



Hand&Gesture detektáló

M.I. és Haladó programozás beadandó – 2024. 11. 26.

Rezümé

A projektünk célja egy kézjeleket detektáló alkalmazás, amely extra funkcióként arckitakaró funkcióval is rendelkezik. Ezen funkciókhoz társul egy egyszerű felhasználói felület is, amely egyértelművé teszi az applikáció használatát.

A beadandó során különös figyelmet fektettünk arra, hogy gyakorlatban könnyedén hasznosítható megoldás készüljön, amely az élet több területén is képes helytállni az egyszerű szórakoztatástól kezdve, az oktatáson át egészen orvosi alkalmazhatóságig.

Erőforráshiány miatt modellfeltanítás nem volt lehetséges így egy előre tanított modellt alkalmaztunk.

Csapattagok

Bori András Balázs	FC2BA9
Hajdu Gergő	KMZ7M1

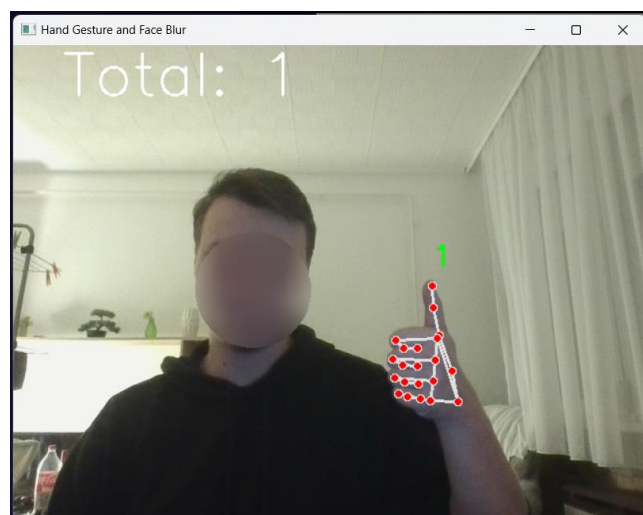
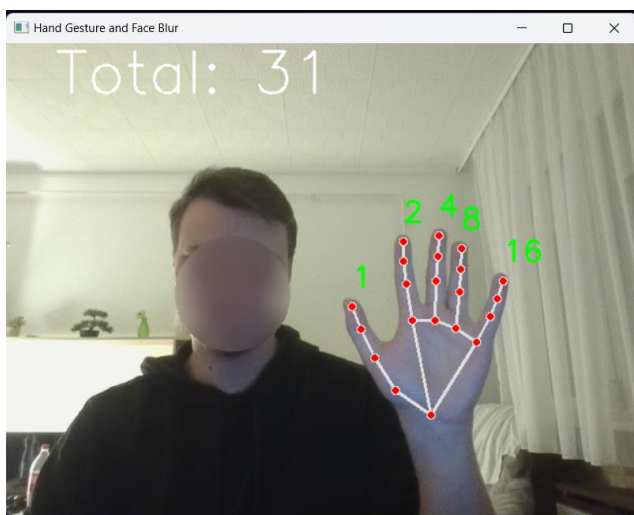
Tartalomjegyzék

Rezümé.....	1
Alkalmazás célja	2
Alkalmazási területek.....	2
Telepítés	3
Felhasznált könyvtárak	3
Az alkalmazás	3
Felület kinézete	3
Kód jellemzése	4
Felmerült problémák	6
Felhasznált források	6
Repository	6

Alkalmazás célja

Az applikáció jelenleg egy egyszerű reprezentációja egy komplexebb programnak, amely ezen kézfelfismeréssel képes különböző funkciókat végrehajtani.

Jelenlegi funkciója, hogy a kéz adott ujjaihoz a kettő hatványait rendeli a kamerakép alapján. Ennek a megvalósításához képes külön-külön azonosítani az ujjakat. Ezzel lehetővé téve, hogy egy kézzel 32 különböző parancsot tudjunk futtatni. Mivel mindkét kezet képes érzékelni ezért összesen 64 különböző funkciót is rendelhetünk hozzá (Az alábbi képen látható ez a folyamat bemutatása 1 kéz esetén).



A képen látható az aktív arckitakarás és kézdetektálás. A „Total” érték az ujjakhoz rendelt értékek összege. Ezen számok célja az, hogy több funkciót tudjunk aktiválni a lehető legegyszerűbb módon.

Az arckitakarás funkció aktívan leköveti a képen szereplő személy arcát. Ez külön is megfelelő funkció lehet többek között személyiségi jogok megőrzése érdekében.

Alkalmazási területek

Kézdetektáló funkció alkalmazási területei:

- Szórakoztatás: Interaktív játékok esetén vezérlés, tartalomgyártók számára kameravezérléshez.
- Oktatás: Gyermekek számára segítené a számok tanulását.
- IoT: Okosotthon vezérlés biztonsági kamerák segítségével.

Arckitakaró funkció alkalmazási területei:

- Személyiségi jogok védelme videókon
- Továbbfejlesztve akár arcfelismerésre is használható

Telepítés

Előzetes követelmény az alkalmazás futtatásához Windows 10 vagy 11 operációs rendszer és webkamera megléte. MacOS rendszer esetén néhány funkció esetlegesen nem megfelelően működik. Visual Studio Code használata ajánlott.

Az alkalmazás optimális futtatása érdekében a python 3.11.9-es verziójának telepítése szükséges. Ez lehetséges a Python hivatalos oldaláról vagy a következő parancs segítségével:

```
curl -o python-3.11.9.exe https://www.python.org/ftp/python/3.11.9/python-3.11.9-amd64.exe
```

```
python-3.11.9.exe /quiet InstallAllUsers=1 PrependPath=1
```

Sikeres telepítés után a futtatáshoz telepítenünk kell a felhasznált könyvtárakat az alábbi sor segítségével:

```
pip install opencv-python mediapipe numpy tk
```

A parancsok sikeres végrehajtása után az alkalmazás készen áll a futtatásra.

Felhasznált könyvtárak

- CV2: Az OpenCV könyvtár a számítógépes látás és képfeldolgozás feladatainak elvégzésére.
- MediaPipe: A MediaPipe könyvtár a kéz és arc felismeréséhez és feldolgozásához.
- Tkinter: A Tkinter könyvtár és annak testreszabható elemeit (ttk) használja az alkalmazás GUI-jának létrehozásához.
- NumPy: A képfeldolgozáshoz szükséges mátrix műveletekhez.

Az alkalmazás

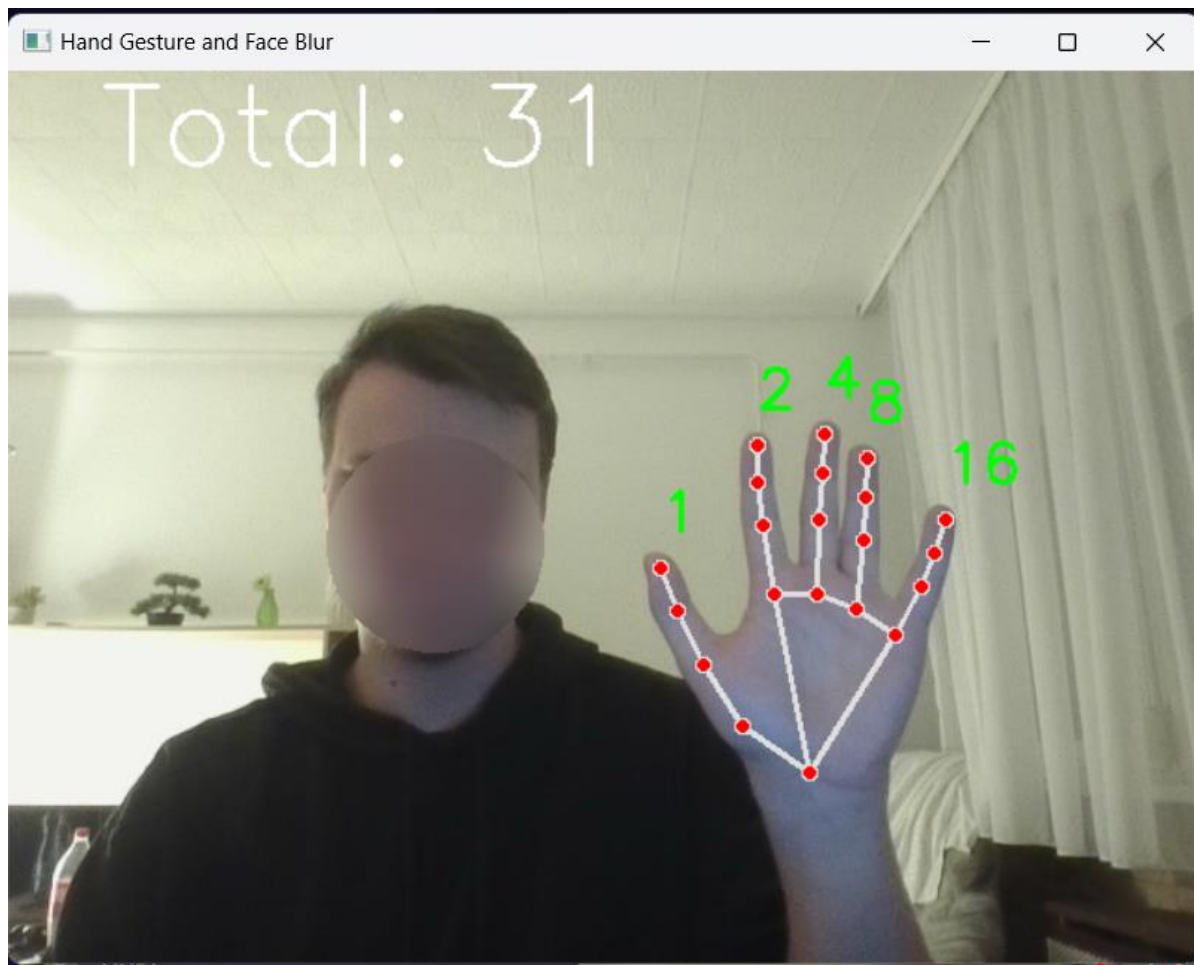
Felület kinézete

Egyszerű felhasználás érdekében az alkalmazás egy egyszerűsített, sötét háttérrel rendelkező vezérlőfelületből és a detektálást kijelző felületből áll.

A vezérlőfelületen kiválaszthatjuk, hogy mely funkciókat szeretnénk alkalmazni majd elindítjuk az alkalmazást vagy ki is léphetünk.

Ezen funkciók MacOS esetén nem megfelelően működnek a rendszer sajátosságai miatt. Erre megoldást nem sikerült találni, így csak korlátozottan lehetséges használni Macbookokon.





A képen látható a fő felület, ahol történik a detektálás. Jelen esetben az arckitakaró funkció és a kézdetektálás is aktív.

Kód jellemzése

A kód létrehoz egy egyszerű felhasználói felületet (GUI), amely python könyvtárak segítségével (Tkinter, OpenCV és MediaPipe) lehetővé teszi a kéz- és arcfelismerést. A GUI felületen lehet ki- és bekapcsolni ezek elérhetőségét és megnyitni a kamerát. A kamerán belül a kéz- és arcfelismerésen kívül a feltett ujjak értékeit adja össze a kód. Kód részletes leírása:

- GUI létrehozása:
 - A Tk osztály létrehoz egy ablakot, amelyet a root változó tárol.
 - Az ablak háttere fekete, és a szövegek fehér színűek, a stílust a ttk.Style segítségével testreszabják.
 - A program logót tölt be a logo.png fájlból, ha az elérhető. A logó átméretezésre kerül, és egy Label widgetben megjelenik a GUI-ban.
- Kamera inicializálása:
 - Az OpenCV cv2.VideoCapture(0) segítségével elindítja a számítógép elsődleges kameráját. Ha nem sikerül a kamera indítása, hibát jelez és bezárja az alkalmazást.

- MediaPipe modellek inicializálása:
 - A mediapipe könyvtár Hands modullal elindítja a kéz követését, és az FaceDetection modullal az arcfelismerést. Ezeket a modelleket a későbbiekben használják a kéz és arc felismerésére.
 - Funkciók testreszabása:
 - A GUI-n lehetőség van két funkció (kéz követése, arc kitakarása) be- és kikapcsolására a Checkbutton widgetek segítségével.
 - A BooleanVar típusú változók tárolják a beállításokat, amelyek a felhasználó választását tükrözik.
- Kép és arcfeldolgozás elindítása:
 - A start_processing függvény indítja el a kamera olvasását és a képfeldolgozást.
 - A kamera folyamatosan képkockákat olvas be, amelyeket RGB formátumban dolgoz fel.
 - Ha a kéz követése be van kapcsolva, a MediaPipe Hands modullal detektálja a kézpozíciókat és megjeleníti az ujjakat.
 - Az arcfelismerés során, ha egy arcot talál, a kód elmosódásokat alkalmaz a képen az arc körüli területen, hogy homályosítsa azt.
- Kézpontszám és ujjállapotok:
 - A kéz figyelése során a különböző ujjbegyek és az ujjak közötti pozíciók alapján meghatározza a kéz pontszámát.
 - Ha egy ujj "felfelé" mutat, annak értéke hozzáadódik a pontszámhoz. A pontszámot minden képkockán megjeleníti.
- Képkockák megjelenítése:
 - Az OpenCV imshow függvényét használja a feldolgozott képkockák megjelenítésére egy ablakban, amely folyamatosan frissül.
 - A felhasználó a "q" billentyű lenyomásával vagy az ablak bezárásával kiléphet a feldolgozásból.
- Kilépés az alkalmazásból:
 - Az exit_app függvény leállítja a kamerát és bezárja az alkalmazást, ha a felhasználó kilép.
- GUI vezérlők:
 - A GUI-n található egy "Start" gomb, amely elindítja a feldolgozást, és egy "Exit" gomb, amellyel kiléphetünk az alkalmazásból.
 - A fő eseményhurok (root.mainloop()) biztosítja, hogy a GUI interaktív maradjon, és figyeli a felhasználói műveleteket.

Összegzés: A kód egy interaktív alkalmazást hoz létre, amely a felhasználotól függően kéz- és arcfelismerést végez egy élő videóban. A felhasználó a GUI segítségével aktiválhatja vagy deaktiválhatja a különböző funkciókat, mint a kéz követését vagy az arc homályosítását.

Felmerült problémák

- **Különböző operációs rendszerek közötti el térések:**

Mivel fejlesztés után MacOS rendszeren is megpróbáltuk futtatni, problémák merültek fel a felhasználó felülettel kapcsolatban. Egyes gombok nem működnek csak első kattintás során, ablakbezárás ikon nem elérhető, ablakot nem képes bezárni a kilépés gomb, esetleges alkalmazás lefagyások. Ennek okai jelenleg ismeretlenek, de valószínűsíthetően a MacOS szigorú szabályainak tudható be. Felmerülő problémákra megoldás a python kernel újraindítása.

- **Csak az adott python verzió esetén működő alkalmazás:**

Nem lehet egyértelműen beazonosítani a hiba okát (lehetséges a MacOS gond okozta ezt is), de a különböző Python verziók esetén nem jelent meg a kamerakép, sem a felhasználói felület. Erre megoldás a megadott Python 3.11.9 telepítése.

- **Hüvelykujj detektálási gondok:**

A hüvelykujj érzékelése folyamatos gondokat jelentett és már internetes keresés után sem sikerült teljesen kiküszöbölni a gondokat, így valamennyi hiba folyamatosan megjelenik. Megoldást jelenthet a kéz elforgatása kis mértékben.

Felhasznált források

1. Kaggle - <https://www.kaggle.com/code/mohamedgobara/100-in-hand-gesture-recognition>
2. GitHub forráskódok
3. AI segítségék – GitHub Copilot, ChatGPT

Repository

<https://github.com/Andris375/Mesterseges-intelligencia-Halado-programozas-beadando.git>

Alkalmazás készítése során offline módon dolgoztunk egy számítógépet használva, így a GitHub repositoryban a commit-ok jelentős része csak a csapat egyik tagjától származnak.