Gesture Recognition for Smart TV Control: Model Development and Performance Analysis

- **Pavani Rao Nileshwar – Group Facilitator**
- **Pawan Patidar**

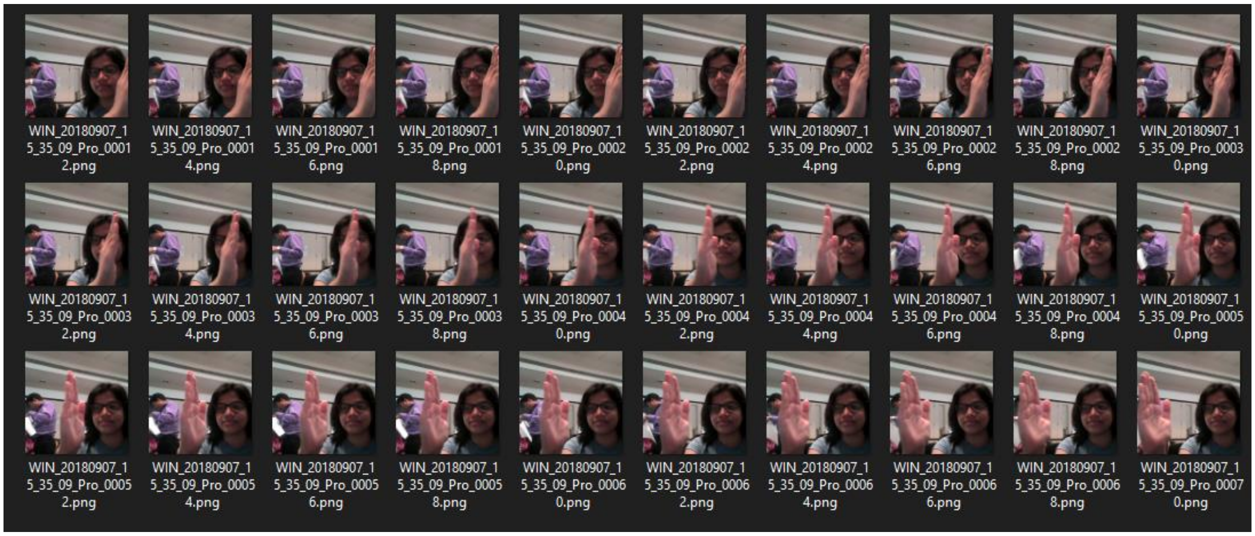
Introduction

In the era of advanced smart devices, gesture recognition offers an innovative way for users to interact with technology. For this project, a gesture-based control system was developed to allow users to control various functions on a smart TV, such as adjusting volume, skipping video sections, and pausing playback. This system recognizes five gestures—thumbs up, thumbs down, left swipe, right swipe, and stop—each corresponding to a specific TV command.

Dataset and Preprocessing

The dataset comprised video sequences divided into 30 frames per gesture. Each video captured one of the five distinct gestures performed in front of a webcam, simulating a real-world application environment. Key steps in data preprocessing included resizing, cropping, and normalizing images to ensure gestures were distinct and less affected by background noise. Data augmentation, such as slight rotations, was also applied to increase model robustness by accounting for variations in hand positioning.

Dataset: https://drive.google.com/uc?id=1ehyrYBQ5rbQQe6yL4XbLWe3FMvuVUGiL


WIN_20180907_15_35_09_Pro_00012.png
WIN_20180907_15_35_09_Pro_00014.png
WIN_20180907_15_35_09_Pro_00016.png
WIN_20180907_15_35_09_Pro_00018.png
WIN_20180907_15_35_09_Pro_00020.png
WIN_20180907_15_35_09_Pro_00022.png
WIN_20180907_15_35_09_Pro_00024.png
WIN_20180907_15_35_09_Pro_00026.png
WIN_20180907_15_35_09_Pro_00028.png
WIN_20180907_15_35_09_Pro_00030.png
WIN_20180907_15_35_09_Pro_00032.png
WIN_20180907_15_35_09_Pro_00034.png
WIN_20180907_15_35_09_Pro_00036.png
WIN_20180907_15_35_09_Pro_00038.png
WIN_20180907_15_35_09_Pro_00040.png
WIN_20180907_15_35_09_Pro_00042.png
WIN_20180907_15_35_09_Pro_00044.png
WIN_20180907_15_35_09_Pro_00046.png
WIN_20180907_15_35_09_Pro_00048.png
WIN_20180907_15_35_09_Pro_00050.png
WIN_20180907_15_35_09_Pro_00052.png
WIN_20180907_15_35_09_Pro_00054.png
WIN_20180907_15_35_09_Pro_00056.png
WIN_20180907_15_35_09_Pro_00058.png
WIN_20180907_15_35_09_Pro_00060.png
WIN_20180907_15_35_09_Pro_00062.png
WIN_20180907_15_35_09_Pro_00064.png
WIN_20180907_15_35_09_Pro_00066.png
WIN_20180907_15_35_09_Pro_00068.png
WIN_20180907_15_35_09_Pro_00070.png

Model Architectures Explored

1. **3D Convolutional Neural Networks (Conv3D):** This architecture extended 2D convolutions to 3D, processing video frames as a sequence across spatial (x, y) and temporal (z) dimensions.

However, the Conv3D model encountered high computational costs and overfitting, yielding a validation accuracy of 46% and validation loss of 1.26 with 3.8 million parameters.

2. **Conv2D + GRU:** This architecture combined Conv2D for feature extraction with GRU for temporal sequence processing. While efficient with only 420,117 parameters, the model's performance remained limited, with a validation accuracy of 25% and high validation loss (3.76), showing challenges in effectively capturing temporal dynamics.
3. **Conv2D + LSTM:** A Conv2D-based feature extraction followed by an LSTM layer was tested for sequence prediction, yielding better results than Conv3D and Conv2D + GRU. This model achieved 69% validation accuracy and a validation loss of 0.77 with 3.15 million parameters, displaying improved generalization.
4. **Transfer Learning: MobileNet + LSTM:** For optimized performance, MobileNet was employed as a feature extractor through transfer learning, combined with an LSTM for temporal recognition. This model emerged as the most effective, achieving a training accuracy of 86.1%, validation accuracy of 74%, and validation loss of 0.60 with 297,733 trainable parameters out of a total 3.5 million, indicating strong generalization and efficient parameter usage.

Training and Optimization Strategies

The models were trained using batch normalization, dropout layers, and early stopping to counteract overfitting. The Adam optimizer was chosen for improved stability and convergence speed. Regularization and data augmentation further improved generalization, and early stopping prevented unnecessary training when validation loss plateaued.

Batch size adjustments were critical for managing GPU memory constraints, balancing model accuracy and computational efficiency.

Model Performance Summary

Below is a summary table of the performance metrics for each tested architecture:

Model Architecture	Training Accuracy	Validation Accuracy	Parameters	Remarks
Conv3D	40.8%	46.0%	3,793,221	Baseline model with high parameter count; performance limited by overfitting, moderate gains despite dropout layers and early stopping.

Conv2D + GRU	30.4%	25.0%	420,117	Struggled to generalize; validation loss remained high, and categorical accuracy was low despite parameter efficiency.
Conv2D + GRU	65.2%	69.0%	420,117	Higher accuracy with LSTM, showed effective validation loss reduction, and moderate overfitting improvement.
Transfer Learning: MobileNet + LSTM	86.1%	74.0%	3,528,645 (297,733 trainable)	Higher accuracy with LSTM, showed effective validation loss reduction, and moderate overfitting improvement.

Final Model: CNN + LSTM with Transfer Learning

The MobileNet + LSTM model with transfer learning was selected as the best-performing model, achieving the following metrics:

- **Training Accuracy:** 86.1%
- **Validation Accuracy:** 74%
- **Validation Loss:** 0.60
- **Parameters:** 297,733 trainable (3.5 million total)

This model demonstrated superior performance, balancing parameter efficiency with high accuracy and minimal overfitting. Its strong results confirm the effectiveness of transfer learning with MobileNet as a feature extractor for sequence-based gesture recognition tasks.

Observations

Key insights from model training and evaluation include:

- **Performance Efficiency:** The MobileNet + LSTM model achieved the best accuracy-efficiency balance, with optimized parameter use and high validation accuracy.
- **High Accuracy and Low Validation Loss:** The model achieved 86.1% training accuracy and 74% validation accuracy with a validation loss of 0.60, showing robust generalization to unseen data.
- **Effective Parameter Utilization:** With only 297,733 trainable parameters, the model was both efficient and scalable, suitable for real-time applications.

Conclusion

This project demonstrates that a MobileNet + LSTM architecture is a viable solution for gesture recognition in smart TVs, providing users with an intuitive, remote-free control method. Future improvements could explore even more advanced architectures, alternative transfer learning models, or hyperparameter tuning to further enhance model accuracy and scalability.