Real-Time Hand Gesture Recognition for Human-Computer Interaction

MOHAN VAMSI OLIPI

904047187

Git Hub Link

Why Is Real-Time Gesture Recognition is Important?

- **Evolution of Human-Computer Interaction (HCI):**
  - Traditional input devices (mouse, keyboard) are becoming outdated.
  - Gesture recognition provides a natural and intuitive way for humans to interact with technology.
- **Touchless Interaction:**
  - Enables hands-free control, improving accessibility and hygiene (e.g., smart home, virtual reality).
- **Improved User Experience:**
  - Users interact with devices in a more natural and fluid way, similar to interacting with people.
- **Faster and More Intuitive:**
  - Allows quicker responses and commands compared to traditional input methods.

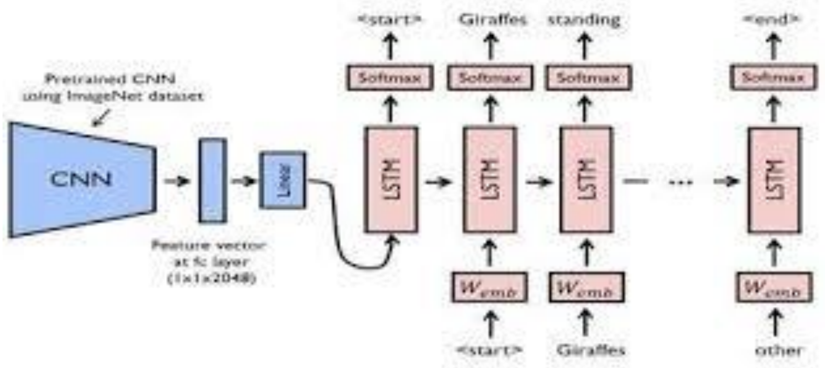
Project Objective

- **Goal:**
- Build a real-time hand gesture recognition system that provides efficient control for various applications (HCI, VR, Smart Homes).
- Achieve high accuracy and low latency (<33ms per frame).

- **Key Challenges Addressed:**
- Traditional methods are not fast enough for real-time applications.
- Focus on achieving high accuracy with low inference time, ideal for resource-constrained environments.

Model Overview

- **Hybrid CNN-LSTM Architecture:**
- **CNN for spatial features:** Efficient extraction of spatial features like hand shape and gesture context.
- **LSTM for temporal features:** Model the sequence of gestures to understand dynamic hand movements.



- **Flow of the Model:**
- **Input:** Video → Frames → CNN extracts features → LSTM models sequences → **Output:** Gesture Class Prediction.

Data Preprocessing

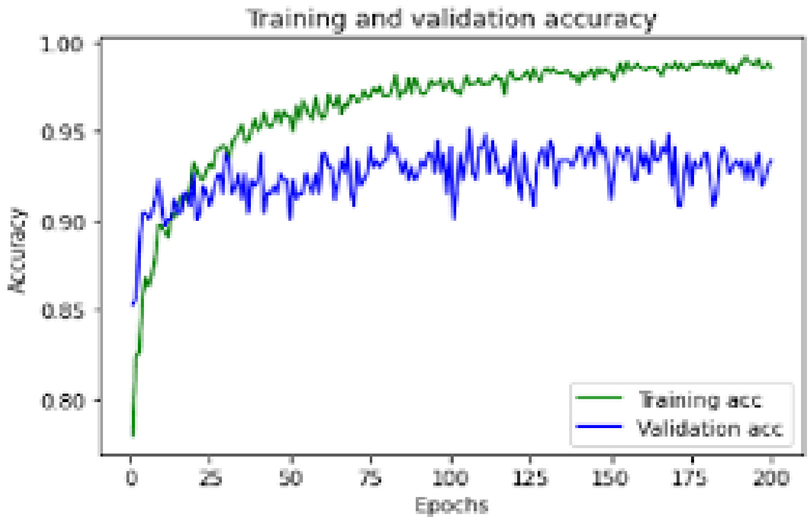
- **Dataset:** Jester Dataset from Kaggle
- **Size:** 148,092 videos, 27 gesture classes.
- **Gestures:** Swiping, zooming, pointing, etc.
- **Data Conversion:** Videos are split into individual frames (224x224) for processing.
- **Preprocessing Steps:**
- Frame extraction from videos.
- Normalization: Convert pixel values to a 0-1 range for better model performance.
- One-hot encoding for gesture labels (27 classes).

CNN-LSTM Model

- **CNN Backbone:** MobileNetV2
- Lightweight, efficient architecture for fast processing.
- **Layers:** Convolutional layers for feature extraction, pooling layers for spatial dimension reduction.
- **LSTM Layer:**
- Captures the sequence of gestures across frames.
- Helps with context understanding (e.g., a swipe gesture is different from a zoom gesture).
- **Output Layer:**
- Softmax layer with 27 neurons, each representing one gesture class.

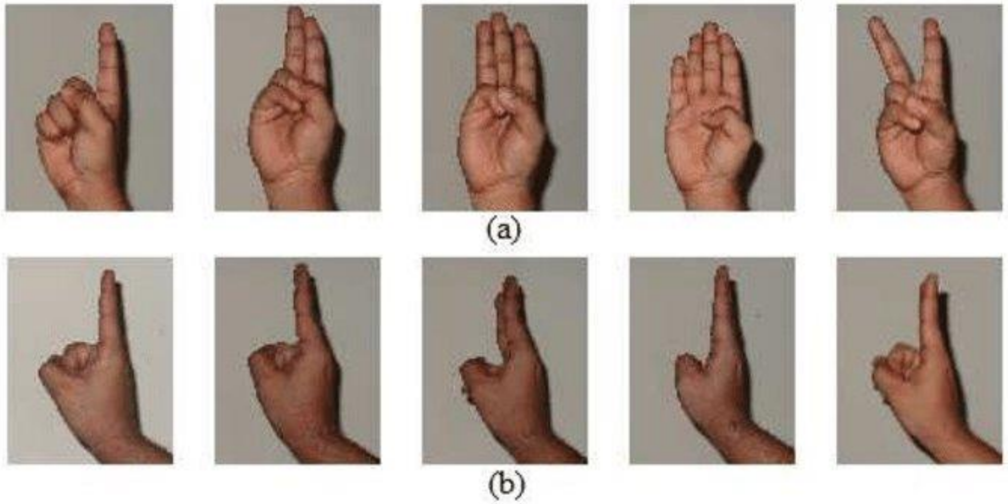
Model Training

- **Training Process:**
- Split dataset: 80% training, 10% validation, 10% test.
- Optimizer: Adam optimizer for faster convergence.
- Loss Function: Categorical cross-entropy.
- Metrics: Accuracy, Precision, Recall, and F1-Score.
- **Expected Results:**
- Training accuracy: >90%.
- Inference speed: <33ms per frame.



Results and Evaluation

- **Accuracy:** Measures the correct predictions vs total predictions.
- **Precision, Recall, F1-Score:** Assess the model's performance for each gesture class.
- **Model Comparison:**



- Compare the CNN-LSTM hybrid model's performance (accuracy and inference time) with traditional CNN models (like MobileNetV2 without LSTM).

Use Cases

- **Virtual Reality (VR):**
- Gesture-based control of virtual objects (e.g., controlling a VR game environment with hand movements).
- **Smart Home Devices:**
- Control lights, fans, music, etc., through hand gestures.
- Example: "Swiping" gestures to change TV channels or adjust the thermostat.
- **Human-Computer Interaction (HCI):**
- Touchless computer control for accessibility, making it easier for people with disabilities.

Challenges & Future Work

- **Challenges:**
  - **Real-Time Processing:** Maintaining fast inference times on resource-constrained devices like smartphones or edge devices.
  - **Environmental Factors:** Handling gestures in different lighting conditions or noisy backgrounds.
- **Future Work:**
  - Improve performance with more diverse training data.
  - Handle multi-hand gestures and improve gesture recognition accuracy in complex scenarios.
  - Expand to mobile devices with optimized model architectures for mobile inference.

Conclusion

- **Real-Time Performance**: The model successfully recognizes hand gestures in real-time with an inference time of less than 33 ms per frame, making it suitable for applications like human-computer interaction (HCI) in virtual reality and smart home systems.
- **High Accuracy**: The hybrid CNN-LSTM model achieves an accuracy of over 90%, demonstrating robust performance in dynamic gesture recognition.
- **Seamless Integration**: By incorporating both spatial and temporal features, the model provides a seamless and intuitive gesture control system, eliminating the need for physical touch in interactive systems.
- **Scalability**: The approach is highly scalable and can be adapted to different gesture datasets and real-time applications with minimal changes.
- **Future Improvements**: There is potential to enhance the model by incorporating more gesture categories, improving training with more diverse datasets, and optimizing for deployment on edge devices with limited computational resources.