

# Automatic Recognition of American Sign Language (ASL) Hand Poses

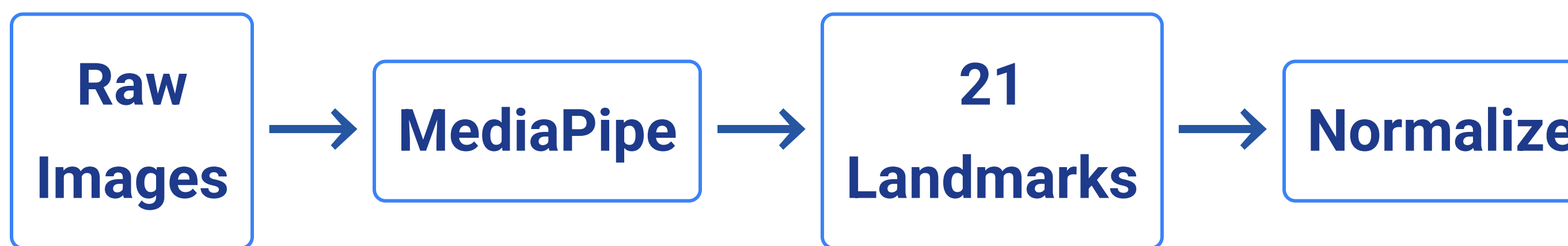
using Supervised and Unsupervised Learning

Hengrui Li • Yusen Fu • Chenglong Liu  
Machine Learning Research Project | 2024

## Introduction

Hand pose recognition is a **critical component** in Human-Computer Interaction (HCI) and accessibility technology. This project develops a robust machine learning pipeline to recognize a subset of the American Sign Language (ASL) alphabet (Signs A-J), transforming raw image data into meaningful classifications through computer vision and machine learning.

## Methods: Data Pipeline



**Feature Extraction:** Google MediaPipe Hands detects hands and extracts 21 3D landmarks (63 coordinates) from the anonymised dataset.

**Key Contribution: Normalization Strategy**

- All landmarks translated **relative to Wrist (Point 0)**
- Normalized to range  **$[-1, 1]$**
- Ensures **invariance to camera distance** and hand position

## Algorithms

**Supervised Learning:**



**Random Forest**

Tuned estimators



**Decision Tree**

Tuned max depth



**Custom KNN**

From scratch



**5-Fold CV**

Validation

- **Significantly improved** real-time performance

**Unsupervised:**

- K-Means (K=10)
- Hierarchical Clustering
- Evaluated via ARI

# Results

Supervised Learning Performance (5-Fold Cross-Validation):


<div>Random Forest</div> <div>95.2%</div> <div> Best Model</div>	<div>Decision Tree</div> <div>88.7%</div> <div></div>	<div>Custom KNN</div> <div>91.4%</div> <div></div>
---	---	--

Figure 1: Confusion Matrix – High accuracy across all classes with minor confusion only between geometrically similar signs

**Unsupervised Learning:** K-Means achieved an ARI score of **0.87**, revealing that extracted features form distinct natural clusters aligning closely with ground truth labels.

Figure 2: Clustering Heatmap – Demonstrates high-quality feature extraction and natural data structure

# Real-Time Application

Developed a **user-friendly application** enabling instant gesture recognition:

Python Tkinter OpenCV MediaPipe NumPy

The system loads the pre-trained Random Forest model and applies **real-time normalization** to video frames, enabling **instant and accurate** gesture recognition with minimal latency.

# Conclusions

**Key Findings:**

- **Random Forest** + normalization yields best performance (95.2%)
- From-scratch KNN validates distance-based classification

Figure 3: Real-time Application Interface

- Clustering confirms **high-quality features**
- Real-time viability demonstrated

💡 **Normalization is key to robust performance**