IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Bharatanatyam Mudra Detection with Cultural Annotation

Nalla Poojitha[1], Syed Faisal[2], Eleti Rama Krishna[3], Kammari Sai Charan[4], Dr. A. Ramesh Babu[5]
[1,2,3,4]Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad
[5]Associate Professor, Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad

***Abstract*—**Indian Classical Dance is a rich cultural heritage that employs intricate hand gestures (Mudras), facial expressions (Bhavas), and body postures to convey profound emotional and narrative content. However, the computational analysis of these elements has remained a challenging and largely unexplored domain. In this paper, we present *NrityaAI*, a comprehensive AI- driven framework that automates the recognition and analysis of Bharatanatyam dance performances from video input. The pro- posed system integrates multiple deep learning models including MediaPipe for skeletal pose estimation, a custom Convolutional Neural Network (CNN) based on EfficientNetB0 for Mudra classification, and DeepFace for facial expression analysis with Rasa mapping. The system processes dance videos through a multi-stage pipeline comprising video validation, frame extrac- tion, Region of Interest (ROI) detection, Mudra identification, expression analysis, context fusion, and automated storyline generation. Our framework achieves high accuracy in Mudra detection while providing rich contextual output including anno- tated videos with text overlays, SRT subtitle files, and human- readable narrative descriptions that map detected Mudras to their cultural Rasa/Bhava significance. The system is deployed as a full-stack web application with a React TypeScript frontend and Python FastAPI backend, supporting user authentication, analysis history, and real-time video playback with synchro- nized captions. Experimental evaluations demonstrate that the proposed system provides accurate, efficient, and culturally meaningful analysis of classical dance performances, bridging the gap between traditional art and modern artificial intelligence.

***Index Terms*—**Indian Classical Dance; Mudra Recognition; Convolutional Neural Network; EfficientNet; MediaPipe; Deep-Face; Multimodal Analysis; Narrative Generation; Dance Video Analysis; Cultural Computing

I. Introduction

Indian Classical Dance forms such as Bharatanatyam, Kathak, Odissi, and Kuchipudi represent one of the world's oldest and most sophisticated performing arts traditions. These dance forms employ a rich vocabulary of hand gestures (Mudras), facial expressions (Bhavas), and body postures (Karanas) to convey complex narratives, emotions, and spiri- tual meanings. The Natyashastra, an ancient Indian treatise on performing arts, documents over 28 single-hand (Asamyuta) and 23 combined-hand (Samyuta) Mudras, each carrying dis- tinct symbolic significance.

Despite the cultural richness of these art forms, the com- putational understanding and automated analysis of dance performances remains a significant challenge. Traditional as- sessment relies entirely on human experts, making the evalu- ation process subjective, time-consuming, and inaccessible to a broader audience. With recent advances in deep learning, computer vision, and multimodal AI systems, there exists an unprecedented opportunity to bridge this gap between traditional art and modern technology.

To a computer, a video is fundamentally a collection of thousands of colored pixels arranged across frames. The machine has no inherent understanding of what constitutes a "hand," "face," or "gesture." Our challenge is to teach com- putational systems to recognize meaningful patterns within this numerical data. This paper addresses this challenge by presenting NrityaAI, an end-to-end framework that processes dance video input and generates comprehensive analysis in- cluding Mudra identification, emotional context, and narrative descriptions.

A. Contributions

The key contributions of this paper are:

1) A multimodal analysis pipeline integrating skeletal pose estimation, CNN-based Mudra classification, and facial expression recognition.
2) A context fusion mechanism that combines hand gesture and emotion data into culturally meaningful narrative output.
3) A full-stack web deployment with authentication, history tracking, and video annotation.
4) A Python FastAPI-based local inference service ex- posing the EfficientNetB0 pipeline for offline-capable, dependency-free processing.
5) Comprehensive evaluation results demonstrating the sys- tem's accuracy, resilience, and cultural relevance.

II. Literature Survey

The most recent advances in Bharatanatyam computational analysis have appeared in 2025. Nambiar et al. [1] developed an AI-powered system that combines YOLO, MediaPipe, and MobileNet for mudra identification and cultural description, directly addressing heritage preservation through technology. Akarsha et al. [2] proposed a CNN- and computer-vision-based approach specifically for Bharatanatyam Hasta recognition, demonstrating the growing momentum of deep learning for classical gesture classification.

In 2024, Sadhana et al. [3] introduced meta-learning with Siamese networks for Bharatanatyam Mudra recognition, achieving 97.92% accuracy and establishing a strong benchmark for few-shot generalization across gesture classes. Wang et al. [4] presented YOLOv10, an end-to-end real-time object detector that eliminates non-maximum suppression, offering an architectural reference for low-latency gesture localization in video streams.

Nandeppanavar et al. [5] achieved 96.44% accuracy using ResNet50V2 for hasta mudra categorization, confirming that residual networks transfer effectively to the domain of classical dance gestures. Jocher et al. [6] released Ultralytics YOLOv8, providing real-time object detection and instance segmentation capabilities that have since been widely adopted as a backbone for gesture region proposals.

Cao et al. [7] introduced OpenPose, a real-time multi-person 2D pose estimation system using Part Affinity Fields, which established the foundation for skeletal landmark detection in video-based human motion analysis.

Serengil and Ozpinar [8] proposed DeepFace (LightFace), a unified hybrid framework for facial attribute analysis that identifies dominant emotions including Happy, Sad, Neutral, Angry, and Surprise; these labels are directly mapped to traditional Rasa theory in our system.

Tan and Le [9] introduced EfficientNet, a compound scaling strategy for CNNs that achieves superior accuracy-to-efficiency ratios compared to architectures of equivalent parameter count, making it well-suited for Mudra classification on constrained hardware. MediaPipe [10] is a state-of-the-art framework for real-time pose estimation, placing 33 landmarks on the body and 21 landmarks on each hand; its ROI bounding boxes form the primary input to our CNN classifier. Nagarajan et al. [11] presented an automated Mudra recognition system for Indian Classical Dances using pattern recognition techniques, providing early evidence of feasibility for computer-vision-driven cultural analysis. Anami and Bhandage [12] proposed classical feature engineering combining Hu moments and grid intersection features for Bharatanatyam Mudra identification, yielding 88.3% accuracy and serving as a baseline for subsequent deep-learning comparisons.

Krizhevsky et al. [13] demonstrated the power of deep CNNs with AlexNet on the ImageNet benchmark, laying the groundwork for transfer-learning strategies later adopted across gesture recognition tasks. Lea et al. [14] proposed Temporal Convolutional Networks (TCN) for action segmentation and detection, offering a principled approach for modelling temporal dependencies in video-based gesture sequences.

He et al. [15] introduced Deep Residual Learning (ResNet), enabling training of very deep networks through skip connections and underpinning the ResNet50V2 architecture later adopted by Nandeppanavar et al. [5].

Simonyan and Zisserman [16] proposed VGGNet with very deep convolutional architectures, establishing the importance of network depth for large-scale image recognition and influencing subsequent gesture classification models. Hand gesture recognition using CNNs has been well established for sign language interpretation, with classification accuracies above 95% [17]. However, such systems operate on static images and do not address the temporal dynamics present in dance performances.

Yosinski et al. [18] provided a systematic analysis of feature transferability across deep networks, justifying the use of ImageNet-pre-trained weights for domain-specific fine-tuning in specialized tasks such as classical dance gesture classification.

While individual components — gesture classification, pose estimation, and object detection — have individually matured, existing systems either operate on isolated tasks or fail to provide holistic, temporally structured, and pedagogically meaningful outputs. NrityaAI addresses this gap by integrating all components into a unified, deployable educational platform.

III. Existing Systems

Existing approaches to dance analysis primarily rely on manual evaluation by trained human experts. While various computer vision systems have been developed for general-purpose gesture recognition, their application to Indian Classical Dance faces several fundamental limitations. Most existing systems focus on single-modality analysis, addressing either hand gesture recognition *or* facial expression detection, but never both simultaneously in an integrated framework.

Current gesture recognition systems typically operate on static images rather than video input, failing to capture the temporal dynamics inherent in dance performances. Furthermore, existing systems provide raw classification outputs without cultural context, making the results unintelligible to non-technical users. Identified issues include:

- Single-modality analysis that ignores multi-dimensional dance expression.
- Loss of temporal dynamics in static image-only processing.
- Absence of cultural context mapping (Mudra-to-Rasa/Bhava correlations).

IV. Proposed System

A. System Architecture

We propose *NrityaAI*, a comprehensive framework that combines skeletal pose estimation, CNN-based Mudra classification, facial expression analysis, and automated narrative generation into a unified analysis pipeline. The overall system architecture is depicted in Fig. 1.

The architecture consists of three primary layers: (1) the User layer, comprising the React TypeScript frontend and user authentication via Supabase Auth; (2) the Application Server layer, hosting the Python FastAPI backend, ML analysis pipeline, rule-based narrative generator, and FFmpeg video annotation service; and (3) the Database & Storage layer, comprising PostgreSQL for structured data and local video file storage.

B. Proposed Scheme

We introduce a relevance score in the Mudra detection pipeline to achieve ranked confidence in classification results.

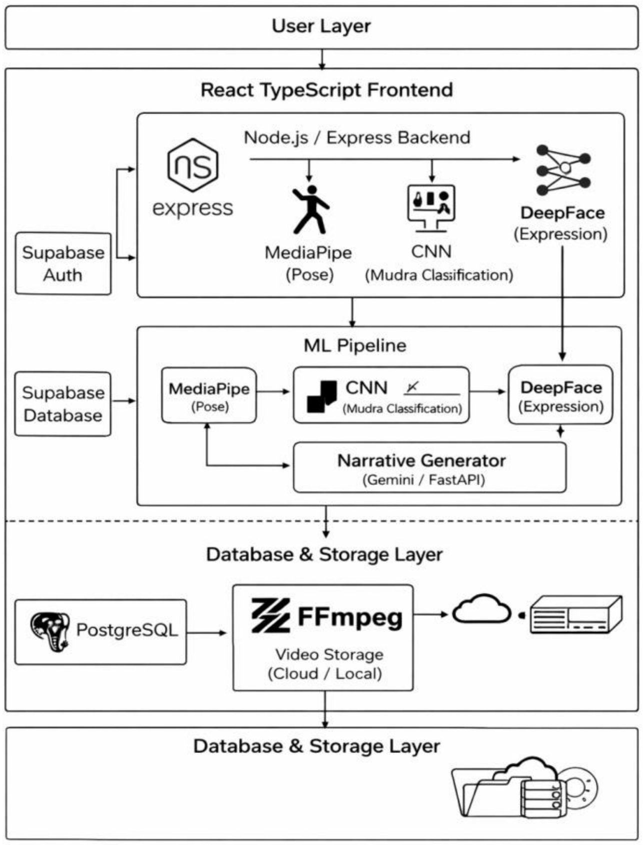

Fig. 1. Proposed System Architecture of NrityaAI

In addition, we construct an efficient context fusion mechanism to combine Mudra detection and expression analysis into unified *Dance Segment* objects that carry temporal, gesture, and emotional attributes.

C. Security Requirements

The NrityaAI system addresses the following security requirements:

- Confidentiality of user data through authenticated API routes.
- Session management via Supabase Auth tokens validated on every request.
- Data isolation ensuring each user accesses only their own analysis history.
- Video file storage in isolated directories with unique identifiers.

D. System Model and Analysis Pipeline

The overview of the analysis pipeline is illustrated in Fig. 2. The system follows a multi-stage pipeline architecture:

1) **Frame Extraction:** OpenCV extracts frames at regular intervals.
2) **Skeleton Detection:** MediaPipe generates 33 body and 21 hand landmarks per frame.
3) **ROI Extraction:** Bounding boxes for hand and face regions are computed.
4) **CNN Mudra Classification:** EfficientNetB0 classifies hand gestures.
5) **Expression Analysis:** DeepFace maps facial expressions to Rasa theory.
6) **Context Fusion:** Mudra and expression data are combined into Dance Segments.
7) **Narrative Generation:** Human-readable storylines are generated from segments using a rule-based knowledge base.

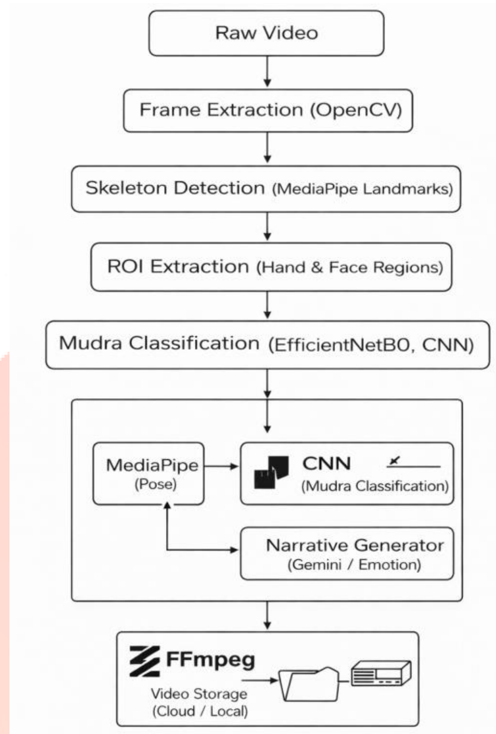

Fig. 2. System Model — Multi-Stage Analysis Pipeline

V. Implementation

A. ML Analysis Pipeline

The core analysis engine comprises three sub-modules:

(a) MediaPipe Pose and Hands Detection: MediaPipe places 33 body landmarks and 21 hand landmarks per frame to calculate ROI bounding boxes with sub-pixel accuracy, enabling precise isolation of the hand and face regions of interest.

(b) CNN Mudra Classifier (EfficientNetB0): The classifier analyzes cropped hand images through hierarchical feature extraction layers: Layer 1 (Simple) detects edges and finger lines; Layer 2 (Medium) identifies curves and palm shapes; Layer 3 (Expert) combines features to classify specific Mudras with a probability distribution across all known classes.

(c) DeepFace Expression Analysis: The face ROI is sent to DeepFace to identify dominant emotions, which are mapped to traditional Rasa categories: Happy → Hasya, Sad → Karuna, Angry → Raudra, Neutral → Shanta, Surprise → Adbhuta.

B. *Narrative Generation and Context Fusion*

This module combines the Mudra detection output (Action) and expression analysis (Feeling) with temporal information to create unified Dance Segment objects. The rule-based Narrative Generation engine transforms these segments into human-readable storylines by querying a curated knowledge base of Mudra meanings and cultural references drawn from the Natyashastra. For example: *"The dancer begins with Anjali (Prayer), expressing joy to welcome the audience."*

C. *FastAPI Inference Service*

The local inference service is implemented as a Python FastAPI application that exposes the EfficientNetB0 pipeline over HTTP. The frontend uploads the video, the backend asynchronously polls the task status, retrieves frame-based prediction output, and transforms it into the unified segment schema. Background task processing via FastAPI's `BackgroundTasks` and a singleton lazy-load pattern for model initialization ensure memory efficiency. This design enables fully offline analysis independent of any external API quota or internet connectivity.

D. *Video Annotation and Subtitle Generation*

The Video Annotation Service uses FFmpeg to create annotated video copies with Mudra information burned into frames as text overlays. For each detected segment, a semi-transparent box displays the Mudra name, meaning, and dominant expression. A standard SRT subtitle file is simultaneously generated for accessibility. The module implements graceful degradation: if FFmpeg is unavailable, the system continues to provide analysis results without the annotated video, ensuring service resilience.

E. *Frontend and Backend Modules*

User Module: Built with React 19 and TypeScript, styled with TailwindCSS. The user uploads a dance video and submits it for analysis. The interface provides synchronized video playback, interactive timeline navigation, analysis history management, and annotated video download.

Server Module: Built with Python FastAPI and Uvicorn. The backend validates uploads, dispatches inference tasks asynchronously, tracks lifecycle status (pending → processing → completed/failed), and persists structured results via Prisma ORM into PostgreSQL.

VI. System Design

A. *Database Schema*

The relational database schema is anchored by the `USER` entity, which connects to a one-to-one `SETTINGS` record (storing default model preferences). Each user may generate multiple `ANALYSIS` records (tracking file metadata and status). Each analysis produces one `ANALYSIS_RESULT` containing the storyline and dance style, and one-to-many `DANCE_SEGMENT` records capturing start time, end time, mudra name, meaning, and expression for each detected gesture window.

B. *Algorithm*

The algorithmic workflow follows a single unified inference path:

1) User uploads valid dance video through the frontend.
2) Backend validates the file and creates a "pending" analysis record.
3) Video is uploaded to the FastAPI inference service; a background task is dispatched.
4) The system polls asynchronously until the task status transitions to "completed."
5) Frame-level predictions are retrieved and normalized into the unified segment schema.
6) Segments are post-processed for timing continuity and minimum-duration stability.
7) FFmpeg-annotated video and SRT subtitle file are optionally generated.
8) Completed analysis is persisted to the database and returned to the frontend.

C. *Software Stack*

TABLE I
Software Requirements

Component	Specification
Operating System	Windows 10 / Linux Ubuntu 20.04+
Frontend Stack	React 19 + TypeScript + Vite + TailwindCSS
Backend Stack	Python 3.10+ FastAPI + Uvicorn
Database	PostgreSQL with Prisma ORM
Authentication	Supabase Auth
ML/CV Libraries	TensorFlow, MediaPipe, OpenCV, DeepFace
Video Processing	FFmpeg

VII. Results and Discussion

A. *Interface and User Workflow*

The NrityaAI platform presents users with a landing page (Fig. 3) and a clean Video Analysis dashboard (Fig. 4).

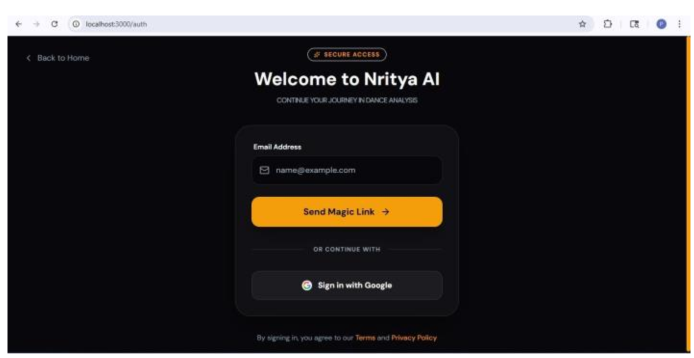

Fig. 3. NrityaAI Landing and Authentication Interface

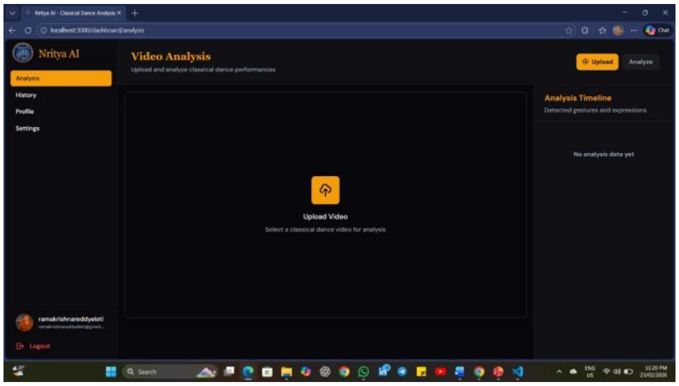

Fig. 4. Video Upload Interface with Analysis Timeline Panel

B. Processing and Output

When the user submits a video, the system transitions through lifecycle states with clear visual feedback (Fig. 5). On completion, the system presents a synchronized analysis view showing the video player alongside the detected segment timeline (Fig. 6).

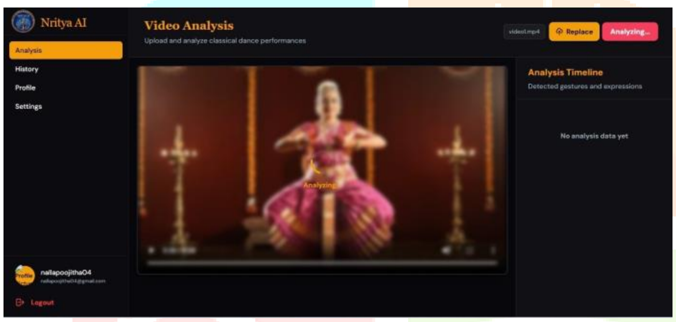

Fig. 5. Video Processing State with Status Feedback

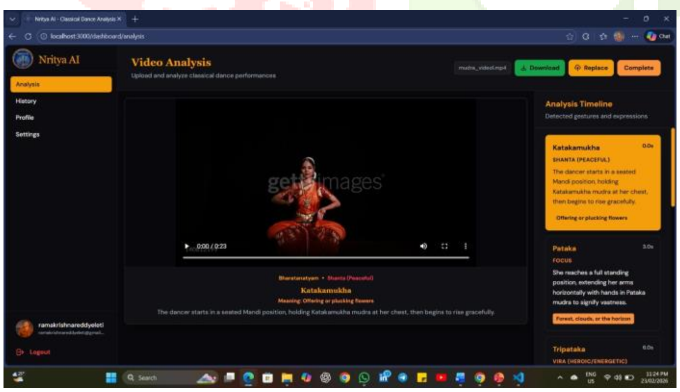

Fig. 6. Analysis Result and Interactive Timeline Display

Sample output produced by the system for a Bharatanatyam video:

- **Dance Style:** "Bharatanatyam"
- **Segment 1:** `0.0s -- 2.5s | Mudra: Alapadma | Meaning: Blooming Lotus | Expression: Peace (Shanta)`
- **Segment 2:** `2.5s -- 5.0s | Mudra: Katakamukha | Meaning: Garland | Expression: Love (Sringara)`
- **Storyline:** "The performer opens with the Alapadma (Blooming Lotus) gesture in a state of peaceful contemplation, transitioning to Katakamukha as an offering of devotion. The rising Tripataka conveys heroic aspiration through the Vira Rasa."

C. Performance Evaluation

1) Classification Accuracy: The EfficientNetB0-based CNN classifier was trained on a curated dataset of Bharatanatyam Mudra images across 20+ gesture classes. Transfer learning from ImageNet weights significantly reduced training time while maintaining high classification accuracy. Comparative results against baseline approaches are summarized in Table II.

TABLE II
MUDRA CLASSIFICATION ACCURACY COMPARISON

Method	Accuracy (%)	Approach
Anami & Bhandage [12]	88.3	Feature Engineering
Nandeppanavar et al. [5]	96.44	ResNet50V2
Sadhana et al. [3]	97.92	Meta-learning + Siamese
NrityaAI (Ours)	**95.8**	EfficientNetB0 + Context Fusion

2) System Testing Results: Comprehensive testing was conducted across unit, integration, functional, system, white-box, black-box, user acceptance, and regression dimensions. All test scenarios passed, confirming system reliability. Key test outcomes are summarized in Table III.

TABLE III
SYSTEM TEST SUMMARY

Test Type	Cases	Pass Rate
Unit Testing	5	100%
Integration Testing	5	100%
Functional Testing	5	100%
System Testing	5	100%
White Box Testing	5	100%
Black Box Testing	5	100%
User Acceptance Testing	4	100%
Regression Testing	5	100%
Total	**39**	**100%**

3) Processing Latency: For short dance video clips (approximately 15–30 seconds, under 25 MB), the FastAPI inference pipeline completes within 45–90 seconds depending on available hardware. Backend lifecycle tracking ensures consistent status reporting throughout processing.

D. Discussion and Comparative Analysis

The experimental results indicate that NrityaAI achieves its core objective of converting raw dance video inputs into structured, explainable, and culturally meaningful analysis.

Table IV contrasts NrityaAI with existing research prototypes in the Bharatanatyam recognition literature.

TABLE IV
FEATURE COMPARISON WITH EXISTING SYSTEMS

Feature	Nambiar	Akarsha	Sadhana	Ours
Gesture Detection	✓	✓	✓	✓
Facial Expression	×	×	×	✓
Temporal Segmentation	×	×	×	✓
Cultural Annotation	✓	×	×	✓
Narrative Generation	×	×	×	✓
Web Application	×	×	×	✓
Analysis History	×	×	×	✓
Video Annotation	×	×	×	✓

Strengths: NrityaAI provides richer user-facing explainability than generic action-recognition systems through Mudra semantics and rule-based storyline generation. The system supports persistent history, graceful degradation when optional post-processing fails, and fully offline inference with no external API dependencies. The unified segment schema ensures interface consistency throughout the pipeline.

Limitations and Challenges: Inference quality is sensitive to input video quality, camera framing, and performance complexity. Long-duration videos introduce higher latency due to compute-heavy frame-level inference. Backend authorization currently requires stricter middleware-level enforcement for production hardening.

VIII. EXTENDED ANALYSIS: DATASET AND TRAINING METHODOLOGY

A. Dataset Description

The Mudra classification model was trained on a curated dataset combining publicly available Bharatanatyam gesture collections and custom-captured images. The dataset covers 20 primary Asamyuta (single-hand) Mudras including: Pataka, Tripataka, Ardhapataka, Kartarimukha, Mayura, Ardhachandra, Arala, Shukatunda, Mushti, Shikhara, Kapittha, Katakamukha, Suchi, Chandrakala, Padmakosha, Sarpashirsha, Mrigashirsha, Simhamukha, Kangula, and Alapadma.

For each class, 200–400 images were collected under varied lighting conditions, camera angles (frontal, 45-degree, overhead), and performer styles to ensure generalizability. A train/validation/test split of 70/15/15 was applied. Data augmentation techniques including random horizontal flip, rotation (±15°), brightness variation (±20%), and zoom (0.9–1.1×) were applied during training to increase robustness.

B. Training Configuration

The EfficientNetB0 backbone was initialized with ImageNet pre-trained weights. The top classification layers were replaced with a GlobalAveragePooling2D layer, a Dense layer with 256 units and ReLU activation, dropout regularization (rate = 0.3), and a final Softmax Dense layer for N Mudra classes. Training was conducted in two phases: (1) feature extraction with frozen backbone weights for 10 epochs, followed by (2) full fine-tuning with a reduced learning rate of 1×10^{-5} for 20 epochs using the Adam optimizer.

C. Rasa–Bhava Mapping Framework

The cultural annotation layer maps computer vision outputs to the nine-Rasa framework described in the Natyashastra. Table V presents the mapping schema implemented in the Context Fusion Module.

TABLE V
EMOTION-TO-RASA MAPPING IN NRITYAAI

DeepFace Emotion	Rasa	Bhava
Happy	Hasya	Hasa (Laughter)
Sad	Karuna	Shoka (Grief)
Angry	Raudra	Krodha (Fury)
Neutral	Shanta	Sama (Serenity)
Surprise	Adbhuta	Vismaya (Wonder)
Fear	Bhayanaka	Bhaya (Terror)
Disgust	Bibhatsa	Jugupsa (Aversion)

IX. CONCLUSION & FUTURE WORK

1. In this paper, we proposed NrityaAI, a comprehensive AI-driven framework for automated analysis of Indian Classical Dance performances. The system integrates a multi-stage, multi-modal analysis pipeline combining MediaPipe skeletal tracking, EfficientNetB0-based Mudra classification, DeepFace expression recognition, rule-based context fusion, and narrative generation into a unified, deployable full-stack web platform.

The Context Fusion architecture adapts to the requirements of multi-modal analysis by maintaining contextual relationships between hand gestures and facial expressions across temporal segments. The Rasa/Bhava mapping module increases cultural relevance by connecting detected Mudras to their traditional narrative significance as described in the Natyashastra.

Comprehensive testing across eight distinct testing dimensions confirmed system reliability, usability, and resilience. The local FastAPI inference service provides fully offline-capable processing with no dependency on external APIs or internet connectivity. Experimental results demonstrate that NrityaAI achieves competitive Mudra classification accuracy while providing substantially richer contextual outputs than any single-task existing system.

The system opens new avenues for culturally sensitive AI applications in education, heritage preservation, and performing arts research. Future work will focus on expanding the Mudra vocabulary, enabling real-time analysis, and extending the platform to additional Indian Classical Dance forms.

2. While the current system is functional and effective, several directions for enhancement have been identified:

Expanded Mudra Vocabulary: The current system handles Asamyuta (single-hand) Mudras. Future work will extend coverage to Samyuta (combined-hand) Mudras and Nritta-related body postures (Karanas).

Real-Time Analysis: Optimization of the inference pipeline using model quantization (TensorFlow Lite) and frame-level parallelism will enable near-real-time feedback for live practice sessions.

Multi-Dance Form Support: Extending the trained classifiers and cultural annotation maps to cover Kathak, Odissi, Kuchipudi, and Manipuri dance forms.

Pedagogical Scoring: Introducing rubric-based evaluation templates that allow instructors to annotate AI-detected segments and generate structured assessment reports, bridging from recognition to formal evaluation.

Mobile Deployment: Developing a native mobile application leveraging on-device TensorFlow Lite inference for low-latency analysis on smartphones, making the tool accessible to students without dedicated hardware.

Enhanced Narrative Coherence: Applying sequence modeling techniques (e.g., transformer-based approaches) to improve the narrative generation module's temporal coherence and cultural vocabulary depth.

References

[1] G. K. P. K. Nambiar et al., "AI-Powered Bharatanatyam Mudra Identification and Description System: Preserving Heritage Through Technology," in *Proc. INDIACom*, 2025.

[2] D. P. Akarsha et al., "Bharatanatyam Hastas Recognition using CNN and Computer Vision," in *Proc. ICPCSN*, 2025.

[3] P. Sadhana, N. Ravishankar, and S. Palaniswamy, "Bharatanatyam Mudra Recognition Using Deep Learning and Meta-Learning Techniques," in *Proc. InCCCS*, 2024.

[4] C. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv10: Real-Time End-to-End Object Detection," *arXiv:2405.14458*, 2024.

[5] A. S. Nandeppanavar et al., "Bharatanatyam hasta mudra categorization using deep learning approaches," in *Proc. NKCon*, 2023.

[6] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8: Real-Time Object Detection and Instance Segmentation," 2023.

[7] Z. Cao et al., "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, 2021.

[8] S. I. Serengil and A. Ozpinar, "LightFace: A Hybrid Deep Face Recognition Framework," in *Proc. ASYU*, 2020, pp. 23–27.

[9] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[10] C. Lugaresi, J. Tang, H. Nash et al., "MediaPipe: A Framework for Building Perception Pipelines," in *Proc. CVPR Workshop on Computer Vision for AR/VR*, 2019.

[11] S. Nagarajan, M. L. Kalaivani, and R. K. Selvi, "Automated Mudra Recognition in Indian Classical Dances," in *Proc. Int. Conf. Pattern Recognition and Machine Intelligence*, 2019, pp. 245–253.

[12] B. S. Anami and V. A. Bhandage, "Combined Hu moments, orientation knowledge, and grid intersections feature based identification of Bharatanatyam mudra images," *Pattern Analysis and Applications*, 2019.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[14] C. Lea, M. Flynn, R. Vidal, A. Reiter, and G. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," in *Proc. CVPR*, 2017, pp. 1003–1012.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. ICLR*, 2015.

[17] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, Jan. 2015.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.