



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

SZAKDOLGOZAT

**OE-NIK
2021**

Hallgató neve:
Hallgató törzskönyvi száma:

**Csörgő Erik
T/006161/F112904/N**



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS INFORMATIKAI KAR

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertervezés és -fejlesztés Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Csörgő Erik**
Törzskönyvi száma: T/006161/FI12904/N
Neptun kódja: ZKL37Z

A dolgozat címe:

Álmoság detektálása sofőrökön gépi látás segítségével
Drowsiness detection on drivers using computer vision

Intézményi konzulens: Dr. Vámosy Zoltán
Külső konzulens:

Beadási határidő: 2021. december 15.

A záróvizsga tárgyai: Számítógép architektúrák
Szoftvertervezés és -fejlesztés
specializáció



A feladat:

Tervezzen meg és készítsen el egy olyan rendszert, ami egy videokamera által biztosított bemenet alapján képes felismerni az álmoság tüneteit a pislogás frekvenciája, ásítás, szem zárvatartása és más jellemzők alapján. Detektálja az arcot és azon belül a száj és szemek pozícióját. A megoldás érdekében vizsgálja meg a hasonló rendszerek jellemzőit, tekintse át, hogy ezek a rendszerek milyen megoldási lehetőségeket alkalmaznak és ezekkel milyen eredményeket értek el. Az elemzett megoldások alapján tervezze meg saját rendszerének felépítését, majd készítse el azt! Megoldását tesztelje és eredményeit értékelje, ezek ismeretében pedig vonja le a szükséges következtetéseket. Határozza meg a rendszer hatékonyságát teszt adatok segítségével és hasonlítsa össze a korábban kiemelt rendszerekhez. Végül pedig vázolja fel az esetleges továbbfejlesztési lehetőségeket.

A dolgozatnak tartalmaznia kell:

- a feladat leírását,
- releváns fejlesztések ismertetését, valamint azok értékelését,
- a megvalósítandó rendszer tervét,
- tesztelési, mérési adatokat,
- az eredmények értékelését, más megvalósításokkal való összehasonlítását,
- továbbfejlesztési lehetőségek ismertetését.



Vámossy Zoltán

Dr. Vámossy Zoltán
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

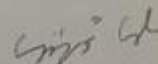
Vámossy Zoltán
.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2024.05.14.


hallgató aláírása



Tartalomjegyzék

1. Bevezetés	9
1.1. Jármű alapú mérések.....	10
1.2. Viselkedés alapú mérések.....	10
1.3. Fiziológiai alapú mérések	11
2. A szakdolgozatban megvalósítandó rendszer	12
3. Hasonló rendszerek	13
3.1. Két kamerás álmoság detektáló rendszer	13
3.2. Szemüveg eltávolító rendszer	14
3.3. EEG hullám alapú álmoság detektálás.....	15
3.4. Konklúzió	16
4. A rendszer részletes specifikációja	18
5. Lehetséges megoldási módszerek	19
5.1. Detektálás	19
5.1.1. Viola-Jones objektum detektáló algoritmus	19
5.1.2. Neurális hálók.....	21
5.1.3. Helyi bináris minta	23
6. Tervezés	24
6.1. Előfeldolgozás	25
6.2. Arcok és jellemzők detektálása	26
6.3. Álmoság ellenőrzés	27
6.4. Eszközök.....	28
7. Implementálás	30
8. Eredmények bemutatása	35
9. Eredmények értékelése	40



10. Továbbfejlesztési lehetőségek	42
11. Összefoglalás	43
12. Summary	43
13. Irodalomjegyzék	45
14. Ábrajegyzék	49



1. BEVEZETÉS

Napjainkba a közúti balesetek nagyrészt sokszor közrejátszik az emberi fáradtság. Magyarországi közúti balesetek 16627 volt 2019-ben és ebből 530 halálos kimenetelű balesettel végződött. Így az ilyen esetekben fontos hamarabb detektálni és figyelmeztetni a célszemélyt. De először is tudnunk kell, hogy milyen állapotai vannak az alvásnak és milyen tünetei vannak egy álmos embernek, amit fel tudunk használni a rendszerünkbe.

Az alvást több fázisra lehet felbontani ebben a sorrendben [1]:

- Éber,
- Gyors-szemmozgás-nélküli (NREM - Non-rapid eye movement) fázis,
- gyors szemmozgás (REM - Rapid eye movement) fázis.

A Gyors-szemmozgás-nélküli fázist még tovább lehet bontani az alábbi állapotokra [2]:

1. Az éber állapot és alvó állapot átmenete (álmos)
2. Könnyű alvás
3. Mély alvás

Mi ezt az „NREM” első fázisát keressük. Az ilyen típusú balesetek általában ezekkel a karakterisztikákkal rendelkeznek [1]:

- Késő este vagy kora délután,
- Nagy sebességű utak,
- Sofőr általában egyedül van,
- Fiatal sofőrökre jellemző,
- Nincsenek féknyomok,
- Gyakori a tragikus kimenetelű baleset.

Ezek nem egyértelműen azonosítják a baleset típusát az esetek komplexitásának köszönhetően. Mivel az előbb említésre került, hogy általában egyedül vannak a sofőrök, így szükséges lenne egy rendszerre, ami jelezné a sofőr állapotát. Ezt már a modern luxus gépjárművekben beágyazott rendszerekkel megvalósították, azonban a régebbi gépjárművekbe nem egyszerű beszerezni.



Az álmoság detektálására több módszert, mértéket fejlesztettek ki és ezeket az alábbi kategóriákba lehet sorolni:

- Jármű alapú mérések,
- Viselkedés alapú mérések,
- Fiziológiai alapú mérések.

Ezekon felül még lehetnek más módszerek is, de ezek a fő irányok.

1.1. Jármű alapú mérések

A jármű alapú, a megközelítésen alapuló rendszerek főleg a gépjármű mozgását és környezetét vizsgálják. Ezek általában egy összetett rendszerként üzemelnek és egyfolytában figyelik a változásokat és reagálnak rájuk.

Ilyen megoldásnak számít a sávváltásra figyelmeztető rendszerek. Egy szenzoron keresztül figyelik az utat képfeldolgozó algoritmusok segítségével és valamilyen formában jelzi a sofőrnek. Az automata fékező segédfunkció is ide sorolható, ami figyelik a jármű előtti területet és ha egy objektum kerül a szenzor(ok) elé, akkor magától elkezd fékezni.

A következő rendszer a sofőr kormány mozgási szokásait figyelik. Fairclough és Graham egy kutatás során megfigyelték, hogy minél álmosabb az ember annál kevesebb mikro-korrigálásokat végez vezetés közben. Sajnos ez nem a legjobb megoldás, mert függ az autó- és az út karakterisztikájától is [3].

Összeségében a jármű alapú megközelítések jól használhatók korrigáló rendszerekhez, de ez a problémája is egyben mivel az állapot bekövetkezte után észlel és cselekszik. Nem egyértelműen az álmoságra nyújt megoldást.

1.2. Viselkedés alapú mérések

Az álmoságnak különböző tünetei lehetnek, amit egy személyen észre lehet venni [2]:

- túl sok vagy túl kevés pislogás,
- bólintás,
- sűrű ásítás.



A viselkedés alapú rendszerek ezek a tulajdonságok alapján próbálják meghatározni az álmoság mértékét. Az ilyen kutatások során főleg a pislogásra fókuszálnak, de még kiemelendő az ásítás is. A PERCLOS (megadja, hogy a szemhéj hány százalékban fedi a pupillát egy periódusban) mértéknek köszönhetően relatív pontosan lehet az álmoságot észlelni.

A problémát a fényhatások okozzák, mivel normál kamera éjszakai képeit nehéz hatékonyan feldolgozni. Másik hibalehetőséget a szemüvegek generálhatnak. Bizonyos algoritmusok még az arc pozíciójára is érzékenyek. Erre egy neurális háló viszonylag jó megoldást tud adni.

Nagy előnye, hogy akár egy mobil telefon kamerája is elég lehet a megoldáshoz. Teszt környezetben magas valószínűséggel eltalálja az állapotot, de sajnos több mindenre kell figyelni valós környezetben, így nem triviális a megvalósítása.

1.3. Fiziológiai alapú mérések

Korábbi metódusoknál az a probléma jelentkezett, hogy csak álmoság bekövetkezte után tudnak jelezni. A fiziológiai megközelítések már hamarabb veszik észre a problémát, mert a fiziológiai változások már korai állapotokban kimutathatóak és ezzel fel lehet fedezni hamis-pozitív eredményeket.

Az álmoság észleléséhez több élettani jelet is fel lehet használni, mint például [4]:

- Elektrokardiográfia (EKG),
- Elektromiográfia (EMG),
- Elektroencefalográfia (EEG),
- Electrooculogram (EOG),
- Pulzus.

Néhány kutatónak az electrooculogram jelek használatával sikerült álmoságot detektálni. Az elektromos potenciálkülönbség egy elektromos mezőt generál a szaruhártya és a retina között és ezt EOG segítségével meg lehet mérni. Jól alkalmazható lassú- és gyors szemmozgások megfigyelésére [4]. A pulzus is felhasználható az alvási fázisok észleléséhez az EKG segítségével. A másik hasonló jel a szívfrekvencia-variabilitás, ami leírja a szív alkalmazkodó képességét a külső- és belső hatásokra. Az EKG alacsony- és magas frekvencia aránya elkezd csökkenni álmos állapotban.



EEG frekvenciák felhasználásával is detektálható a fáradtság. Az EEG felbontható öt frekvencia tartományba (Lásd 1. ábra) [4]:

- Delta: Alvó állapot,
- Théta: Álmos állapot,
- Alfa: Kipihent állapot,
- Béta: Éber állapot,
- Gamma.

Önmagában nem elég pontos az EEG frekvenciák mérése, de EMG jelek megfigyelésével már pontosabb megoldást kapunk. [4].

A négy legáltalánosabb EEG mintázat paraméterei

hullámforma	frekvencia Hz	amplitúdó μV	EEG minta
α -ritmus	7 - 13	15 - 100	
β -ritmus	13 - 30	5 - 50	
δ -ritmus	0.5 - 3	100-200	
θ -ritmus	4 - 7	0.5 - 20	

1. ábra: EEG frekvencia tartományok [19]

2. A SZAKDOLGOZATBAN MEGVALÓSÍTANDÓ RENDSZER

A cél egy olyan rendszer, amely képes egy kamera segítségével észlelni a sofőr állapotát bizonyos tulajdonságok alapján. Ilyen tulajdonságokba tartozik a PERCLOS, a pislogások frekvenciája és az ásítások száma. Kritérium, hogy a program feldolgozási sebessége elegendő legyen valós idejű használatra. Lassú feldolgozás esetén a program pontossága csökkenhet drasztikusan. Számos körülmények ronthatják az eredményt, mint például az aktuális fényhatások vagy a szemüveg. Ezen felül még fontos tényezők lehetnek a használt kamera felbontása és a sebessége.



3. HASONLÓ RENDSZEREK

A programozás szépsége, hogy egy adott problémára nincs egyértelmű válasz, mivel több megközelítés alapján is meg lehet oldani az adott feladatot különböző előnyökkel és hátrányokkal. Nekünk kell meghatározni, hogy milyen kompromisszumokat tudunk elfogadni és mit nem.

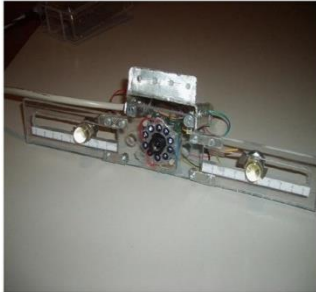
3.1. Két kamerás álmoság detektáló rendszer

Korábban említésre került, hogy az ilyen típusú balesetek nagy része kora délután vagy este történtek, így fontos erre valamilyen megoldást találni változó körülményben való detektálásra. Egyik ilyen lehet a hisztogram kiegyenlítés, de léteznek még egyéb technikák is [5].

Például 2010-ben Marco Javier Flores, José María Armingol és Arturo de la Escalera két kamerás rendszert terveztek a megvilágításbeli problémákra. Egyik kamera a nappali körülményekért felelős, a másik pedig az éjszakai körülményekért. Érdemes kiemelni, hogy mindkét kamera szürkeárnyalatos felvételeket készít [5].

A nappali kamera kezdetben Viola-Jones objektum detektort használ. Ugyan fejlesztés közben rájöttek, hogy ez a módszer érzékeny a fej pozíciójára, így végül egy „Condensation Algorithm” -ot használtak egy neurális hálóval megtámogatva. Ez a megoldás már kevésbé volt érzékeny az említett problémára. A szem állapotára pedig egy SVM-et (Support Vektor Machine) használtak. [5]

Az éjszakai rendszer egy infravörös kamerát használ (2. ábra). A kamera két kép mezőt generál: egy párost és egy páratlant. Az első kép úgy jön létre, hogy minden pixelen végig haladva kiszámolja a páros és páratlan pixel különbségét. Így keletkezik egy kép külső megvilágítások nélkül és a pupilla világosan szerepel a képen. Miután a kép átesett több előfeldolgozó algoritmuson, akkor egy SVM segítségével detektálja az arcot. A szemek követésére szintén a korábban használt „Condensation Algorithm” -ot alkalmazza. [5]



2. ábra: a) infravörös kamera b) sofőrről fotó c) páratlan(felső) kép d) páros(alsó) kép [5]

Álmosság paramétereként alkalmazták a PERCLOS-t és a pislogási frekvenciát. Ezen felül még figyeli, hogy a sofőr figyelme a vezetésen van-e. Ha valamelyik paraméter nem felel meg, akkor jelez a felhasználónak [5].

Előnye, hogy stabilan jól alkalmazható a változó körülmények között, de cserébe igényel egy extra infravörös kamerát és ezek között be kell állítani, hogy mikor melyik üzemeljen. Ez a két kamera csak egy szögből figyeli a sofőrt. Akár előnyösebb lehet, ha ezt két különböző szögből végzi.

3.2. Szemüveg eltávolító rendszer

A szemüveg komoly problémát tud okozni az objektum felismerő algoritmusoknak. Erre készített egy megoldást Wen-Chang Cheng, Hsien-Chou Liao, Min-Ho Pan és Chih-Chuan Chen [6].

Az arc detektálásához hagyományos megoldásokat használtak, amik megtalálhatóak az OpenCV könyvtárban. Ezt követően az alábbi módszer szerint megpróbálják eltávolítani a szemüveget a képről. Az arc területét elnevezi FR -nek, míg az FR oldalait ezen jelölések használatával különbözteti meg: FR_{left} , FR_{right} , FR_{top} és FR_{bottom} . A szem területét (RI) a 3. ábra által feltüntetett képletekkel számolja [6].

$$RI_{left} = FR_{left} + FR_{width}/8$$

$$RI_{top} = FR_{top} + FR_{height}/4$$

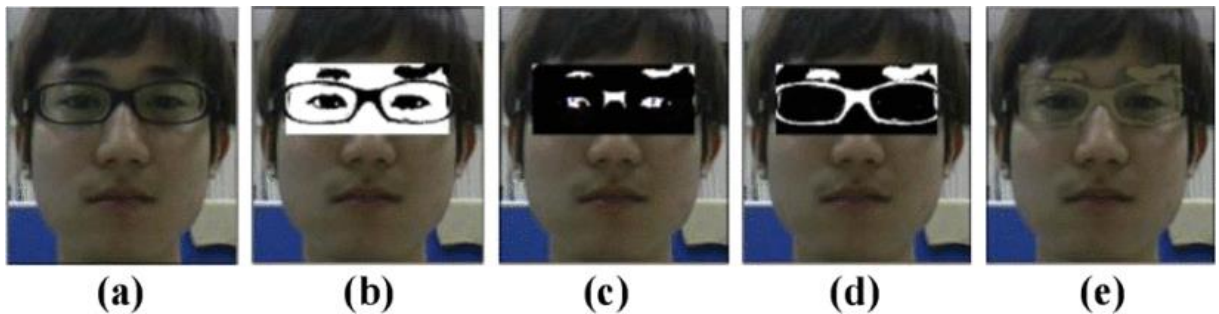
$$RI_{right} = FR_{right} + FR_{width} - FR_{width}/8$$

3. ábra: Szemüveg eltávolításhoz használt képletek



Miután meghatározta a szem területét, akkor felbontja jobb és bal szemre. A következő lépés a szemüveg maszkolása (4. ábra) [6]:

1. Először szürkeárnyaltos és utána bináris képpé alakítja egy 40-es határral.
2. Dilatáció műveletet végez el a képen. Mindkét szem kiválasztja a centerhez legközelebbi objektumot.
3. Invertálja a bináris képet.
4. A két objektumon lefut a határ kitöltő funkció, így eltünteti a szemgolyót. A fehér terület lesz a szemüveg maszkja.
5. A maszk alapján kiszínezi a szemüveget a bőrszínnel.



4. ábra: A szemüveg eltávolítás folyamatai [6]

A szem követésére OpenCV beépített kaszkád osztályozóját használták fel. A két szemet külön-külön azonosítja. A korábban létrejött szem területeket használja mintának a követéshez. Ha elveszti az arcot, akkor kezdődik előlről a folyamat.

A fáradtság mérése két lépésből áll. Első lépés, hogy a képet RGB szín modellből át kell váltani HSL-be. Az S csatornán pedig hisztogram kiegyenlítést kell végezni. Végül az S csatornás képet bináris formátumba átalakítja. Legvégül pedig a szem formáját vizsgálja. Ha lapos az objektum, akkor csukva van különben nyitva.

Ez a megközelítés részben megoldja a szemüveg problémáját, de cserébe az algoritmus több álpozitív eredményt generál. Az arc pozíciójára még mindig érzékeny marad, mivel 30 fok fölött nem detektálja az objektumot.

3.3. EEG hullám alapú álmoság detektálás

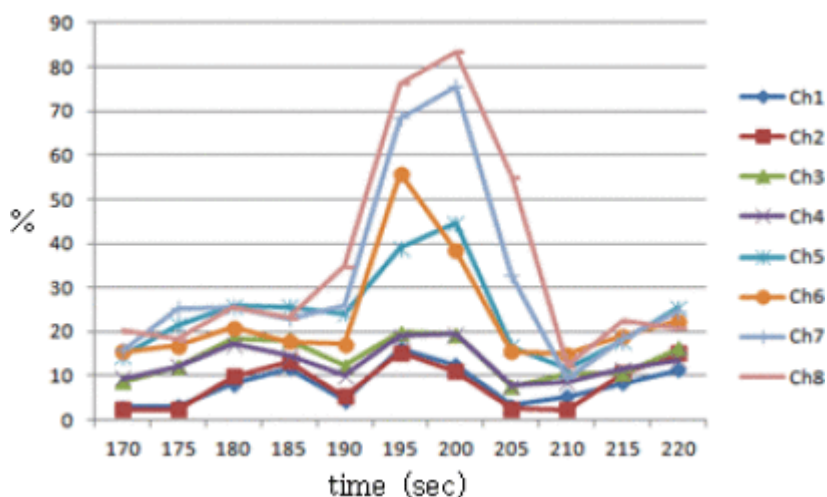
A bevezetésben említésre került, hogy az ember fizikális vizsgálatával is észlelhető a fáradtság. 2012-ben Dajeong Kim, Hyung-Seob Han, Sangjin Cho és Uipil Chong



bemutatták a saját megvalósításukat. Az ő megoldásuk az EEG hullámokon alapszik [7].

A méréshez a nyakszirti lebenyhez és a homloklebenyhez EEG elektródákat helyeztek el, mivel ezek a részek felelnek az alvó állapotokért. Ha a személy nyugodt és csukva van a szeme, akkor az alfa hullámok kezdenek felerősödni (5. ábra). Ahogy elkezdi kinyílni az alfa hullámok is eltűnnek és a helyüket a béta hullámok veszik át [7].

Nagyon pontos eredményre képes, habár a mérő eszközök viselete kényelmetlen.



5. ábra: Alfa hullámok változása álmos állapotban [7]

Sokszor akár korlátozhatják a sofőr fejmozgását a kábelek.

3.4. Konklúzió

A korábbi pontokban említésre kerültek hasonló rendszerek. Mindegyik megold egy problémát, azonban ezek általában valamilyen hátrányokat is hordoznak magukban. Ezeket összefoglalva az 1. táblázat mezőiben láthatjuk:



	Előny	Hátrány
Kétkamerás rendszer	<ul style="list-style-type: none">• Bármilyen fénykörülmények között használható• Nem érzékeny a fej pozíciójára	<ul style="list-style-type: none">• Két kamerára van szükség• Össze kell hangolni a két kamerát
Szemüveg eltávolító rendszer	<ul style="list-style-type: none">• Megoldja a szemüveg problémát	<ul style="list-style-type: none">• Sok álpozitív eredményt generál• Érzékeny a fej pozíciókra
EEG hullám alapú álmoság detektálás	<ul style="list-style-type: none">• Nagy pontossággal rendelkezik	<ul style="list-style-type: none">• Kényelmetlen szenzorok, kifejezetten vezetés közben

1. táblázat: Összefoglaló táblázat



4. A RENDSZER RÉSZLETES SPECIFIKÁCIÓJA

Szükségünk van egy olyan rendszerre, amely az alábbiakra képes:

- képes az arc azonosítására változó körülmények között,
- elég legyen egy hagyományos kamera a monitorozáshoz,
- valós időben fusson,
- a szemén kívül az ásítást is figyelje,
- erőforrásigénye lehetőleg alacsony legyen.
- Három állapotot jelezzen a rendszer: először is a kezdő állapot az éber állapot. Következő állapot az álmos állapot. Ez abban az esetben következik be, ha az álmoságot leíró jellemzők paraméterei kritikus érték alá esnek. Ha a sofőr két másodpercig nem nyitja ki a szemét, akkor kerül alvó állapotba.
- Álmos állapotban csak valamilyen enyhe jelzést közvetít a felhasználó felé. Alvó állapot esetén intenzív jelzést ad ki, hogy azonnal felébredjen a felhasználó (2. táblázat).

Állapot	Bekövetkezés	Jelzés intenzitás
Éber	Alapértelmezett	-
Álmos	Valamelyik paraméter kritikus tartományba kerül	enyhe
Alvó	A felhasználó nem nyitja ki a szemét 2 másodpercen belül	intenzív

2. táblázat: Állapotok leírása



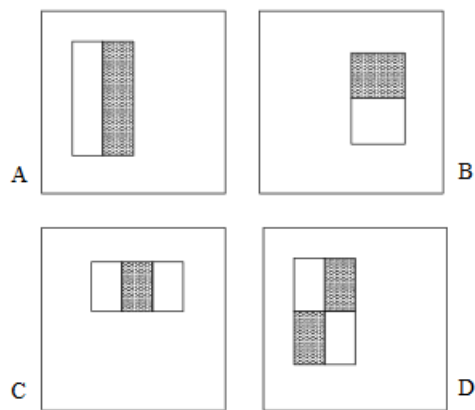
5. LEHETSÉGES MEGOLDÁSI MÓDSZEREK

5.1. Detektálás

5.1.1. Viola-Jones objektum detektáló algoritmus

Paul Viola és Michael Jones mutatta be ezt a módszert még 2001-ben. Elsődlegesen az arcfelismerés inspirálta a projektet. Habár ez a megoldás már idejétmúlt és nem olyan pontos, mint az újabb megoldások, de még mindig jó sebességet lehet vele elérni valós idejű rendszereknél köszönhetően az egyszerűségének [8].

A képet érdemes átalakítani szürkeárnyalatos képpé, mivel az ilyen képeken nagyobb teljesítményt lehet elérni. Nem közvetlen a kép intenzitás értékekkel dolgozik, hanem jellegzetességek készletét használja. Pontosabban a Haar-jellegzetességeket használja (6. ábra).



6. ábra: Haar-jellegzetességek [8]

Egy ilyen jellegzetesség két vagy több azonos méretű és szomszédos téglalapokat vizsgál. Mindkettő területnek kiszámolja az intenzitását és megvizsgálja a különbségüket egymáshoz képest. Például a szem régiója alapvetően sötétebb, mint az arc, így az egyik szem jól beazonosítható egy két-téglalapos jellegzetesség sötétebb részével [9].

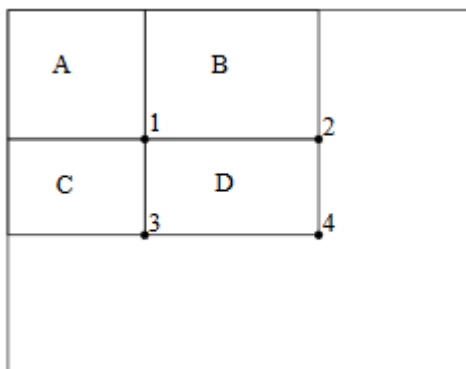


A jellegzetességek felgyorsítása érdekében egy úgy nevezett integrál képet használ. Az integrál kép (x, y) pontjában a pont feletti pixel összegét tartalmazza az (x, y) pixelt is beleértve (**Hiba! A hivatkozási forrás nem található.**).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

7. ábra: Egy pont kiszámítás az integrál képen
(ii – integrál kép, i – eredeti kép) [8]

Az előnye, hogy miután egyszer kiszámolta az integrál képet onnantól kezdve már a jellegzetesség konstans időben kiszámítható [9]. Ehhez elég nyolc értéket kiolvasni a két téglalap esetén (lásd 8. ábra).



8. ábra A D terület kiszámításához ki kell
vonni az A , B és C területek értékét [8]

Még szüksége van valamilyen gépi tanuló megoldásra, ami osztályozni fogja a nagy mennyiségű jellegzetességeket. Viola-Jones egy bizonyos AdaBoost nevű gépi tanulási algoritmust használ a gyenge osztályozójuk teljesítményének javítására. Minden menetben meghív egy új gyenge osztályozó és frissíti a súlyokat. Viola-Jones esetén egy gyenge osztályozóhoz kiválasztja azt a jellegzetességet, amely legjobban szétválasztja a negatívát a pozitívtól. A tanítási folyamat során a legtöbb jellegzetesség elvetésre kerül és csak néhány száz vagy ezer marad meg [9].

Az osztályozók növekvő komplexitása miatt kaszkád szerkezetet alkalmaz a sebesség növelésére. Lényegében egy ablakról van szó, amely megmondja mekkora



valószínűséggel tartalmazza az objektumot és ha egy bizonyos érték alatt van, akkor azt elengedi és a komplexebb számításokat csak az ígéretesebb régiókon végzi el [9].

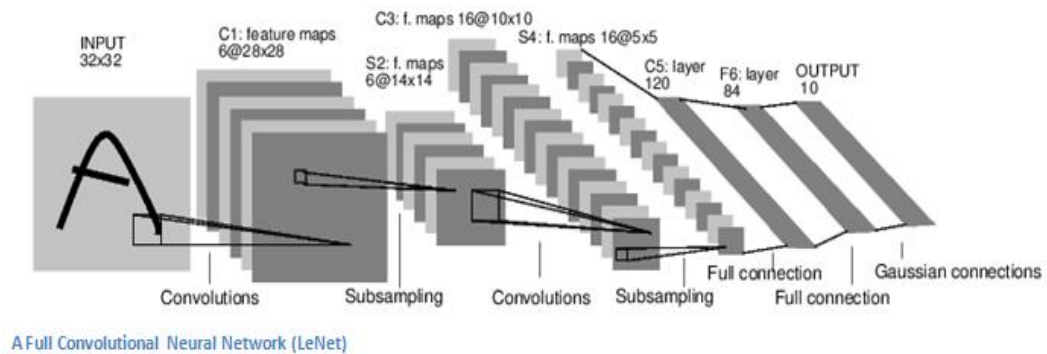
Az előnye a sebessége és relatív alacsony álpozitív eredmények aránya, de sok tanulási idő árán éri el és érzékeny a fej pozíciójára.

5.1.2. Neurális hálók

A neurális hálók az emberi agy működését próbálják lemodellezni és így egy hatékonyabb rendszert alkotni a hagyományos megoldásokhoz képest. Alkalmas adatok csoportosítására, meglévő adatok alapján jóslásra és objektum felismerésre, habár még lehetne sorolni.

Konvolúciós neurális hálózatok (CNN – Convolution neural network)

A konvolúciós neurális hálózatok a „deep-learning” egy speciális típusa, amelyet főleg képek feldolgozására fejlesztettek ki. Az ilyen típusú neurális hálók 3 típusú rejtett réteggel rendelkeznek (Konvolúciós, Összevonó, Teljesen összekötött) (9. ábra):



9. ábra: CNN felépítése [10]

A konvolúciós réteg általában több aktivációs térképpel rendelkezik, amelyet filterek segítségével számol ki. A mélysége a szűrők számától függ. Ha a kapott eredmény egy nagy szám, akkor a filter által keresett objektum nagy valószínűséggel megtalálható (10. ábra) [10]



Visualization of the
receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive
field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)

10. ábra: Filter használata [10]

Az első rétegnél a bemenet az maga a kép és következő rétegben az előző réteg lesz a bemenet. A rétegek elején csak egyszerű vonalakat vagy formákat képes felismerni, de minél mélyebbre jutunk annál komplexebb alakokat képes azonosítani a szűrőknek köszönhetően [10].

A következő réteg az összevonó réteg. Általában két konvolúciós réteg között helyezkedik el. Használható a konvolúciós réteg dimenziójának csökkentésére, de akár a jellegzetességek kiemelésére is alkalmas. Egyik ilyen módszer a „max pooling” megkeresi egy területen a maximum értéket és az lesz az új érték. Hasonlóan működik az „average pooling” csak a terület átlagát írja be az új helyre [10].

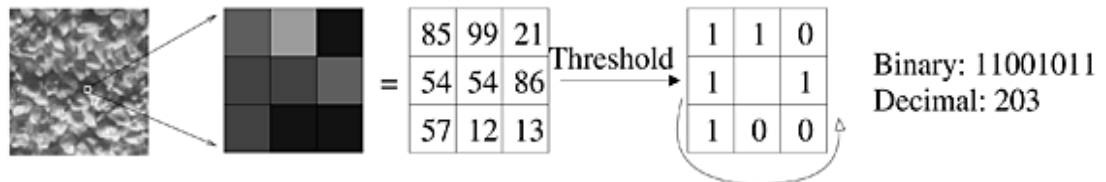
Az utolsó pedig a teljesen összekötött réteg. Össze van kötve az előző réteg összes neuronjával. A feladat az osztályozás a kapott eredményekből. A kimeneteken határozza meg, hogy egy objektum típus mekkora valószínűséggel található meg a képen.

Nagy előnye, hogy sok környezeti tényezőtől független, mint például a fej pozíciója vagy a szemüveg. A nagy pontosságért cserébe nagy teljesítmény igénye van és nem triviális a megértése. Videokártya használata ajánlott.



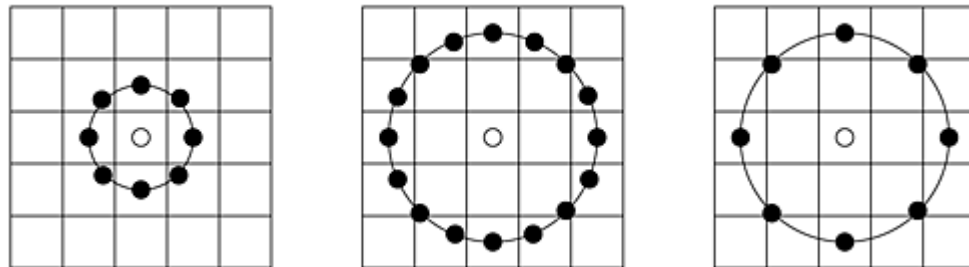
5.1.3. Helyi bináris minta

Eredetileg textúrák leírására alkalmazták, de arc felismerésre is felhasználható. Minden pixelre kiszámít egy bináris kódot és átváltja decimális számmá. Ha a szomszédnak nagyobb vagy egyenlő az értéke, akkor 1-es kerül be különben pedig nulla. Ezt egy sorrendbe kiolvassa, mint bináris kód és konvertálja decimális számrendszerbe (11. ábra) [11].



11. ábra: A bináris kód számításának folyamat [11]

Később ezt kibővítették úgy, hogy nagyobb sugarú szomszédokat vizsgáltak egy megadott sugarú körben. Ez segített a különböző méretű textúrák leírásában (12. ábra).



12. ábra: Módosított változat [11]

Ez a módszer akár több más osztályozó algoritmussal is használható, mint például az SVM vagy egyéb gépi tanuló algoritmus.

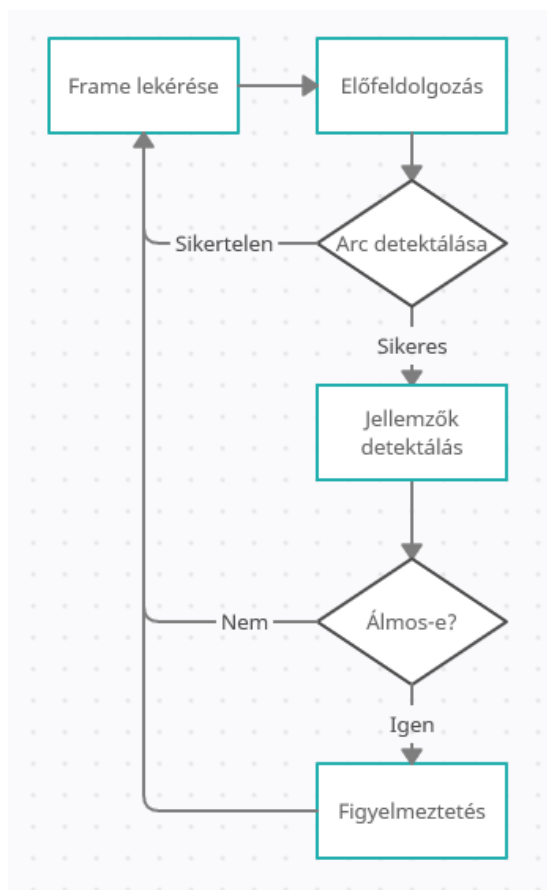
Számítás szempontjából egyszerűbb, így egyben gyorsabb is. A hátránya, hogy csak szürkeárnyalatú vagy bináris képen alkalmazható és nagyobb területen a hibák jelentősen növekednek.



6. TERVEZÉS

A fentebb specifikált kritériumoknak megfelelően az alábbi rendszervázlat készült el (13. ábra):

1. Frame lekérése
2. Kép előfeldolgozása
3. Arc detektálása
4. Arcjellemzők detektálása
5. Álmoság ellenőrzése
6. Figyelmeztetés



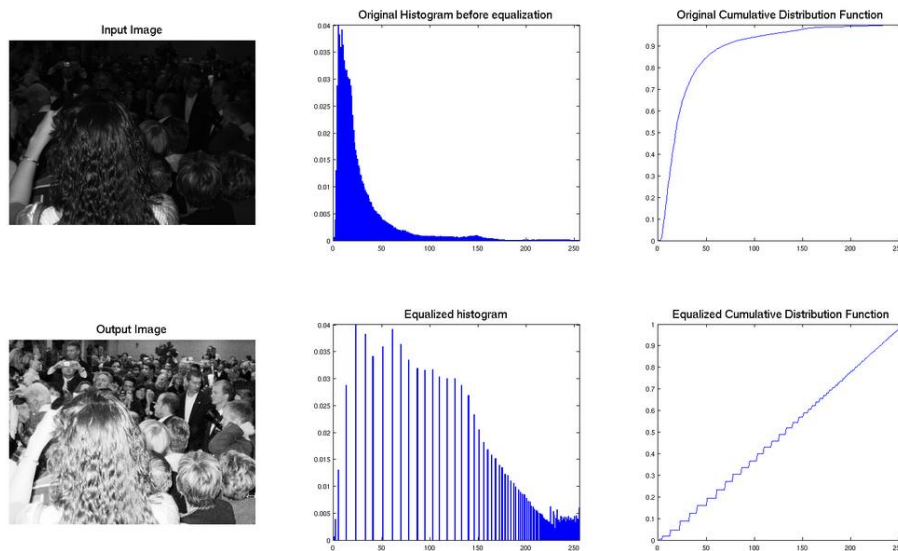
13. ábra: A rendszer folyamatábrája



6.1. Előfeldolgozás

Először a képet ajánlott valamilyen szűrő által feldolgozni a korábban említett különböző fényhatások miatt. Erre alkalmas módszer a hisztogram kiegyenlítés (HE – Histogram equalization). Az alábbi módszer a kép intenzitás hisztogramját használja a kontraszt módosításához. Ez a hisztogram tárolja, hogy egy bizonyos intenzitás érték hányszor jelent meg a képen. A kiegyenlítés egy nem lineáris leképezés segítségével egyenletesen eloszlatja az új intenzitás értékeket és egy közel lapos hisztogramot hoz létre (14. ábra). Egy RGB képnél minden színnek van egy saját hisztogramja, így ebben az esetben minden táblára ki kell számolni az új hisztogramokat. Ezért érdemes valamilyen más formátumba átalakítani a képet (Például: HSV - Hue, Saturation, Value). [12].

Ennek a tovább fejlesztett változata az Adaptív (vagy lokális) Hisztogram kiegyenlítés, amely annyiban különbözik, hogy egy ablak használatával megy végig a pixeleken és csak a szomszédos pixel értékeket veszi figyelembe. Az előnye, hogy nem az egész kép alapján módosít.

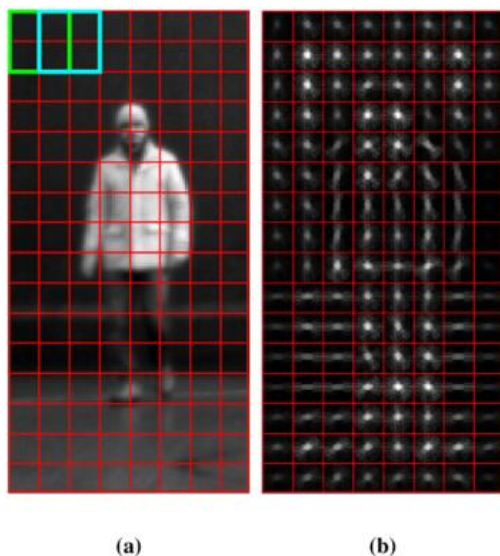


14. ábra: Hisztogram kiegyenlítés [8]



6.2. Arcok és jellemzők detektálása

Miután megvan az előfeldolgozás, akkor a következő lépés az arc észlelése, de ugyan ez a módszer alkalmazható a szem és a száj felismerésére is. Erre a célra használható a Hisztogram-orientált színátmenetek (HOG -Histogram oriented gradient) jellemző leíró módszere. Az objektum formájára fókuszál hasonlóan, mint az élek detektálásánál. A különbség a két megközelítés között, hogy a HOG esetében az éleknek irányuk is van. Ezeket az éleket úgy számolja ki, hogy kisebb blokkokra bontja és ezeknek a blokkoknak számolja ki az irányát és a színátmeneteket. A végén minden részének kiszámolja a hisztogramját irány és színátmenetek segítségével (15. ábra).

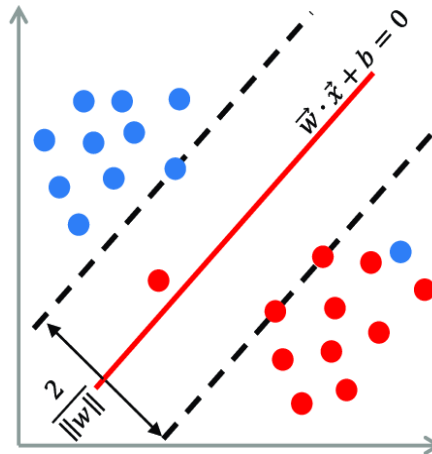


15. ábra: a) blokkokra bontás b) irány és átmenetek meghatározása [21]

Még szükség van valamilyen osztályozó rendszerre az arcfelismeréshez. Ilyen rendszer lehet a SVM (Support Vector Machine), amely egy felügyelt tanuló algoritmus. Ezt főleg osztályozásra (Lineáris) szokták használni, de használható még regresszió-analízishez (nem lineáris) is. Bizonyos minták tanítása után képes két csoportba kategorizálni a kapott bemenetet (16. ábra). A feladat szempontjából teljesen jól használható, mivel az egyik csoportot emberi arcként osztályozza és a másik csoportot kedvezőtlen értékekkel elveti. Bizonyos tulajdonságok alapján próbálja meghatározni,



hogy melyik címkébe tartozik a kép egy határ alapján, amely két síkra bontja az adatokat.



16. ábra: Lineáris SVM [22]

6.3. Álmoság ellenőrzés

Ebben részben már olyan fontos tulajdonságokra kell fókuszálni, mint például a száj és a szem. A szem figyelésére több módszer is van, amelyet az úgynevezett „Facial landmarkokkal” könnyű meghatározni. Először ilyen a pislogás számolása. De előbb meg kell határozni, hogy mi számít pislogásnak.



17. ábra: a) Bal szem b) Jobb szem indexelése [10]

A 17. ábra mutatja, hogy a példának a dlib „facial landmark” algoritmus, hogyan jelöli a szem pontjait. Ezen pontok alapján az alábbi módon lehet számolni a szemoldalarányát (EAR – Eye-aspect ratio) [13]:



$$EAR = \frac{(42 - 38) * (41 - 39)}{(2 * (40 - 37))}$$

Így kapunk egy 0 és 1 közötti értéket. Ha ez az érték kisebb, mint a kijelölt határérték akkor van a szem zárva különben nyitva van. Így már lehet számolni a pislogások számát, frekvenciáját és PERCLOS-t is meg lehet határozni.

Fontos tudni, hogy éber állapotban a pislogás frekvenciája körülbelül 10-20 pislogás/perc. Ha ez az érték a tartományon kívülre kerül, akkor van rá esély, hogy a sofőr álmos, de ez nem garantált. Az alábbi képlettel lehet kiszámolni [14]:

$$f_{BFreq} = n/N$$

Az n az a nyitott állapotból a zárt állapotba váltások száma és N pedig a képkockák száma egy periódus alatt.

A következő fontos paraméter a PERCLOS. Fentebb említésre került, hogy ez a mérték megadja, hogy a szemhéj hány százalékban fedi a pupillát egy periódusban. Kiszámítása így történik [14]:

$$f_{PERCLOS} = n_{close}/N_{total}$$

Az n_{close} ebben az esetben csukott szemű képkockák számát jelöli, míg N_{total} pedig az összes képkockák számát jelöli egy perióduson belül.

Miután detektáltuk a szájat a képen meg kell határoznunk az ajkak közötti távolságot. Ha ez a távolság átlép egy határt akkor vesszük ásításnak. A számítás hasonlóan történik, mint a szem-oldalarányánál.

Fontos megjegyezni, hogy ezek a mértékek nem teljesen pontosak, de közelítőleg jó eredményt adnak. A hatásfokot növelni tudjuk, ha ezeket a paramétereket egyszerre alkalmazzuk.

6.4. Eszközök

A választott programozási nyelv a c++, mivel az OpenCV és a dlib a c++ használja. Azonban lehetne használni ezen könyvtárak Pythonban megírt változatait is, de a c++ az alapértelmezett.



Dlib C++ könyvtár

Olyan fejlesztői eszköz, amely gépi tanuló algoritmusokat tartalmaz és egyéb eszközöket, amelyekkel komplexebb c++ programokat lehet létrehozni. A könyvtár „open-source”, így szabadon alkalmazható. Sok dokumentációval és példa kóddal rendelkezik. Számos témakörben vannak algoritmusok implementálásai, mint például numerikus algoritmusok és képfeldolgozó algoritmusok. Rendelkezik grafikus felhasználói felülettel. Ezen felül még magába foglal egy moduláris unit test keretrendszert [15].

OpenCV C++ könyvtár

Hasonlóan Dlib-hez ez a könyvtár is open-source [16]. Ez a könyvtár is a képfeldolgozás algoritmusával foglalkozik. Tartalmaz egy neurális hálókval foglalkozó modult is, amely DNN névre hallgat. Itt számos előre betanított modell is felhasználható és támogat más külsős könyvtárakat, mint például a Tensorflow.



7. IMPLEMENTÁLÁS

Első lépés, hogy a kamerától lekérjük az aktuális frame-et. Ha ez megtörtént, akkor elindul az előfeldolgozás. Pontosabb feldolgozás érdekében hisztogram kiegyenlítést alkalmaz a képen, így megkönnyítve az arc detektálását.

Ekkor a képet át kell alakítani a Hisztogram-orientált színátmenet módszer segítségével. Akár más módszereket is lehetne alkalmazni, mint például a bináris képpé konvertálás, de HOG az alapértelmezett módszer a dlib könyvtárban. Az osztályozásra lineáris SVM-et használ, de ezt bármikor le lehet cserélni, ha nem megfelelő az algoritmus teljesítménye, mivel a dlib és az OpenCV is többféle algoritmust valósít meg, mint például a konvolúciós neurális hálókon alapuló arc felismerést, amelyet később láthatunk. Ha nem sikerül az algoritmusnak megtalálnia az objektumot, akkor visszalép az első szakaszba.

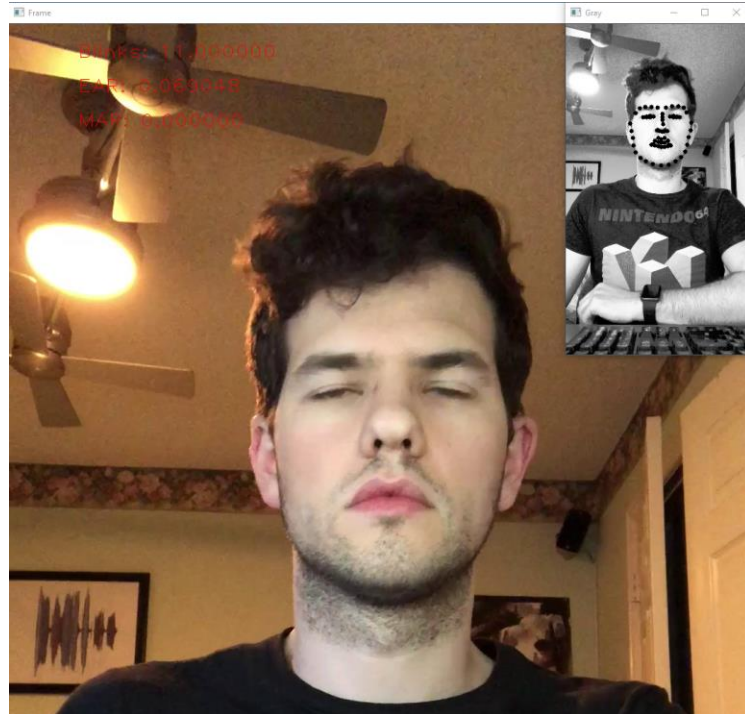
A következő lépés az arcpontok detektálása a kijelölt területen. Sikertelen próbálkozás esetén itt is visszalép a kezdő állapotba. Sikeres észlelés esetén megkezdhető a paraméterek kiolvasása. Ebben a fázisban segítségünkre lehet a dlib-ben megvalósított „facial landmark” detektor. A detektornak köszönhetően a fontos pontok koordinátáit megkapjuk.

A meglévő adatokból kiszámolja a rendszer a szem-oldalarányát és meghatározza a szem állapotát. Ezután következnek a paraméterek kiszámítása (PERCLOS, pislogási frekvencia) és ellenőrzi, hogy két másodpercnél tovább volt-e csukva a felhasználó szeme. Természetesen az ásításra is figyel.

Ha valamelyik paraméter kritikus tartományba esik, akkor a korábbiakban megállapított figyelmeztetés intenzitása szerint figyelmeztet és a folyamat kezdődik újra az elejétől, míg a felhasználó meg nem szakítja.



18. ábra mutatja meg, hogy pontosan mit lát a felhasználó és az algoritmus. Éppen pislogást láthatunk, amely még az aktuális EAR értéken is észrevehető. Ilyenkor fontos figyelnie az algoritmusnak, hogy egy pislogást ne észleljen többnek. A detektáló bizonytalanságából adódhatnak pontatlanságok működés közben. Ezért az algoritmusnak figyelembe kell vennie a korábbi képkockák eredményét.



18. ábra: Nagy képen a program működés közben, kis képen a program ahogy látja az arcot

Fejlesztés közben felmerült egy olyan probléma, hogy ha az algoritmust nagyobb felbontású képen alkalmazzuk, akkor sok időt visz el az arc detektáló algoritmus, így ebben az állapotban nem alkalmas valós idejű alkalmazásra. Két különböző megoldás került kipróbálásra az előző probléma kiküszöbölésére. Az első megoldásban újra méretezzük a képet. Ebben az esetben már elfogadható eredményeket tudunk felmutatni a felbontás kárára. A második megoldás az alkalmazás felépítése módosulna úgy, hogy feltételezzük a szűrő fej pozíciója fix, így csak egyszer keressük meg arcot az alkalmazás legelején. Ezzel számítások jó részét meglehetősen kerülni. De ez a dlib facial landmark algoritmusában problémát okoz, mivel az arcra sokszor rosszul vagy egyáltalán nem helyezi rá a 68 pontos maszkot. Így végül az első megoldás került kiválasztásra az alacsonyabb felbontás ellenére.



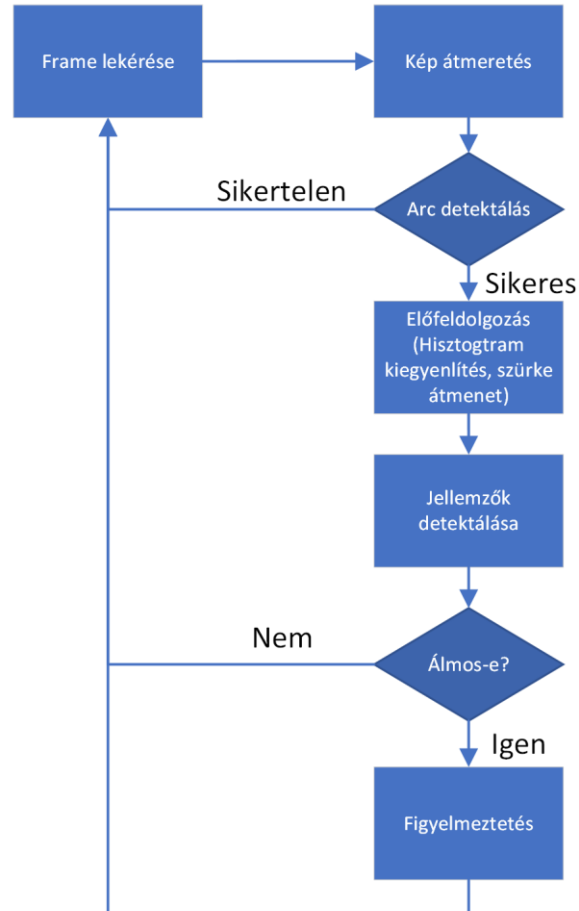
Később előkerült egy hiba a rendszerben. Eredetileg előre definiált értékkel volt meghatározva az EAR határértéke, így a program nem vette figyelembe az eltérő szem méreteket. Erre olyan megoldás született, hogy a program első percében csak mintát vesz az alkalmazás és később futás közben ezt újra kiértékeli. Miután végzett kiszámolja az átlagot és veszi körülbelül a két harmadát. Ez a paraméterbeállítással jelentős mértékben pontosabb lett változó szemméretek esetén.

A továbbiakban egy módosított algoritmus is fog szerepelni, amely Caffé modellt használ az arc detektáláshoz. Ez egy deep learning alapú előre betanított modell. Az felépítése a ResNet10 architektúrán alapszik és Single Shot Detector(SSD) elven működik. Az SSD-t kifejezetten valós idejű feldolgozáshoz fejlesztették. Nagyobb sebességért cserébe viszont az algoritmus pontosságát áldozza be.

Az OpenCV DNN moduljába tölti be a modellt. A DNN modult csak futtatáshoz lehet használni, ellenben más könyvtárakkal, ahol tanítani is lehet a modelleket. Viszont nagy előnye, hogy sebessége jelentősen gyorsabb magasabb felbontások, mint a korábbi megoldás még processzoron is.

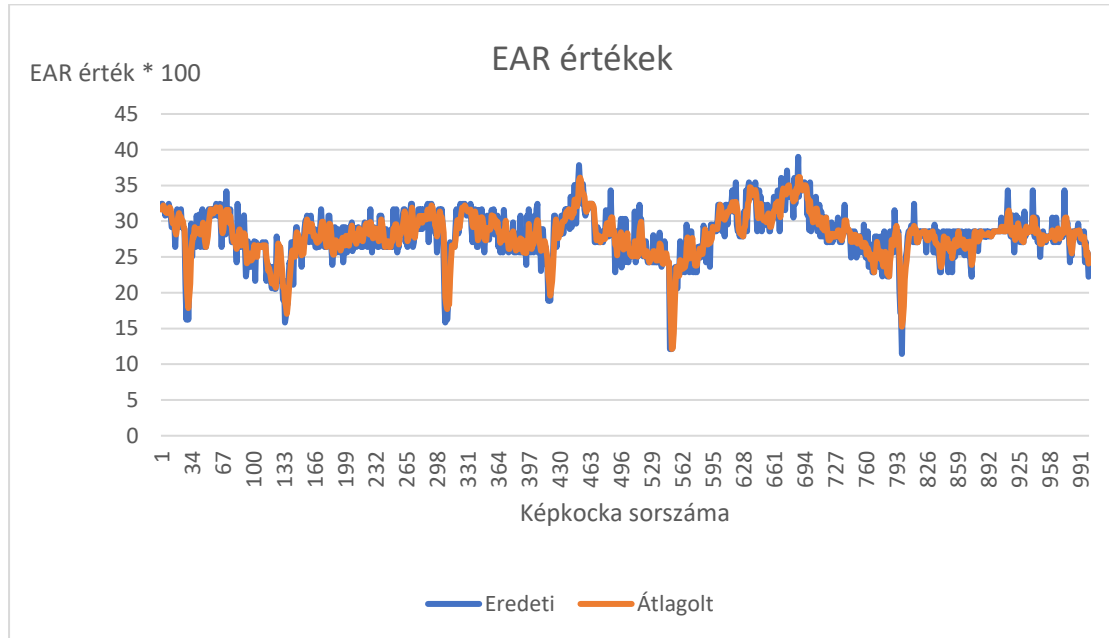
Fontos, hogy bemenetként színes 300x300-as képet vár el, amelynek köszönhetően minimálisan át kell strukturálni a korábbi algoritmust.

A 19. ábra felépítésén láthatjuk, hogy szét lett választva két különböző részre. Az arc detektálás előtt csak átméretezi a képet és az előfeldolgozás eddigi műveleteit csak a detektálás után végzi, de még a jellemzők detektálása előtt. Kiemelendő, hogy az arc jellemzők detektálásán az eredeti képet elemzi ki és nem az átméretezettet.



19. ábra: Módosított Caffè megoldás

Első megoldásban csak egyetlen EAR érték alapján volt meghatározva a szem állapota. Ellenben az utolsó verzióban már az utolsó három EAR érték átlagát használja fel, így csökkentve a pontatlanságnak köszönhető hirtelen kiugrásokat. Így a kiugrások is alacsonyabbak, ezért az EAR határértékét is növelni kell, amely a pislogás érzékenységét fokozza.



3. táblázat: EAR értékek 17-es video első 1000 képkockáiból

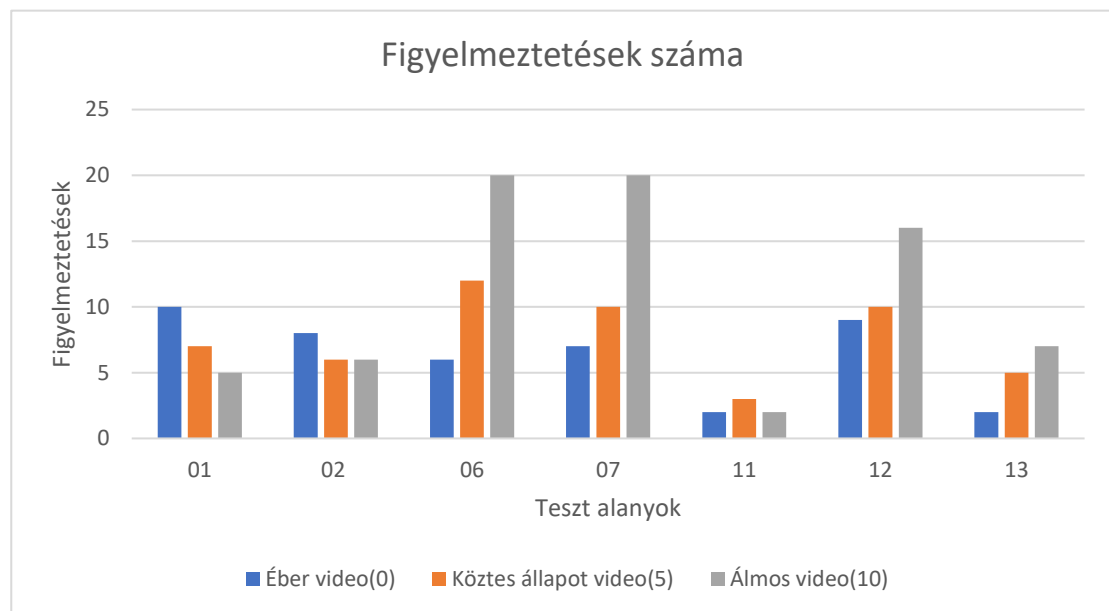
A 3. táblázaton láthatunk egy minta EAR diagramot, amely a 17-es számú teszt alany álmós videója. Kék színnel láthatjuk az eredeti (nem átlagolt) értékeket, míg a narancssal az átlagoltat. Az alacsonyabb kiugró értékek a pislogások. A diagramból kiolvasható, hogy az eredeti az erősebb pislogásokat gond nélkül detektálja, de problémát okozhat a határérték körül mozgó pislogás, így pedig fals pozitív pislogást generálhat ezzel torzítva a végeredményt. Ezáltal kevesebb fals pozitív eredményt fog generálni az átlagolt EAR érték számítás.



8. EREDMÉNYEK BEMUTATÁSA

A program tesztelésére az UTA (University of Texas Arlington) álmoság adatbázisa [17] nyújtott segítséget, amely közel 30 órányi videót tartalmaz több jelentkezőről. A videók több kategóriákba vannak sorolva a fáradtság mértékétől függően, ezzel megkönnyítve a tesztelést. Jelentkezőket megkérték, hogy csináljanak 3 videót különböző éberségi állapotokban. A videók közben valamilyen tevékenységet kellett csinálniuk, mint például olvasás vagy videók megtekintése, de kötelező volt egy olyan időszak is, amikor nem csináltak semmit. Így az alábbi tevékenységekkel próbálták szimulálni a sofőrök állapotait. 0-tól 10-ig terjedő skálán vannak megjelölve a videók. Míg 0 az éber videókat jelzi, addig a 10 az álmosakat. A tesztelés során az adatbázis hat jelentkezőre lett letesztelve. Minden futtatás után lementésre került a figyelmeztetések száma, amit az alábbi táblázatban láthatunk összehasonlítva egymáshoz képest:

A 4. Táblázatban láthatjuk a mérési eredményeket személyenként csoportosítva. Ezek már a korábban említett módosításokkal készültek, így már nincsenek akkora kiugró eredmények, mint korábban.



4. Táblázat: Figyelmeztetések száma teszt alanyokra bontva



Továbbiakban a két megoldást kell összehasonlítani: a HOG+SVM és a Caffé modell alapú megoldást. Három markánsan különböző tesztalany került kiválasztásra (20. ábra). Ezáltal megtekinthető a két algoritmus előnye és hátránya másféle szituációkban. Alapvetően a videók nem tartalmazznak pislogásról adatokat, így sajátkezűleg kellett a pislogás idejét meghatározni a képkocka sorszáma szerint. Ennek okán előfordulhatnak mérési pontatlanságok, habár jó közelítő eredményt szolgáltat.

Továbbá több objektum detektálási metrika kerül bemutatásra [18]. Először is a detektálási eredményeket négy különböző kategóriába lehet sorolni:

- True positive (Valós Pozitív),
- False positive (Fals pozitív),
- False negative (Fals negatív),
- True negative (Valós negatív).

Valós pozitív esetén sikeres detektálja a pislogást a használt modell. Azonban, ha olyan állapotot is pislogásnak vesz, amely valójában nem pislogás, akkor azokat fals pozitív eredménynek kell számítani. Természetesen, ha teljesen elvétette az algoritmus az állapotot, akkor a fals negatív eredmények számát fogja növelni. Végezetül maradtak a valós negatív eredmények, melyeknél sikeresen nem történt detektálás. Az utóbbi eredményeket nem szokás felhasználni, mivel ezek sokszor nem határozhatóak meg egyszerűen.

A korábban bemutatott eredménytípusok alapján számos metrikát lehet meghatározni. Ilyen a Precision (precízió), mely a pontosság arányát határozza meg a modellnek. Az alábbi módon számítható:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

A Recall metrika a valós pozitív detektálások számának arányát írja le az összes detektálható pislogáshoz viszonyítva. Következő módon számítható:

$$Recall = \frac{True\ positive}{True\ positive + False\ negativ}$$

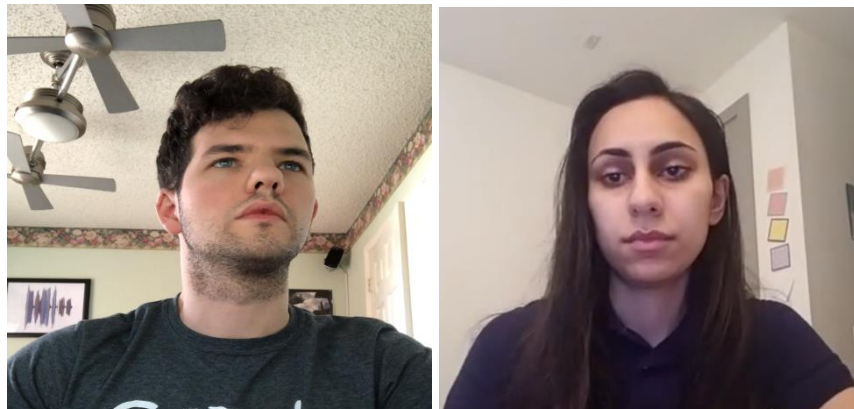


Akkor tökéletes az alábbi két metrika, ha mindkettő értéke egyet vesz fel. Azonban a legtöbb esetben ez nem érhető el. Sokszor valamely értéket be kell áldozni a másik kárára. Ezt az egyensúlyt kell jól meg választani az optimális működés érdekében.

A következő metrika pedig az F1 score (F1 érték). Az F1 score valójában a Precision és a Recall súlyozott átlaga. Ezáltal minden eredmény típust számításba vesz a pontosság meghatározásánál. Számítási módja:

$$F1 = 2 \times \frac{Precision \times Recall}{(Precision + Recall)}$$

A 20. ábra baloldalán látható 02-es és a jobb oldalon pedig 17-es azonosítójú tesztalany látható. A három tesztalany kiválasztásánál fontos tényező volt, hogy minél változatosabb legyen, hogy a legtöbb szituációban láthatóak legyen a program működésének előnyei és hátránya.



20. ábra: Kettő a három teszt alanyból (harmadik személy arcképe nem szerepelhet publikációkban)



A 5. táblázat oszlopaiban látható a tesztalanyok álmos és éber videókra bontott végeredménye. A sorokban pedig a két algoritmus összehasonlítását láthatjuk a korábban említett metrikák szerint százalékosan megadva.

		02		07		17	
		Éber	Álmos	Éber	Álmos	Éber	Álmos
HOG+SVM	Recall százalékosan	88%	95%	84%	76%	73%	90%
	Precision százalékosan	78%	92%	73%	87%	82%	75%
	F1 score százalékosan	83%	93%	78%	80%	77%	82%
	Fals pozitív aránya	22%	7%	27%	13%	18%	25%
Caffe/DNN	Recall százalékosan	89%	95%	84%	86%	80%	90%
	Precision százalékosan	88%	93%	72%	85%	70%	78%
	F1 score százalékosan	88%	94%	77%	85%	75%	84%
	Fals pozitív aránya	12%	7%	28%	15%	30%	22%

5. táblázat: HOG+SVM és Caffe/DNN megoldás összehasonlítása metrikák alapján



	02		07		17	
	Éber 1080 x1920	Álmos 1080 x1920	Éber 1920 x1080	Álmos 1920 x1080	Éber 480 x640	Álmos 240 x426
HOG+SVM FPS	26	26	30	30	32	71
Caffe/DNN FPS	33	33	33	33	50	55

6. táblázat: Algoritmusok összehasonlítása FPS alapján

A 6. táblázat mezőiben a két algoritmus FPS számait láthatjuk videókra felbontva sokszor különböző felbontásban. Ezen eredmények is a korábbi teszt alanyok videoiból lett meghatározva.



9. EREDMÉNYEK ÉRTÉKELÉSE

Az 5. táblázat 02-es alany eredményein láthatjuk, hogy a két megoldás szinte teljesen megegyező eredményeket hozott Recall érték alapján. Azonban az éber videónál a falszpozitív eredmények közel a duplája HOG+SVM-nek a Caffe modell alapú megoldással szemben. Ez valamelyest annak is köszönhető, hogy a Caffe modell nem használ átméretezést a jellemzők detektálásánál, így pontosabb eredményt tudott elérni. Közben ezt a pontosságot minimálisan, de nagyobb FPS számmal tudta elérni CPU-n. A 07-es teszt alanyánál már eltérő eredmények tekinthetőek meg. Az éber video eredményei közel azonosok, míg az álmos videónál már láthatóan egy nagyobb a differencia Recall érték szempontjából. Összehasonlítva az előző tesztalannyal, nehezebb feladatuk volt az előző tesztalanyhoz képest. Abból is látható, hogy miközben a Recall érték csökkent, addig a falszpozitív detektálások aránya megduplázódott. Részben köszönhető a tesztalany nagyon sűrű pislogásának, amely közelítőleg 20 kép kockánként történt meg a többi tesztalany 100-120 képkockánként történő pislogásához képest. Az utolsó alanyánál módosításra volt szükség a HOG+SVM algoritmuson mivel az álmos és az éber videó alapfelbontása jelentősen alacsonyabb a korábbiakhoz, ennél fogva átméretezve már átlépte azt a határt, amely felbontáson már nem képes stabilan működni az algoritmus. Az éber videónál észrevehető, hogy a Caffe modell nagyobb Recall értéket képes elérni, de ezt csak nagyobb falszpozitív aránnyal képes fenntartani, míg a HOG+SVM alacsonyabb Recall értéket és jobb falszpozitív arányt sikerült felmutatni. Az álmos videó esetén közel azonos végkifejlet következett, habár a HOG+SVM-nek minimálisan jobb a falszpozitív aránya. Viszont FPS számban már nagyobb a különbség a két algoritmus között a HOG+SVM javára. Az alábbi váratlan eredményt az alacsony felbontás okozhatta, mivel ez már közel van a Caffe modell átméretezett felbontásához és így az eddigi minimális előnye megfordult a HOG+SVM-mel szemben.

Az előzőleg kielemezett táblázatot láthatjuk összefoglalva a 7. táblázat soraiban a Precision, a Recall és F1 érték alapján.



	HOG+SVM	Caffe/DNN
Átlagos Recall érték százalékosan	84%	87%
Átlagos Precision érték százalékosan	81%	81%
Átlagos F1 score százalékosan	82%	84%
Átlagos fals pozitív arány százalékosan		

7. táblázat: Összesített táblázat a két algoritmusról

Összességében elmondható a 7. táblázat alapján, hogy a Caffe modell alapú megoldás pontosság szempontjából minimálisan, de jobb értékeket ér el miközben kiegyensúlyozottabb is, amely a 3% százalékkal nagyobb átlag pontosságból és a magasabb minimum Recall értékből is észrevehető. Ezért cserébe átlagosan 1%-kal magasabb fals pozitív arányt volt képes felmutatni. Teljesítmény szempontjából is minimálisan 3 FPS-el a HOG+SVM előtt jár a Caffe modell alapú megoldás CPU-n futtatva, miközben egy deep learning algoritmusról van szó, amely drasztikusan felgyorsítható GPU segítségével.

Azonban még fontos megtekinteni az alkalmazás viselkedését nem csak teszt környezetben, hanem valós körülmények között is. A rendszert egy 640x480 felbontású webkamerán lett tesztelve (21. ábra), amely összességében 30 FPS-re volt képes. A felbontásnak köszönhetően a stabil látótávolsága közelítőleg 40-50 centiméter lehet. Azonban ezt lehet növelni, ha egy nagyobb felbontású kamerát használunk, mivel az algoritmus szempontjából az arc méretnek kell lennie minimum 96x96 pixelben. Továbbá a kamera FPS száma ideális esetben minél magasabb, de minimum el kell, hogy érje az algoritmus teljesítményét. Különben torzíthatja az a program működését. Magasabb FPS szám előnye, hogy ha képes az algoritmus többször



mintavételezni akkor pontosabb eredményt érhet el. Hátránya, hogy a meghatározott paramétereket hozzá kell igazítani.



21. ábra: Működés valós környezetben

10. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A fejlesztés folyamán nagyon fontos volt az ideális EAR határérték megtalálása, mivel akár 10-20% százalékokat is tudott rontani egy rosszul beállított határérték, mert sok esetben ezen érték körül mozogtak, amelyet könnyen elvéthet az algoritmus. Rengeteget segített a program elején futtatott mintavételezés, de így is sokszor módosítani kellett az ideális tartomány megtalálásához. Teszt során előjött, hogy a pislogások száma nagyban függ a személytől, így a továbbiakban mindig a személyhez kéne igazítani a pislogási frekvenciákat. Ebben az esetben több mintára is lenne szükség a megvizsgált személytől.

Az egész folyamat rengeteg számítással járt, amelyet többféle módon fel lehetne gyorsítani. Először is erre a feladatra tervezett cél hardver nyújthatna fejlődési lehetőséget. Hátránya, hogy költséges és sok időt igényel ennek a legyártása. Habár a



Coffe modell gyorsítható GPU-val, viszont ez a jelenlegi piacon költséges megoldás. A másik opció, hogy ezeket a számításokat áthelyezzük a felhőbe, ezáltal megszűnik a teher a kamerát kezelő eszközről. Habár itt ma technológiai limitekbe ütközünk, hiszen valószínűsíthetőleg sávszélesség és lefedettségi problémákba ütköznénk. Ezen felül sok múlna azon, hogy a felhasználó milyen messze van az adatközponttól.

11. ÖSSZEFOGLALÁS

Összefoglalva a szakdolgozatom egy olyan alkalmazásról szól, amely a sofőrökön képes megfigyelni az álmoság állapotát és erre figyelmezteti is, ezáltal elkerülve több tragikus kimenetelű balesetet.

Elmélyülve az gépi látás világában témérdek megoldást találtam, amelyeket végül felhasználtam a rendszertervemben. Több más kutatást megvizsgáltam, amelyek különböző problémákat boncolgattak, mint például a szemüveg probléma detektálásnál vagy pedig a sötét környezetben történő használata a kamerának.

Végül elkészítettem a programomat, amelyet az OpenCV és a dlib könyvtárak segítségével implementáltam. Fejlesztés során több optimalizációkat hajtottam végre a programon. Ilyen volt a képek átméretezése, mintavételezés bevezetése és az átlagolt EAR érték számítás.

Végezetül elmondhatom, hogy sikerült megvalósítani a specifikációkat és még sok fejlődési lehetőséget rejt magában.

12. SUMMARY

In summary my thesis is about drowsiness detection on drivers which can detect drowsiness and alert the driver. And this could prevent a great deal of tragical car accidents.

Through my journey of the computer vision world, I have found a lot of possible solutions for my system specification. I have reviewed other thesis and I have found different kind of problems like detecting with eyeglasses or detecting in the dark



In the end I have implemented my plan which have used OpenCV and dlib libraries. To get better I used optimizations like resizing images, using sample period before the main the main program or calculating avarged Eye Aspect Ratio instead of simple Eye Aspect Ratio.

Finally I can say that i have succesfully implemented my specification and it still have potential to evolve.



13. IRODALOMJEGYZÉK

- [1] National Institute of Neurological Disorders and Stroke, „Brain Basics: Understanding Sleep,” [Online]. Available: <https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Understanding-Sleep>. [Hozzáférés dátuma: 02 11 2020].
- [2] M. Murugappan, K. Sundaraj és A. Sahayadhas, „Detecting Driver Drowsiness Based on Sensors: A Review. *Sensors*,” *Sensors*, %1. kötet12, %1. szám12, pp. 16937-16953., 2012..
- [3] S. H. Fairclough és R. Graham, „Impairment of Driving Performance Caused by Sleep Deprivation or Alcohol: A Comparative Study,” *Human Factors The Journal of the Human Factors and Ergonomics Society*, %1. kötet41., %1. szám1., pp. 118-128., 1999..
- [4] M. Akin, M. B. Kurt, N. Sezgin és M. Bayram, „Estimating vigilance level by using EEG and EMG signals,” *Neural Comput & Applic*, %1. kötet17., p. 227–236., 2008.
- [5] M. J. Flores, J. M. Armingol és A. d. l. Escalera, „Driver Drowsiness Warning System Using Visual Information for Both Diurnal and Nocturnal Illumination Conditions,” *EURASIP Journal on Advances in Signal Processing*, %1. kötet2010., pp. 1-19., 2010..
- [6] W.-C. Cheng, H.-C. Liao, M.-H. Pan és C.-C. Chen, „A fatigue detection system with eyeglasses removal,” in *2013 15th International Conference on Advanced Communications Technology (ICACT)*, PyeongChang, South Korea, 2013.01.27-2013.01.30.



- [7] D. Kim, H. Han, S. Cho és U. Chong, „Detection of drowsiness with eyes open using EEG-based power spectrum analysis,” in *2012 7th International Forum on Strategic Technology (IFOST)*, Tomsk, 2012.
- [8] P. Viola és M. Jones, „Rapid Object Detection using a Boosted Cascade of Simple Features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001.
- [9] M. B. Varga és M. Tenk, „Valós idejű objektum felismerés gépi látássegítségével,” 2013.. [Online]. Available: <https://tdk.bme.hu/VIK/DownloadPaper/Valos-ideju-objektum-felismeres-gepi-latas>. [Hozzáférés dátuma: 13. 12. 2020.].
- [10] A. Deshpande, „A Beginner's Guide To Understanding Convolutional Neural Networks,” [Online]. Available: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>. [Hozzáférés dátuma: 30 04 2020].
- [11] T. Ahonen, A. Hadid és M. Pietikäinen, „Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, %1. kötet28., %1. szám12., pp. 2037-2041., 2006.
- [12] T. McReynolds és D. Blythe, *Advanced Graphics Programming Using OpenGL*, Morgan Kaufmann, 2005..
- [13] B. K. Savaş és Y. Becerikli, „Real Time Driver Fatigue Detection Based on SVM Algorithm,” in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, Istanbul, Turkey, 2018.10.25-2018.10.27.



- [14] F. Zhang, J. Su, L. Geng és Z. Xiao, „Driver Fatigue Detection Based on Eye State Recognition,” in *2017 International Conference on Machine Vision and Information Technology (CMVIT)*, Singapore, 2017..
- [15] D. E. King, „Dlib-ml: A Machine Learning Toolkit,” *Journal of Machine Learning Research*, %1. kötet10., pp. 1755-1758., 2009.
- [16] „OpenCV,” OpenCV team, [Online]. Available: <https://opencv.org/>. [Hozzáférés dátuma: 09 05 2022].
- [17] R. Ghoddoosian, M. Galib and V. Athitsos, “A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 2019.
- [18] K. E. Koech, „Towards Data Science,” [Online]. Available: <https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e>. [Hozzáférés dátuma: 09 05 2022].
- [19] SZTE, TTIK, Élettani, Szervezettani és Idegtudományi Tanszék, [Online]. Available: <https://phys.bio.u-szeged.hu/DT/elettan/ch12.html>. [Hozzáférés dátuma: 27 04 2020].
- [20] T. Guo, J. Dong, H. Li és Y. Gao, „Simple convolutional neural network on image classification,” in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, China, 2017.03.10-2017.03.12.
- [21] A. Kumar, A. Kaur és M. Kumar, „Face detection techniques: a review,” *Artificial Intelligence Review*, %1. kötet52., %1. szám1., p. 927–948., 2019..
- [22] S. K. Lal és A. Craig, „A critical review of the psychophysiology of driver fatigue,” *Biological psychology*, %1. kötet55., %1. szám3., pp. 173-194., 2001.



- [23] S. Albawi, T. A. Mohammed és S. Al-Azawi, „Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017..
- [24] E. P. Fotiadis, M. Garz és A. Barrientas, „Human Detection from a Mobile Robot Using Fusion of Laser,” *Sensors (Basel)*. 2013;13(9): Published 2013 Sep 4. doi:10.3390/s130911603, %1. kötet13., %1. szám9., pp. 11603-11635., 2013.
- [25] G. Liu, S. Mao és J. H. Kim, „A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis,” *Sensors*, %1. kötet19., %1. szám9., p. 2023., 2019..



14. ÁBRAJEGYZÉK

1. ábra: EEG frekvencia tartományok [19]	12
2. ábra: a) infravörös kamera b) sofforról fotó c) páratlan(feslő) kép d) páros(alsó) kép [5]	14
3. ábra: Szemüveg eltávolításhoz használt képletek	14
4. ábra: A szemüveg eltávolítás folyamatai [6]	15
5. ábra: Alfa hullámok változása álmos állapotban [7]	16
6. ábra: Haar-jellegzetességek [8]	19
7. ábra: Egy pont kiszámítás az integrál képen (ii – integrál kép, i – eredeti kép) [8]	20
8. ábra A D terület kiszámításához ki kell vonni az A, B és C területek értékét [8]...	20
9. ábra: CNN felépítése [10].....	21
10. ábra: Filter használata [10]	22
11. ábra: A bináris kód számításának folyamat [11]	23
12. ábra: Módosított változat [11]	23
13. ábra: A rendszer folyamatábrája.....	24
14. ábra: Hisztogram kiegyenlítés [8]	25
15. ábra: a) blokkokra bontás b) irány és átmenetek meghatározása [21]	26
16. ábra: Lineáris SVM [22].....	27
17. ábra: a) Bal szem b) Jobb szem indexelése [10]	27
18. ábra: Nagy képen a program működés közben, kis képen a program ahogy látja az arcot	31
19. ábra: Módosított Caffé megoldás	33
20. ábra: Kettő a három teszt alanyból (harmadik személy arcképe nem szerepelhet publikációkban)	37
21. ábra: Működés valós környezetben	42