

Önálló feladat

A feladatom kitalálása során törekedtem arra, hogy akár a hétköznapiakban is felhasználható legyen és fejlesztések útján akár a saját mindennapjaimba is integrálni tudjam a későbbiekben.

Az általam kitalált feladat egy olyan szoftver írása, amely az eszközön elérhető kamerán keresztül vizsgálja az eszközt használó személynek az álmosági szintjét.

Mint diák és munkavállaló, aki az idejének jó részét a számítógép előtt tölti, észleltem, hogy hajlamosabbá válok figyelmen kívül hagyni a fáradtságra utaló jeleket. A fáradtság mértékének növekedésével a produktivitás szintje is csökken, amit munka során jobb elkerülni. Ezzel a problémával véleményem szerint sok más ember is szembesült már.

A leggyakoribb elterjedése napjainkban adott felhasználású gépjárművekben található meg, amik figyelik a gépjármű vezetőinek a fáradtsági szintjét. Ezek a kamerák és feldolgozó egységek általában fuvarozó vagy nagyobb tömegközlekedési járművek szélvédőjén vannak jelen egy kompakt kiserelésben, amik felhő alapú összeköttetésben is vannak a cég rendszerével külső figyelmeztetés vagy kiértékelés érdekében.

Az általam kitalált és megalkotandó szoftver jelen esetben a számítógéphez kötött webkamera élő képe alapján detektálna arcot és vizsgálná a pislogások gyakoriságát, az egy perc alatt megtett pislogásoknak a mennyiségét, ami 12 pislogásnak felel meg percenként [1]. Későbbiekben ez a szoftver továbbfejleszthető pislogás időhosszának figyelembevételével is, ami átlagosan 1/3 másodpercig tart [1].

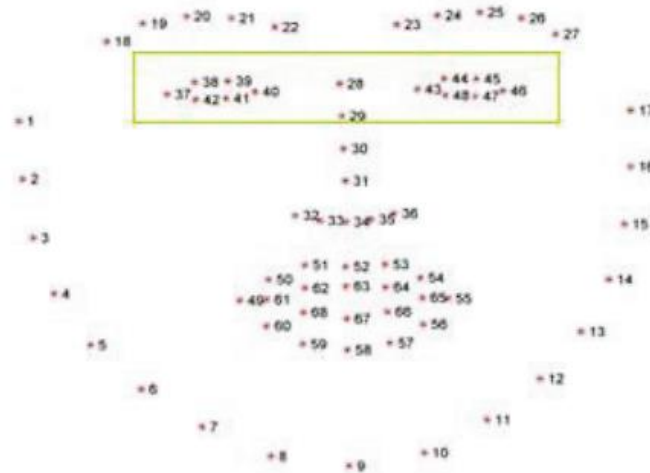
Továbbfejlesztésnél lehet vizsgálni a napszaktól vagy környezettől eltérő pislogási viselkedési normákat. Emellett figyelembe vehető az ásítás detektálása is, vagy akár a szem nyitottságának nagyságát figyelembe vevő osztályozó rendszer, a szemfelnyitási szintjének a megbecslése is. Ezekkel a további implementációkkal szerintem egy egészen pontos rendszert lehet kapni és a rendszer tanításához elegendő lenne az, ha egy kisebb közösségben elterjedne a géphasználók körében. A laptopok esetében az alpból beépített webcamera hasznos funkciót tudna ellátni, ami a vezetés esetén akár életet is menthet.

A rendszer megvalósításához tisztában kell lenni a felépítéséhez szükséges lehetőségekkel és limitációkkal. Limitáció alatt értendő az, hogy a kamerának kellőképpen kell látnia a felhasználónak a szemeit és az integrált funkciók függvényében akár a részleteit magas felbontás és FPS mellett.

Észlelési fázishoz több módszernek az ötvözetét szokták használni. Ebben az esetben használható a Haar Cascade algoritmus, integrált kép, az AdaBoost, a regressziós fák együttese módszerek és sok egyéb. Abban az esetben, ha valaki közel rendszerfüggetlen, vagy adott hardverre optimalizált, vagy általánosságba véve optimalizált rendszert szeretne alkotni akkor ezek és sok más egyéb technikák használata javasolt.

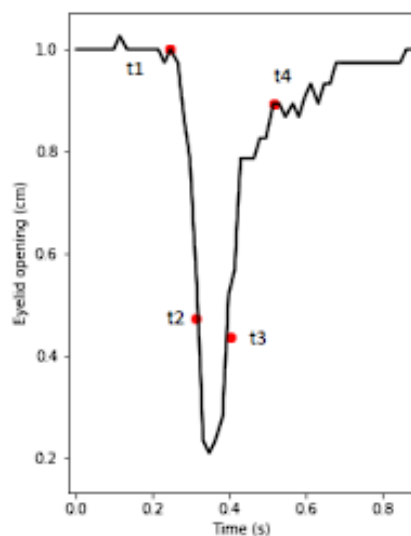
Napjainkban nem feltétlenül szükséges ezeket a lépéseket végrehajtanunk és mindnek a metodikáját részleteiben ismernünk fejlesztéshez, mert publikusan elérhető mindenki számára képi osztályozási modellek. Ezek közül a számomra a legismertebbek az OpenCV és a TensorFlow. Ezek használatának az előnye, hogy előre betanított osztályozókkal rendelkeznek, mint például az arc, a szem, a száj felismerése.

Az arc maszk használata során meg kell határoznunk azt, hogy a maszk mely pontjai a száj vagy a szem. Ezt különböző technikákkal meg lehet határozni ilyen például a pontfelhő adott pontértékeinek a közvetlen megadása is. Ez hasznos lehet abban az esetben, ha egy rendszert betanítani szeretnénk és nincs sok, eltérő személynek az arcáról videó anyagunk. A manuális megadással kiszűrhető a rendszer automatikus kijelölésében keletkező zajokat.



1. ábra Arc tájékozódási pontok [3]

Miután megkaptuk a szem körül elhelyezkedő pontokat meg kell határoznunk a pislogás detektálásához a szemhéjtávolságokat. Érthető módon a szemhéjtávolságok változásának függvényében határozhatjuk meg a pislogás, vagy csak a zaj közötti különbséget. „Ehhez tetszőleges mennyiségű pontot fel kell venni a szem az egyes pislogásoknak a jellemzéséhez. E pontok kiszámításához először a csukott szemeket kell lokalizálni a jel első rendű különbsége és egy adaptív küszöbérték alapján. Ezután, minden egyes észlelt hely esetében a jelet megkeressük azelőtt és a völgy után, amíg meg nem szűnik növekedni egy meghatározott értéken belül, így megkapva a t1 és tx pontokat.” [2]



3. ábra A pislogás jellemzése [2]

Az általam megvalósításra kerülő szoftver esetén a Python 3-at használom fel OpenCV és tkinter vagy azon más változatának egyikével. A UI elkészítéséhez fogom felhasználni a tkinter vagy annak másik változatát és az arc domborzatát pedig az OpenCV-vel dolgozom fel. A megvalósításban elsődlegesen a percnkénti pislogás gyakoriságból meghatározom a fáradtságot és egy egyszerű UI-al megjelenítem a folyamat működését. Emellett ugyan azon a UI-on belül jelzek a felhasználó számára.

Hivatkozások

- [1] K.-A. Kwon, R. J. Shipley, M. Edirisinghe, D. G. Ezra, G. Rose, S. M. Best és R. E. Cameron, „National Library of Medicine,” *Journal of the Royal Society Interface*, %1. kötet10, %1. szám85, p. 20130227, 06 08 2013.
- [2] L. Oliveira, J. S. Cardoso, A. Lourenço és C. Ahlström, „Driver drowsiness detection: a comparison between intrusive and non-intrusive signal acquisition methods,” *IEEE Xplore*, pp. 1-6, 2018.
- [3] M. Siwach, S. Mann és D. Gupta, „A Practical Implementation of Driver Drowsiness Detection Using Facial Landmarks,” *IEEE Xplore*, pp. 1-4, 2022.

Ábrajegyzék

1. ábra Arc tájékozódási pontok [3]	2
1. ábra Arc tájékozódási pontok [3]	2
2. ábra A pislogás jellemzése [2].....	3