Model :

Keypoint classifier: (For hand gestures)

This folder consists of :

1. keypoint_classifier_label.csv

   Consists of various labels we are using. By label I mean the hand gestures.

2. keypoint.csv

   This CSV stores the data for the gestures in the form of Handmark information or simply the coordinates. By handmark information what I mean is the set of key points on the hand that the app tracks (e.g., tips of fingers, wrist, etc.).During the training phase, when we train our model on any new gesture, this document gets auto updated to a new feature.

   - The first column of this CSV acts as index and maps to various labels mentioned in keypoint_classifier_label.csv starting from 0.

3. Keypoint_classifier.py

   Here the KeyPointClassifier class is designed to classify key points (such as hand or body joints) using a pre-trained **TensorFlow Lite model**. It provides an efficient way to detect and classify specific patterns or gestures based on keypoints extracted from images or video frames.
   (Further explanations in code)

Point history classifier: (For finger gestures)

This folder consists of :

1. point_history_classifier_label.csv

   Consists of various labels we are using. By label I mean the finger gestures.

2. point_history.csv

   This CSV stores the data for the finger gestures in the form of Handmark information (coordinates). By keeping track of how the hand's position and orientation change, the model can classify more complex gestures that involve motion (like a swipe or rotation). During the training phase, when we train our model on any new gesture, this document gets auto updated to a new feature.

   - The first column of this CSV acts as index and maps to various labels mentioned in keypoint_classifier_label.csv starting from 0.

3. point_history_classfier.py

   Here the PointHistoryClassifier class is designed to classify a sequence of points, representing a history of past positions (such as points in time from a moving object). This class uses a pre-trained **TensorFlow Lite model** to classify the historical data and determine whether the sequence matches a predefined pattern or action.
   (Further explanations in code)

Utils :

1. cvfpscalc.py

   The CvFpsCalc class is designed to calculate the frames per second (FPS) for a given video stream or sequence of images using OpenCV.

app.py :

This file is a hand gesture recognition application that uses **MediaPipe** for hand tracking and **TensorFlow Lite models** for classifying gestures based on hand landmarks and their history.

- Here **Hand landmark** means the set of key points on the hand that the app tracks (e.g., tips of fingers, wrist, etc.)
- History means the sequence or movement of these landmarks over time. By keeping track of how the hand's position and orientation change, the model can classify more complex gestures that involve motion (like a swipe or rotation).

keypoint_classification.py :

So this is the main training scripts used to train a model for classifying hand keypoints, based on data stored in a CSV file keypoint.csv. The script uses tensorflow for model building, training, and conversion to TensorFlow Lite (TFLite) for mobile/embedded deployment (incase if we want to).
(Refer code comments for explanation)

point_history_classification.py :

It's similar to keypoint_classification.py but there is a small difference. Keypoint_classification.py is assigned to train the model for static hand gestures like open palm, victory sign etc. On the other hand point_history_classification.py is designed for classifying sequences of hand keypoint data over time like the clockwise movement of fingers etc.