

```
# collect_keypoints_to_csv.py

import cv2
import mediapipe as mp
import csv
import os

# Configuration
CSV_PATH = "keypoints_data.csv"
LABELS = list("ABCDEFGHIJKLMNOPQRSTUVWXYZ")

# Initialize 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=0.5)
mp_drawing = mp.solutions.drawing_utils

# Data collection control
collecting = False
current_label = "A"

# Create CSV file with headers if not exists
if not os.path.exists(CSV_PATH):
    with open(CSV_PATH, mode='w', newline='') as f:
        writer = csv.writer(f)
        header = [f"x{i}" for i in range(21)] + [f"y{i}" for i in range(21)] + ["label"]
        writer.writerow(header)

# Open webcam
cap = cv2.VideoCapture(0)
```

```

print("\n→ Press a letter (A–Z) to set the label")
print("→ Press 'c' to toggle data collection")
print("→ Press 'q' to quit\n")

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

    frame = cv2.flip(frame, 1)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    result = hands.process(rgb)

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

            # Save keypoints
            if collecting:
                row = []
                for lm in hand_landmarks.landmark:
                    row.extend([lm.x, lm.y])
                row.append(current_label)

                with open(CSV_PATH, mode='a', newline='') as f:
                    writer = csv.writer(f)
                    writer.writerow(row)

            print(f"✓ Saved keypoints for label {current_label}")

```

```
# Show status on screen

cv2.putText(frame, f"Label: {current_label} Collecting: {collecting}", (10, 30),
             cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0) if collecting else (0, 0, 255), 2)

cv2.imshow('Keypoint Capture', frame)


key = cv2.waitKey(10)

if key == ord('q'):
    break

elif key == ord('c'):
    collecting = not collecting

elif key != -1 and chr(key).upper() in LABELS:
    current_label = chr(key).upper()


cap.release()

cv2.destroyAllWindows()

print("\n Data saved to:", CSV_PATH)
```