```
!pip install --upgrade pip
!pip install mediapipe==0.10.5 opencv-python==4.8.0.76
import os
os.kill(os.getpid(), 9)
```

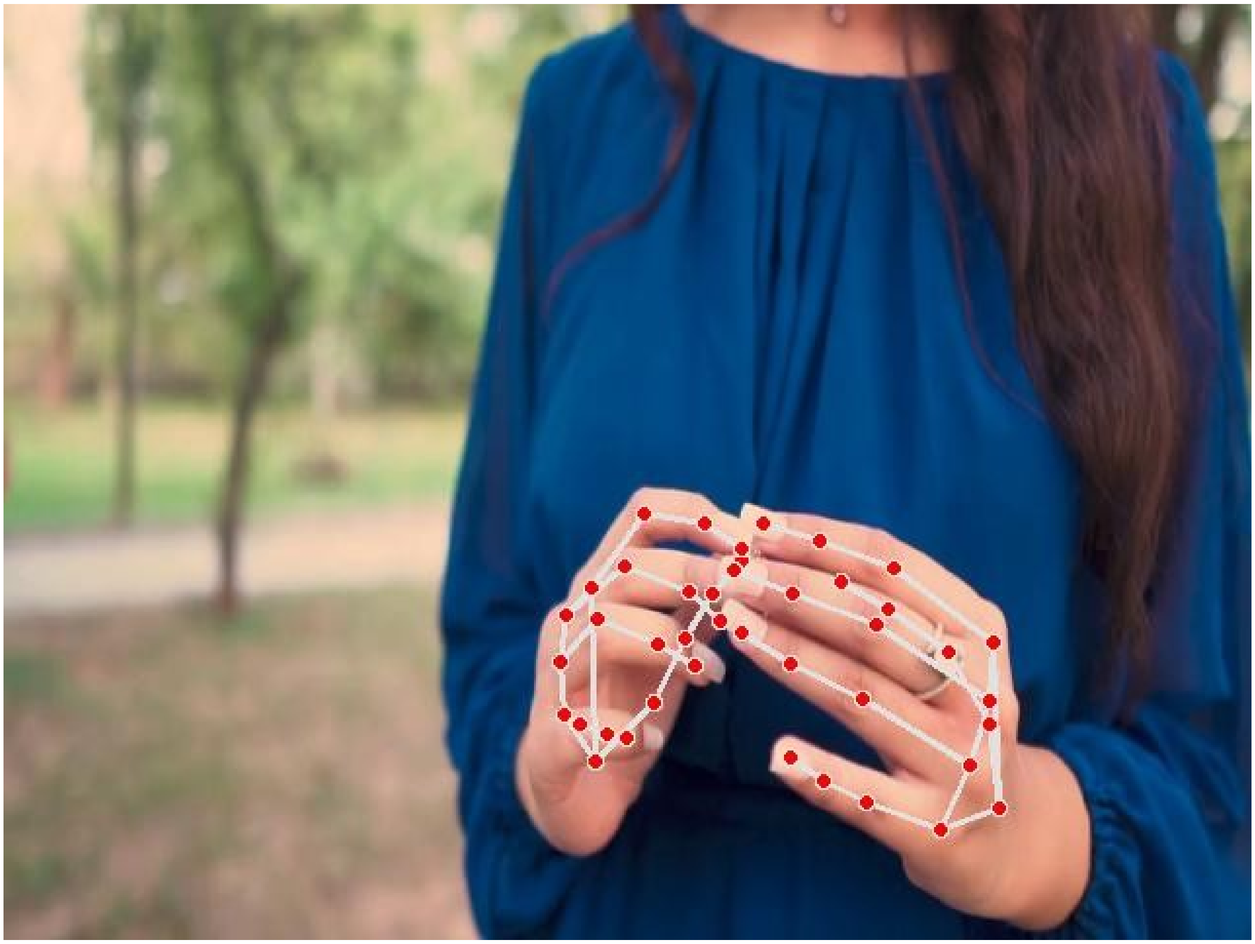
```
Requirement already satisfied: pip in /usr/local/lib/python3.11/dist-
packages (24.1.2)
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 21.1 MB/s eta
0:00:00
pting uninstall: pip
    Found existing installation: pip 24.1.2
    Uninstalling pip-24.1.2:
      Successfully uninstalled pip-24.1.2
Successfully installed pip-25.0.1
Collecting mediapipe==0.10.5
  Downloading mediapipe-0.10.5-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.6 kB)
Collecting opencv-python==4.8.0.76
  Downloading opencv_python-4.8.0.76-cp37-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Requirement already satisfied: absl-py in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(1.4.0)
Requirement already satisfied: attrs>=19.1.0 in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(25.3.0)
Requirement already satisfied: flatbuffers>=2.0 in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(25.2.10)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(3.10.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(1.26.4)
Requirement already satisfied: opencv-contrib-python in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(4.11.0.86)
Collecting protobuf<4,>=3.11 (from mediapipe==0.10.5)
  Downloading protobuf-3.20.3-py2.py3-none-any.whl.metadata (720
bytes)
Requirement already satisfied: sounddevice>=0.4.4 in
/usr/local/lib/python3.11/dist-packages (from mediapipe==0.10.5)
(0.5.1)
Requirement already satisfied: CFFI>=1.0 in
/usr/local/lib/python3.11/dist-packages (from sounddevice>=0.4.4-
>mediapipe==0.10.5) (1.17.1)
```

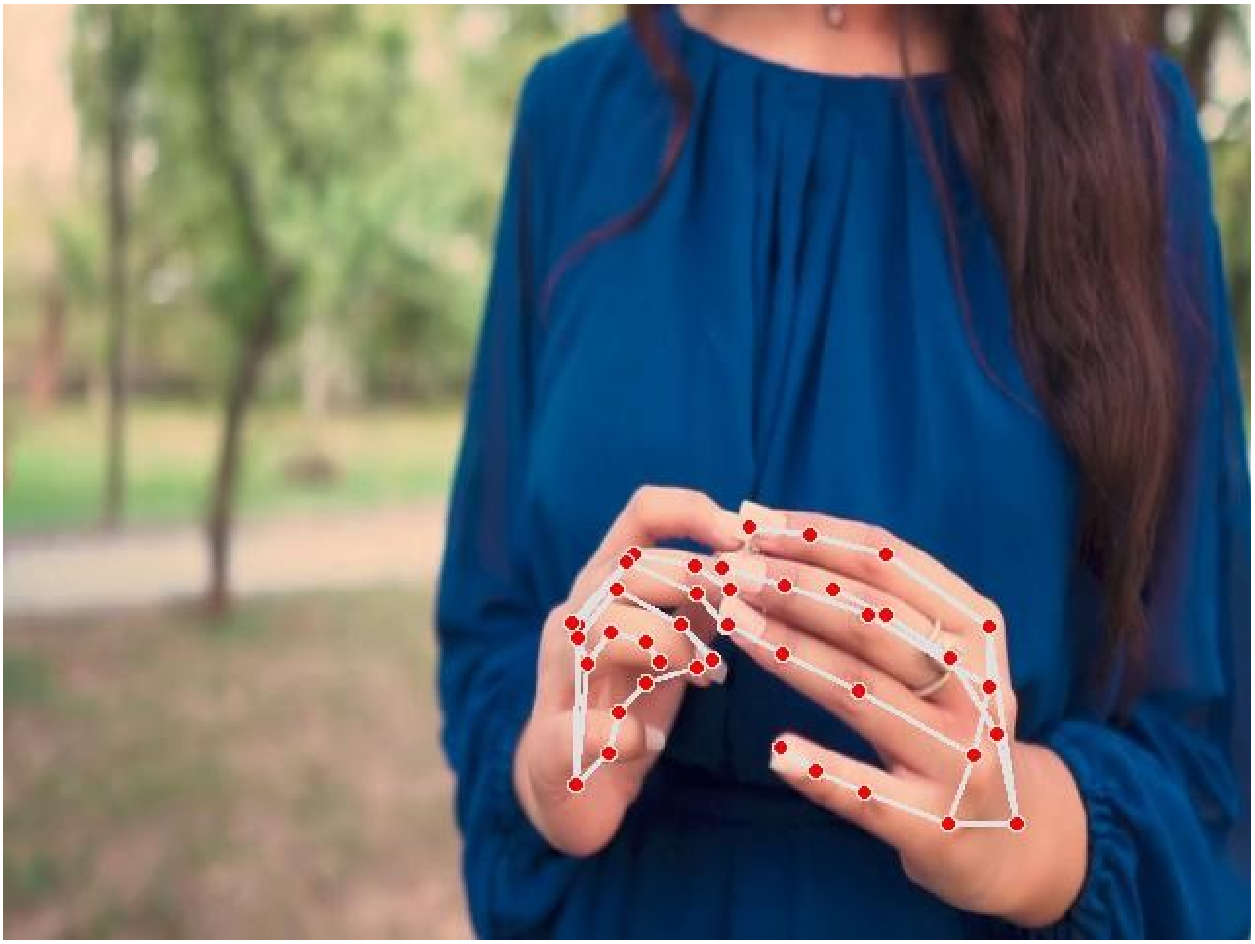
```
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (1.3.2)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (24.2)
Requirement already satisfied: pillow>=8 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/dist-packages (from matplotlib-
>mediapipe==0.10.5) (2.8.2)
Requirement already satisfied: pycparser in
/usr/local/lib/python3.11/dist-packages (from CFFI>=1.0-
>sounddevice>=0.4.4->mediapipe==0.10.5) (2.22)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7-
>matplotlib->mediapipe==0.10.5) (1.17.0)
Downloading mediapipe-0.10.5-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (33.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 33.5/33.5 MB 56.6 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (61.7 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.7/61.7 MB 60.1 MB/s eta
0:00:00
ediapipe
  Attempting uninstall: protobuf
    Found existing installation: protobuf 4.25.6
    Uninstalling protobuf-4.25.6:
      Successfully uninstalled protobuf-4.25.6
  Attempting uninstall: opencv-python
    Found existing installation: opencv-python 4.11.0.86
    Uninstalling opencv-python-4.11.0.86:
      Successfully uninstalled opencv-python-4.11.0.86
  Attempting uninstall: mediapipe
    Found existing installation: mediapipe 0.10.21
    Uninstalling mediapipe-0.10.21:
      Successfully uninstalled mediapipe-0.10.21
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
grpcio-status 1.71.0 requires protobuf<6.0dev,>=5.26.1, but you have
protobuf 3.20.3 which is incompatible.
tensorflow-metadata 1.17.1 requires protobuf<6.0.0,>=4.25.2;
python_version >= "3.11", but you have protobuf 3.20.3 which is
incompatible.
ydf 0.11.0 requires protobuf<6.0.0,>=5.29.1, but you have protobuf
3.20.3 which is incompatible.
Successfully installed mediapipe-0.10.5 opencv-python-4.8.0.76
protobuf-3.20.3

import cv2
import mediapipe as mp
from google.colab.patches import cv2_imshow
```

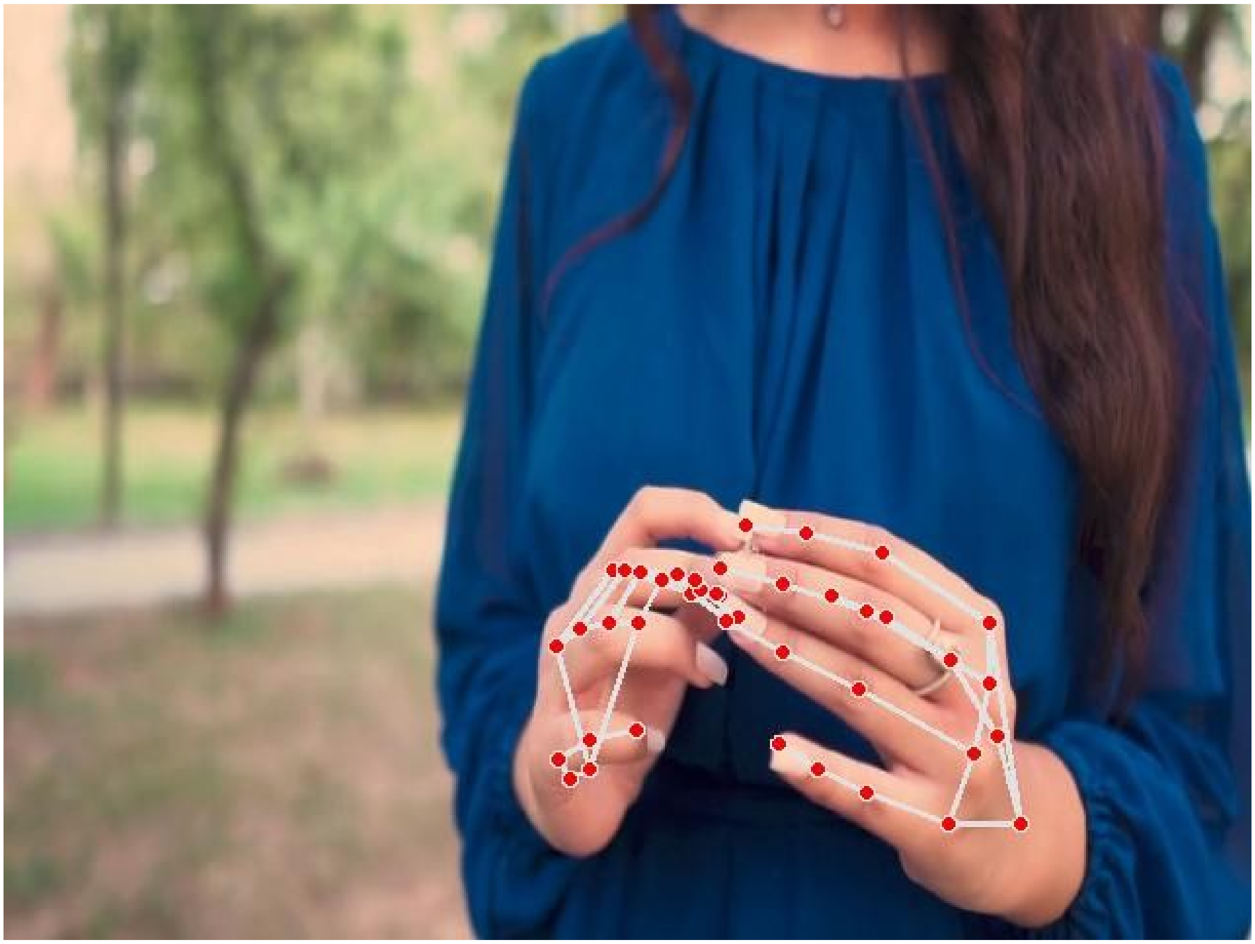
Finger tracking in video

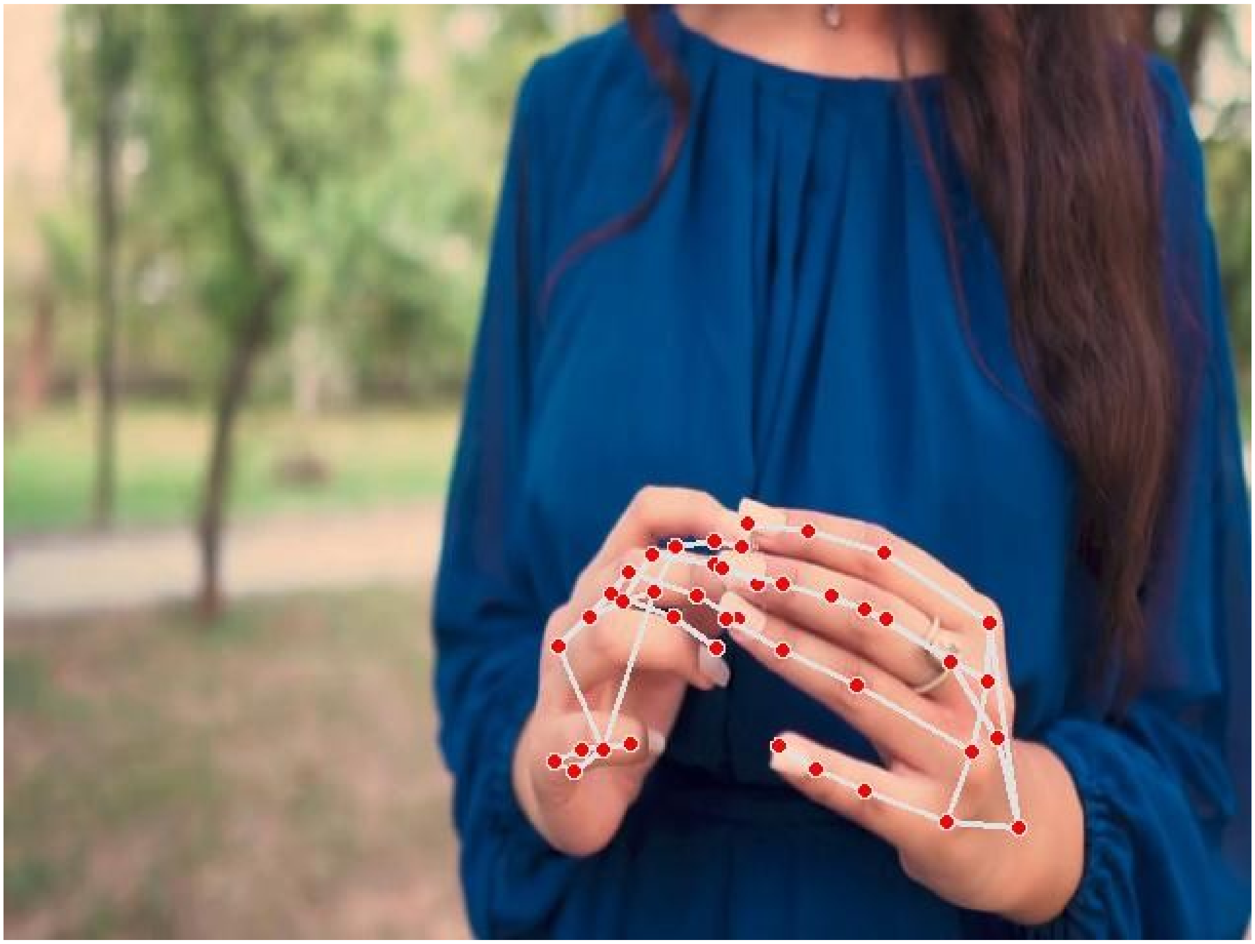
Here I used MediaPipe to track the fingers in a video so that maybe I can further try to reach ring detection. And this is done by making frames out of video only 5 I have printed.

```
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
hands = mp_hands.Hands(static_image_mode=False,
                       max_num_hands=2,
                       min_detection_confidence=0.7)

video_path = "/content/146131-788410158_small.mp4"

cap = cv2.VideoCapture(video_path)
frame_count = 0
while cap.isOpened() and frame_count < 5:
    ret, frame = cap.read()
    if not ret:
        break

    frame = cv2.resize(frame, (640, 480))
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(rgb)

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)

    cv2_imshow(frame)
    frame_count += 1

cap.release()
```







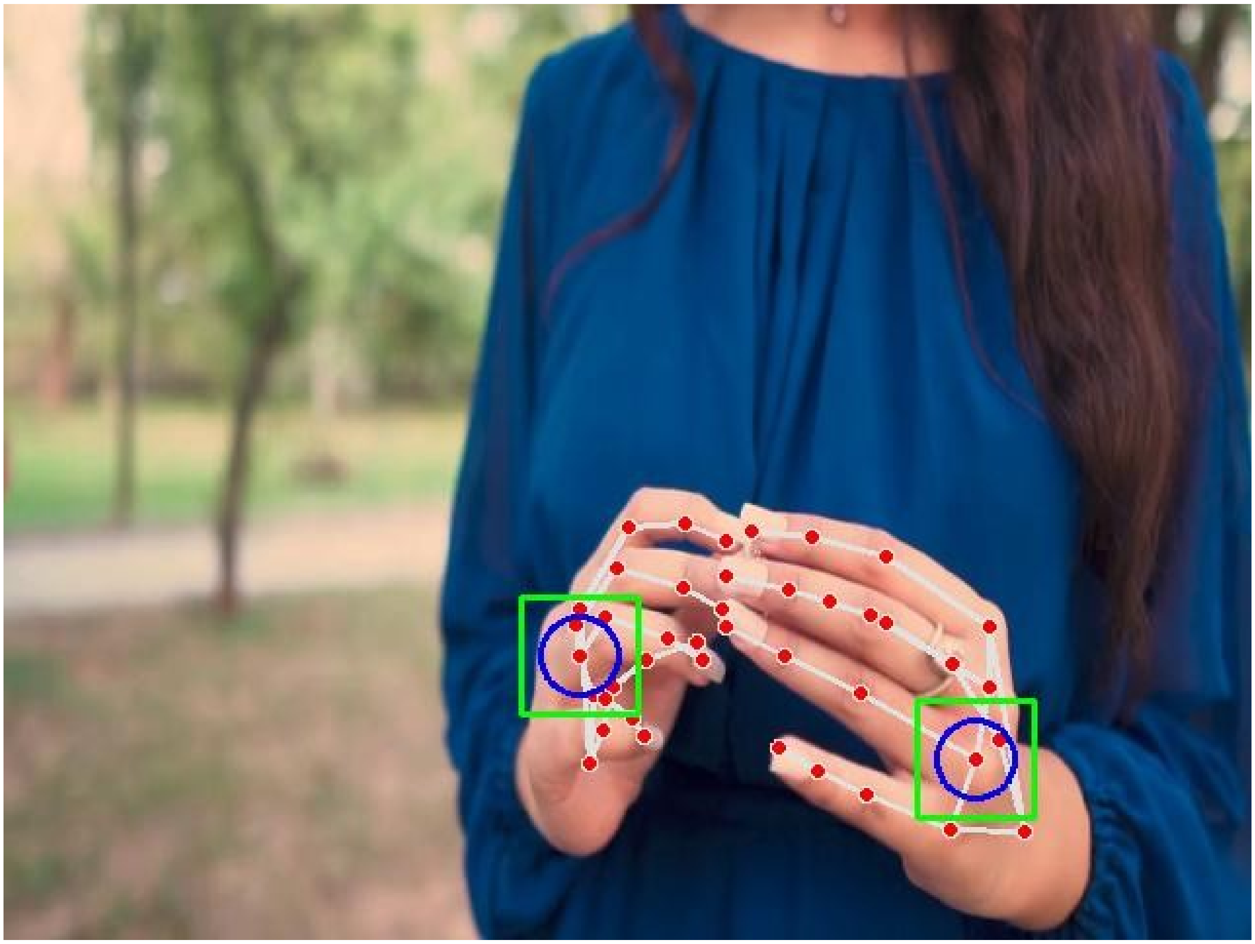


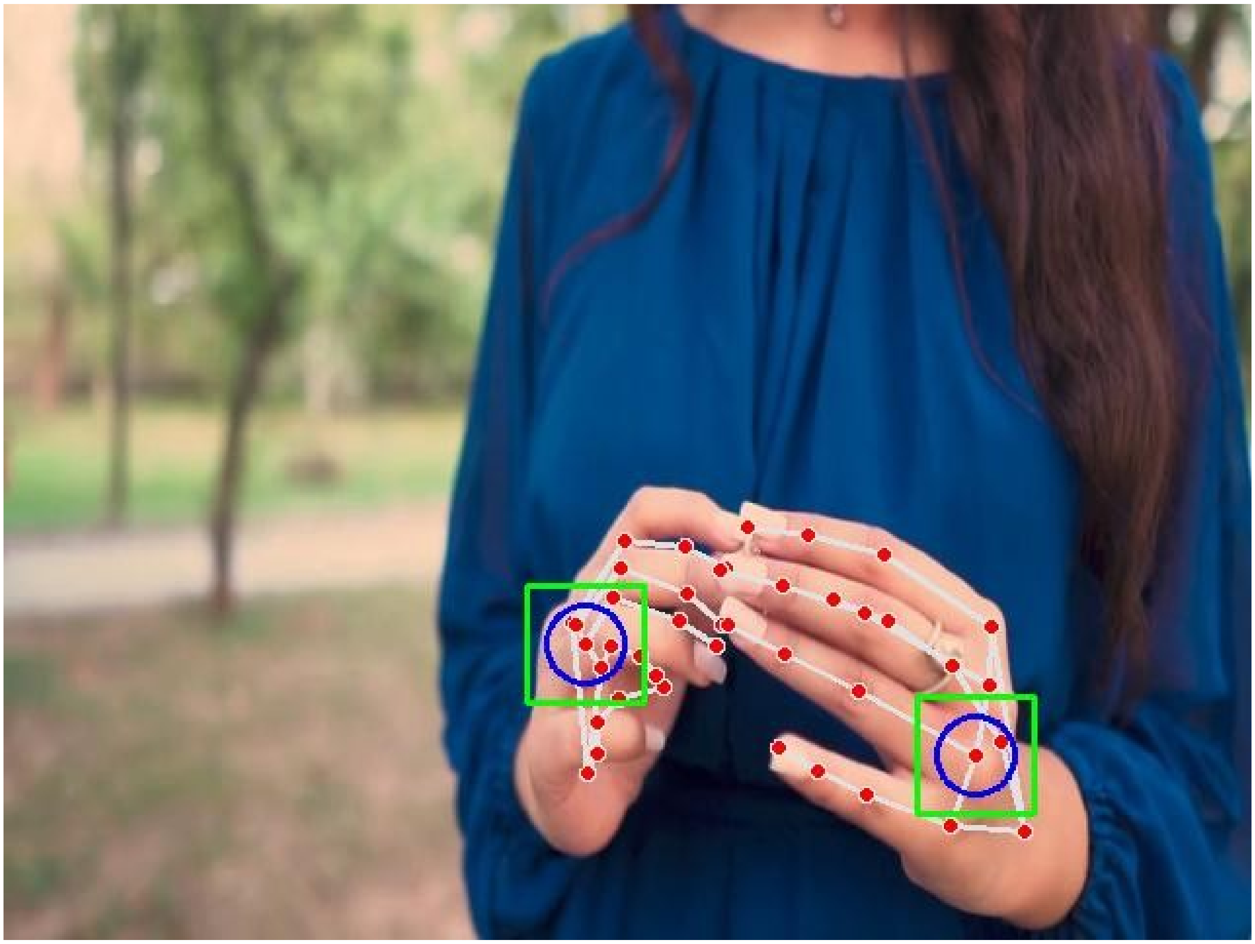
Ring detection

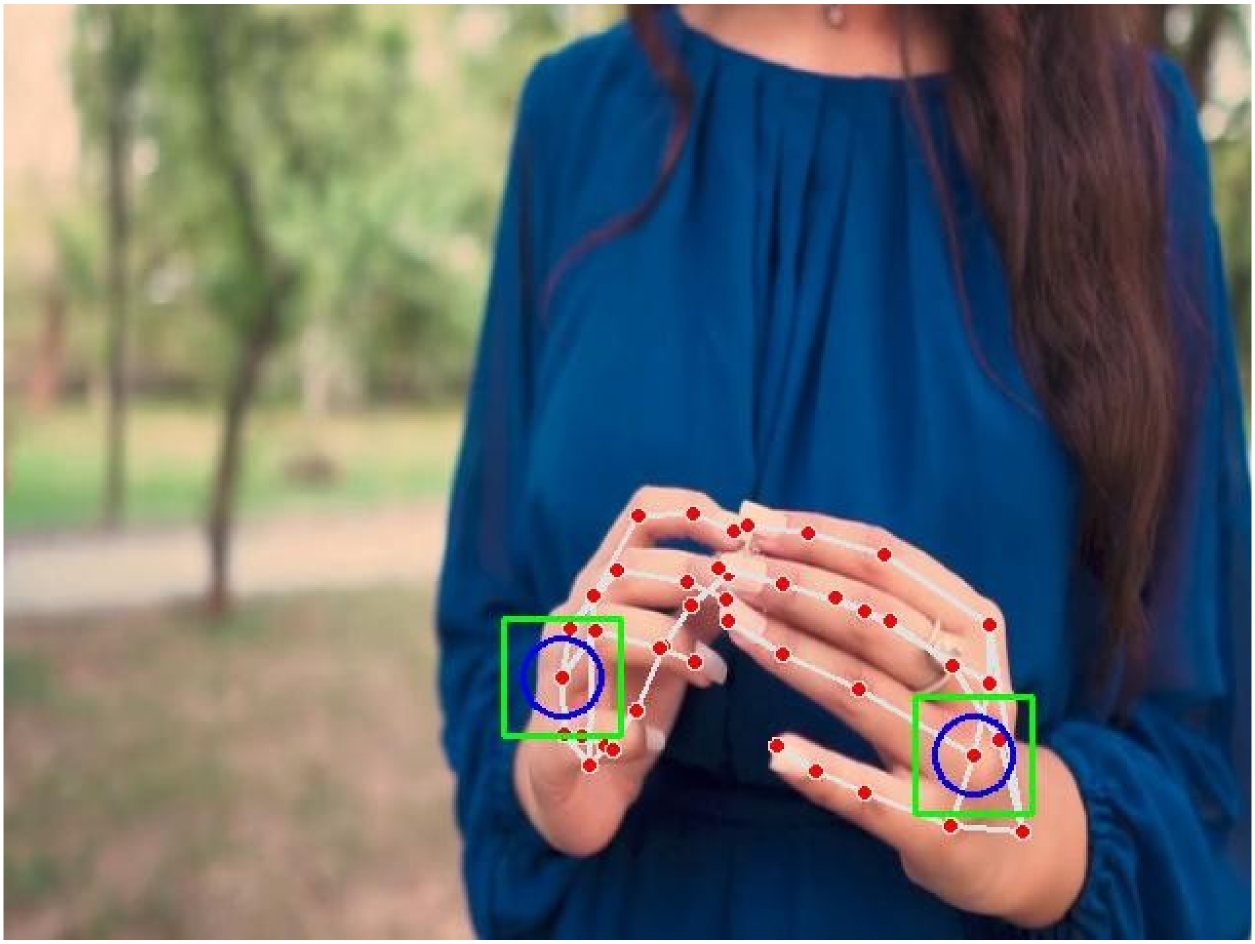
In the following part I wanted to detect the ring, which was a failure as you can see from the results the MediaPipe is circling even those fingers whcih has no ring at all.

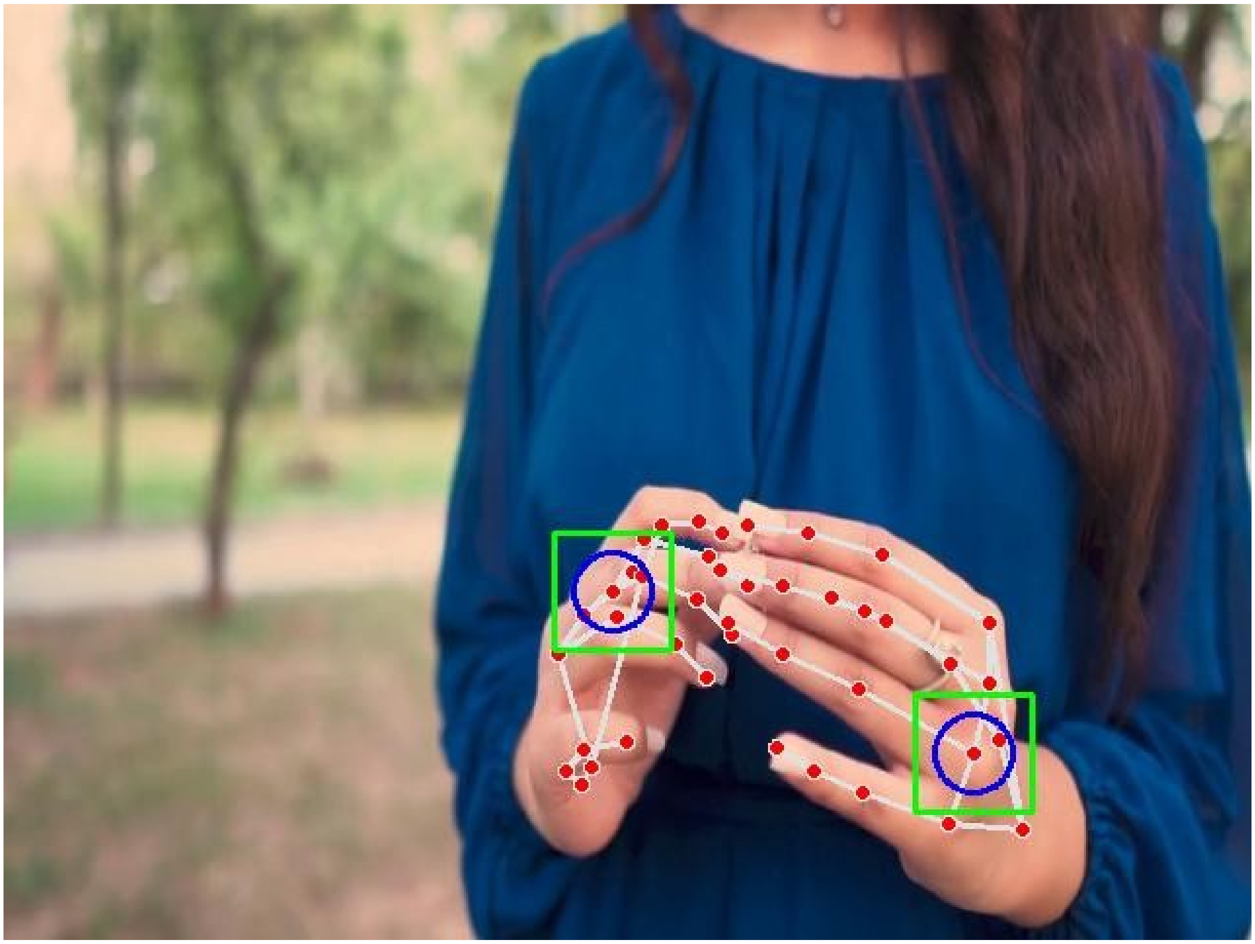
```
import numpy as np

def draw_ring_zone(frame, hand_landmarks):
    for hand in hand_landmarks:
        ring_base = 
hand.landmark[mp_hands.HandLandmark.RING_FINGER_MCP]  # base of the 
ring finger
        h, w, _ = frame.shape
        x, y = int(ring_base.x * w), int(ring_base.y * h)
        cv2.circle(frame, (x, y), 20, (255, 0, 0), 2)  # Circle around 
the base of the ring finger
        cv2.rectangle(frame, (x-30, y-30), (x+30, y+30), (0, 255, 0), 
2)

cap = cv2.VideoCapture(video_path)
```

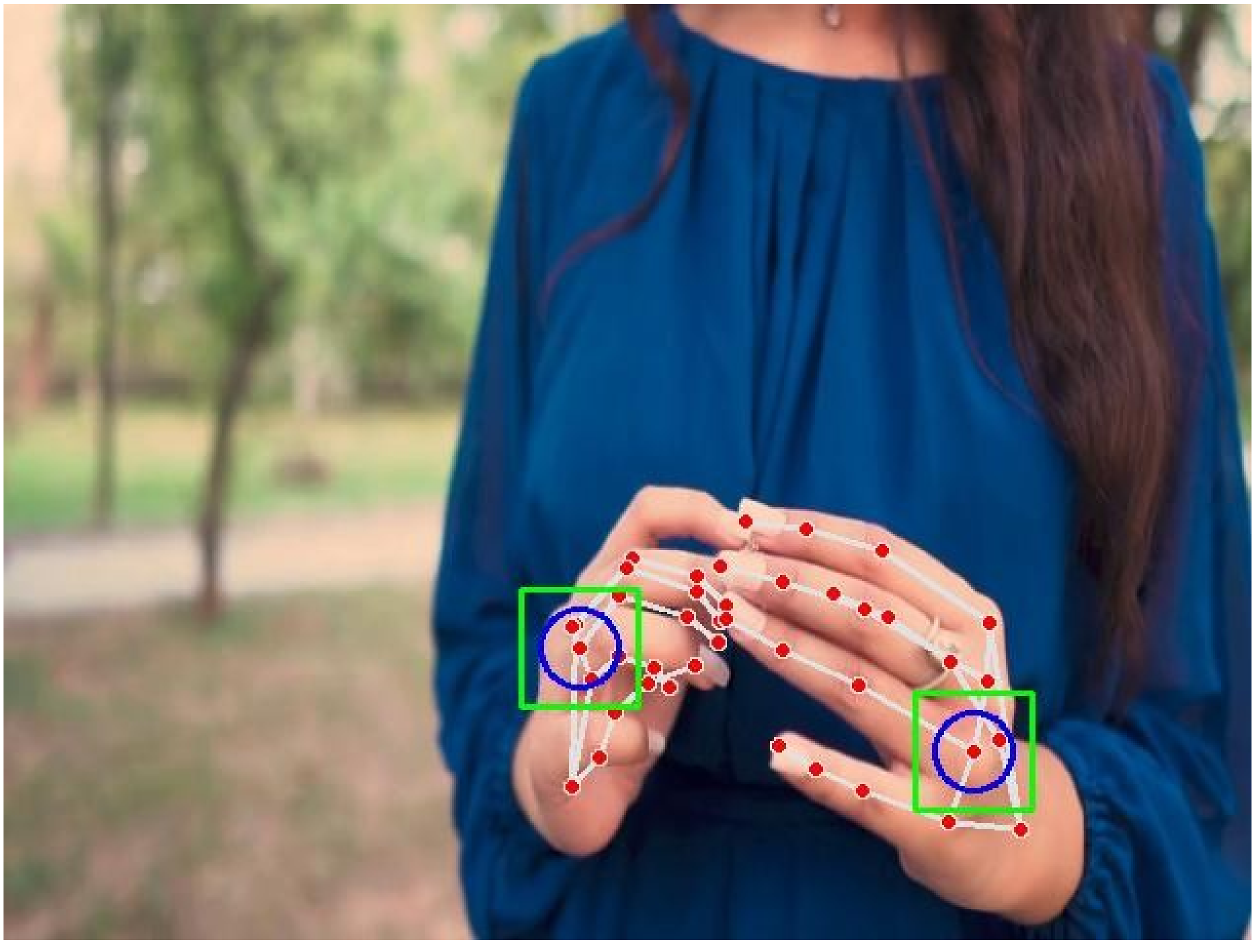
```
frame_count = 0
while cap.isOpened() and frame_count < 5:
    ret, frame = cap.read()
    if not ret:
        break

    frame = cv2.resize(frame, (640, 480))
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(rgb)

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
            draw_ring_zone(frame, results.multi_hand_landmarks)

    cv2_imshow(frame)
    frame_count += 1

cap.release()
```









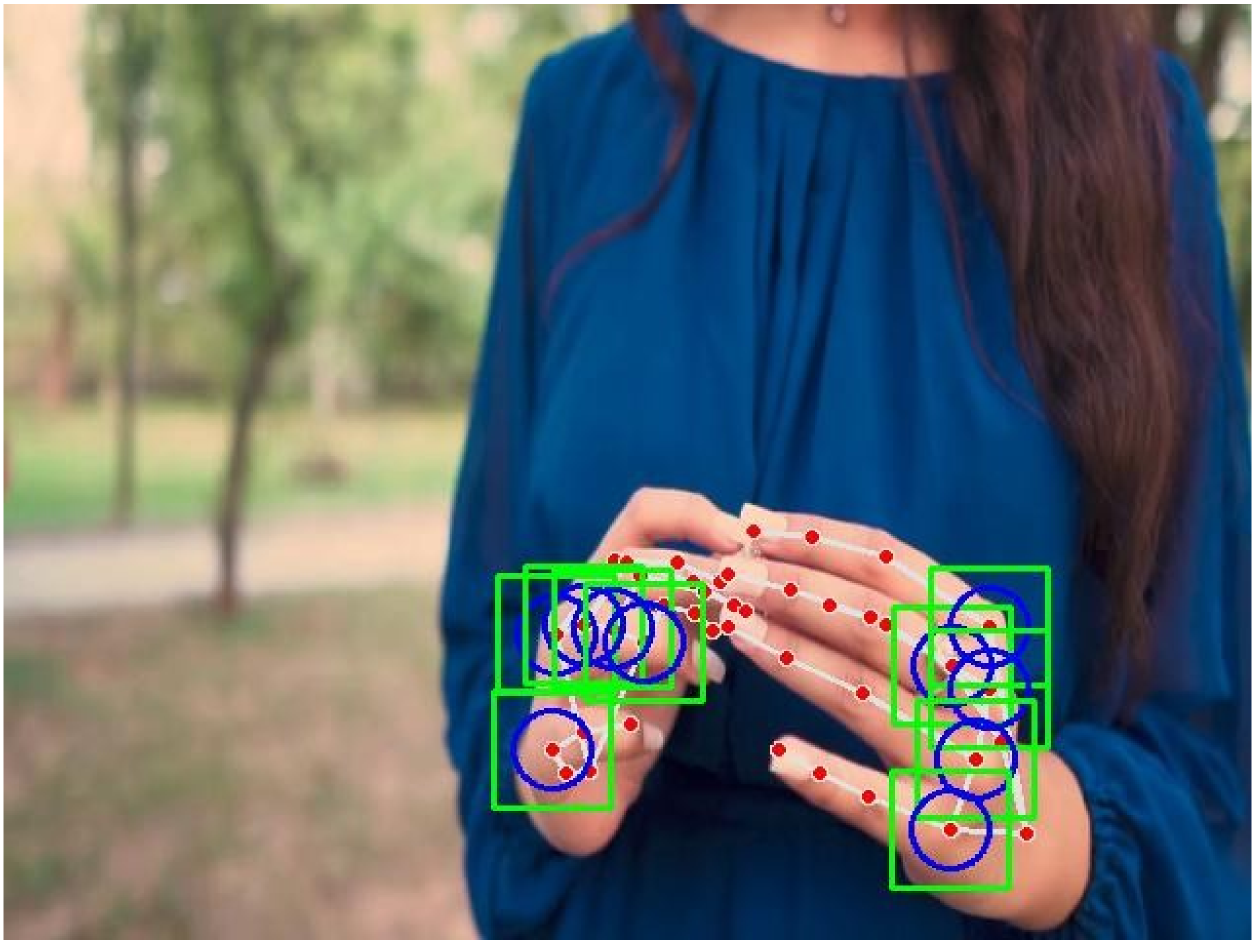


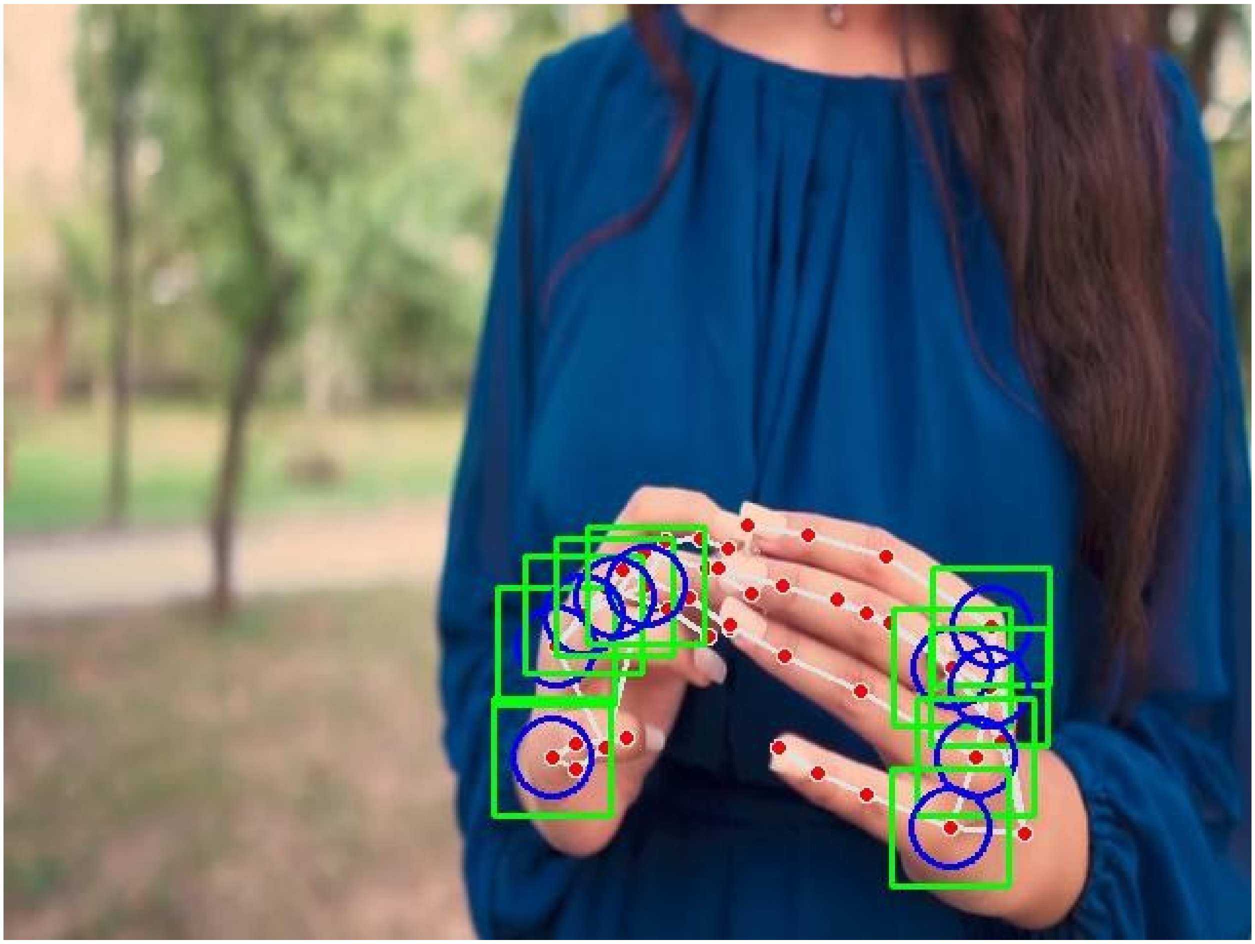
All finger base tracking

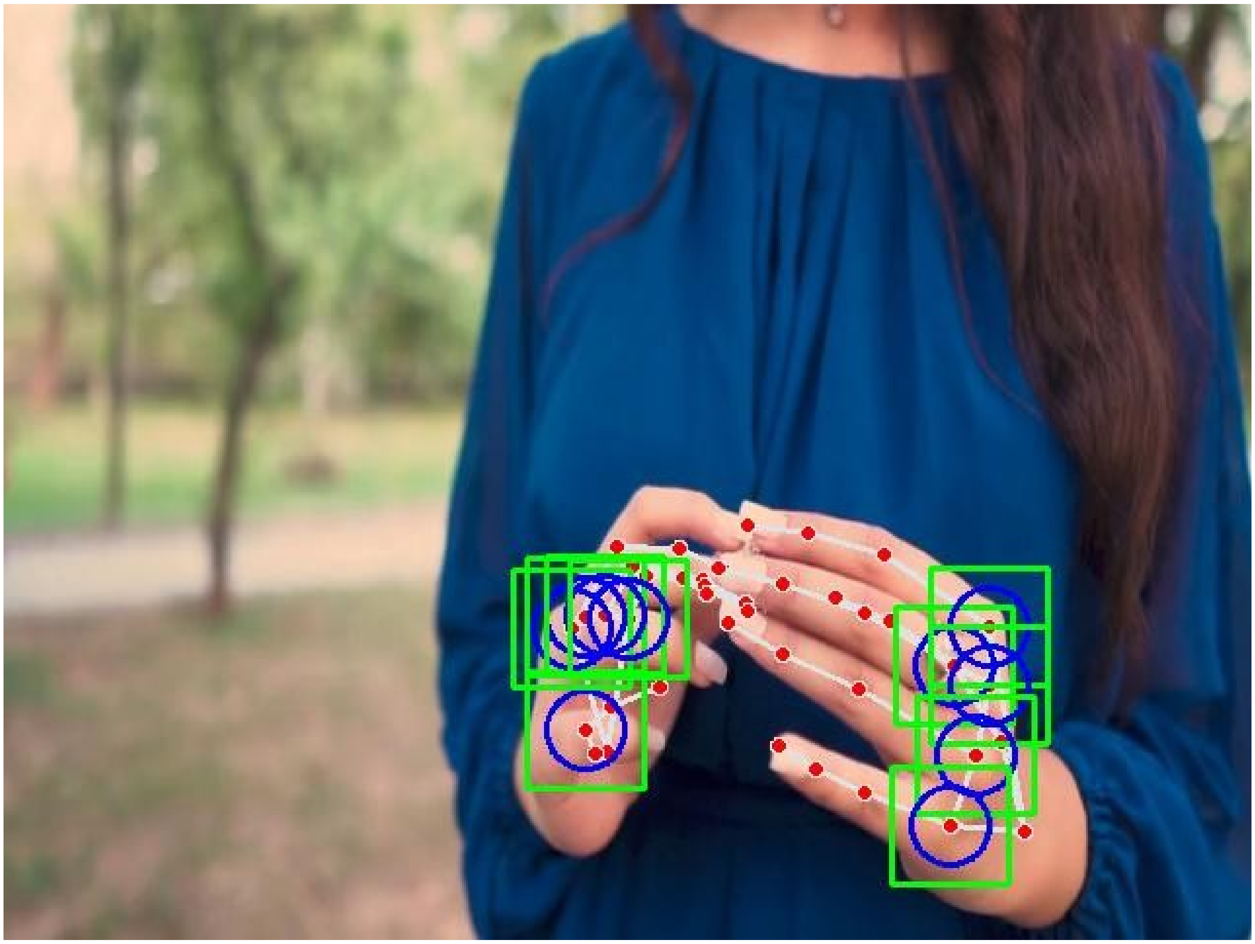
Then I thought maybe by tracking all the finger base and then detecting ring based on the color difference would be possible. So, this code snippet does circle on all the finger bases.

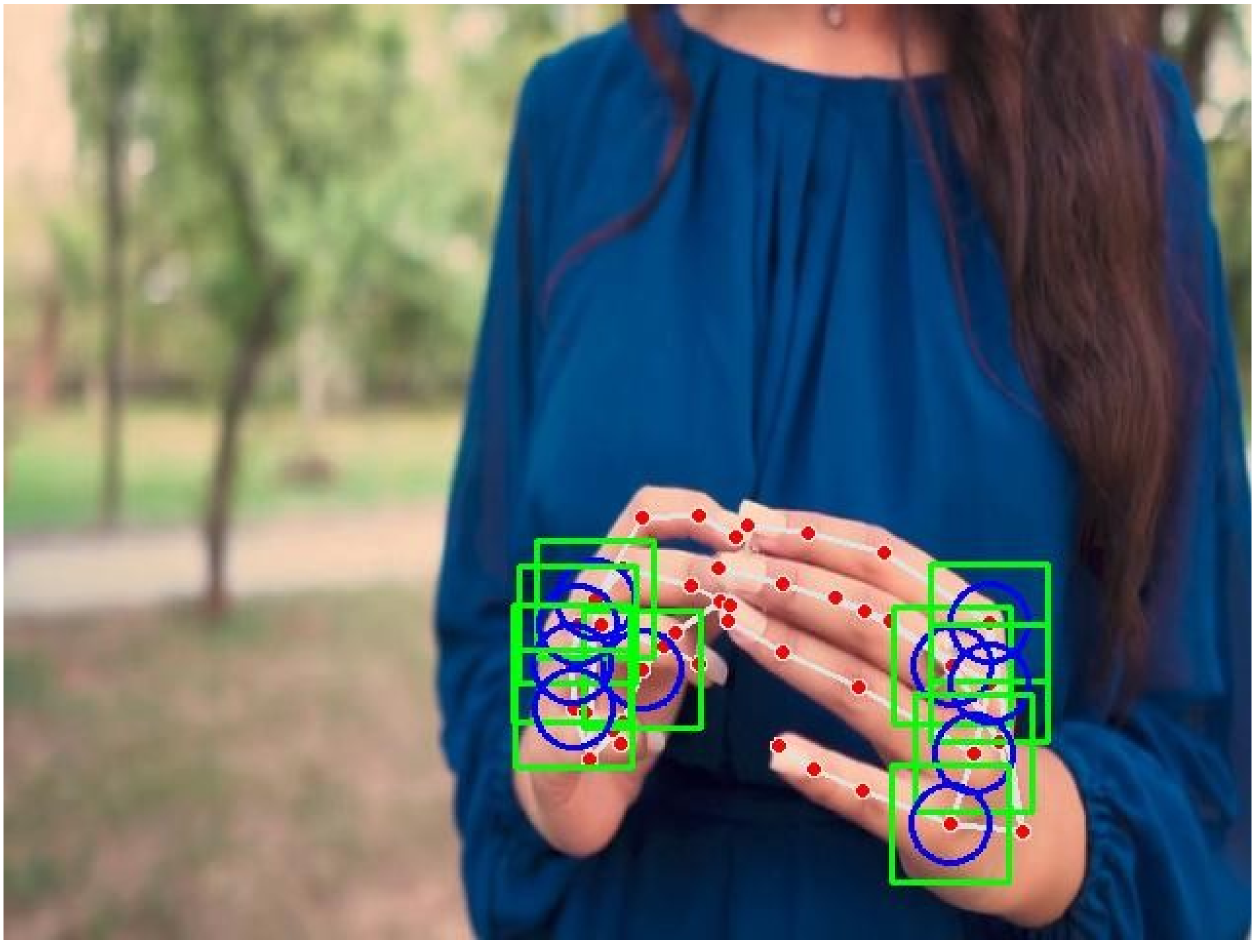
But, this dosen't work as well color was not differentiating the hand and rings due to lighting effects.

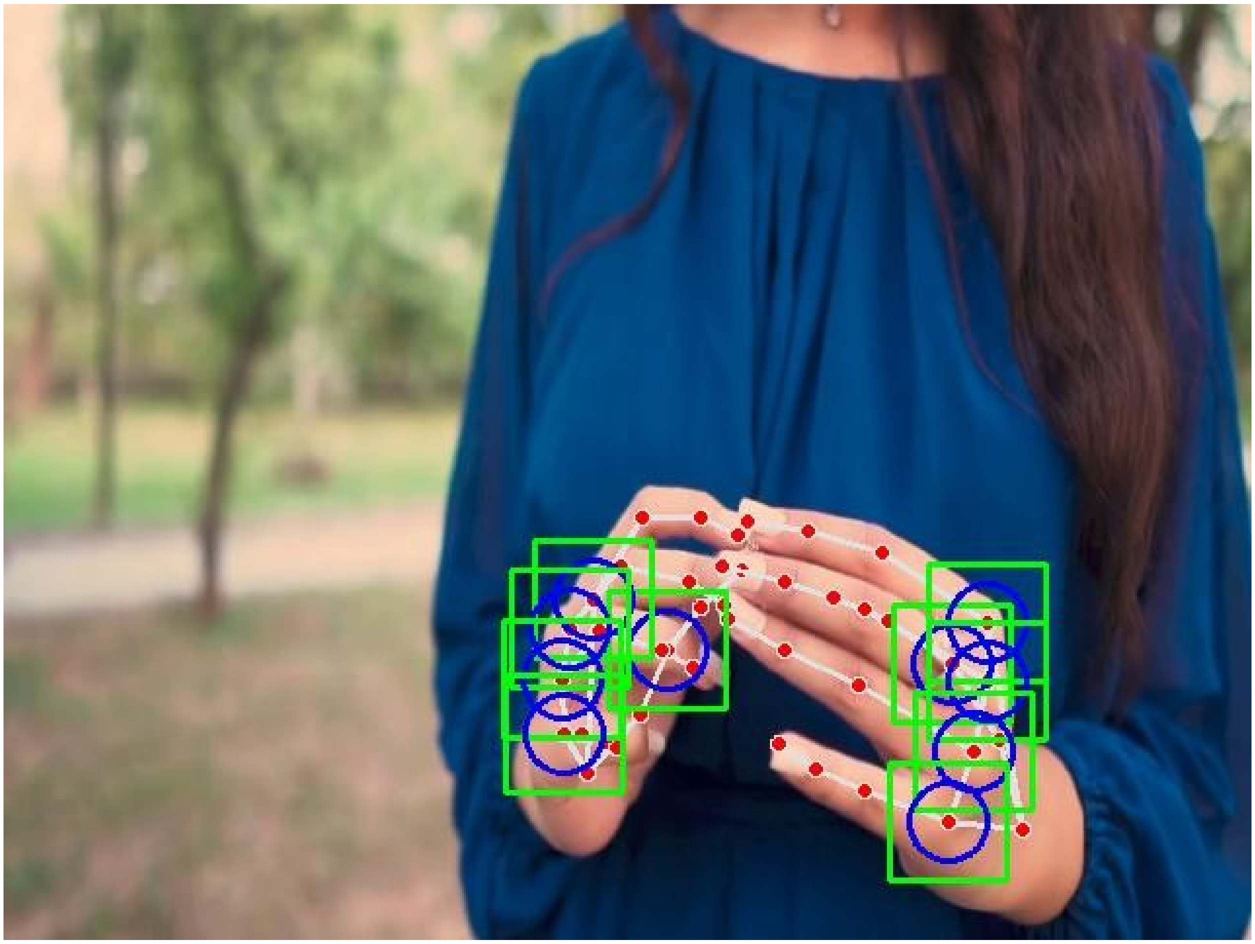
```
def draw_ring_zone_all_fingers(frame, hand_landmarks):
    for hand in hand_landmarks:
         for finger_landmark, finger_name in zip(
            [mp_hands.HandLandmark.THUMB_MCP,
             mp_hands.HandLandmark.INDEX_FINGER_MCP,
             mp_hands.HandLandmark.MIDDLE_FINGER_MCP,
             mp_hands.HandLandmark.RING_FINGER_MCP,
             mp_hands.HandLandmark.PINKY_MCP],
            ['Thumb', 'Index', 'Middle', 'Ring', 'Pinky']):

            finger_base = hand.landmark[finger_landmark]
            h, w, _ = frame.shape
            x, y = int(finger_base.x * w), int(finger_base.y * h)

            cv2.circle(frame, (x, y), 20, (255, 0, 0), 2)  # Circle
            cv2.rectangle(frame, (x-30, y-30), (x+30, y+30), (0, 255,
0), 2)

cap = cv2.VideoCapture(video_path)
frame_count = 0
while cap.isOpened() and frame_count < 5:
    ret, frame = cap.read()
    if not ret:
        break

    frame = cv2.resize(frame, (640, 480))
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(rgb)

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
            draw_ring_zone_all_fingers(frame,
results.multi_hand_landmarks)

    cv2_imshow(frame)
    frame_count += 1

cap.release()
```











Hand tracking

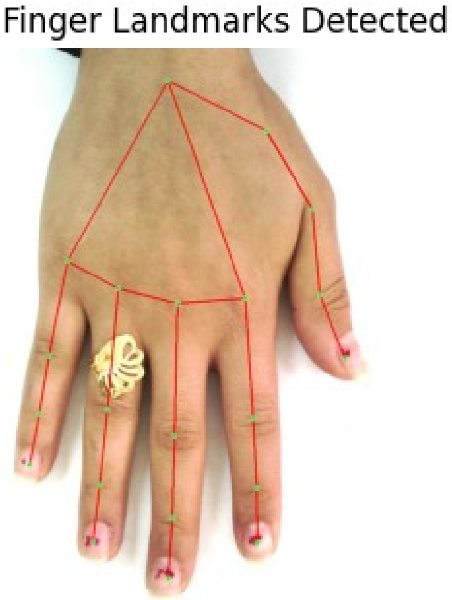
Then I moved from complete video to just hand to see if with only hand it is possible to track the rings with minimus lightning.

```
file_path = "/content/Hand_0005437.jpg"
image = Image.open(file_path)
img_np = np.array(image)

plt.imshow(img_np)
plt.axis('off')
plt.title("Input Hand Image")
plt.show()
```

Finger tracking

MediaPipe was successful to track all the finger in a hand but for the ring detection it still fails here.

```
import cv2
import mediapipe as mp
import matplotlib.pyplot as plt


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


with mp_hands.Hands(static_image_mode=True,
                    max_num_hands=2,
                    min_detection_confidence=0.5) as hands:
    results = hands.process(img_np)

    img_annotated = img_np.copy()

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(
                img_annotated,
                hand_landmarks,
                mp_hands.HAND_CONNECTIONS,
                mp_drawing.DrawingSpec(color=(0, 255, 0), thickness=2,
circle_radius=2),
                mp_drawing.DrawingSpec(color=(255, 0, 0), thickness=2)
            )
    else:
        print("No hands detected.")

plt.figure(figsize=(6, 6))
plt.imshow(img_annotated)
plt.axis('off')
plt.title("Finger Landmarks Detected")
plt.show()
```



Finger Landmarks Detected

```
import cv2
from google.colab.patches import cv2_imshow

video_path = "/content/219228_small.mp4"
cap = cv2.VideoCapture(video_path)

for i in range(5):
    ret, frame = cap.read()
    if not ret:
        break
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    plt.imshow(frame_rgb)
    plt.title(f"Frame {i}")
    plt.axis('off')
    plt.show()
cap.release()
```

Frame 0

Frame 1

Frame 2

Frame 3

Frame 4

```python
cap = cv2.VideoCapture(video_path)
frame_count = 0

with mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=2,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:

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

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        results = hands.process(frame_rgb)

        if results.multi_hand_landmarks:
            for hand_landmarks in results.multi_hand_landmarks:
                mp_drawing.draw_landmarks(
                    frame,
                    hand_landmarks,
                    mp_hands.HAND_CONNECTIONS)

        if frame_count % 20 == 0:
            cv2_imshow(frame)

        frame_count += 1

cap.release()
```