

Box ütések detektálása

A program egy nagyobb volumenű projekt alapjait fekteti le. A teljes projekt célja egy kamera felvétel alapján box mozdulatok felismerése, értékelése, valamint ez alapján technikai szintű visszajelzés adása a felhasználó számára. Jelenleg a felismerést valósítja meg a program, mediapipe test-koordináták távolsága és más szabályok alapján.

Felhasznált könyvtárak

Cv2 - Képfeldolgozásért felelős könyvtár. Ezen keresztül dolgozzuk fel a kamera felvételt, kódoljuk a program által érthető formába az információt, majd dekódoljuk.

Mediapipe – A testek felismeréséért felelős könyvtár, mely felruházza a detektált testet koordinátákkal, mely által az adott pozíciót határozzuk meg.

Math – Egy matematikai függvényeket tartalmazó könyvtár, mely a távolság meghatározását egyszerűsíti meg.

A program működése

A kiválasztott kamerán keresztül elindul a felvétel. Ha testet detektál, a mediapipe felruházza a test különböző részeit koordinátákkal. Optimális működés érdekében a teljes felsőtestnek szerepelnie kell a képben, valamint a test jobb oldalát mutatni. A szabály alapú figyelés folyamatosan aktív, menti az x és y koordinátáit a könyököknek, vállnak, csuklóknak, valamint az ornak.

Jab (egyenes) detektálás

Annak érdekében, hogy ne detektáljon több ütést, mint ami valójában történt, az r_arm_ready, valamint l_arm_ready változókra van szükségünk. Ezek boolean változók, melyek azt határozzák meg, hogy a kar jelenleg testközelben van-e, azaz készen áll-e egy ütésre. Ez a válltól való távolságától függ.

```
if r_arm_ready:
    if r_dist > punch_threshold and r_wrist_y < (r_shoulder_y + raised_threshold):
        l_jab_count += 1
        r_arm_ready = False
else:
    if r_dist < reset_threshold:
        r_arm_ready = True
```

Ha megfelel a feltételnek, azaz vissza van húzva a kar, akkor a Pitagórasz tétel alapján, a csukló és váll koordináták távolságát ismét megméri. Ha túlmegy az adott határon, amit a punch_threshold-ban tárolunk, akkor egyenes ütésként detektálja és inkrementálja a számlálót. Innentől kezdve csak akkor detektál újabb ütést, ha a kart visszahúzzuk a testhez, és utána ütünk még egyet.

Uppercut (felütés) detektálás

A felütés hasonló logikán alapszik. A feltételek változnak, miszerint a csuklónak az orr felett kell lennie, a könyöknek a váll alatt kell lennie, valamint a kar nem lehet teljesen kinyújtva. Itt is csak is egy ütést detektál a program, amíg nincs visszahúzva a kar.

```
if l_upper_ready:
    if (l_wrist_y < nose_y and
        l_elbow_y > l_shoulder_y and
        l_dist < punch_threshold):

        r_upper_count += 1
        l_upper_ready = False
else:
    if l_wrist_y > (nose_y + 0.15):
        l_upper_ready = True
```

Lehetséges fejlesztések

Jelenlegi állapotában több szempontból limitált a program. Az ütések több formában is megtudnak mutatkozni, szinte mindenkor stílus másképpen használja, és szimpla szabály alapon több komplikáció is visszafogja a kiteljesedéstől a programot. Mesterséges intelligencia alapú integráció lenne optimális, mely többféle ütésfelismerési módot használhat, és így többféle ütésekkel is képes lesz reagálni. Ez megkönnyíti a technikai visszajelzési aspektust is, mivel a hibákat a mesterséges intelligencia detektálná és értelmezné.

Emellett ideális lenne több eszközt felhasználni. Több kamerával több szögből lehetne felvételt készíteni, mely pontosabb adatokat adna. Külső eszközök integrációja, mint akár giroszkópok is segíthetnék a program pontosságát.