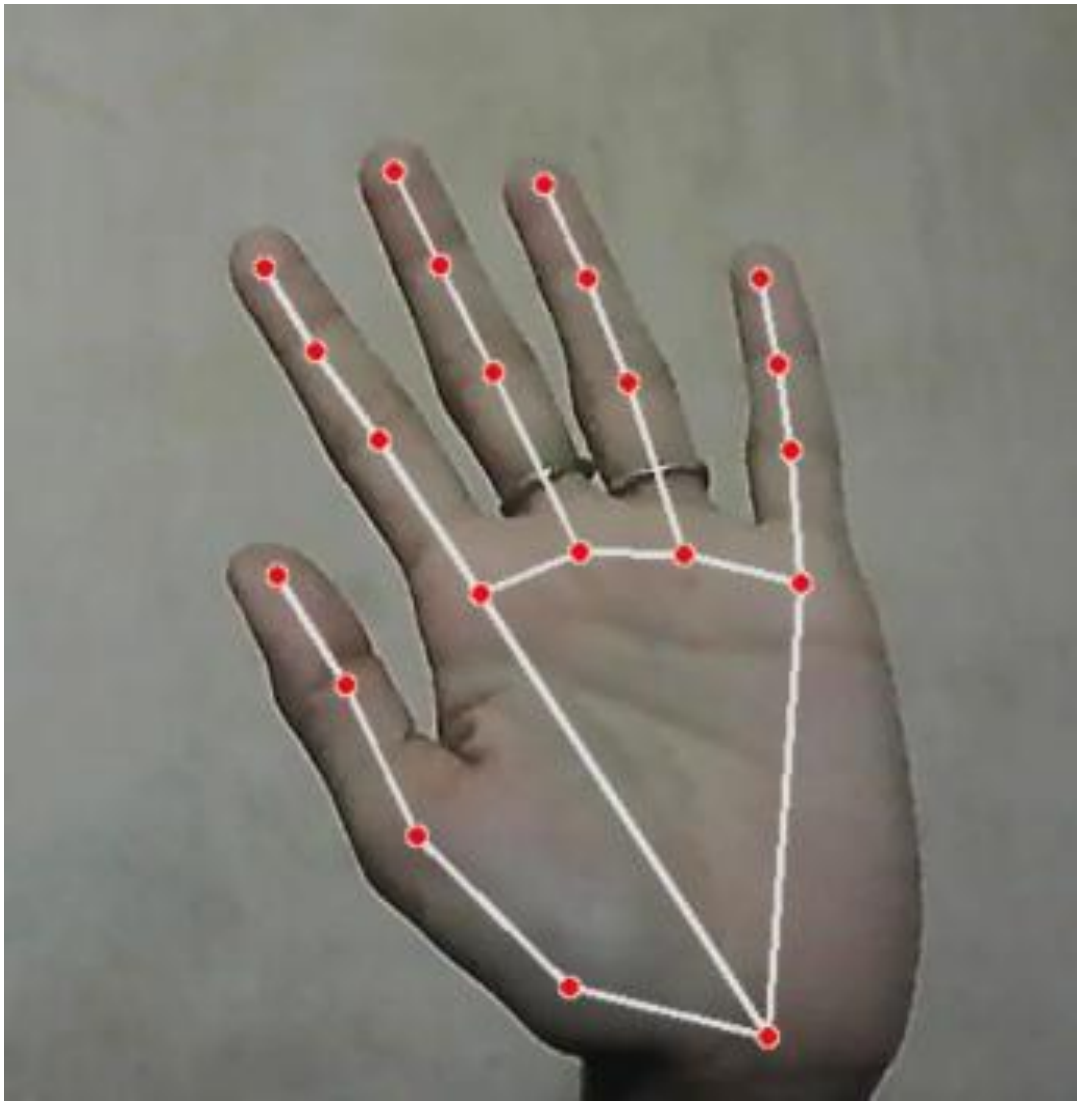
CODE FOR WEBCAM (Hand gestures)

```
********************************************************************************
import cv2
import numpy as np
import tensorflow as tf
import mediapipe as mp

# ================= LOAD MODEL =================
model = tf.keras.models.load_model("hand_gesture_cnn.h5", compile=False)

class_names = [
    'c', 'down', 'fist', 'fist_moved', 'index',
    'l', 'ok', 'palm', 'palm_moved', 'thumb'
]

IMG_SIZE = 224

# ================= MEDIAPIPE =================
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.7
)

mp_draw = mp.solutions.drawing_utils

# ================= WEBCAM =================
cap = cv2.VideoCapture(0)
```

```
print("Press Q to quit")

while True:
    ret, frame = cap.read()
    if not ret:
        break

    frame = cv2.flip(frame, 1)
    h, w, _ = frame.shape

    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    result = hands.process(rgb)

    if result.multi_hand_landmarks:
        for hand_landmarks in result.multi_hand_landmarks:
            x_list = []
            y_list = []

            for lm in hand_landmarks.landmark:
                x_list.append(int(lm.x * w))
                y_list.append(int(lm.y * h))

            xmin, xmax = max(min(x_list) - 20, 0), min(max(x_list) + 20, w)
            ymin, ymax = max(min(y_list) - 20, 0), min(max(y_list) + 20, h)

            hand_img = frame[ymin:ymax, xmin:xmax]

            if hand_img.size != 0:
                hand_img = cv2.resize(hand_img, (IMG_SIZE, IMG_SIZE))
                hand_img = hand_img / 255.0
```

```
            hand_img = np.expand_dims(hand_img, axis=0)

            preds = model.predict(hand_img, verbose=0)
            class_id = np.argmax(preds)
            confidence = preds[0][class_id]

            label = f"{class_names[class_id]} ({confidence:.2f})"

            cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0,255,0), 2)
            cv2.putText(
               frame, label,
               (xmin, ymin - 10),
               cv2.FONT_HERSHEY_SIMPLEX,
               0.8, (0,255,0), 2
            )

        mp_draw.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)

    cv2.imshow("Hand Gesture Recognition", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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



OUTPUT : HAND GESTURES RECOGNITION

****