**Gesture recognition system using MediaPipe, Vision Transformers (ViT), and LSTM-based deep learning models involves a multi-stage pipeline. Here's an outline of how you can build such a system**

**1. Hand Landmark Detection with MediaPipe**

* **MediaPipe** is an excellent framework for **real-time hand tracking**. It provides an easy way to detect hand landmarks (like knuckles, fingertips, etc.), which you can use as inputs for your gesture recognition model.

Steps:

* + Use the **MediaPipe Hands** solution to extract 21 hand landmarks in real time.
  + For each frame, MediaPipe provides the coordinates of these landmarks, which represent the position of key points on the hand.

**2. Feature Extraction with Vision Transformers (ViT)**

* After obtaining the hand landmarks from MediaPipe, feed them into a **Vision Transformer (ViT)** model for feature extraction. Since ViT works well with image patches, you can convert the hand landmark data into a **2D patch representation**.

Steps:

* + Flatten the hand landmark coordinates and segment them into small patches. You can treat each frame or a series of frames as a "sequence of patches."
  + Use a pre-trained ViT model (fine-tune it) or train a custom one for your gesture recognition task.

**3. Temporal Modeling with LSTM**

* Once you have features from the ViT model, you can pass them to an **LSTM (Long Short-Term Memory)** network to capture the **temporal dependencies** between the frames. LSTMs are excellent for handling sequential data, such as a series of hand movements over time, which is critical in dynamic gesture recognition.

Steps:

* + Feed the ViT features (from multiple frames) into the LSTM network.
  + The LSTM will output a sequence that represents the temporal dynamics of the gesture.

**4. Training the Model**

* **Data Preparation**: For training, you need labeled gesture datasets where each frame has corresponding gesture labels (e.g., "swipe left," "swipe right"). Use the landmarks extracted from MediaPipe as input data, and the labels represent different gestures.
* **Training Strategy**:
  + Train the ViT model to extract meaningful features from each frame.
  + Train the LSTM model to capture the temporal flow of gestures from the sequence of frames.

**5. Inference**

* For real-time gesture recognition, you'll continuously process frames through MediaPipe, extract features using ViT, and predict gestures using the LSTM. You can use a sliding window over a sequence of frames to ensure that the system responds quickly and accurately.

**6. Performance Optimization**

* For deployment on resource-constrained devices, consider using techniques like **model quantization** and **pruning** to reduce the size of the models.

This hybrid system leveraging **MediaPipe for real-time hand tracking**, **ViT for spatial feature extraction**, and **LSTM for temporal modeling** is well-suited for complex gesture recognition tasks.

There are existing projects that combine **MediaPipe**, **LSTM**, and other deep learning models for gesture recognition. For example:

1. **Sign Language Recognition Project**: This project uses **MediaPipe** for extracting hand landmarks and an **LSTM** model for recognizing different sign language gestures in real-time video. The system can detect various hand signs like "hello," "thank you," and "goodbye," using sequences of keypoints from the video frames to predict actions​
2. **Real-Time Hand Gesture Recognition**: Another project uses **MediaPipe’s hand model** to extract hand and body landmarks, then feeds these features into an **LSTM neural network** for hand gesture classification. The pre-trained model detects gestures like "like" and "ok"​
3. **Gesture Recognition with LSTM and MediaPipe**: This project implements a **deep learning-based hand gesture recognition** using **LSTM** and **MediaPipe** for keypoint extraction. The model can be trained on custom datasets and can handle real-time gesture recognition​

These projects could serve as great starting points if you're planning to build or extend a gesture recognition system using these technologies. You can customize them to add support for **Vision Transformers (ViT)** alongside the existing LSTM framework to enhance performance further.

 If you need **efficiency** and are working with smaller datasets or lower computational power, **Inception-v3** is likely the better choice.

 For **complex** or **large-scale** gesture recognition tasks, particularly if you have access to large datasets and more powerful hardware, **Vision Transformers** are expected to outperform CNNs like Inception-v3.