**Mudra recognition**

Immerse in the world of Bharatanatyam using the mudra recognition tool. Use the tool to recognise mudras and learn about them through viniyogas from Abhinaya Dharpana by Nandikeshwara.

///

Code:

!pip install mediapipe

# import necessary packages

import cv2

import numpy as np

import mediapipe as mp

import tensorflow as tf

from keras.models import load\_model

import tf\_keras as k3

from google.colab import drive

drive.mount('/content/drive')

# initialize mediapipe

mpHands = mp.solutions.hands

hands = mpHands.Hands(max\_num\_hands=1, min\_detection\_confidence=0.7)

mpDraw = mp.solutions.drawing\_utils

# Load the gesture recognizer model

model = k3.models.load\_model('/content/drive/My Drive/Higher Education/Gesture Recognition/mp\_hand\_gesture')

# Load class names

f = open('/content/drive/ My Drive/Higher Education/Gesture Recognition/mp\_hand\_gesture/gesture.names', 'r')

classNames = f.read().split('\n')

f.close()

print(classNames)

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='photo.jpg', quality=0.8):

  js = Javascript('''

    async function takePhoto(quality) {

      const div = document.createElement('div');

      const capture = document.createElement('button');

      capture.textContent = 'Capture';

      div.appendChild(capture);

      const video = document.createElement('video');

      video.style.display = 'block';

      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);

      div.appendChild(video);

      video.srcObject = stream;

      await video.play();

      // Resize the output to fit the video element.

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.

      //await new Promise((resolve) => capture.onclick = resolve);

      await new Promise(resolve => setTimeout(resolve, 5000));

      const canvas = document.createElement('canvas');

      canvas.width = video.videoWidth;

      canvas.height = video.videoHeight;

      canvas.getContext('2d').drawImage(video, 0, 0);

      stream.getVideoTracks()[0].stop();

      div.remove();

      return canvas.toDataURL('image/jpeg', quality);

    }

    ''')

  display(js)

  data = eval\_js('takePhoto({})'.format(quality))

  binary = b64decode(data.split(',')[1])

  return binary

  with open(filename, 'wb') as f:

    f.write(binary)

  return filename

from types import NoneType

while True:

    image\_bytes = take\_photo()

    image\_np = np.frombuffer(image\_bytes, dtype=np.uint8)

    frame = cv2.imdecode(image\_np, cv2.IMREAD\_COLOR)

    x, y, c = frame.shape

    # Now 'frame' holds the image data as a NumPy array

    # You can perform further operations on 'frame' here, e.g., display it:

    # from google.colab.patches import cv2\_imshow

    # cv2\_imshow(frame)

    framergb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

    result = hands.process(framergb)

    className = '' # Initialize className before the if block

    if result.multi\_hand\_landmarks:

      landmarks = []

      for handslms in result.multi\_hand\_landmarks:

          for lm in handslms.landmark:

              # print(id, lm)

              lmx = int(lm.x \* x)

              lmy = int(lm.y \* y)

              landmarks.append([lmx, lmy])

          # Drawing landmarks on frames

          mpDraw.draw\_landmarks(frame, handslms, mpHands.HAND\_CONNECTIONS)

          # Predict gesture

          prediction = model.predict([landmarks])

          # print(prediction)

          classID = np.argmax(prediction)

          className = classNames[classID]

          print(className)

print('MUDRA DETAILS')

print('=============')

if(className == "thumbs up") or (className == 'thumbs down'):

      print("Mudra is Shikhara")

      print('''Viniyoga Sloka:

    Madana kamuka sthambaecha Nishcaya pitrukarmani

    Oshtra pravishtaroopani  Radhana prashnabhavanae

    Linga nastheethivachanae  Samarana katibandakarshana

    Parirambhavidikrama Gandani nada Shikarayujyatahe Bharatadibi

    Meaning:

    God of love, a Bow, a Pillar, To decide, Making offering to Manes,

    Lips, To Enter or to pour, Tooth, Questioning,

    Shiva Lingam, Saying “I don’t know”, An act of remembering, To act, To tie around the waist,

    To embrace, Ringing of bells, Peak''')

elif(className == "fist"):

    print("Mudra is Mushti")

elif(className == "stop" or (className == 'live long')):

    print("Mudra is Patakam")

else:

  print("Mudra not mapped yet")

    if cv2.waitKey(1) == ord('q'):

      break