Hand Sign Recognition

1. What is the idea?

- Idea Introduction:

  Sign language is manual communication commonly used by people who are deaf. Sign language is not universal; people who are deaf from different countries speak different sign languages. The gestures or symbols in sign language are organized in a linguistic way. Each individual gesture is called a sign. Each sign has three distinct parts: the handshape, the position of the hands, and the movement of the hands.

  Hand Sign recognition has become an important task in computer vision.

- Toolkit:

  - Mediapipe
  - CV2
  - Sklearn

Objective:

- Create a Real Time program for hand sign recognition.

Phase 1: Initiation:

First we need to define body parts which need to be detected (Hand, face, pose,etc..)

- Main functions:

```
      results = mp.solutions.holistic
```

One of the mediapipe pipelines contains optimized hand components which allows for holistic tracking,thus enabling the model to simultaneously detect hand landmarks. Results contain a number of hand landmarks.

```
cap = cv2.VideoCapture(0)

while cap.isOpened():

        ret, frame = cap.read()
```

Opens Webcam for real time video and starts to captures frames

Phase 2:Create Csv file For landmarks:

- Main functions:

```
num_coords = len(results.right_hand_landmarks.landmark)

landmarks = ['class']

for val in range(1, num_coords+1):

    landmarks += ['x{}'.format(val),
'y{}'.format(val),'z{}'.format(val), 'v{}'.format(val)]

with open('coords.csv', mode='w', newline='') as f:

    csv_writer = csv.writer(f, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)

csv_writer.writerow(landmarks)
```

1. Save number of landmarks in num_coords which equals 21
2. Create a list (Landmarks) which carries the header of the csv file.
3. Create a csv file and write the list which carries column names.

Phase 3:Create Dataset for signs you need:

- ### Main function:

  It's typically like the first phase plus the part which saves landmark values in a csv file.

```
try:

            # Extract Hand landmarks

            hand = results.right_hand_landmarks.landmark

            hand_row = list(np.array([[landmark.x, landmark.y,
landmark.z, landmark.visibility] for landmark in hand]).flatten())

            # Concate rows

            row = hand_row

            # Append class name

            row.insert(0, class_name)

            # Export to CSV

            with open('D:\ApplAi\Hand recognition\coords.csv',
mode='a', newline='') as f:

                csv_writer = csv.writer(f, delimiter=',',
quotechar='"', quoting=csv.QUOTE_MINIMAL)

                csv_writer.writerow(row)

        except:

            pass
```

1. We need to convert results from object to 1D array using numpy, So np.array converts it.
2. Why 1D? When converting results to np array it becomes a 2D array so, it won't append correctly in csv. To make it append correctly, First flatten 2D to become in 1D so it can be inserted as a row.

Phase 3:Data preprocessing:

Main functions:

1. Load data into data frame using pandas
2. Define Features (x) and Label or class(y)
3. Split data for training and testing using train_test_split(), 70% training 30% testing.
4. Create pipeline for feature scaling using StandardScalar()

```
df = pd.read_csv('D:\ApplAi\Hand recognition\coords.csv')

X = df.drop('class', axis=1) # features

y = df['class'] # target value

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=1234)

pipelines = {

    'lr':make_pipeline(StandardScaler(), LogisticRegression()),

    'rc':make_pipeline(StandardScaler(), RidgeClassifier()),

    'rf':make_pipeline(StandardScaler(), RandomForestClassifier()),

}
```

Phase 4:Machine Learning Models:

1. Main idea is to classify hand Sign based on input landmarks. Which makes it classification problem
2. Models:
    1. Logistic regression
    2. Random Forest
    3. Ridge Classifier
3. Save model in pickle file.

```
with open('handsign.pkl', 'wb') as f:

    pickle.dump(fit_models['rf'], f)
```

Phase 5 :Make real time :

Same as the Third phase plus the part which predicts class of hand sign based on landmarks using Machine Learning saved model.

Load:

```
with open('handsign.pkl', 'rb') as f:

    model = pickle.load(f)
```

prediction:

```
try:

            # Extract Hand landmarks

            hand = results.right_hand_landmarks.landmark

            hand_row = list(np.array([[landmark.x, landmark.y, landmark.z,
landmark.visibility] for landmark in hand]).flatten())

            # Concate rows

            row = hand_row

            X = pd.DataFrame([row])

            sign_class = model.predict(X)[0]

            sign_prop = model.predict_proba(X)[0]
```

1. Load model from pickle file for prediction.
2. Convert data into data frame
3. Predict class using model.predict