Best MediaPipe Sign Language Recognition Models with Clear Guides

Based on my research, here are the **top 3 best models** that have clear guides, proven results, and easily accessible code using MediaPipe:

1. JaspreetSingh's MediaPipe + MLP Model (RECOMMENDED) [1] [2]

Why it's the best for your project:

- **Proven superior performance** over CNNs for real-time recognition
- **Simple MLP architecture** - easier to train than LSTM
- **Complete project structure** with all necessary files
- **Modern approach** (2025) with documented better results than MobileNetV2

Key Features:

- Real-time ASL letter recognition using MediaPipe Hands
- MLP (Multi-Layer Perceptron) trained on landmark coordinates
- Dynamic sentence formation with SPACE and DELETE controls
- Complete Jupyter notebooks for training and testing

Project Structure:

```
 SIGN_TO_SENTENCE_PROJECT/
├── Asl_Sign_Data/ # Raw ASL dataset
├── asl_mediapipe_keypoints_dataset.csv # Preprocessed features
├── asl_mediapipe_mlp_model.h5 # Trained model
├── Mediapipe_Training.ipynb # Training script
├── Combined_Architecture.ipynb # Real-time demo
├── requirements.txt
```

Code Example:

```
import mediapipe as mp
import numpy as np
import tensorflow as tf

# Load trained MLP model
mlp_model = tf.keras.models.load_model("asl_mediapipe_mlp_model.h5")

# Initialize MediaPipe Hands
```

```
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(min_detection_confidence=0.7, min_tracking_confidence=0.7)

# Predict sign from landmarks
def predict_sign(landmarks):
    input_data = np.array(landmarks).flatten().reshape(1, -1)
    prediction = mlp_model.predict(input_data)
    return np.argmax(prediction)
```

GitHub: https://github.com/JaspreetSingh-exe/Sign-Language-Recognition-System [2]

2. Nicholas Renotte's LSTM Tutorial (COMPREHENSIVE) [3] [1]

Why it's excellent:

- **Most comprehensive tutorial** with 2+ hour detailed walkthrough
- **470+ GitHub stars** - highly validated by community
- **Complete pipeline** from data collection to real-time inference
- **Professional-level implementation** using LSTM for sequence modeling

What you'll learn:

- Extract MediaPipe Holistic keypoints (face + pose + hands)
- Build sequence datasets (30 frames per gesture)
- Train LSTM deep learning model with Keras/TensorFlow
- Real-time sign language prediction
- Model evaluation with confusion matrices

Key Steps:

1. Install dependencies (MediaPipe, TensorFlow, OpenCV)
2. Extract keypoints using MediaPipe Holistic
3. Collect sequences for training (30 sequences × 30 frames per action)
4. Build LSTM model with 2-3 layers
5. Train and evaluate model
6. Real-time inference with smoothing

Architecture:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(30, 1662)),  # Holistic features
    LSTM(128, return_sequences=True),
    LSTM(64),
    Dropout(0.3),
    Dense(64, activation='relu'),
```

```
    Dense(num_actions, activation='softmax')
])
```

YouTube Tutorial: https://youtu.be/doDUihpj6ro (2:27:13 complete walkthrough)[1]
GitHub: https://github.com/nicknochnack/ActionDetectionforSignLanguage

3. Computer Vision Engineer's Scikit-Learn Approach (BEGINNER-FRIENDLY)[4]

Why it's perfect for beginners:

- **Simple classification** using Random Forest/SVM
- **Fast training** - no deep learning complexity
- **Clear 55-minute tutorial** with step-by-step explanation
- **Immediate results** for quick prototyping

Workflow:

1. **Data Collection**: Capture hand landmarks using MediaPipe Hands
2. **Feature Extraction**: Flatten 21 landmarks × 3 coordinates = 63 features per hand
3. **Model Training**: Use Random Forest Classifier
4. **Real-time Testing**: Webcam inference with confidence thresholds

Code Structure:

```
import mediapipe as mp
import cv2
import numpy as np
from sklearn.ensemble import RandomForestClassifier
import pickle

# Extract landmarks
def extract_landmarks(results):
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            landmarks = []
            for lm in hand_landmarks.landmark:
                landmarks.extend([lm.x, lm.y, lm.z])
            return np.array(landmarks)
    return np.zeros(63)

# Train classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

YouTube Tutorial: https://youtu.be/MJCSjXepaAM (55:37)[4]
GitHub: https://github.com/computervisioneng/sign-language-detector-python

Recommended Implementation Strategy for Your 4 Models:

Phase 1: Start Simple (Week 1-2)

Use **Computer Vision Engineer's approach** to build:[4]

- English letters classifier (A-Z)
- Arabic letters classifier (32 letters)

Phase 2: Scale Up (Week 3-4)

Implement **JaspreetSingh's MLP approach** for:[2]

- English words (start with 10-20 words)
- Arabic words (start with 10-20 words)

Phase 3: Advanced (Week 5-6)

Optionally upgrade to **Nicholas Renotte's LSTM** for better sequence modeling if needed.[1]

Why These Are The Best Options:

1. **Proven Results**: All have documented high accuracy (85-99%)
2. **Complete Code**: Full implementations available on GitHub
3. **Clear Documentation**: Step-by-step tutorials with explanations
4. **Active Community**: Well-maintained with community support
5. **Real-time Performance**: Optimized for live webcam inference
6. **Scalable**: Easy to add new gestures/languages

Quick Start Recommendation:

Start with **JaspreetSingh's MediaPipe + MLP approach** because:[2]

- **Most practical** for graduation project timeline
- **Better real-time performance** than LSTM
- **Complete project structure** already provided
- **Easier to understand and modify**
- **Modern architecture** with proven superiority over CNNs

The repository includes everything you need: data collection scripts, training notebooks, pre-trained models, and real-time inference code. You can directly adapt this for both English and Arabic by changing the dataset and class labels.

⁂

1. https://www.youtube.com/watch?v=doDUihpj6ro

2. https://github.com/JaspreetSingh-exe/Sign-Language-Recognition-System
3. https://www.youtube.com/watch?v=Bpaa560qN9U
4. https://www.youtube.com/watch?v=MJCSjXepaAM
5. https://arxiv.org/html/2506.11154v1
6. https://www.freecodecamp.org/news/create-a-real-time-gesture-to-text-translator/
7. https://www.youtube.com/watch?v=pDXdlXlaCco
8. https://github.com/gspagare/Real-Time-Gesture-Recognition-Using-Mediapipe-and-BiLSTM
9. https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer/python
10. https://blog.tensorflow.org/2021/11/3D-handpose.html
11. https://github.com/AmElmo/sign_language_detector
12. https://mediapipe.readthedocs.io/en/latest/solutions/hands.html
13. https://github.com/AkramOM606/American-Sign-Language-Detection
14. https://github.com/evarghese563/Sign-Language-Detection-Using-LSTM
15. https://www.kaggle.com/code/vnefedov/asl-recognition-mediapipe-3
16. https://dev.to/yoshan0921/developing-an-asl-app-with-kaggles-top-model-and-customized-mediapipe-gesture-model-5eaa
17. https://github.com/mhash1m/sign-language-detection
18. https://www.youtube.com/watch?v=pG4sUNDOZFg
19. https://www.ijraset.com/best-journal/sign-language-detection-using-mediapipe-and-ml-184
20. https://github.com/youngsoul/sign-language-detection-with-tfod2
21. https://ijrar.org/papers/IJRAR1DUP055.pdf
22. https://www.techrxiv.org/users/798477/articles/1151368/master/file/data/Navendu_iSeS_2024-2/Navendu_iSeS_2024-2.pdf
23. https://paperswithcode.com/task/sign-language-recognition?page=8&q=
24. https://github.com/topics/sign-language-recognition-system
25. https://github.com/SomyanshAvasthi/Sign-Language-Detection-using-MediaPipe
26. https://github.com/AvishakeAdhikary/Realtime-Sign-Language-Detection-Using-LSTM-Model
27. https://github.com/kinganupamdutta27/SIGN-LANGUAGE-RECOGNITION-WITH-MACHINE-LEARNING
28. https://metehanozdeniz.com/projects/deep-learning/2025-06-05-sign-language-recognition/
29. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker/python