```
input_path = "/content/v_PushUps_g01_c01.avi"
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
!ls /content/drive/MyDrive/
```

```
 1669048453831354467693568519913.jpg
'Colab Notebooks'
 Document.rtf.gdoc
'evolution of secu and pri con.rtf.gdoc'
 IMG-20220219-WA0377.jpg
 IMG_20221118_153149.jpg
 IMG_20221121_221048.jpg
 IMG_20221121_221146.jpg
 IMG-20221121-WA0010.jpg
 IMG-20221121-WA0011.jpg
 IMG-20231016-WA0019.jpg
 IMG-20231016-WA0023.jpg
 IMG-20231017-WA0002.jpg
 IMG-20231017-WA0004.jpg
 IMG-20250219-WA0037.jpg
 mini_project.py
 Screenshot_2024-09-28-08-47-22-921_com.google.android.apps.nbu.paisa.user.jpg
 ssl.rtf.gdoc
 ucf3class_model.h5
 ufcdataset
'VISWANTH R (1)-1 (1).pdf'
'VISWANTH R (1)-1.pdf'
'VISWANTH R (1) (1).pdf'
'VISWANTH R (1).pdf'
'Viswanth resume (1).pdf'
'Viswanth resume.pdf'
'vulnerablity asssesment.rtf.gdoc'
'web app security.rtf.gdoc'
```

```
!ls -la "/content/drive/MyDrive/ufcdataset"
```

```
total 4
drwx------ 34 root root 4096 Dec 28  2024 UCF50
```

```
!ls -la "/content/drive/MyDrive/ufcdataset/UCF50/PushUps"
```

```
-rw------- 1 root root 206868 Oct  1  2010 v_PushUps_g10_c04.avi
```

```
-rw------- 1 root root 206868 Oct  1  2010 v_PushUps_g19_c04.avi
-rw------- 1 root root 283674 Oct  1  2010 v_PushUps_g20_c01.avi
-rw------- 1 root root 262366 Oct  1  2010 v_PushUps_g20_c02.avi
-rw------- 1 root root 266362 Oct  1  2010 v_PushUps_g20_c03.avi
-rw------- 1 root root 270416 Oct  1  2010 v_PushUps_g20_c04.avi
-rw------- 1 root root 243334 Oct  1  2010 v_PushUps_g21_c01.avi
-rw------- 1 root root 436934 Oct  1  2010 v_PushUps_g21_c02.avi
-rw------- 1 root root 378248 Oct  1  2010 v_PushUps_g21_c03.avi
-rw------- 1 root root 288854 Oct  1  2010 v_PushUps_g21_c04.avi
-rw------- 1 root root 172544 Oct  1  2010 v_PushUps_g22_c01.avi
-rw------- 1 root root 318262 Oct  1  2010 v_PushUps_g22_c02.avi
-rw------- 1 root root 181566 Oct  1  2010 v_PushUps_g22_c03.avi
-rw------- 1 root root 272144 Oct  1  2010 v_PushUps_g22_c04.avi
-rw------- 1 root root 238150 Oct  1  2010 v_PushUps_g23_c01.avi
-rw------- 1 root root 228568 Oct  1  2010 v_PushUps_g23_c02.avi
-rw------- 1 root root 229034 Oct  1  2010 v_PushUps_g23_c03.avi
-rw------- 1 root root 221610 Oct  1  2010 v_PushUps_g23_c04.avi
-rw------- 1 root root 116176 Oct  1  2010 v_PushUps_g24_c01.avi
-rw------- 1 root root 113298 Oct  1  2010 v_PushUps_g24_c02.avi
-rw------- 1 root root 113888 Oct  1  2010 v_PushUps_g24_c03.avi
-rw------- 1 root root 112492 Oct  1  2010 v_PushUps_g24_c04.avi
-rw------- 1 root root 260800 Oct  1  2010 v_PushUps_g25_c01.avi
-rw------- 1 root root 310132 Oct  1  2010 v_PushUps_g25_c02.avi
-rw------- 1 root root 317926 Oct  1  2010 v_PushUps_g25_c03.avi
-rw------- 1 root root 282498 Oct  1  2010 v_PushUps_g25_c04.avi
-rw------- 1 root root 232846 Oct  1  2010 v_PushUps_g26_c01.avi
-rw------- 1 root root 228914 Oct  1  2010 v_PushUps_g26_c02.avi
-rw------- 1 root root 236992 Oct  1  2010 v_PushUps_g26_c03.avi
-rw------- 1 root root 220152 Oct  1  2010 v_PushUps_g26_c04.avi
```

```
!find "/content/drive/MyDrive/ufcdataset" -name "*.avi"
```

```
/content/drive/MyDrive/ufcdataset/UCF50/TrampolineJumping/v_TrampolineJumping_g25_c03.avi
```

```
/content/drive/MyDrive/ufcdataset/UCF50/TrampolineJumping/v_TrampolineJumping_g25_c03.avi
/content/drive/MyDrive/ufcdataset/UCF50/TrampolineJumping/v_TrampolineJumping_g25_c04.avi
/content/drive/MyDrive/ufcdataset/UCF50/TaiChi/v_TaiChi_g01_c01.avi
/content/drive/MyDrive/ufcdataset/UCF50/TaiChi/v_TaiChi_g01_c02.avi
/content/drive/MyDrive/ufcdataset/UCF50/TaiChi/v_TaiChi_g01_c03.avi
/content/drive/MyDrive/ufcdataset/UCF50/TaiChi/v_TaiChi_g01_c04.avi
/content/drive/MyDrive/ufcdataset/UCF50/TaiChi/v_TaiChi_g02_c01.avi
```

```
!cp "/content/drive/MyDrive/ufcdataset/UCF50/PushUps/v_PushUps_g01_c01.avi" /content/
```

```
!ls /content/*.avi
```

```
/content/v_PushUps_g01_c01.avi
```

```
input_path = "/content/v_PushUps_g01_c01.avi"
```

```
!ls -la "/content/drive/MyDrive/ufcdataset"
```

```
total 4
drwx------ 34 root root 4096 Dec 28  2024 UCF50
```

```
!cp "/content/drive/MyDrive/ufcdataset/UCF50/PushUps/v_PushUps_g01_c01.avi" /content/
```

```
!ls /content/*.avi
```

```
/content/v_PushUps_g01_c01.avi
```

```
input_path = "/content/v_PushUps_g01_c01.avi"
```

Double-click (or enter) to edit

```
!pip install mediapipe
```

```
Requirement already satisfied: mediapipe in /usr/local/lib/python3.11/dist-packages (0.10.21)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from mediapipe) (1.4.0)
Requirement already satisfied: attrs>=19.1.0 in /usr/local/lib/python3.11/dist-packages (25.3.0)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.11/dist-packages (from mediapipe) (25.2.10)
Requirement already satisfied: jax in /usr/local/lib/python3.11/dist-packages (from mediapipe) (0.5.2)
Requirement already satisfied: jaxlib in /usr/local/lib/python3.11/dist-packages (from mediapipe) (0.5.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from mediapipe) (3.10.0)
Requirement already satisfied: numpy<2 in /usr/local/lib/python3.11/dist-packages (from mediapipe) (1.26.4)
Requirement already satisfied: opencv-contrib-python in /usr/local/lib/python3.11/dist-packages (from mediapipe) (4.11.0.86)
Requirement already satisfied: protobuf<5,>=4.25.3 in /usr/local/lib/python3.11/dist-packages (from mediapipe) (4.25.8)
Requirement already satisfied: sounddevice>=0.4.4 in /usr/local/lib/python3.11/dist-packages (from mediapipe) (0.5.2)
Requirement already satisfied: sentencepiece in /usr/local/lib/python3.11/dist-packages (from mediapipe) (0.2.0)
Requirement already satisfied: CFFI>=1.0 in /usr/local/lib/python3.11/dist-packages (from sounddevice>=0.4.4->mediapipe) (1.17.1)
Requirement already satisfied: ml_dtypes>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from jax->mediapipe) (0.4.1)
Requirement already satisfied: opt_einsum in /usr/local/lib/python3.11/dist-packages (from jax->mediapipe) (3.4.0)
Requirement already satisfied: scipy>=1.11.1 in /usr/local/lib/python3.11/dist-packages (from jax->mediapipe) (1.16.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib->mediapipe) (2.9.0.
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from CFFI>=1.0->sounddevice>=0.4.4->mediapipe)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib->mediapip
```

```
import mediapipe as mp
```

```
!find /content/drive/MyDrive -name "*.h5"
```

```
/content/drive/MyDrive/ucf3class_model.h5
```

```
!cp "/content/drive/MyDrive/ucf3class_model.h5" /content/
```

```
from tensorflow.keras.models import load_model
model = load_model("/content/ucf3class_model.h5")
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until
```

```
import cv2
import numpy as np
import mediapipe as mp

# Load model
from tensorflow.keras.models import load_model
model = load_model("/content/ucf3class_model.h5")

# Define classes
actions = ['PushUps', 'PullUps', 'JumpRope']  # Add more if trained

# MediaPipe setup
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
mp_drawing = mp.solutions.drawing_utils

SEQUENCE_LENGTH = 30
sequence = []

# Load video
input_path = "/content/v_PushUps_g01_c01.avi"  # Replace with your input video
cap = cv2.VideoCapture(input_path)

# Output video setup
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('/content/output_prediction.avi', fourcc, 20.0,
                      (int(cap.get(3)), int(cap.get(4))))

# Keypoint extraction
def extract_keypoints(results):
    if results.pose_landmarks:
        return np.array([[lm.x, lm.y, lm.z] for lm in results.pose_landmarks.landmark]).flatten()
    return np.zeros(33 * 3)

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

    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(image)
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

    keypoints = extract_keypoints(results)
    sequence.append(keypoints)
    sequence = sequence[-SEQUENCE_LENGTH:]

    if len(sequence) == SEQUENCE_LENGTH:
        input_data = np.expand_dims(sequence, axis=0)
        prediction = model.predict(input_data)[0]
        predicted_label = actions[np.argmax(prediction)]
        confidence = np.max(prediction)

        cv2.putText(image, f'{predicted_label} ({confidence:.2f})', (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    out.write(image)

cap.release()
out.release()
```

```
cv2.destroyAllWindows()
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
```

```
from google.colab import files
files.download("/content/output_prediction.avi")
```

```
import cv2
import numpy as np
import mediapipe as mp
from tensorflow.keras.models import load_model

# Load model and labels
model = load_model("/content/ucf3class_model.h5")
actions = ['PushUps', 'PullUps', 'JumpRope']  # Update based on your training

# MediaPipe setup
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
mp_drawing = mp.solutions.drawing_utils

SEQUENCE_LENGTH = 30
sequence = []

# Function to extract keypoints
def extract_keypoints(results):
    if results.pose_landmarks:
        return np.array([[lm.x, lm.y, lm.z] for lm in results.pose_landmarks.landmark]).flatten()
    else:
        return np.zeros(33 * 3)

# 🔁 CHANGE THIS: User video path (e.g., upload.mp4)
input_path = "/content/user_video.mp4"  # 👈 replace this with the uploaded video
cap = cv2.VideoCapture(input_path)

# Prepare output video
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('/content/output_prediction.avi', fourcc, 20.0,
                      (int(cap.get(3)), int(cap.get(4))))

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

    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(image)
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

    keypoints = extract_keypoints(results)
    sequence.append(keypoints)
    sequence = sequence[-SEQUENCE_LENGTH:]

    if len(sequence) == SEQUENCE_LENGTH:
        input_data = np.expand_dims(sequence, axis=0)
        prediction = model.predict(input_data)[0]
        predicted_label = actions[np.argmax(prediction)]
        confidence = np.max(prediction)

        cv2.putText(image, f'{predicted_label} ({confidence:.2f})', (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    out.write(image)

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

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until
```

UPLOADING THE VIDEO

```
from google.colab import files
uploaded = files.upload()  # Upload again here
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving 3127085-uhd_3840_2160_24fps.mp4 to 3127085-uhd_3840_2160_24fps.mp4
```

Double-click (or enter) to edit

```
!ls
```

```
3127085-uhd_3840_2160_24fps.mp4  output_prediction.avi  ucf3class_model.h5
drive                            sample_data            v_PushUps_g01_c01.avi
```

```
!mv "3127085-uhd_3840_2160_24fps.mp4" user_video.mp4
```

```
!ls /content/user_video.mp4
```

```
/content/user_video.mp4
```

EXAMPLE

```
import cv2
import numpy as np
import mediapipe as mp
from tensorflow.keras.models import load_model

# Load your trained model
model = load_model("/content/ucf3class_model.h5")

# Define the actions your model was trained on
actions = ['PushUps', 'PullUps', 'JumpRope']  # 👉 replace with your actual labels

# MediaPipe pose setup
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
mp_drawing = mp.solutions.drawing_utils

SEQUENCE_LENGTH = 30
sequence = []
```

```
# Function to extract keypoints from pose landmarks
def extract_keypoints(results):
    if results.pose_landmarks:
        return np.array([[lm.x, lm.y, lm.z] for lm in results.pose_landmarks.landmark]).flatten()
    else:
        return np.zeros(33 * 3)

# ✅ Use your renamed video
input_path = "/content/user_video.mp4"
cap = cv2.VideoCapture(input_path)

# Setup output video
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('/content/output_prediction.avi', fourcc, 20.0,
                      (int(cap.get(3)), int(cap.get(4))))

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

    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(image)
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

    keypoints = extract_keypoints(results)
    sequence.append(keypoints)
    sequence = sequence[-SEQUENCE_LENGTH:]

    if len(sequence) == SEQUENCE_LENGTH:
        input_data = np.expand_dims(sequence, axis=0)
        prediction = model.predict(input_data)[0]
        predicted_label = actions[np.argmax(prediction)]
        confidence = np.max(prediction)

        # Draw prediction
        cv2.putText(image, f'{predicted_label} ({confidence:.2f})', (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    out.write(image)

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

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
```

```
from google.colab import files
files.download("/content/output_prediction.avi")
```

```
Start coding or generate with AI.
```

Double-click (or enter) to edit