

Code for the DIP Component

```
cap = cv2.VideoCapture(VIDEO_TEST)
current_stage = ""
prediction_probability_threshold = 0.6

with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, image = cap.read()

        if not ret:
            break

        image = rescale_frame(image, 50)

        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        results = pose.process(image)

        if not results.pose_landmarks:
            print("No human found")
            continue

        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_pose.POSE_CONNECTIONS, mp_drawing.DrawingSpec(color=(244, 117, 66),
thickness=2, circle_radius=2), mp_drawing.DrawingSpec(color=(245, 66, 230), thickness=2,
circle_radius=1))

INPUT_SCALER_PATH = '/Users/likeshkoya/code/PROJECT/plank_input_scaler.pkl'
with open(INPUT_SCALER_PATH, "rb") as f2:
    input_scaler = pickle.load(f2)

try:

    row = extract_important_keypoints(results)
    X = pd.DataFrame([row, ], columns=HEADERS[1:])
    X = pd.DataFrame(input_scaler.transform(X))

    prediction = deep_learning_model.predict(X)
```

```

predicted_class = np.argmax(prediction, axis=1)[0]

prediction_probability = max(prediction.tolist()[0])

if predicted_class == 0 and prediction_probability >=
prediction_probability_threshold:
    current_stage = "Correct"
elif predicted_class == 2 and prediction_probability >=
prediction_probability_threshold:
    current_stage = "Low back"
elif predicted_class == 1 and prediction_probability >=
prediction_probability_threshold:
    current_stage = "High back"
else:
    current_stage = "Unknown"

cv2.rectangle(image, (0, 0), (300, 60), (80, 150, 210), -1)

cv2.putText(image, "DETECTION", (95, 12), cv2.FONT_HERSHEY_COMPLEX, 0.5,
(0, 0, 0), 1, cv2.LINE_AA)
cv2.putText(image, current_stage, (90, 40), cv2.FONT_HERSHEY_COMPLEX, 1,
(255, 255, 255), 2, cv2.LINE_AA)

cv2.putText(image, "PROB", (15, 12), cv2.FONT_HERSHEY_COMPLEX, 0.5, (0, 0,
0), 1, cv2.LINE_AA)
cv2.putText(image, str(round(prediction_probability, 2)), (10, 40),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

except Exception as e:
    print(f"Error: {e}")

cv2.imshow("CV2", image)

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

cap.release()
cv2.destroyAllWindows()

for i in range(1, 5):
    cv2.waitKey(1)

```