"
```

end if
end function

The final news fetch node incorporates the normalization logic:

function fetch_news_node(state)
query ← state[“classification_label”]

{1. Normalization Logic}
if query = “Punch” **then**
query ← “boxing”
end if

{2. Context Prioritization}
if state[“refined_context”] is not None **and then**
state[“refined_context”] ≠ ‘‘’’
query ← state[“refined_context”]
end if

{Execute search API call...}
{return {“news_articles”: results, “final_query”: query}}

end function

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[2] E. J. Hu et al., “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022.

[3] J. Redmon et al., “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.