ASL HAND GESTURE RECOGNITION

Using MediaPipe Landmarks + MLP

2702315223 – Supandi

2702322853 – Evan F Hartono

BACKGROUND

- Deaf users face communication barriers with non–sign-language speakers
- ASL alphabet recognition is a practical first step for sign-to-text systems
- Pixel-based CNNs are heavy; landmark-based models are faster
- Research goal: test practicality + real-time performance on consumer hardware
- MediaPipe provides robust pretrained hand pose estimation

Goal: build a lightweight real-time ASL alphabet recognition system

Approach: MediaPipe Hands (pretrained model) + Multilayer Perceptron

Output: real-time gesture prediction from webcam input

Framework: FastAPI + OpenCV

HOW MEDIAPIPE WORKS

1. Palm Detection (SSD-Lite CNN)

- Detects palm region, easier than detecting full hand
- Provides bounding box + orientation

2. ROI Alignment

- Hand crop normalized to fixed orientation
- Stabilizes input → higher landmark accuracy

3. Landmark Regression Model (MobileNetV3-like CNN)

- Predicts 21 landmarks (x, y, z) directly
- Trained on millions of annotated hand poses
- Learns geometry (contours/shape), not skin color

4. Tracking + Temporal Smoothing

- Uses previous frame to skip heavy detection
- Enables 20–30 FPS on CPU

DATASET & FEATURES

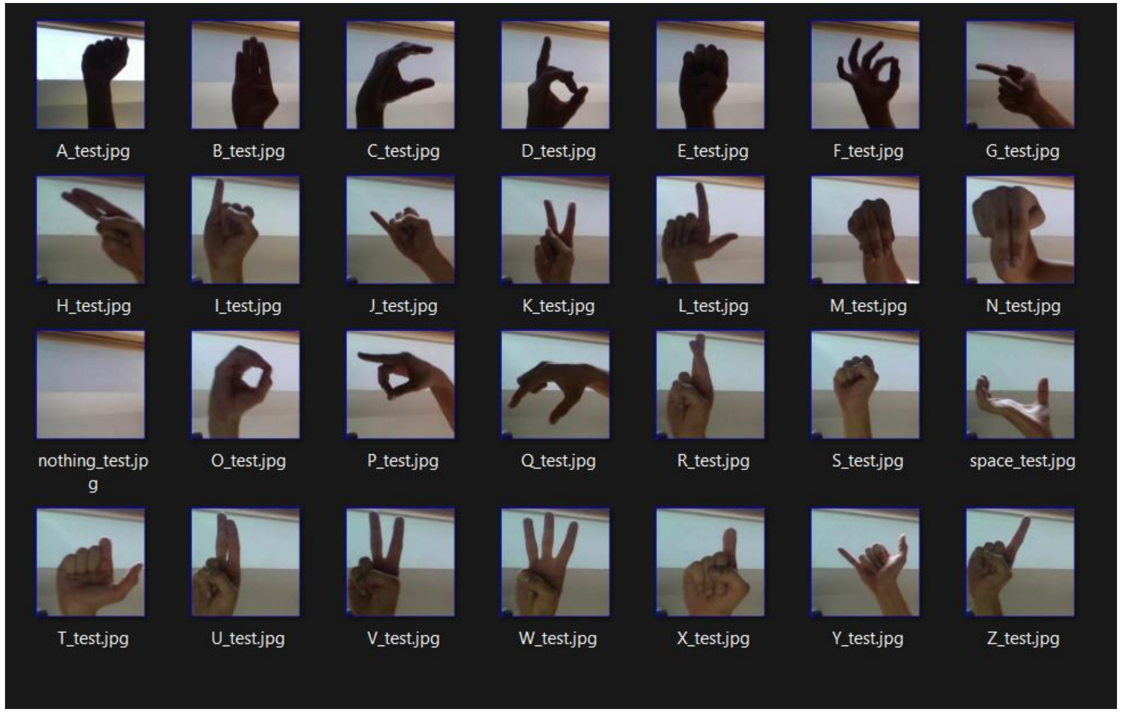
Dataset: ASL A–Z + “space”, “nothing”

Balanced at ±2000 samples each

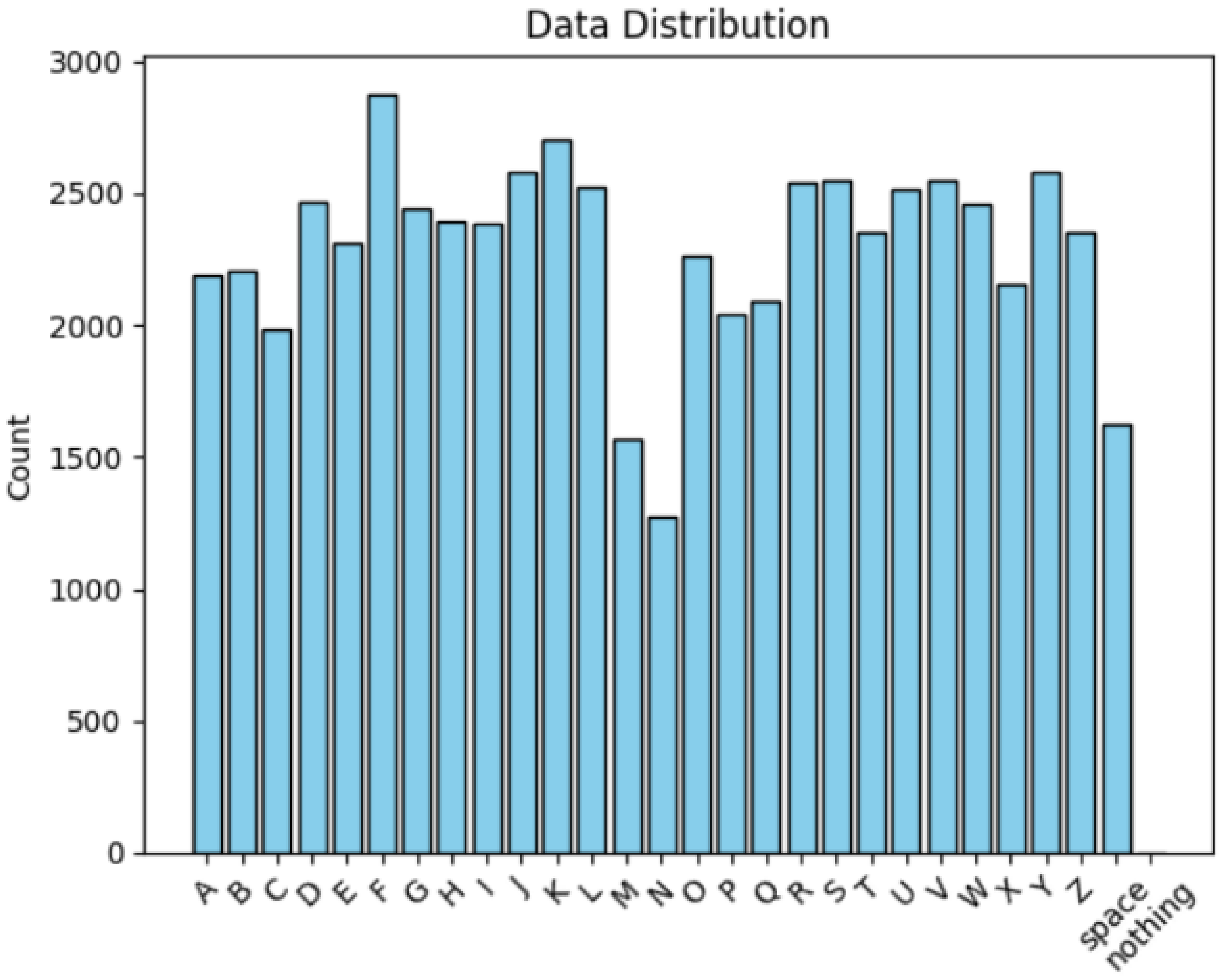
Preprocessing:

- Extract 21 landmarks → 63D feature vector
- Standardization (mean, std)
- Label encoding for gesture classes

Split: 80% train, 10% val, 10% test



CLASS DATASET DISTRIBUTIO N



MODEL & TRAINING

Multilayer Perceptron (MLP)

Input: 63 landmark features

Hidden layers: ReLU

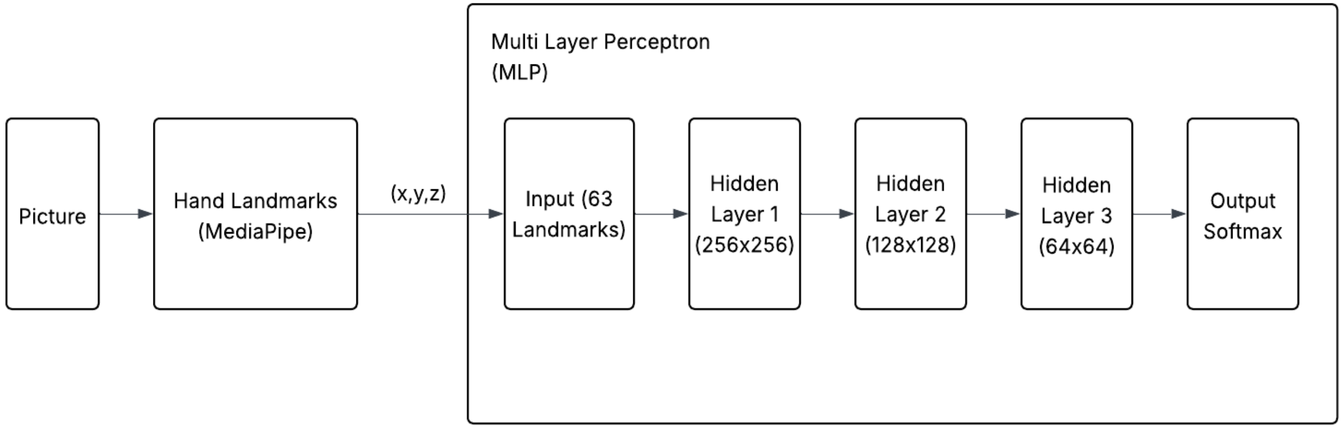
Output: 28 classes

Training: batch 32, early stopping, LR scheduling

Results:

Test Accuracy: 99.23%

Test Loss: 0.033

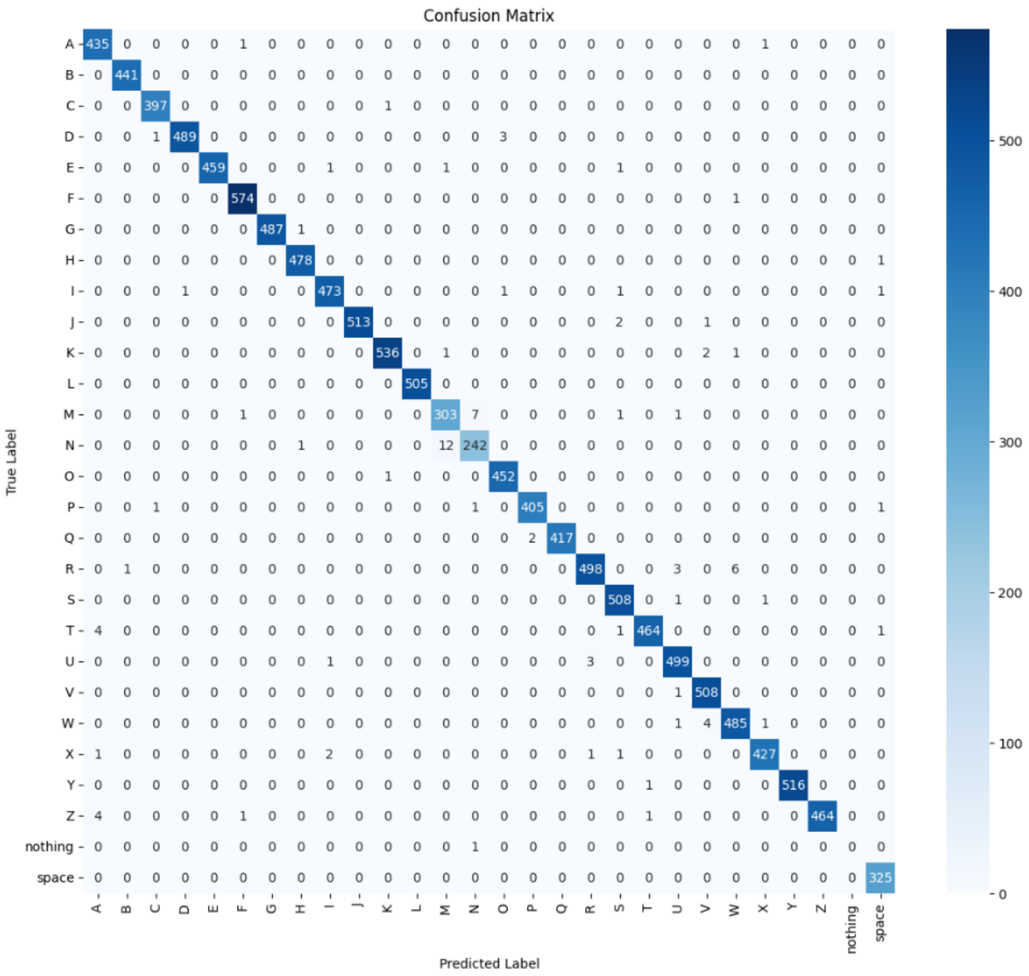


TRAINING CONFIGURATION

Component	Value
Data training	X_train, y_train
Batch size	32
Epochs (maks.)	100
Validation data	(X_val, y_val)
Callbacks	callbacks (lihat tabel di bawah)
Verbose	1

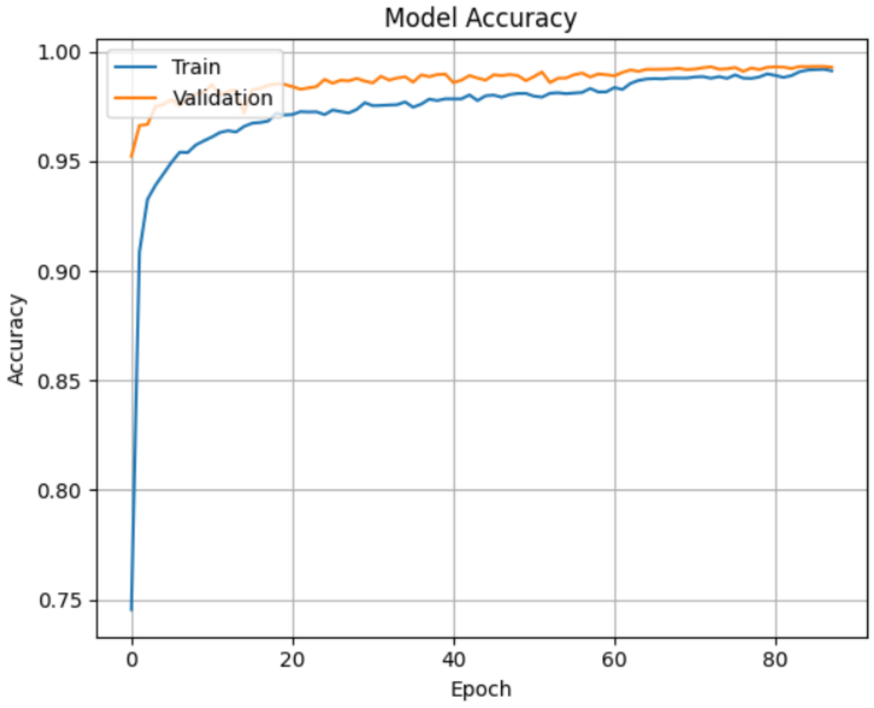
Callback	Parameter	Nilai
EarlyStopping	monitor	val_loss
	patience	15
	restore_best_weights	True
ReduceLROnPlateau	monitor	val_loss
	factor	0.5
	patience	10
	min_lr	1e-7

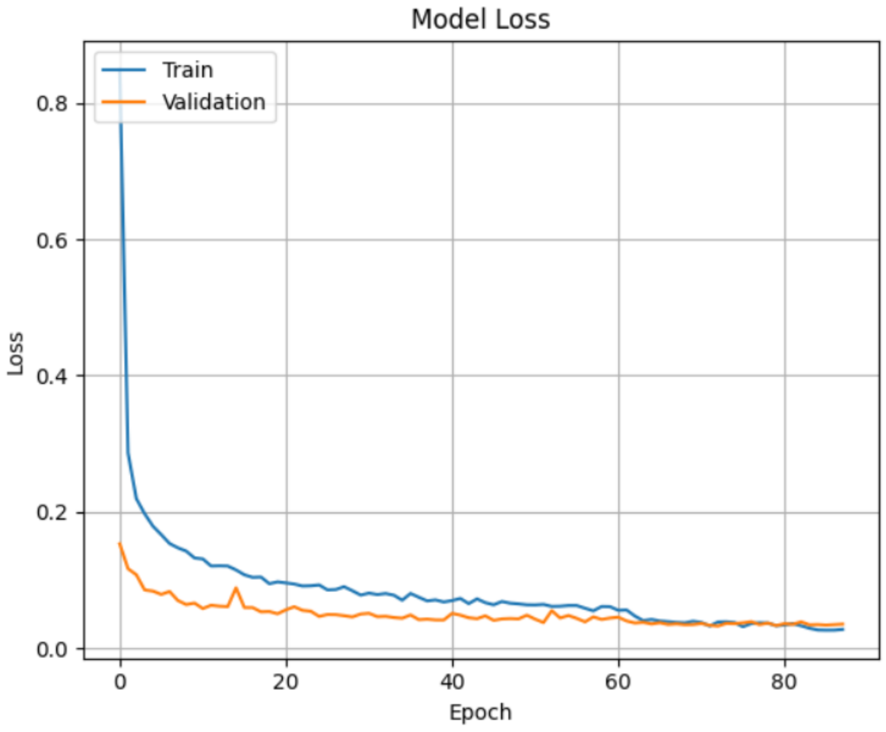
EVALUATION



- **Overall pattern:** the matrix is strongly **diagonal**, meaning the model is **highly accurate** for most classes (most off-diagonal values are 0 or very small).
- Best performing classes: **F, L, S, Y** (high confidence, perfect or near-perfect).
- Most confused pair: **M ↔ N** (visually similar gestures or data overlap).
- Weakest class: “nothing” (likely due to class imbalance or labeling issue).

EVALUATION (ACCURACY & LOSS)





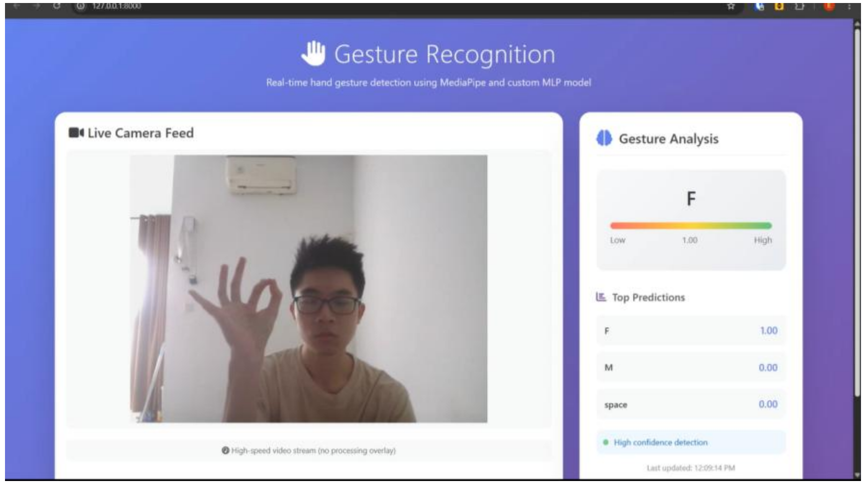
EVALUATION & REAL-TIME RESULTS

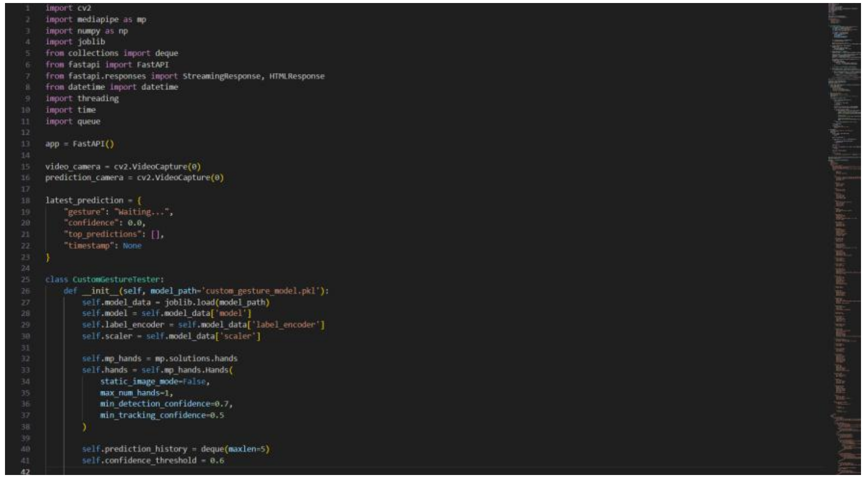
Strengths:
•Responsive predictions (<1s stabilization)
•Smooth video stream
•Reliable detection in normal lighting & simple backgrounds

Limitations:
•Lower accuracy in poor lighting
•Dataset trained only on left hand → right-hand accuracy drops
•MediaPipe is main CPU bottleneck at high FPS
•Angle/rotation variance impacts landmark stability

SYSTEM APP IMPLEMENTATION

- Framework: **FastAPI + OpenCV**
- Two-thread architecture:
  - Thread 1: high-FPS webcam streaming
  - Thread 2: MediaPipe + MLP prediction
- Shared state: gesture, confidence, top-3 predictions
- Temporal smoothing reduces flicker
- UI designed for minimal latency





REFLECTION & NEXT STEPS

What worked:
•Landmark-based pipeline extremely lightweight
•High offline accuracy with simple MLP
•Real-time performance usable with threading

Issues found:
•Lighting + orientation sensitivity
•Left-hand dataset bias
•MediaPipe jitter in non-ideal environments

Improvements:
•Add mirrored right-hand data
•Augment lighting/angle variations
•Consider hybrid model: landmarks + image features
•Migrate MediaPipe to GPU/ONNX for speed