Install And Import Dependencies

In [1]:

```
# pip install mediapipe opencv-python
```

In [2]:

```
import mediapipe as mp
import cv2
import numpy as np
import uuid # uniform unique identifier
import os
```

Detect Hands

In [4]:

```
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands
```

In [7]:

```
with mp_hands.Hands(min_detection_confidence=0.8,min_tracking_confidence=0.5) a
s hands:
    cap = cv2.VideoCapture(0)
    while cap.isOpened():
        ret , frame = cap.read()

        # BGR 2 RGB
        image = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB)

        # Set Flag
        image.flags.writeable = False

        # Detection
        results = hands.process(image)

        # Set Flag
        image.flags.writeable = True

        # RGB 2 BGR
        image = cv2.cvtColor(image , cv2.COLOR_RGB2BGR)

        # Rendering the Result
        if results.multi_hand_landmarks:
            for num , hand in enumerate(results.multi_hand_landmarks):

                mp_drawing.draw_landmarks(image , hand , mp_hands.HAND_CONNECTI
ONS, mp_drawing.DrawingSpec(color=(121,22,76) , thickness=2 ,circle_radius=4),#
(B , G ,R)
mp_drawing.DrawingSpec(color=(121,44,250) , thickness=2 , circle_radius=2))


        cv2.imshow('Hand Tracking',image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

Output Images

In [8]:

```
os.mkdir("Output Images")
```

In [9]:

```
with mp_hands.Hands(min_detection_confidence=0.8,min_tracking_confidence=0.5) a
s hands:
    cap = cv2.VideoCapture(0)
    while cap.isOpened():
        ret , frame = cap.read()

        # BGR 2 RGB
        image = cv2.cvtColor(frame , cv2.COLOR_BGR2RGB)

        # Set Flag
        image.flags.writeable = False

        # Detection
        results = hands.process(image)

        # Set Flag
        image.flags.writeable = True

        # RGB 2 BGR
        image = cv2.cvtColor(image , cv2.COLOR_RGB2BGR)

        # Rendering the Result
        if results.multi_hand_landmarks:
            for num , hand in enumerate(results.multi_hand_landmarks):

                mp_drawing.draw_landmarks(image , hand , mp_hands.HAND_CONNECTI
ONS, mp_drawing.DrawingSpec(color=(121,22,76) , thickness=2 ,circle_radius=4),#
(B , G ,R)
mp_drawing.DrawingSpec(color=(121,44,250) , thickness=2 , circle_radius=2))

        Save Image
        cv2.imwrite(
            os.path.join(
                "Output Images",
                '{}.jpg'.format(uuid.uuid1())),
                image)

        cv2.imshow('Hand Tracking',image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

In [ ]: