**Experiment no.1**

Aim Develop a hand gesture recognition model that can accurately identify and classify different hand gestures from image or video data, enabling intuitive human-computer interaction and gesture-based control systems.

Source : IDE used: Kaggle Notebook

Dataset: https://lnkd.in/dUu7hxEm

Github link: <https://lnkd.in/gbVds9tF>

Theory: Developing a hand gesture recognition model involves several steps, including data collection, preprocessing, model selection, training, and evaluation. Below is a comprehensive guide to building such a model

**1.Define the Problem statement**

**Objective**: Accurately identify and classify different hand gestures.

**Scope**: Determine the types of gestures to recognize (e.g., thumbs up, peace sign, etc.) and the context of use (e.g., controlling a device, interacting with a software application).

2. **Data Collection**

**Gesture Dataset**:

* Collect a diverse set of hand gesture images or videos. Ensure the dataset includes various hand shapes, sizes, and skin tones.
* Use existing datasets like the **"Hand Gesture Recognition Dataset"**, **"ASL Alphabet"**, or **"HandNet"** if applicable. Alternatively, collect your own data using cameras or smartphones.

3. **Data Preprocessing**

**Normalization**:

* Resize images or video frames to a consistent size.
* Normalize pixel values (e.g., scaling to the range [0, 1]).

4. **Model Selection**

**Feature Extraction**:

* For image-based gestures, use Convolutional Neural Networks (CNNs) for feature extraction.
* For video-based gestures, consider using models that combine CNNs with Recurrent Neural Networks (RNNs) or Transformers to capture temporal patterns.

5. **Training the Model**

Train the model on the training dataset while monitoring performance on the validation set to avoid overfitting

6. **Evaluation**

Evaluate the model using metrics such as accuracy, precision, recall, and F1 score

7. **Deployment**

Deploy the model in a real-time application using frameworks like TensorFlow Lite or ONNX for efficient inference.

8. **Continuous Improvement**

Collect user feedback and continuously refine the model based on real-world performance.

Conclusion: